

حل مسائل جاوا

(حل ۶۰۰ برنامه - مرجع کامل)



مولفین: دکتر رمضان عباس نژادورزی
دکتر فاطمه عبدی سقاواز

برخی از عناوین مهم

آشنایی با جاوا (دستورات ورودی و خروجی)
ساختار تصمیم و حلقه تکرار
متدها در جاوا
آرایه‌ها و رشته‌ها
کلاس‌ها و وراثت
فایل‌های ورودی و خروجی

حل مسائل جاوا

(حل ۶۰۰ برنامه - مرجع کامل)

تألیف:

دکتر رمضان عباس نژادورزی
دکتر فاطمه عبدی سقاواز



فن آوری نوین

سرشناسه	عباس نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	حل مسائل جاوا (حل ۶۰۰ برنامه - مرجع کامل)/تالیف رمضان عباس نژاد ورزی، فاطمه عبدی سقاواز.
مشخصات نشر	بابل: فناوری نوین، ۱۳۹۷.
مشخصات ظاهری	۴۵۶ ص.: مصور، جدول.
شابک	۴۲۵۰۰۰ ریال ۲-۱۵-۷۲۷۲-۶۰۰-۹۷۸:
وضعیت فهرست نویسی	فیا
موضوع	جاوا (زبان برنامه نویسی کامپیوتر) -- مسائل، تمرین ها و غیره.
موضوع	Java (Computer program language) -- Problems, exercises, etc.
شناسه افزوده	عبدی سقاواز، فاطمه، ۱۳۵۶ -
رده بندی کنگره	۱۳۹۷ ۱۷ع۲/۷۶/۷۳QA
رده بندی دیویی	۰۰۵/۱۳۳
شماره کتابشناسی ملی	۵۱۳۰۳۲۷



فن آوری نوین

www.fanavari-novin.net

بابل، کدپستی ۷۳۴۴۸-۴۷۱۶۷

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

حل مسائل جاوا (حل ۶۰۰ برنامه - مرجع کامل)
تألیف: رمضان عباس نژاد ورزی، فاطمه عبدی سقاواز

نوبت چاپ: چاپ اول

سال چاپ: بهار ۹۷

شمارگان: ۱۰۰۰

قیمت: ۴۲۵۰۰ تومان

نام چاپخانه و صحافی:

شابک: ۹۷۸-۶۰۰-۷۲۷۲-۱۵-۲

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

فهرست مطالب

فصل اول: آشنایی با جاوا (دستورات ورودی و خروجی).....	۵
فصل دوم: ساختار تصمیم و حلقه تکرار.....	۴۰
فصل سوم: متدها در جاوا.....	۱۴۱
فصل چهارم: آرایه و رشته‌ها.....	۲۳۲
فصل پنجم: کلاس‌ها و وراثت.....	۳۷۲
فصل ششم: فایل‌های ورودی و خروجی.....	۴۲۸
منابع:.....	۴۵۶

مقدمه

جاوا توسط شرکت سان میکرو سیستم، به عنوان یک زبان شیء گرا توسعه داده شده است. معماری بی طرف و امنیت زبان برنامه نویسی جاوا باعث شده اند که این زبان به محبوبیت برسد. معماری بی طرف، یعنی این که می توانید با جاوا برنامه های بنویسید که بر روی تمام سیستم عامل ها به خوبی کار کند. به همین دلیل در برخی از دانشگاه های ایران در برنامه سازی پیشرفته جاوا تدریس می شود. یکی از راه های آموزش زبان های برنامه نویسی، آموزش مفاهیم برنامه به همراه مثال های متعدد است. کتاب حاضر شامل ۶ فصل است که عبارت اند از:

فصل اول، شامل ۷۵ برنامه است که مفاهیمی از قبیل آشنایی با جاوا، عملگرها و دستورات ورودی و خروجی را آموزش می دهد.

فصل دوم، شامل ۲۳۵ برنامه است که مفاهیمی از قبیل ساختارهای تصمیم (switch, if) و ساختارهای تکرار (continue, break, while, do while, for) را آموزش می دهد.

فصل سوم، شامل ۱۰۷ برنامه است که متدهای کتابخانه ای و طرز کار آن ها و تعریف متدها و فراخوانی آن ها را آموزش می دهد.

فصل چهارم، شامل ۱۳۵ برنامه است که مفاهیم از قبیل آرایه یک بعدی، دوبعدی و رشته را آموزش می دهد. فصل پنجم، شامل ۳۰ برنامه است که مفاهیمی از قبیل کلاس ها، شیء گرایی، وراثت، چندریختی و پیاده سازی مجدد و عملگرها را آموزش می دهد.

فصل ششم، شامل ۴۰ برنامه است که ورودی و خروجی فایل های متنی و باینری را آموزش می دهد.

بخشی از کتاب جهت کاهش حجم آن به صورت پیوست الکترونیکی ارائه گردیده است.

از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک fanavarienovin@gmail.com ارسال نمایند.

در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

مؤلفین

fanavarienovin@gmail.com

نام کتاب	لینک خرید چاپی	لینک خرید الکترونیکی	لینک فایل نمونه
مبانی رایانه و برنامه‌نویسی به زبان ++C	http://daneshnegar.com/book/380238.html	http://ktbr.ir/b/30588	http://ketabesabz.com/dl/52319
آشنایی با مبانی امنیت شبکه (امنیت اطلاعات)	http://daneshnegar.com/book/371137.html	http://ktbr.ir/b/30327	
اصول طراحی پایگاه داده‌ها	http://daneshnegar.com/book/371655.html	http://ktbr.ir/b/29943	http://ketabesabz.com/dl/52155
آموزش کام‌به‌کام برنامه‌نویسی پایتون	http://daneshnegar.com/book/392582.html	http://ktbr.ir/b/29984	http://ketabesabz.com/dl/52181
حل مسائل ++C (آزمایشگاه کامپیوتر مرجع کامل)	http://daneshnegar.com/book/392262.html	http://ktbr.ir/b/29982	http://ketabesabz.com/dl/53546
آموزش کام‌به‌کام LINQ با C#	http://daneshnegar.com/book/369388.html	http://ktbr.ir/b/28451	
ساختمان داده‌ها با ++C	http://daneshnegar.com/book/379094.html	http://ktbr.ir/b/29676	http://ketabesabz.com/dl/52156
طراحی سیستم‌های شی‌گرا با زبان C#	http://daneshnegar.com/book/374658.html	http://ktbr.ir/b/29621	http://ketabesabz.com/dl/52126
مدیریت استراتژیک فناوری اطلاعات	http://daneshnegar.com/book/374659.html	http://ktbr.ir/b/29779	http://ketabesabz.com/dl/52125
گرافیک رایانه‌ای با زبان برنامه‌نویسی C#	http://daneshnegar.com/book/376021.html	http://ktbr.ir/b/29674	http://ketabesabz.com/dl/51049
درس و کنکور پایگاه داده پیشرفته	http://daneshnegar.com/book/392578.html	http://ktbr.ir/b/29644	http://ketabesabz.com/dl/52102
فیزیک الکترونیسته	http://daneshnegar.com/book/379161.html	http://ktbr.ir/b/29680	
تجارت الکترونیکی	http://daneshnegar.com/book/379188.html	http://ktbr.ir/b/29619	http://ketabesabz.com/dl/51013
راهنمای کاربردی کاربری OPNET برای شبکه‌های شبیه‌سازی کامپیوتر	http://daneshnegar.com/book/392583.html	http://ktbr.ir/b/28504	
درس و کنکور سیستم عامل پیشرفته	http://daneshnegar.com/book/392580.html	http://ktbr.ir/b/28505	http://ketabesabz.com/dl/52180
شبکه‌های کامپیوتری با رویکرد کاربردی، آزمایشگاه شبیه‌سازی شبکه	http://daneshnegar.com/book/392254.html	http://ktbr.ir/b/28528	
آزمایشگاه پایگاه داده با SQL Server ۲۰۱۲	http://daneshnegar.com/book/377301.html	http://ktbr.ir/b/28503	http://ketabesabz.com/dl/53447

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۷

http://ketabesabz.com/dl/۵۳۴۴۶	http://ktbr.ir/b۲۸۴۵۰	http://daneshnegar.com/book/۳۷۵۸۹۲.html	کاربرد رایانه در مدیریت و حسابداری
http://ketabesabz.com/dl/۵۳۴۸۸	http://ktbr.ir/b۲۸۴۴۹	http://daneshnegar.com/book/۳۶۸۹۲۹.html	آموزش گام به گام برنامه نویسی بانک اطلاعاتی با ویژوال بیسیک نت
	http://ktbr.ir/b۲۸۴۵۲	http://daneshnegar.com/book/۳۸۰۲۳۸.html	آموزش گام به گام برنامه نویسی به زبان ++C
http://ketabesabz.com/dl/۵۱۰۴۷	http://ktbr.ir/b۲۸۴۴۸	http://daneshnegar.com/book/۳۶۸۴۸۶.html	دانلود کتاب آموزش گام به گام برنامه نویسی بانک اطلاعاتی با #C
	http://ktbr.ir/b۲۸۳۹۸		حل مسائل پاسکال
http://ketabesabz.com/dl/۵۱۰۴۸	http://ktbr.ir/b۲۸۴۰۱	http://daneshnegar.com/book/۳۹۲۲۶۲.html	حل مسائل ++C
http://ketabesabz.com/dl/۵۱۰۱۱	http://ktbr.ir/b۲۸۳۹۹	http://daneshnegar.com/book/۳۷۴۶۵۷.html	دانلود کتاب حل مسائل #C
	http://ktbr.ir/b۳۲۶۷۲	http://daneshnegar.com/book/۳۹۲۸۱۸.html	۶۰۰ برنامه ++C با حل کامل آنها (حل مسائل - ++C مرجع کامل)
	http://ktbr.ir/b۳۲۶۷۱	http://daneshnegar.com/book/۳۹۲۸۱۷.html	حل مسائل جاوا (حل ۶۰۰ برنامه - مرجع کامل)
		http://daneshnegar.com/book/۳۹۲۸۱۵.html	۶۵۰ برنامه #C با حل کامل آنها (حل مسائل #C- مرجع).
		http://daneshnegar.com/book/۳۹۲۸۱۹.html	حل مسائل پایتون (حل ۶۵۰ برنامه - مرجع کامل)
		http://daneshnegar.com/book/۳۹۲۸۱۴.html	دانشی ثروتمند شدن
		http://daneshnegar.com/book/۳۹۲۸۱۶.html	برنامه سازی پیشرفته به زبان #C
		http://daneshnegar.com/book/۳۹۲۸۱۳.html	الگوریتم ها و محاسبات موازی
	http://ktbr.ir/b۲۸۳۹۷		دانلود کتاب حل مسائل C

فصل

۱

آشنایی با جاوا (دستورات ورودی و خروجی)

۱. دستوراتی که متغیرهای m و n را با نوع int ، $ch1$ و $ch2$ را با نوع $char$ ، $d1$ و $d2$ را با نوع $double$ و ثابت pl را با نوع $float$ و مقدار 3.14 تعریف می‌کند.

```
int m, n;
char ch1, ch2;
double d1, d2;
const float p = 3.14;
```

۲. دستورات زیر چند بایت از حافظه را اشغال می‌کنند (در محیط‌های ۳۲ بیتی)؟

```
long double x, y;
char ch;
float f1, f2;
```

این دستورات، ۲۹ بایت از حافظه را اشغال می‌کنند. زیرا هر یک از متغیرهای x و y ، ۱۰ بایت (نوع $long double$)، ch یک بایت (نوع $char$) و $f1$ و $f2$ هر یک ۴ بایت (نوع $float$) را اشغال می‌کنند. لذا، این دستورات مجموعاً ۲۹ (۸ + ۱ + ۲۰) بایت را اشغال می‌کنند.

۳. عبارت زیر را به ازای $x=5$ و $y=4$ ارزیابی کنید:

$$m = x + (y * 2) / 6 + y / 8;$$

مقادیر را در عبارت قرار می‌دهیم:

$$m = 5 + (4 * 2) / 6 + 4 / 8$$

$$m = 5 + 8 / 6 + 4 / 8$$

$$m = 5 + 1 + 4 / 8$$

$$m = 5 + 1 + 0 = 6$$

۴. عبارت زیر را به ازای $x=4$ و $y=3$ ارزیابی کنید:

$$m = x > y ? x * 3 : y * 3;$$

ابتدا مقادیر را در x و y قرار می‌دهیم:

$$m = 4 > 3 ? 3 * 3 : 4 * 3;$$

عبارت $4 > 3$ ارزیابی می‌شود. چون این شرط ارزش درستی دارد، $3 * 3$ (۹) در m قرار می‌گیرد.

۵. پس از اجرای دستورات زیر، چه مقداری در x قرار می‌گیرد؟

```
int x;
long double y;
x = sizeof y;
```

دستور اول، متغیر x را با نوع int تعریف می‌کند، دستور دوم، متغیر y را با نوع $long double$ تعریف می‌کند و دستور سوم، طول نوع متغیر y را محاسبه کرده، در متغیر x قرار می‌دهد. بنابراین، ۱۰ در x قرار می‌گیرد (نوع $long double$ ، ۱۰ بایت را اشغال می‌کند).

۶. عبارت زیر را بدون استفاده از عملگر $*$ انجام دهید:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۹

$$y = 255 * x - 18 * x + 15;$$

ابتدا $255 * x$ را محاسبه می‌کنیم. برای انجام این کار، x را ۸ بار به سمت چپ شیفت داده، از x کم می‌کنیم. زیرا، با هر بار شیفت، مقدار x در ۲ ضرب می‌شود. پس مقدار x در 2^8 (۲۵۶) ضرب می‌گردد. سپس، مقدار $18 * x$ را محاسبه کرده، (x را ۴ بار به سمت چپ شیفت داده دو بار با x جمع کرده، علامت آن را به منفی تبدیل می‌کنیم). در پایان، این مقادیر را با ۱۵ جمع می‌کنیم. یعنی، عبارت زیر را داریم:

$$\begin{aligned} y &= x \ll 8; \\ y1 &= y - x; \\ y2 &= x \ll 4; \\ y2 &+= (x + x); \\ y &= y1 - y2 + 15; \end{aligned}$$

۷. پس از اجرای عبارت زیر مقدار متغیر m چه خواهد شد؟

$$m = (y = 0, x = ++y, m = y, y + 2);$$

ابتدا y برابر با صفر می‌شود و در عبارت دوم به y یک واحد اضافه شده، در x قرار می‌گیرد. پس اکنون مقدار x و y برابر یک است. در عبارت بعدی، y در m قرار می‌گیرد و در نتیجه، m نیز برابر با یک می‌شود. حاصل عبارت $y + 2$ که برابر با ۳ است در m قرار می‌گیرد.

۸. وزن یک مولکول آب 3.0×10^{-23} گرم و وزن یک لیتر آب در حدود ۹۵۰ گرم است. برنامه‌ای که وزن آب را برحسب لیتر از ورودی خوانده، تعداد مولکول‌های آن را محاسبه می‌کند.

متغیر	هدف
m	وزن یک مولکول
l	وزن یک لیتر آب
w	وزن آب به لیتر از رودی خوانده می‌شود.
tedad	تعداد مولکول‌های آب

```
package ch1_8;
import java.util.*;
public class Ch1_8 {
    public static void main(String[] args) {
        double m = 3.0e-23, l = 950, tedad;
        long w;
        System.out.print("Enter w:");
        w = (new Scanner(System.in).next Long());
        tedad = (w * l) / m;
        System.out.printf("Tedad = %1$s" + "\n", tedad);
    }
}
```

خروجی برنامه:

Enter w: ۱۲۳

Tedad = ۳,۸۹۵E۲۷

۹. هرسال برابر با $3/156 \times 10^7$ ثانیه است. برنامه‌ای که سن شما را به سال دریافت کرده، به ثانیه تبدیل می‌کند.

```
package ch1_9;
import java.util.*;
public class Ch1_9 {
    public static void main(String[] args) {
        double ss = 3.156e7, second;
        int age;
        System.out.print("Enter age:");
    }
}
```

هدف	متغیر
هرسال برحسب ثانیه	ss
سن به سال	age
سن برحسب ثانیه	second

```
age = (new Scanner(System.in)).nextInt();
second = age * ss;
System.out.printf("Second = %1$s" + "\n", second);
}
```

خروجی برنامه:

Enter age: ۴۸
Second = ۱,۵۱۴۸۸E۹

۱۰. برنامه‌ای که حقوق کارمندی را خوانده، بیمه، مالیات و دریافتی او را محاسبه کرده، نمایش می‌دهد. بیمه و مالیات به ترتیب ۷ و ۱۰ درصد حقوق می‌باشند.

```
package ch1_10;
import java.util.*;
public class Ch1_10 {
public static void main(String[] args) {
long p,s,m,b;
System.out.print("Enter salary:");
s=(new Scanner(System.in)).nextLong();
b = s * 7 / 100;
m = s / 10;
p = s - b - m;
System.out.printf("s=%1$s b=%2$s m=%3$s p=%4$s " + " \n", s,b,m, p);
}
}
```

متغیر	هدف
s	حقوق
b	بیمه
m	مالیات
p	دریافتی

خروجی برنامه:

Enter salary: ۱۰۰۰۰۰۰

s = ۱۰۰۰۰۰۰ b = ۷۰۰۰۰ m = ۱۰۰۰۰۰ p = ۸۳۰۰۰۰

۱۱. برنامه‌ای که قیمت کالایی را در سال قبل و جاری خوانده، نرخ تورم و قیمت سال بعد آن را چاپ می‌کند. نرخ تورم باید به صورت درصد (مثلاً ۶/۵٪) حساب گردد. (توضیح: برای محاسبه نرخ تورم و قیمت سال بعد به صورت زیر عمل کنید):

$$\text{نرخ تورم} = \frac{\text{قیمت سال قبل} - \text{قیمت سال جاری}}{\text{قیمت سال قبل}}$$

$$\text{نرخ تورم} \times \text{قیمت سال جاری} + \text{قیمت سال قبل} = \text{قیمت سال بعد}$$

متغیر	هدف
T	نرخ تورم
y ^۱	قیمت کالا در سال قبل
y ^۲	قیمت کالا در سال جاری
y ^۳	قیمت تخمینی کالا در سال بعد

```
package ch1_11;
import java.util.*;
public class Ch1_11 {
public static void main(String[] args) {
double t, y1,y2,y3;
System.out.print("Enter price for year1:");
y1 = (new Scanner(System.in)).nextDouble();
System.out.print("Enter price for year2:");
y2 = (new Scanner(System.in)).nextDouble();
t = (float)(y2 - y1) / y1;
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۱

```

y3 = y2 + y2 * t;
System.out.printf("Extera = %1$s\t\t Price next
    year = %2$s", t, y3);
}
}

```

خروجی برنامه :

Enter price for year ۱: ۱۰۰۰۰۰

Enter price for year ۲: ۱۱۰۰۰۰

Price next year = ۱۲۱۰۰۰۰

Extera = ۰,۱

برنامه‌ای که مقدار x را از ورودی خوانده، حاصل عبارت مقابل را محاسبه می‌کند:

$$y = \frac{1}{x^2 + x + 1}$$

برنامه‌ای که دو عدد را از ورودی می‌خواند (عدد اول مقاومت یک مدار الکتریکی و عدد دوم جریان آن را نمایش می‌دهد). با فرمول مقابل ولتاژ را محاسبه کرده، نمایش می‌دهد:

جریان * مقاومت = ولتاژ

مجموع مقاومت‌های R_۱، R_۲ و R_۳ که به‌طور موازی به هم متصل‌اند، با فرمول زیر محاسبه می‌شود:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} = \frac{1}{R} = \frac{R_2 * R_3 + R_1 * R_3 + R_1 * R_2}{R_1 * R_2 * R_3}$$

برنامه‌ای که سه مقاومت را از ورودی خوانده، مقاومت کل را محاسبه می‌کند.

برنامه‌ای که وزن کالایی را برحسب کیلوگرم دریافت می‌نماید و وزن آن را برحسب گرم نمایش می‌دهد. هر کیلوگرم برابر ۱۰۰۰ گرم است.

اگر کارمندی به‌اندازه ۱۵ درصد حقوق ماه گذشته خود پاداش بگیرد، برنامه‌ای که حقوق کارمند را خوانده، پاداش او را محاسبه نموده، نمایش می‌دهد.

برنامه‌ای که ارتفاع و قاعده مثلث را از ورودی خوانده، مساحت آن را محاسبه کرده، به خروجی می‌برد.

قاعده * ارتفاع × ۰,۵ = مساحت مثلث

➤ در شرکتی، سالانه ۱۵۰ خودکار، ۵۰ بسته کاغذ A۴ مصرف می‌شود. در پایان سال این شرکت می‌خواهد بدانند در سال آینده چقدر برای این بخش از تجهیزات اداری، افزایش هزینه دارد. برنامه‌ای که قیمت این اقلام را در امسال از ورودی دریافت می‌نماید، نرخ تورم در سال آینده را نیز می‌گیرد، هزینه اضافه شده شرکت را در این بخش محاسبه می‌کند و به خروجی می‌برد. تورم به صورت درصد وارد می‌گردد که برنامه باید آن را به مقدار اعشاری تبدیل کند. به عنوان مثال، اگر تورم را ۵٫۶ وارد کنیم، برنامه باید از ۰/۰۵۶ استفاده کند.

➤ یک دوچرخه‌سوار با سرعت x کیلومتر بر ساعت شروع به حرکت می‌کند و پس از n دقیقه سرعت آن به k کیلومتر در ساعت می‌رسد. برنامه‌ای که با استفاده از فرمول زیر شتاب او را محاسبه کرده، نمایش می‌دهد:

$$\text{شتاب} = \frac{60 * (\text{سرعت اولیه } x) + (\text{سرعت نهایی } k)}{n(\text{زمان})}$$

➤ برنامه‌ای که مصرف (کیلومتر به لیتر) و گنجایش باک بنزین (لیتر) اتومبیلی را از ورودی می‌خواند و تعیین می‌کند که با یک باک بنزین اتومبیل چه مسافتی را می‌تواند پیماید.

➤ برنامه‌ای که دو عدد را از کاربر دریافت می‌کند و حاصل جمع، ضرب، تقسیم، تفریق و باقی‌مانده تقسیم صحیح آن‌ها را چاپ می‌کند.

➤ برنامه‌ای که دو عدد صحیح را از کاربر دریافت کرده، عدد بزرگ‌تر و سپس عبارت is larger than و عدد کوچک‌تر را چاپ می‌کند. اگر این دو عدد برابر باشند، عبارت "These number are equal" را چاپ می‌کند.

➤ برنامه‌ای که سه عدد را از صفحه کلید می‌گیرد و حاصل جمع، میانگین، حاصل ضرب، بزرگ‌ترین و کوچک‌ترین عدد را چاپ می‌کند.

➤ برنامه‌ای که دو عدد را از صفحه کلید می‌گیرد و تعیین می‌کند که آیا عدد اول بر عدد دوم بخش پذیر است یا نه؟

➤ برنامه‌ای که دو عدد را خوانده، تعیین می‌کند آیا عدد اول مضربی از عدد دوم است یا خیر.

➤ برنامه‌ای که پنج عدد صحیح را خوانده، بزرگ‌ترین و کوچک‌ترین عدد را نمایش می‌دهد.

➤ با استفاده از امکاناتی که در این فصل دیدید برنامه‌ای که توان ۲، توان ۳، و عدد ۰ تا ۱۰ را به صورت زیر جدول بندی می‌نماید؟ (توضیح: همان‌طور که مشاهده نمودید، تعداد

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۳

خطوط این برنامه زیاد شده است که با بیان حلقه‌های تکرار در فصل بعد، تعداد دستورات این قبیل برنامه‌ها کاهش خواهد یافت).

➤ برنامه‌ای که یک عدد صحیح پنج‌رقمی را خوانده، ارقام آن را جدا می‌کند و هر یک از ارقام را با سه فاصله بین آن‌ها چاپ می‌کند. به عنوان مثال، اگر کاربر عدد ۴۲۳۳۹ را وارد کند، خروجی به صورت زیر باشد:

Result is 4 2 3 3 9

➤ هر تن ۳۵۲۷۳,۹۲ اونس است. برنامه‌ای که وزن یک محموله را به اونس گرفته، به تن تبدیل می‌کند. برنامه به کاربر اجازه می‌دهد تا هر زمان می‌خواهد این محاسبه را تکرار کند.

➤ هر لیتر معادل ۰,۲۶۴۱۷۹ گالن است. برنامه‌ای که میزان بنزین مصرفی اتومبیل کاربر را برحسب لیتر و مسافت طی شده آن را به مایل گرفته، سپس مصرف به ازای هر مایل به گالن را نمایش می‌دهد.

➤ برنامه‌ای که عددی را خوانده، مربع و مکعب آن را نمایش می‌دهد.

➤ برنامه‌ای که دو ضلع موازی و ارتفاع (a, b و h) ذوزنقه را دریافت کرده، مساحت آن را با فرمول زیر محاسبه می‌کند:

$$s = h * (a + b) / 2$$

➤ برنامه‌ای که شعاع دایره را خوانده، محیط، مساحت و قطر آن را نمایش می‌دهد (محیط، مساحت و قطر دایره به صورت زیر حساب می‌شوند):

$$\pi * \text{شعاع} * 2 = \text{محیط دایره}$$

$$\pi * (\text{شعاع})^2 = \text{مساحت دایره}$$

$$\text{شعاع} * 2 = \text{قطر دایره}$$

➤ برنامه‌ای که عددی را به سال خوانده، تعیین می‌کند چند ماه، چند روز و چند ثانیه است.

$$\text{سال} * 365,25 = \text{روز}$$

$$60 * 60 * 24 * \text{روز} = \text{ثانیه}$$

➤ برنامه‌ای که m و n را خوانده، حاصل عبارات زیر را محاسبه کند:

$$a = m^2 - n^2$$

$$b = 2 * m * n$$

$$c = m^2 + n^2$$

- برنامه‌ای که دو عدد را خوانده، در متغیرهای a و b قرار می‌دهد، سپس حاصل جمع، تفریق، ضرب، تقسیم، a^2+b^2 و a^3+b^3 را نمایش می‌دهد.
- برنامه‌ای که یک عدد سه‌رقمی را خوانده، اگر مجموع رقم‌های اول و سوم برابر رقم دوم باشد، "Yes"، وگرنه "No" را چاپ می‌کند.
- فرض کنید هر صفحه کتاب ۳۰ سطر (خط) است و هر خط ۸۰ بایت حافظه اشغال می‌کند. برنامه‌ای که تعداد صفحات یک کتاب و ظرفیت یک فلش را به گیگابایت خوانده، تعیین می‌کند که این فلش چند تا از این کتاب‌ها را می‌تواند در خودش ذخیره کند (گیگابایت برابر با 1024^3 بایت است).
- برنامه‌ای که یک عدد چهاررقمی را خوانده، اگر حاصل ضرب رقم‌های اول و چهارم، برابر حاصل جمع ارقام دوم و سوم باشد، "Yes"، وگرنه، "No" را نمایش می‌دهد.
- برنامه‌ای که تعداد کالا و قیمت هر کالا را خوانده، مبلغ فروش را نمایش می‌دهد (مبلغ فروش برابر با تعداد کالا * قیمت کالا است).
- برنامه‌ای که سن‌تان را به سال خوانده، مشخص کند چند دقیقه زندگی کرده‌اید (هر سال ۳۶۵,۲۵ روز است و هر روز 24×60 دقیقه می‌باشد).
- برنامه‌ای که مقداری را به مایل و فوت دریافت کرده، به متر و کیلومتر تبدیل می‌کند و نمایش می‌دهد (هر فوت ۰,۳۰۴۸۰ متر و هر مایل ۱۶۰۹,۳۵ متر است).
- برنامه‌ای که x و y را از ورودی خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$Z = x++ * y / x++;$$
- برنامه‌ای که یک حرف انگلیسی کوچک را خوانده، به حرف بزرگ تبدیل کرده و نمایش می‌دهد (فاصله بین حرف بزرگ و کوچک ۳۲ کاراکتر است. یعنی، چنانچه از یک حرف کوچک ۳۲ تا کم کنیم به حرف بزرگ تبدیل می‌شود).
- برنامه‌ای که یک کاراکتر را خوانده و تعیین می‌کند که بین ۰ تا ۹ است یا خیر؟
- برنامه‌ای که x و y را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$z = x^3 + 2x^2 + 3y - 5$$

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۵

- برنامه‌ای که نمره‌ای دانشجویی را دریافت کرده، اگر نمره کم‌تر از ۱۰ بود، کلمه "Failed" وگرنه کلمه "Passed" را نمایش می‌دهد (در این برنامه متغیر grade، نمره دانشجو است).
- برنامه‌ای که دو عدد را خوانده، بدون استفاده از متغیر کمکی محتوی آن‌ها را تعویض می‌کند.
- برنامه‌ای که یک حرف کوچک را خوانده، با استفاده از عملگر بیتی آن را به حرف بزرگ تبدیل می‌کند (کد اسکی حروف کوچک ۳۲ واحد از حروف بزرگ بیش‌تر است). برای این منظور، اگر کاراکتر حرف کوچک باشد، آن را با مقدار (۳۲ - ۲۵۵) بیت به بیت و (&) بیتی می‌نماییم تا تبدیل به حرف بزرگ شود.
- برنامه‌ای که قاعده و ارتفاع مثلث را خوانده، مساحت مثلث را محاسبه می‌کند (مساحت مثلث برابر با ارتفاع ضربدر نصف قاعده است).
- پروژه برنامه‌نویسی ۱: برنامه‌ای که ۵ عدد را خوانده، بزرگ‌ترین و کوچک‌ترین عدد را نمایش می‌دهد.
- پروژه برنامه‌نویسی ۲: برنامه‌ای که یک عدد ۵ رقمی را خوانده، ارقام عدد را بافاصله نمایش می‌دهد.

ساختار تصمیم و حلقه تکرار

۱. برنامه‌ای که شماره دانشجویی و معدل تعداد N دانشجو را از ورودی خوانده، دانشجویی که دومین معدل را از نظر بزرگی دارد، پیدا می‌کند و به خروجی می‌برد. (توضیح: چون هیچ‌یک از دانشجویان معدلشان کوچکتر از صفر نمی‌باشد، لذا اولین بار معدل دو دانشجویی که بالاترین معدل را دارند، صفر در نظر گرفتیم.)

متغیر	هدف
n	تعداد دانشجویان
max ₁	اولین معدل
max ₂	دومین معدل
i	شمارنده
id ₁	شماره دانشجویی که اولین معدل را دارد.
id ₂	شماره دانشجویی که دومین معدل را دارد.
Aver	معدل دانشجو
Id	شماره دانشجو

```
package ch2_1;
import java.util.*;
public class Ch2_1 {
    public static void main(String[] args) {
        int id1 = -1, id2 = 0, i, n, id;
        float aver, max1 = -1, max2 = 0;
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextInt();
        if (n < 2)
        {
            System.out.println("Please enter a number greater than 1");
            return;
        }
        for (i = 1; i <= n; i++)
        {
            System.out.print("Enter a id:");
            id = (new Scanner(System.in)).nextInt();
            System.out.print("Enter a average:");
            aver = (new Scanner(System.in)).nextFloat();
            if (aver > max1)
            {
                id2 = id1;
                max2 = max1;
                max1 = aver;
                id1 = id;
            }
            else if (aver > max2)
            {
                max2 = aver;
                id2 = id;
            }
        }
        System.out.printf("Max2 = %1$s\tId2 = %2$s"+ "\n", max2, id2);
    }
}
```

خروجی برنامه:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۷

```

Enter n: ۴
Enter a id: ۱
Enter a average: ۱۴, ۵
Enter a id: ۲
Enter a average: ۱۳, ۵
Enter a id: ۳
Enter a average: ۱۸
Enter a id: ۴
Enter a average: ۱۰
Max ۲ = ۱۴, ۵ Id ۲ = ۱
    
```

۲. برنامه‌ای که اعدادی را از ورودی خوانده تشخیص می‌دهد که آیا اعداد موردنظر کامل هستند یا خیر. عددی کامل است که مجموع مقسوم‌علیه‌های آن (به‌جز خودش) برابر با آن عدد باشد، پس از بررسی هر عدد از کاربر سؤال کند که می‌خواهد به کارش ادامه دهد یا خیر.

متغیر	هدف
num	اعداد موردنظر
i	شمارنده از ۱ تا num
Sum	مجموع مقسوم‌علیه‌ها
Yes	تعیین ادامه کار

```

package ch2_2;
import java.util.*;
public class Ch2_2 {
    public static void main(String[] args) {
        int sum, num, i;
        String yes;
        do
        {
            System.out.print("Enter a number:");
            num = (new Scanner(System.in)).nextInt();
            sum = 0;
            for (i = 1; i < num; i++)
            {
                if (num % i == 0)
                {
                    sum += i;
                }
            }
            if (sum == num)
            {
                System.out.println("Perfected");
            }
            else
            {
                System.out.println("Not Perfected");
            }
            System.out.print("Continue ... ?");
            yes = new Scanner(System.in).nextLine();
        } while(yes.compareTo("y") == 0 || yes.compareTo("Y") == 0);
    }
}
    
```

خروجی برنامه:

```

Enter a number: ۶
Perfected
Continue ... ?y
Enter a number: ۱۰
Not Perfected
Continue ... ?y
Enter a number: ۲۸
Perfected
Continue ... ?n
    
```

۳. برنامه‌ای که تعداد n جمله از سری فیبوناچی را تولید می‌کند.
 ۱ = سری فیبوناچی
 ۱ ۱ ۲ ۳ ۵ ۸ ۱۳ ۲۱ ۳۴ ۵۵

```
package ch2_3;
import java.util.*;
public class Ch2_3 {
    public static void main(String[] args) {
        int n, f1 = 1, f2 = 1, f3, i;
        System.out.print("Enter a number:");
        n = (new Scanner(System.in)).nextInt();
        if (n == 1)
        {
            System.out.print(f1);
            return;
        }
        else if (n == 2)
        {
            System.out.printf("%1$s %2$s ", f1, f2);
            return;
        }
        System.out.printf("%1$s %2$s ", f1, f2);
        for (i = 3; i <= n; i++)
        {
            f3 = f1 + f2;
            System.out.printf("%1$s ", f3);
            f1 = f2;
            f2 = f3;
        }
    }
}
```

متغیر	هدف
I	شمارنده
f1	جملات سری
f2	جملات سری
f3	جملات سری
N	تعداد جملات

خروجی برنامه:

Enter a number: ۱۰
 ۱ ۱ ۲ ۳ ۵ ۸ ۱۳ ۲۱ ۳۴ ۵۵

۴. برنامه‌ای که شماره کارمندی و حقوق تعدادی از کارکنان موسسه را دریافت کرده، بر اساس تعریف زیر، مالیات حقوق را محاسبه می‌کند و به خروجی می‌برد. سپس مشخص می‌کند بیشترین دریافتی مربوط به کدام کارمند است.

مالیات از معاف ۴۰۰۰۰۰ ≤ حقوق
 ۱۰ درصد به مازاد بر ۴۰۰۰۰۰ ۵۰۰۰۰۰ ≤ حقوق ≤ ۵۰۰۰۰۰
 ۱۵ درصد مازاد ۵۰۰۰۰۰ ۷۰۰۰۰۰ ≤ حقوق ≤ ۷۰۰۰۰۰
 ۱۷ درصد مازاد ۷۰۰۰۰۰ > حقوق

```
package ch2_4;
import java.util.*;
public class Ch2_4 {
    public static void main(String[] args) {
        long i, salary, tax = 0, pay, max = 0, id1 = 0, id = -1, n;
        System.out.print("Enter a number:");
        n = (new Scanner(System.in)).nextLong();
        for (i = 1; i <= n; i++)
        {
```

متغیر	هدف
id	شماره کارمندی
salary	حقوق
ax	مالیات
pay	دریافتی
i	شمارنده
id1	شماره کارمندی که بیشترین دریافتی را دارد
max	بیشترین دریافتی
n	تعداد کارمندان

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۹

```

System.out.print("Enter a id:");
id = (new Scanner(System.in)).nextLong();
System.out.print("Enter a salary:");
salary = (new Scanner(System.in)).nextLong();
if (salary <= 400000)
{
    tax = 0;
}
else if (salary >= 400001 && salary <= 500000)
{
    tax = (salary - 400000) * 10 / 100;
}
else if (salary >= 500001 && salary <= 700000)
{
    tax=(500000-400001)*10/100+(salary - 500001) * 15 / 100;
}
else if (salary >= 700001)
{
    tax = (500000 - 400001) * 10 / 100 + (salary - 500001) *
        15 / 100 + (salary - 700001) * 17 / 100;
}
pay = salary - tax;
if (pay > max)
{
    max = pay;
    id1 = id;
}
System.out.printf("Salary = %1$s Tax = %2$s Pay= %3$s" +
    "\n", salary, tax, pay);
}
System.out.printf("Max = %1$s Id = %2$s" + "\n", max, id1);
}
}

```

خروجی برنامه:

```

Enter a number: 4
Enter a id: 1
Enter a salary: 1000000
Salary = 1000000 Tax = 130997 Pay= 869003
Enter a id: 2
Enter a salary: 2000000
Salary = 2000000 Tax = 450997 Pay= 1549003
Enter a id: 3
Enter a salary: 1500000
Salary = 1500000 Tax = 290997 Pay= 1209003
Enter a id: 4
Enter a salary: 1700000
Salary = 1700000 Tax = 350997 Pay= 1349003
Max = 1549003 Id = 2

```

۵. برنامه‌ای که سال تولد کاربر و سال فعلی را از ورودی خوانده، مشخص می‌کند چند سال، چند ماه، چند روز، چند ساعت، چند دقیقه و چند ثانیه عمر کرده است.

متغیر	هدف
byy	سال تولد
bmm	ماه تولد
bdd	روز تولد
Cyy	سال جاری

ماه جاری	cmm
روز جاری	Cdd
سالهایی که زندگی کرده	year
ماههایی که زندگی کرده	month
روزهایی که زندگی کرده	Day
تعداد ساعاتی که زندگی کرده	hh
تعداد دقیقاتی که زندگی کرده	mm
تعداد ثانیههایی که زندگی کرده	ss
تعداد کل روزهایی که زندگی کرده	days

```

package ch2_5;
import java.util.*;
public class Ch2_5 {
    public static void
main(String[] args) {
    long byy,bdd,bmm,cyy,cdd,cmm,year,month,day;
    double hh,ss,mm,days;
    System.out.print("Enter birth date(year):");
    byy = (new Scanner(System.in)).nextLong();
    System.out.print("Enter birth date(month):");
    bmm = (new Scanner(System.in)).nextLong();
    System.out.print("Enter birth date(day):");
    bdd = (new Scanner(System.in)).nextLong();
    System.out.print("Enter current date(year):");
    cyy = (new Scanner(System.in)).nextLong();
    System.out.print("Enter current date(month):");
    cmm = (new Scanner(System.in)).nextLong();
    System.out.print("Enter current date(day):");
    cdd = (new Scanner(System.in)).nextLong();
    if (cdd < bdd)
    {
        cmm--;
        cdd += 30;
    }
    day = cdd - bdd;
    if (cmm < bmm)
    {
        cyy--;
        cmm += 12;
    }
    month = cmm - bmm;
    year = cyy - byy;
    days = day + month * 30 + year * 365;
    hh = days * 24;
    mm = hh * 60;
    ss = mm * 60;
    System.out.printf("Old is: %1$s/%2$s/%3$s" + "\n", year,
    month, day);
    System.out.printf("Hour is(hh:mm:ss): %1$s:%2$s:%3$s" +
    "\n", (long) hh, (long) mm, (long) ss);
    }
}

```

خروجی برنامه:

```

Enter birth date(year) : ۱۳۴۸
Enter birth date(month) : ۱۰
Enter birth date(day) : ۰۴

```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۲۱

Enter current date (year): ۱۳۹۶
 Enter current date (month): ۱۱
 Enter current date (day): ۲۳
 Old is: ۴۸/۱/۱۹
 Hour is (hh:mm:ss): ۴۲۱۶۵۶:۲۵۲۹۹۳۶۰:۱۵۱۷۹۶۱۶۰۰

۶. فرض کنید بخواهید از بانک ۱۰۰۰۰۰۰۰۰ ریال وام با بهره ۱۵٪ با مدت بازپرداخت ۱۸ ماهه دریافت کنید، بهره وام به صورت زیر محاسبه می‌شود:

$$\text{بهره وام} = \frac{10000000 \times 18 \times 0.15}{12} = \frac{\text{نرخ بهره} \times \text{تعداد اقساط} + \text{مبلغ پرداختی وام}}{12}$$

مبلغ بهره (۱۲۲۵۰۰۰۰۰) ریال به متقاضی پرداخت می‌گردد. حال چنانچه متقاضی ۱۰۰۰۰۰۰۰۰۰ ریال نیاز داشته باشد، چقدر وام باید به او پرداخت شود. برنامه‌ای که مبلغ مورد نیاز متقاضی، تعداد اقساط و مبلغ بهره را دریافت می‌کند، سپس وامی که باید به متقاضی پرداخت شود و قسط هر ماه را چاپ می‌کند. (توضیح: مبلغ وام پرداختنی به صورت زیر محاسبه می‌گردد).

$$\text{بهره نرخ درصد} \times \text{تعداد اقساط} \times \text{مبلغ وام پرداختنی} - \text{مبلغ وام پرداختی} = \text{مبلغ متقاضی} = \text{مبلغ}$$

مبلغ متقاضی وام - مبلغ وام پرداختی

$$\text{مبلغ متقاضی} \times 12 = \text{درصد نرخ بهره} \times \text{تعداد اقساط} \times \text{مبلغ وام} - \text{مبلغ وام} \times 12$$

$$\text{مبلغ متقاضی} \times 12 = \frac{\text{نرخ بهره} \times \text{تعداد اقساط} \times \text{مبلغ کل وام}}{12}$$

تذکره: برنامه تا هر زمان که کاربر بخواهد ادامه می‌یابد.

متغیر	هدف
m	مبلغ مورد نیاز متقاضی
s	نرخ بهره
k	کل وام پرداختی
tedad	تعداد اقساط
p	پرداختی هر قسط
ansi	شرط ادامه

```

package ch2_6;
import java.util.*;
public class Ch2_6 {
    public static void main(String[] args) {
        long m, s, k, tedad, p;
        String ansi;
        for (;;)
        {
            System.out.print("Enter m:");
            m = (new Scanner(System.in)).nextLong();
            System.out.print("Enter tedad:");
            tedad = (new Scanner(System.in)).nextLong();
            System.out.print("Enter s:");
            s = (new Scanner(System.in)).nextLong();
            k = 12 * m / (12 - tedad * s / 100);
            p = k / tedad;
            System.out.printf("k = %1$s\tp = %2$s" + "\n", k, p);
            System.out.print("Do you want to continue(y/n):");
            ansi = new Scanner(System.in).nextLine();
            ansi = ansi.toLowerCase();
            if (ansi.compareTo("n") == 0)
            {
                break;
            }
        }
    }
}
    
```

خروجی برنامه:

Enter m: ۸
 Enter tedad: ۷
 Enter s: ۱۲

```
k = ۸ p = ۱
Do you want to continue(y/n):y
Enter m:۸
Enter tedad:۱۰
Enter s:۱
k = ۸ p = ۰
Do you want to continue(y/n):n
```

۷. برنامه‌ای که عدد اعشاری را از ورودی خوانده، هر یک از قسمت‌های صحیح و اعشاری آن را به صورت یک عدد صحیح به خروجی می‌برد. به عنوان مثال، ۱۳/۴۲ به صورت دو عدد صحیح ۱۳ و ۴۲ به خروجی برود.

متغیر	هدف
num	عدد خوانده شده
num۱	قسمت صحیح
num۲	قسمت اعشاری
temp	متغیر کمکی
i	شمارنده
j	شمارنده
k	شمارنده

```
package ch2_7;
import java.util.*;
public class Ch2_7 {
    public static void main(String[] args) {
        double num, temp, num2 = 0, num1;
        long i, k, j = 0;
        System.out.print("Enter a number:");
        num = (new Scanner(System.in)).nextDouble();
        num1 = Math.floor(num);
        long num11 = (long) num;
        System.out.printf("Number one is %1$s " + "\n", num11);
        temp = num;
        i = 0;
        while (temp != j)
        {
            temp = temp * 10;
            num2 += (((long) Math.floor(temp) % 10) *
                Math.pow(10, i));
            j = (long) temp;
            ++i;
        }
        num2 = Math.pow(10, i) * (num - num1);
        System.out.printf("Number two is %1$s "+" \n", (long) num2);
    }
}
```

خروجی برنامه:
Enter a number:۱۲۳۲,۶۶۶
Number one is ۱۲۳۲
Number two is ۶۶۶

۸. برنامه‌ای که سه مقدار را از ورودی خوانده، آن‌ها را به ترتیب صعودی در خروجی چاپ می‌کند.

```
package ch2_8;
import java.util.*;
public class Ch2_8 {
    public static void main(String[] args) {
        int a, b, c, temp;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b:");
        b = (new Scanner(System.in)).nextInt();
        System.out.print("Enter c:");
        c = (new Scanner(System.in)).nextInt();
        if (a > b)
```

متغیر	هدف
a, b, c	اعداد خوانده شده
temp	متغیر کمکی

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۲۳

```

{
    temp = a;
    a = b;
    b = temp;
}
if (a > c)
{
    temp = a;
    a = c;
    c = temp;
}
if (b > c)
{
    temp = b;
    b = c;
    c = temp;
}
System.out.printf("Sorted is %1$s %2$s %3$s "+ "\n", a, b, c);
}
}

```

خروجی برنامه:

```

Enter a: ۱۲
Enter b: ۴
Enter c: ۸
Sorted is ۴ ۸ ۱۲

```

➤ برنامه‌ای که سه مقدار را از ورودی خوانده، آن‌ها را به ترتیب صعودی در خروجی چاپ می‌کند.

➤ برنامه‌ای که شماره کارمندی، ساعت کارکرد و دستمزد ساعتی کارکنان موسسه‌ای را خوانده، حقوق آن‌ها را محاسبه می‌کند. اگر کارمندی بیش از ۴۰ ساعت کار کرده باشد، اضافه‌کار به او تعلق می‌گیرد. به ازای هر ساعت کاری، $\frac{3}{2}$ دستمزد ساعتی به‌عنوان اضافه‌کاری پرداخت می‌شود.

➤ یکی از مسائل مهم شرکت‌ها و ادارات پیش‌بینی بودجه چند سال بعد است. برنامه‌ای که قیمت کالا (ارزش پروژه) و نرخ تورم کالا را گرفته، قیمت کالا را برای چند سال بعد پیش‌بینی می‌نماید (برنامه تعداد سال‌هایی که قیمت کالا باید پیش‌بینی شود، از کاربر دریافت می‌نماید).

➤ برنامه‌ای که مجموع ده جمله اول سری زیر را محاسبه و چاپ می‌کند.

$$s = \frac{1}{x} - \frac{1}{x + 2x^2} + \frac{1}{x + 2x^2 + 3x^3} - \frac{1}{x + 2x^2 + 3x^3 + 4x^4} + \dots$$

➤ برنامه‌ای که مجموعه‌ای از اعداد دوتایی صحیح را خوانده، حاصل ضرب آن‌ها را بدون استفاده از عملگر ضرب نمایش می‌دهد. چنانچه کاربر به جای دو عدد، صفر وارد کند، برنامه خاتمه یابد.

- برنامه‌ای که عددی را خوانده، مشخص می‌کند آیا عدد متقارن است یا خیر. نمونه‌های از اعداد متقارن عبارت‌اند از: ۱۲۴۲۱ و ۳۲۴۲۳. (توضیح: چنانچه عددی برابر با مغلوبش باشد، متقارن است. پس، ابتدا مغلوب عدد را محاسبه کرده، با خودش مقایسه می‌کنیم.)
- برنامه‌ای که m و n دو عدد صحیح و مثبت را خوانده، با استفاده از عملگر $+$ ، m^n را محاسبه و چاپ می‌نماید.
- برنامه‌ای که تعدادی عدد را خوانده، هر یک از اعداد را که بر ۹ بخش پذیر باشد، در خروجی چاپ می‌کند (از روش مجموع ارقام استفاده کنید).
- برنامه‌ای که اطلاعات تعدادی کارمند از قبیل شماره کارمندی، سال استخدام، کد جنسیت، (۰: زن، ۱: مرد) پایه استخدام (۹-۱) و حقوق را خوانده، موارد زیر را در خروجی چاپ می‌کند.
 - الف. تعداد کارمندان مرد
 - ب. تعداد کارکنان زن
 - ج. میانگین حقوق به تفکیک زن و مرد
- فروشگاه‌های را در نظر بگیرید که درصدی از فروش به فروشنده اختصاص می‌دهد. میزان درصد به صورت زیر محاسبه می‌گردد:
 - کالای درجه ۱: اگر میزان فروش کم‌تر یا مساوی ۱۰۰۰۰۰۰ باشد، ۶ درصد به فروشنده تخصیص می‌یابد. اگر میزان فروش ۱۰۰۰۰۰۰ تا ۲۰۰۰۰۰۰ باشد ۷٪ و گرنه ۱۰٪ به فروشنده اختصاص می‌یابد.
 - کالای درجه ۲: اگر میزان فروش کم‌تر از ۱۰۰۰۰۰ باشد، ۴٪ و گرنه ۶٪ به فروشنده اختصاص می‌یابد.
 - کالای درجه ۳: در این نوع کالا، ۴۱٪ میزان فروش به فروشنده تخصیص می‌یابد.
 - کالای درجه ۴: در این نوع کالا، ۵٪ میزان فروش به فروشنده اختصاص می‌یابد.
- برنامه‌ای که تعداد اقلام فروش را دریافت می‌کند. سپس به تعداد خوانده شده درجه کالا، قیمت واحد کالا، تعداد فروش را می‌خواند و مجموع فروش و مبلغی را چاپ می‌نماید که باید به فروشنده تخصیص یابد.
- برنامه‌ای که خروجی زیر را ایجاد می‌کند.

```
file:///D:/BookCSharp/newCSharpPractic/2/p18/p18/bin/Debug/p18.EXE
7 6 5 4 3 2 1
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

- بازی سنگ، کاغذ و قیچی دو بازیکن دارد. بازیکن اول می‌تواند قیچی (مقدار ۱)، سنگ (مقدار ۲) و یا کاغذ (مقدار ۳) را انتخاب کند و بازیکن دوم همین‌طور. اگر انتخاب دو

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۲۵

بازیکن یکسان باشد، مساوی‌اند. اگر متفاوت باشد، یک نفر برنده است و برنده یک امتیاز را کسب می‌کند. برای مثال:

اگر بازیکن اول قیچی و بازیکن دوم کاغذ را انتخاب کند، بازیکن اول برنده است. چون قیچی کاغذ را می‌برد.

اگر بازیکن اول قیچی و بازیکن دوم سنگ را انتخاب کند، بازیکن دوم برنده است. زیرا سنگ قیچی را می‌شکند.

اگر بازیکن اول سنگ و بازیکن دوم کاغذ را انتخاب کند، بازیکن دوم برنده است. زیرا کاغذ دور سنگ می‌پیچد.

برنامه‌ای که این مسابقه را ۵ دور بین دو بازیکن انجام می‌دهد و برنده را نمایش می‌دهد (بازیکن‌ها از انتخاب هم دیگر خبر ندارند).

➤ برنامه‌ای که تمام اعداد بین ۱۰۰۰ تا ۱۱۰۰ که بر ۹ بخش‌پذیرند را نمایش می‌دهد (عددی بر ۹ بخش‌پذیر است که مجموع ارقام آن بر ۹ قابل قسمت باشد).

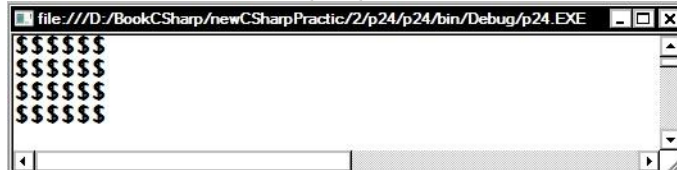
➤ برنامه‌ای که کاراکتری را که نشان‌دهنده رنگی می‌باشد، از ورودی خوانده، می‌گوید چه رنگی را انتخاب کردید. مثلاً اگر 'r' یا R را وارد کردید، برنامه به شما می‌گوید دوست دارید رنگ قرمز (Red) را انتخاب کنید. برای خروج کاربر باید e یا E را وارد کنید.

➤ برنامه‌ای که دو عدد مثبت را از ورودی خوانده، آنگاه عدد بزرگ‌تر را به روش تفریق بر عدد کوچک‌تر تقسیم می‌کند.

➤ برنامه‌ای که n را می‌خواند و حاصل عبارت زیر را محاسبه می‌کند:

$$\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

➤ برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را ایجاد می‌کند:



➤ برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را چاپ می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p25/p25/bin/Debug/p25.EXE
1 1 1 1 1 1 1
2 2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5
6 6 6
7 7
8
```

برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را چاپ می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p26/p26/bin/Debug/p26.EXE
1 1 1 1 1 1 1 1
2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4 4
5 5 5 5 5 5 5
6 6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8
```

برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p27/p27/bin/Debug/p27.EXE
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8 8
```

برنامه‌ای که القاب افراد را از ورودی به صورت یک کاراکتر خوانده، سپس لقب آن را به صورت کامل در خروجی چاپ می‌کند (بر اساس جدول زیر):

علامت ورودی	لقب	لقب لاتین
b, B	بانو	lady
d, D	دوشیزه	miss
p, P	پروفسور	professor
a, A	آقا	mr
j, J	جناب	excellency
m, M	خانم	wife

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۲۷

➤ قیمت اتومبیل در انتهای هر سال ۲۰٪ کاهش می‌یابد (استهلاک اتومبیل ۲۰٪ است). برنامه‌ای که قیمت یک اتومبیل را از ورودی خوانده، تا ده سال آینده قیمت آن را در پایان هر سال نمایش می‌دهد.

➤ برنامه‌ای که عددی را از ورودی خوانده، تشخیص می‌دهد که این عدد جزء سری فیبوناچی است یا خیر؟

➤ برنامه‌ای که تمام حالت‌هایی که یک اسکناس ۱۰۰۰ ریالی را با ۲۰۰، ۵۰۰، ۱۰۰ و ۵۰ ریالی می‌توان خرد نمود می‌نویسد.

➤ برای رانندگان میزان مصرف سوخت خودرویشان از اهمیت ویژه‌ای برخوردار است. راننده‌ای چند بار مقدار بنزین باک خودروی خود را به همراه مقدار مسافتی که رانندگی کرده، ثبت نموده است. برنامه‌ای که مسافت را به مایل و مصرف بنزین را به گالن از کاربر می‌گیرد و نسبت مایل به گالن را برای هر بار محاسبه می‌کند و نمایش می‌دهد. برنامه پس از پردازش همه اطلاعات نسبت مایل به گالن کل را محاسبه نموده، چاپ می‌کند. برای خروج از برنامه به جای گالن مقدار ۰ را وارد کنید.

➤ برنامه‌ای که عدد N را از ورودی خوانده، خروجی زیر را چاپ می‌کند:

```

file:///D:/BookCSharp/newCSharpPractic/2/p33/p33/bin/Debug/p33.EXE
Enter n:10
I      10*I    100*I   1000*I
1       10      100     1000
2       20      200     2000
3       30      300     3000
4       40      400     4000
5       50      500     5000
6       60      600     6000
7       70      700     7000
8       80      800     8000
9       90      900     9000
10      100     1000    10000
    
```

➤ برنامه‌ای که دنباله‌ای از اعداد صحیح را باهم جمع می‌کند. فرض کنید که اولین عدد صحیحی که خوانده می‌شود، تعداد اعدادی را که باید وارد شوند، مشخص می‌کند. به‌عنوان مثال، ورودی به‌صورت زیر می‌باشد:

➤ ۸۰۰ ۲۷۰ ۲۰۰ ۱۷۰ ۱۵۰ ۱۰۰ ۶

➤ برنامه میانگین و مجموع اعداد ۱۰۰، ۱۵۰، ۱۷۰، ۲۰۰، ۲۷۰ و ۸۰۰ را نمایش می‌دهد.

➤ برنامه‌ای که مقدار سری π (سری زیر) را حداکثر تا n جمله محاسبه می‌کند و نمایش می‌دهد (n را از ورودی می‌خواند):

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

➤ برنامه جدولی چاپ می‌کند که حاصل مقدار π را با تقریب یک جمله، دو جمله، سه جمله، ...، n جمله نمایش می‌دهد.

➤ برنامه‌ای که سه ضلع (عدد صحیح غیر صفر) را می‌خواند و تشخیص می‌دهد آیا سه ضلع اضلاع یک مثلث هستند یا خیر. (توضیح: سه ضلع زمانی تشکیل مثلث را می‌دهند که مجموع هر دو ضلع بیش‌تر از ضلع سوم باشد.)

➤ برنامه‌ای که سه عدد صحیح غیر صفر را گرفته، تشخیص دهد این سه عدد تشکیل مثلث قائم‌الزاویه را می‌دهند یا خیر؟ زمانی که توان ۲ یکی از اضلاع برابر مجموع توان ۲ اضلاع دیگر باشد مثلث قائم‌الزاویه است. (توضیح: هدف از این برنامه کاربرد عملگر $\|$ (یا منطقی) می‌باشد.)

➤ نمره دانشجویان یک کلاس به صورت زیر محاسبه می‌شود:
در طول ترم دو کوئیز برگزار می‌گردد که هر کدام ۱۰ نمره دارند.
امتحان پایان ترم ۵۰ درصد نمره کل، امتحان میان ترم ۲۵ درصد نمره کل و ۲۵ درصد دیگر مربوط به کوئیزها است.

برنامه‌ای که نمرات کوئیز میان ترم و پایان ترم یک دانشجو را گرفته، نمره کل او را محاسبه می‌کند و به صورت حروفی نمایش می‌دهد (برای نمرات بزرگ‌تر یا مساوی ۹۰، حروف A، نمرات بزرگ‌تر یا مساوی ۸۰ و کم‌تر از ۹۰ حرف B، نمرات بزرگ‌تر یا مساوی ۷۰ و کم‌تر از ۸۰ حرف C، نمرات بزرگ‌تر یا مساوی ۶۰ و کم‌تر از ۶۰، حرف D و نمرات زیر ۶۰ حرف E را نمایش دهد). (توضیح: در این برنامه، ابتدا نمرات کوئیز را به ۱۰۰ تبدیل می‌کنیم. یعنی، نمره کوئیز اول + نمره کوئیز دوم را در ۵ ضرب می‌کنیم. سپس برای محاسبه نمره کل، نمره به دست آمده از کوئیز را در ۰,۲۵ ضرب کرده، با ۰,۲۵ نمره میان ترم جمع نمود. حاصل را با ۰,۵ پایان ترم جمع می‌نماییم تا نمره کل به ۱۰۰ به دست آید.)

➤ برنامه‌ای که تعیین می‌کند در یک فروشگاه آیا میزان خرید مشتری از سقف اعتبار حسابش بیش‌تر شده است یا خیر. برای هر مشتری اطلاعات زیر موجود می‌باشد:

۱. شماره حساب (عدد صحیح) ۲. تراز اول ماه

۳. مجموع همه اقلامی که در یک ماه توسط این مشتری خریداری شده است.

۴. مجموع همه اعتباراتی که در یک ماه به حساب مشتری وارد شده است.

۵. سقف اعتبار مجاز

این برنامه باید هر یک از اطلاعات ۱ تا ۵ را بگیرد و تراز جدید (= اعتبارات - هزینه‌ها + تراز اولیه) را محاسبه و تعیین کند که آیا تراز جدید از سقف اعتباراتش بیش‌تر شده است. برنامه باید شماره حساب، سقف اعتبارات و تراز مشتری را به همراه پیام 'Credit Limited Exceeded' (از سقف اعتبارات تجاوز کرده است) چاپ کند.

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، مجموع عبارت را نیز نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۲۹

$$1 + 2 + 3 + \dots + n = \text{مجموع}$$

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$1 - 2 + 3 - 4 + \dots \pm n = \text{حاصل عبارت}$$

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$-1 + 2 - 3 + 4 + \dots \pm n = \text{حاصل عبارت}$$

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$1 - \frac{2}{2!} + \frac{3}{3!} - \frac{4}{4!} + \dots \pm \frac{n}{n!} = \text{حاصل عبارت}$$

➤ برنامه‌ای که x و n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \dots \pm \frac{x^n}{n!} = \text{حاصل عبارت}$$

➤ برنامه‌ای که x و n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$\frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \dots \pm \frac{x^n}{n!} = \text{حاصل عبارت}$$

➤ برنامه‌ای که یک عدد، یک کاراکتر و سپس عدد دیگری را خوانده، اعمال زیر را انجام دهد:

اگر کاراکتر ورودی "A"، "a" یا "+" باشد، حاصل جمع دو عدد را نمایش دهد.

اگر کاراکتر ورودی "S"، "s" یا "-" باشد، حاصل تفریق دو عدد را نمایش دهد.

اگر کاراکتر ورودی "X"، "x" یا "*" باشد، حاصل ضرب دو عدد را نمایش دهد.

اگر کاراکتر ورودی "D"، "d" یا "/" باشد، حاصل تقسیم دو عدد را نمایش دهد.

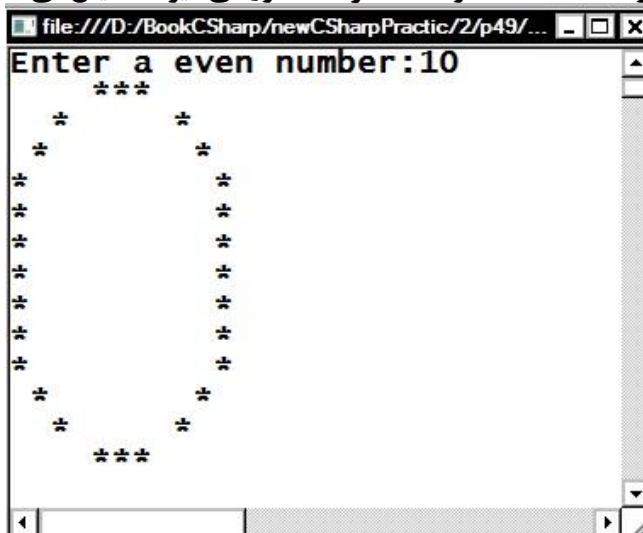
اگر کاراکتر ورودی "M"، "m" یا "%" باشد، باقی‌مانده تقسیم صحیح عدد اول بر عدد دوم را نمایش دهد.

در غیر این صورت، پیغام "Invalid operator" را نمایش دهد.

➤ برنامه قبلی را با دستور switch پیاده‌سازی می‌کند.

➤ برنامه‌ای که دو عدد را خوانده، اگر باقی‌مانده تقسیم هر عدد بر عدد دیگر صفر شود، "Yes" وگرنه، "No" را نمایش دهد.

➤ برنامه‌ای که عدد فرد n را خوانده، خروجی زیر را نمایش می‌دهد:



➤ برنامه‌ای که کاراکتر به همراه کد اسکی تمام کاراکترهای رایانه را نمایش می‌دهد (متغیر i شمارنده‌ای که صفر تا ۲۵۵ را می‌شمارد. کد اسکی کاراکترهای رایانه از صفر تا ۲۵۵ می‌باشد).

➤ برنامه‌ای که خروجی زیر را تولید کرده، نمایش می‌دهد.

1									
2	4								
3	6	9							
4	8	12	16						
5	10	15	20	25					
6	12	18	24	30	36				
7	14	21	28	35	42	49			
8	16	24	32	40	48	56	64		
9	18	27	36	45	54	63	72	81	
10	20	30	40	50	60	70	80	90	100

➤ برنامه‌ای که تعدادی عددی را دریافت کرده، معادل ماه آن‌ها را نمایش می‌دهد (اگر عدد وارد شده خارج از بازه ۱ تا ۱۲ باشد، برنامه خاتمه می‌یابد). در این برنامه month عددی است که ماه معادل آن باید تعیین شود.

➤ برنامه‌ای که رشته‌ای را دریافت کرده، تعداد کاراکترهای رشته را می‌شمارد (رشته به Enter ختم می‌شود).

➤ برنامه‌ای که حقوق تعدادی کارمند را خوانده، اگر کارمند بیش از ۴۰ ساعت کار کرده باشد، به اندازه ۱,۵ برابر هر ساعت به او اضافه کاری داده می‌شود. در پایان، کل حقوق

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۳۱

دریافتی را نمایش می‌دهد (برای خروج از برنامه از کاربر سؤال می‌کند که آیا ادامه می‌دهد یا خیر؟ اگر کاربر صفر را وارد کند، برنامه خاتمه می‌یابد).

➤ برنامه‌ای که نتایج عملگرهای (AND) &&، (OR) ||، (NOT) ! را به عدد تبدیل کرده، نمایش دهد.

➤ برنامه‌ای که عددی را خوانده، در هر مرحله عدد موردنظر، رقم یکان و حاصل ضرب ارقام فرد آن را نمایش می‌دهد. به‌عنوان مثال، اگر کاربر عدد ۳۲۷۶ را وارد نماید خروجی به‌صورت زیر نمایش داده می‌شود:

Number	Digit	Multiply
3276	6	1
327	7	7
32	2	7
3	3	21
Multiply is		21

➤ برنامه‌ای که عددی را خوانده، میانگین ارقام فرد و زوج آن را نمایش می‌دهد. به‌عنوان مثال، اگر عدد ۲۵۶۹ باشد، میانگین ارقام فرد آن، یعنی $(۲+۹+۵)/۲$ است. میانگین ارقام زوج آن $(۲+۶)/۲$ می‌باشد.

➤ برنامه‌ای که کلیه اعداد سه‌رقمی را نمایش می‌دهد که با ارقام ۱، ۲ و ۳ تولید می‌شوند.

➤ برنامه‌ای که کلیه اعداد سه‌رقمی را نشان می‌دهد که در ارقام آن ۱، ۴ یا ۸ باشد.

➤ برنامه‌ای که کلیه اعداد چهاررقمی که فقط از ارقام ۱، ۲، ۴ و ۹ تشکیل شده باشند و هر رقم یک‌بار تکرار شده باشد را نمایش می‌دهد.

➤ برنامه‌ای که یک جمله را خوانده، تعداد ارقام ۰ تا ۹ آن را می‌شمارد (جمله با $Ctrl + z$ یعنی EOF خاتمه می‌یابد).

➤ برنامه‌ای که خروجی زیر را نمایش می‌دهد:

2	4	6	8	10	12
6	8	10	12	14	16
10	12	14	16	18	20
14	16	18	20	22	24

➤ برنامه‌ای که خروجی زیر را نمایش می‌دهد:



➤ برنامه‌ای که x و n را خوانده، حاصل عبارت $\sum_{i=1}^n x^i$ را نمایش می‌دهد (یعنی، حاصل عبارت زیر را نمایش می‌دهد):

$$x^1 + x^2 + x^3 + \dots + x^n$$

➤ برنامه‌ای که x ، n و k را خوانده، حاصل عبارت $F(x, n, k) = \frac{\sum_{j=n}^u x^j}{\sum_{i=1}^k i!}$ را نمایش می‌دهد.

➤ برنامه‌ای که n (عددی بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

➤ برنامه‌ای که n (عددی بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۳۳

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مانند زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p68/p68/bin/Debug/p68.EXE
Enter n:8
*****
*       *
*     *   *
*   *     *
* *       *
* *     *
* *   *
* * *
*****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را از ورودی خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، اضلاع و قطر فرعی را نمایش می‌-

```
file:///D:/BookCSharp/newCSharpPractic/2/p69/p69/bin/Debug/p69.EXE
Enter n:9
*****
*       *
*     *   *
*   *     *
* *       *
* *     *
* *   *
* * *
*****
```

دهد

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p70/p70/bin/Debug/p70.EXE
Enter n:10
*****
*       *
*     *   *
*   *     *
* *       *
* *     *
* *   *
* * *
*****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p71/p71/bin/Debug/p71.EXE
Enter n:7
*
* *
* * *
* * *
* * *
* * *
*****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مانند شکل زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p72/p72/bin/Debug/p72.EXE
Enter n:8
*****
*           *
 *         *
  *       *
   *     *
    *   *
     * *
      *
  
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده و خروجی مانند زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p73/p73/bin/Debug/p73.EXE
Enter n:9
*****
*           *
 *         *
  *       *
   *     *
    *   *
     * *
      *
 *     *
 *   *
 * *
 *
  
```


➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر آخر، ستون آخر و قطر فرعی را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p74/p74/bin/Debug/p74.EXE
Enter n:10
*
 *
  *
   *
    *
     *
      *
       *
        *
 *     *
 *   *
 * *
 *
 *     *
 *   *
 * *
 *
 *     *
 *   *
 * *
 *
 *     *
 *   *
 * *
 *
  
```

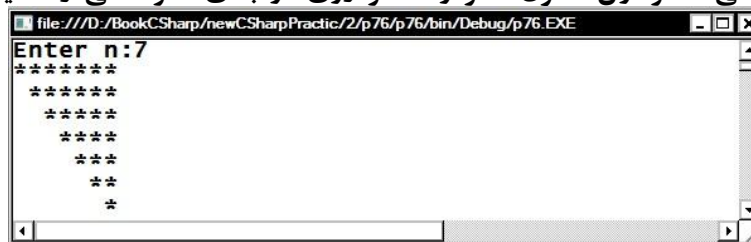
حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۳۵

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر آخر، ستون اول و تمام عناصر روی قطر اصلی و زیر آن را نمایش می‌دهد.



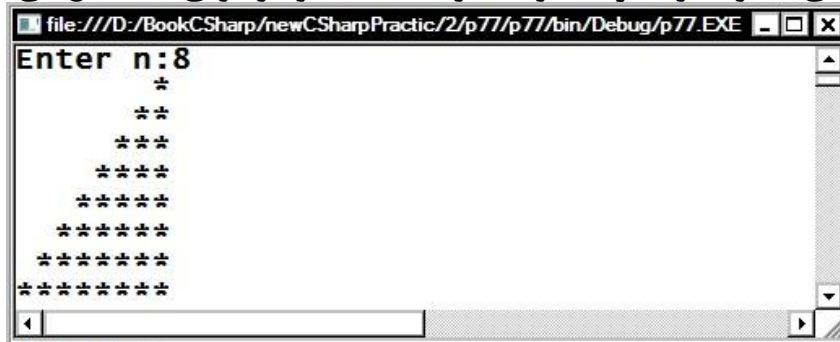
```
file:///D:/BookCSharp/newCSharpPractic/2/p75/p75/bin/Debug/p75.EXE
Enter n:10
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر اول، ستون آخر و عناصر روی و بالای قطر اصلی را نمایش می‌دهد.



```
file:///D:/BookCSharp/newCSharpPractic/2/p76/p76/bin/Debug/p76.EXE
Enter n:7
*****
*****
*****
****
***
**
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر آخر، ستون آخر، عناصر روی و زیر قطر فرعی را نمایش می‌دهد.



```
file:///D:/BookCSharp/newCSharpPractic/2/p77/p77/bin/Debug/p77.EXE
Enter n:8
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

- برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر اول، ستون اول، عناصر روی و بالای قطر فرعی را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p78/p78/bin/Debug/p78.EXE
Enter n:8
*****
*****
*****
*****
****
***
**
*
    
```

- برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p79/p79/bin/Debug/p79.EXE
Enter n:8
*      *
* *    * *
* * *  * * *
* * * * *
* * * * *
* * *  * * *
* *    * *
*      *
    
```

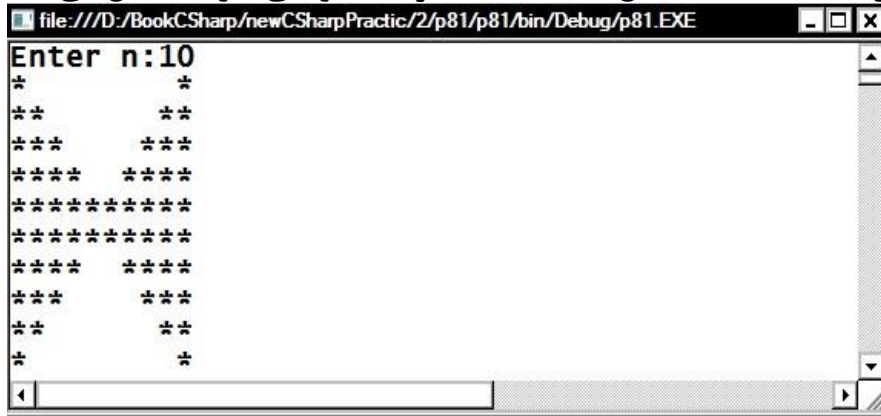
- برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p80/p80/bin/Debug/p80.EXE
Enter n:9
*****
*      *
* *    * *
* * *  * * *
* * * * *
* * * * *
* * *  * * *
* *    * *
*      *
*
    
```

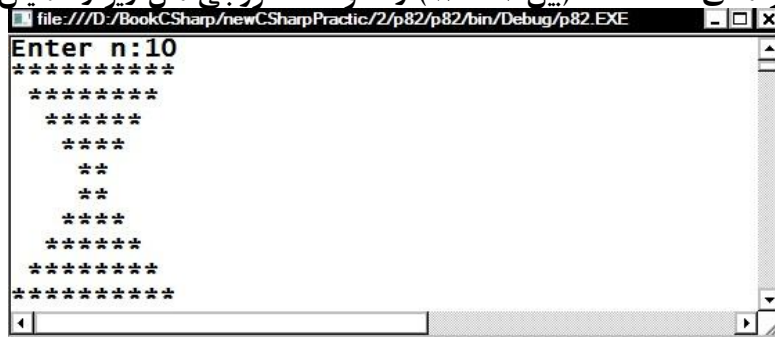
حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۳۷

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.



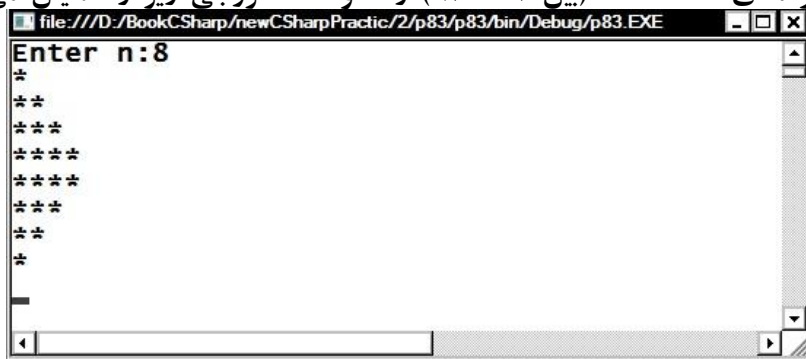
```
file:///D:/BookCSharp/newCSharpPractic/2/p81/p81/bin/Debug/p81.EXE
Enter n:10
*
**
***
****
*****
*****
****
***
**
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.



```
file:///D:/BookCSharp/newCSharpPractic/2/p82/p82/bin/Debug/p82.EXE
Enter n:10
*****
**
  **
   **
    **
   **
  **
 *****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.



```
file:///D:/BookCSharp/newCSharpPractic/2/p83/p83/bin/Debug/p83.EXE
Enter n:8
*
**
***
****
***
**
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p84/p84/bin/Debug/p84.EXE
Enter n:10
*
 **
 ***
 ****
 *****
 **
 ***
 **
 *
    
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p85/p85/bin/Debug/p85.EXE
Enter n:9
*****
*****
*****
***
**
*
    
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```

file:///D:/BookCSharp/newCSharpPractic/2/p86/p86/bin/Debug/p86.EXE
Enter n:9
*
 **
 ***
 ****
 *****
    
```

➤ برنامه‌ای که حقوق کارمند، نوع کسور بیمه و درصد مالیات را خوانده، حقوق خالص کارمند را حساب می‌نماید. برای محاسبه حقوق خالص به صورت زیر عمل می‌گردد

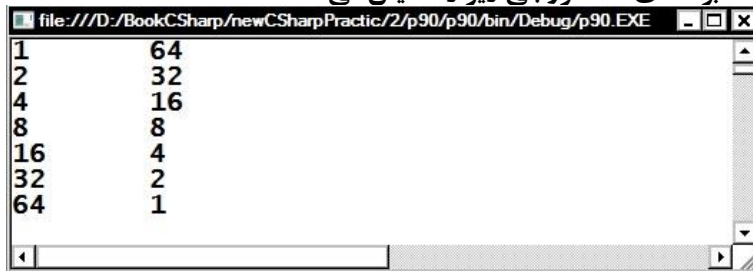
- (برای حل این تمرین از عملگر ؟ استفاده کنید):
- اگر نوع کسور بیمه ۱ باشد، $۱۰۰ / ۲ * \text{حقوق} = \text{بیمه}$
- اگر نوع کسور بیمه ۲ باشد، $۱۰۰ / ۸,۵ * \text{حقوق} = \text{بیمه}$
- اگر نوع کسور بیمه ۳ باشد، $۱۰۰ / ۱۰ * \text{حقوق} = \text{بیمه}$
- $۱۰۰ / \text{درصد مالیات} * \text{حقوق} = \text{مالیات}$
- $\text{مالیات} - \text{بیمه} - \text{حقوق} = \text{حقوق خالص}$

➤ برنامه‌ای که دو عدد صحیح x و y را خوانده، با استفاده از عملگر ضرب، x^y را محاسبه می‌کند و نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۳۹

- برنامه‌ای که تعداد عدد را خوانده و تشخیص می‌دهد هر عدد خوانده شده دارای رقم صفر است یا خیر. برای خروج از برنامه کاربر باید ۱- را وارد کنید.

➤ برنامه‌ای که خروجی زیر را نمایش می‌دهد:



```
file:///D:/BookCSharp/newCSharpPractic/2/p90/p90/bin/Debug/p90.EXE
1      64
2      32
4      16
8       8
16     4
32     2
64     1
```

- برنامه‌ای که شماره خانوار، هزینه روزانه آن‌ها را به روز می‌خواند، هزینه ماهیانه و سالیانه آن‌ها را نمایش می‌دهد. سپس تعیین می‌کند کدام خانوار کم‌ترین هزینه و کدام خانوار بیش‌ترین هزینه را دارد. برای خروج از برنامه به جای شماره خانوار عدد ۹۹- وارد می‌شود.

- برنامه‌ای که عدد n را خوانده (n تعداد کارمندان است). سپس ساعات اضافه کار n کارمند را خوانده و سه کارمند که کم‌ترین اضافه کار را دارند، نمایش می‌دهد.

- برنامه‌ای که سن و جنسیت تعدادی افراد را خوانده، تعداد و درصد مردانی که سن آن‌ها بین ۱۸ تا ۲۰ است را محاسبه می‌کند و نمایش می‌دهد. اگر جنسیت M یا m وارد گردید، این فرد، مرد است (برای خاتمه برنامه به جای سن ۱- وارد می‌شود).

- یک شرکت بیمه به بیمه‌گذاران خود سود سالانه می‌دهد. برنامه‌ای که میزان سود سالانه (درصد سود سالانه) و میزان سرمایه‌گذاری یک بیمه‌گذار و تعداد سال‌های سرمایه‌گذاری را دریافت می‌کند و سود بیمه‌گذار را برای n سال مختلف حساب می‌نماید (بدون پرداخت سود به بیمه‌گذار یعنی سود به سرمایه سالانه اضافه گردد). سرمایه بیمه‌گذار پس از n سال چقدر است.

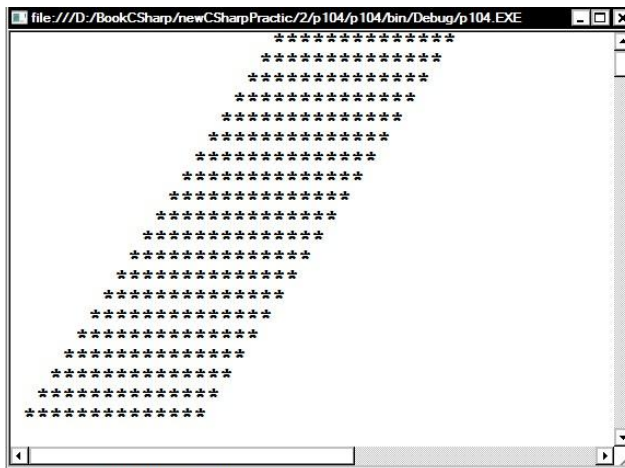
- برنامه‌ای که تمام اعداد چهاررقمی را نمایش می‌دهد که حاصل ضرب ارقام آن‌ها عدد تام (کامل) است. عددی تام است که مجموع مضرب‌های اعداد کوچک‌تر از خودش برابر خودش باشد. به عنوان مثال، دو عدد زیر تام هستند:

$$۶ = ۱ + ۲ + ۳ \qquad ۲۸ = ۱ + ۲ + ۴ + ۷ + ۱۴$$

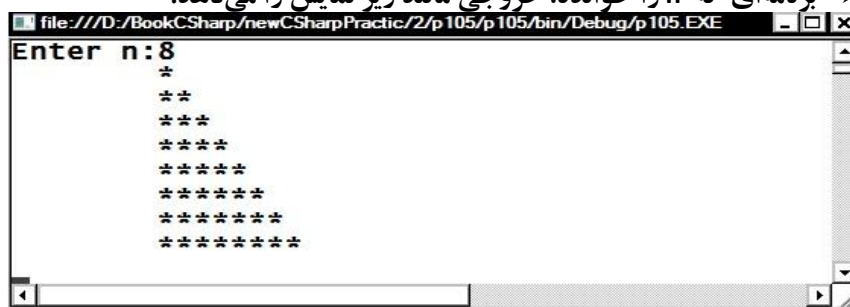
- برنامه‌ای که تعداد رقم‌های صفر اعداد چهاررقمی را می‌شمارد.

- برنامه‌ای که سه عدد را خوانده، اعداد بین اعداد اول و دوم که مجموع ارقام آن‌ها برابر عدد سوم باشد را نمایش می‌دهد. به‌عنوان مثال، اگر کاربر ۵، ۱۰۰۰ و ۱۵ را وارد کند، برنامه باید اعداد از ۵ تا ۱۰۰۰ که مجموع ارقام آن‌ها برابر ۱۵ باشد را نمایش دهد.
- برنامه‌ای که دو عدد را خوانده، اعداد اولی که بین این دو عدد قرار دارند را نمایش می‌دهد (عددی اول است که بر هیچ عدد کوچک‌تر از نصف خودش به جزء یک بخش پذیر نباشد).
- برنامه‌ای که یک عدد صحیح را خوانده و ارقامی از آن عدد که بر ۲ یا ۳ بخش پذیر هستند را نمایش می‌دهد.
- هنگام خرید خانه عوامل مختلفی را باید در نظر گرفت. در هنگام خرید خانه، میزان هزینه تقریبی سوختی که برای یک سال مصرف می‌شود و نرخ مالیات سالانه داده‌شده‌اند. برنامه‌ای که داده‌هایی از قبیل قیمت خانه، هزینه سوخت سالانه و مالیات سالانه را دریافت می‌کند. این برنامه هزینه ۱۰ ساله خانه را محاسبه می‌نماید. برنامه تا زمانی که کاربر به جای قیمت خانه ۹۹۹- وارد کند، ادامه می‌یابد.
- برنامه‌ای که عدد n را خوانده، مقدار $\sqrt{n + \sqrt{n + \sqrt{n + \dots}}}$ را تا دقت 10^{-10} محاسبه می‌کند و نمایش می‌دهد.
- در یک بازی دونفره n چوب‌کبریت وجود دارد. هر یک از بازیکنان می‌توانند در نوبت خود، ۱، ۲ یا حداکثر ۳ چوب‌کبریت را بردارند. بازیکنی که آخرین چوب‌کبریت را برمی‌دارد، بازنده است. برنامه‌ای که این بازی را بین دو بازیکن اجرا می‌کند. این برنامه، ابتدا تعداد چوب‌کبریت‌ها را خوانده و در هر مرحله تعداد چوب‌کبریت‌های باقی‌مانده را نمایش می‌دهد.
- برنامه‌ای که اگر کاربر هر یک از حروف A، B، C و D را وارد کند، پیام Big Character! را نمایش می‌دهد. اگر کاربر یکی از حروف a، b، c یا d را وارد کند، Litter character! را نمایش خواهد داد. ولی اگر کاربر یکی از کاراکترهای ۱، ۲، ۳ یا ۴ را وارد نماید، پیام A Figurer! را نمایش می‌دهد. وگرنه پیام Unknown Character! را نمایش می‌دهد (هدف این برنامه or کردن مقادیر در دستور switch است).
- برنامه‌ای که خروجی زیر نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۴۱



برنامه‌ای که n را خوانده، خروجی مانند زیر نمایش را می‌دهد:



➤ ۱۰۶. برنامه‌ای که سه ضلع مثلث را خوانده و مشخص می‌کند که آیا مثلث متساوی‌الاضلاع P، متساوی‌الساقین است یا مختلف‌الاضلاع می‌باشد (در این برنامه متغیرهای a، b و c اضلاع مثلث را نگه‌داری می‌کنند).

➤ ۱۰۷. برنامه‌ای که سه ضلع مثلث را دریافت کرده، اگر مثلث قائم‌الزاویه بود، مساحت آن، وگرنه محیط آن را محاسبه کرده و نمایش می‌دهد. در این برنامه متغیرهای a، b و c اضلاع مثلث را نگه‌داری می‌کنند، اما متغیر temp کمکی است که محیط یا مساحت را ذخیره می‌نماید.

➤ ۱۰۸. برنامه‌ای که کاراکتری را خوانده و مشخص می‌کند که در کدام یک از دسته‌های زیر قرار دارد؟
 حروف بی‌صدا حروف صدادار ارقام سایر کاراکترها

➤ ۱۰۹. برنامه‌ای که دو عدد صحیح را خوانده و مقسوم‌علیه مشترکشان را نمایش می‌دهد. در این برنامه x و y دو عدد هستند و a شمارنده از ۱ تا x می‌باشد.

➤ ۱۱۰. برنامه‌ای که ابتدا n را خوانده، سپس n معدل را برای n دانشجو می‌خواند و بزرگ‌ترین معدل مشروط را نمایش می‌دهد. در این برنامه n ، تعداد دانشجویان، grade، معدل هر دانشجو و \max بزرگ‌ترین معدل مشروط شده است.

➤ ۱۱۱. برنامه‌ای که معدل n دانشجو را خوانده، میانگین معدل‌های بالاتر از ۱۵ را نمایش می‌دهد. در این برنامه n (تعداد دانشجویان)، grade (معدل هر دانشجو)، sum (مجموع معدل‌های بالای ۱۵)، count (تعداد معدل‌های بالای ۱۵) و i (شمارنده) است.

➤ ۱۱۲. برنامه‌ای که حاصل سری زیر را تا ۴ رقم اعشار محاسبه کرده، نمایش می‌دهد:

$$+\frac{1}{2}-\left(\frac{1}{2}\times\frac{1}{3}\right)+\left(\frac{1}{2}\times\frac{1}{3}\times\frac{1}{4}\right)-\dots$$

➤ ۱۱۳. برنامه‌ای که ابتدا تعداد دانشجویان (n) را خوانده، سپس n معدل را می‌خواند و تعداد دانشجویهایی که معدل آن‌ها بین ۱۰ تا ۱۵ است را شمارش می‌کند و نمایش می‌دهد. در این برنامه n (تعداد دانشجویان)، grade (معدل هر دانشجو)، i (شمارنده) و count (تعداد دانشجویانی که معدل آن‌ها بین ۱۰ تا ۱۵) است.

➤ ۱۱۴. برنامه‌ای که ابتدا n را خوانده، سپس n جمله سری زیر را نمایش می‌دهد. در این سری زیر جمله‌ی سوم به بعد هر جمله میانگین دو جمله قبلی می‌باشد.

$$1, 5, 3, 4, 3, 5, \dots$$

➤ ۱۱۵. برنامه‌ای که ابتدا n را خوانده، سپس n عدد را می‌خواند و واریانس آن را نمایش می‌دهد.

➤ پروژه برنامه‌نویسی ۱: برنامه‌ای که روزی از سال را خوانده، مشخص می‌کند که چه فصلی، چه ماه و چه روزی از سال است. به‌عنوان مثال، روز ۳۵ سال، فصل بهار، ماه اردیبهشت و روز چهارم این ماه است.

➤ پروژه برنامه‌نویسی ۲: برنامه‌ای که x و n را خوانده، حاصل سری زیر را محاسبه می‌کند.

$$\frac{x^n n^x (x+n)^{n+x}}{n! x! (n+x)!} \sum_{i=1}^n \frac{x^i i^x}{n! i! (n+i)!}$$

➤ پروژه برنامه‌نویسی ۳: شرکت مخابرات ایران برای مکالمات راه دور از نرخ‌های زیر استفاده می‌کند:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۴۳

تمام مکالماتی که بین ساعات ۲۳ شب تا ۸ صبح انجام می‌شود، نرخ مکالمات ۵۰٪ محاسبه می‌گردد.

برای مکالماتی که در روزهای تعطیل (روز ۷) انجام می‌شود، نرخ مکالمه ۷۵ درصد محاسبه می‌شود.

برای بقیه ساعات روزهای دیگر مکالمه به‌طور کامل حساب می‌شود.

نرخ هر پالس ۴۶ ریال می‌باشد.

به کلیه مکالمات ۴ درصد مالیات تعلق می‌گیرد.

برنامه‌ای که برای تعدادی مشترک، ساعت شروع مکالمه (از ساعت صفر تا ۲۳)، تعداد پالس-های مکالمه، روز انجام مکالمه را دریافت می‌کند، مبلغ ناخالص مکالمه، مالیات، مبلغ خالص، مجموع مالیات و مجموع مبلغ خاص را محاسبه و نمایش می‌دهد (برای خاتمه کاربر به‌جای ساعت شروع ۹۹- را وارد نماید).

➤ پروژه برنامه‌نویسی ۴: نرخ هر مترمکعب آب با توجه به نرخ مصرف از قبیل مصارف خانگی، تجاری یا صنعتی فرق می‌کند. برنامه‌ای که میزان مصرف آب به مترمکعب و نوع مصرف تعدادی مشترک را خوانده، مبلغ صورت‌حساب آب آن‌ها را محاسبه و چاپ می‌کند. برای محاسبه مصرف صورت‌حساب آب به روش زیر عمل می‌نماید:

اگر نوع مصرف حرف H یا h باشد (مصرف خانگی)، به ازای هر ۱۰۰ مترمکعب مصرف ۵۰۰ ریال دریافت می‌شود.

اگر نوع مصرف حرف I یا i وارد شود (مصرف صنعتی تا چهار میلیون مترمکعب)، به ازای هر ۱۰۰۰ مترمکعب آب ۷۵۰ ریال و به ازای هر مترمکعب بیش از آن ۰/۰۰۰۲۵ به مبلغ قبض اضافه می‌گردد.

اگر نوع مصرف حرف E یا e وارد شود (مصرف تجاری)، در صورتی که مصرف آب تا دو میلیون مترمکعب باشد، به ازای هر ۱۵۰۰ مترمکعب مصرف ۶۰۰ ریال و به ازای هر مترمکعب بیش از دو میلیون ۰/۰۰۰۰۴ به مبلغ صورت‌حساب اضافه خواهد شد. برای خاتمه برنامه کاربر به‌جای مصرف آب، عدد منفی وارد می‌نماید.

فصل

۳

متد در جاوا

۱. برنامه‌ای که **n** جمله سری فیبوناچی را به روش بازگشتی تولید می‌کند. وظایف متدها:
 + متد **main**: این متد متغیرهای برنامه را تعریف کرده، متد **fibonacci** را فراخوانی می‌کند.
 + متد **fibonacci**: این متد **i**امین عدد فیبوناچی را تولید می‌کند. **i** را به‌عنوان پارامتر می‌پذیرد.

```
package ch3_1;
import java.util.*;
public class Ch3_1 {
    private static int fibo(int n)
    {
        if (n == 1)
        {
            return (1);
        }
        if (n == 2)
        {
            return (1);
        }
        return (fibo(n - 2) + fibo(n - 1));
    }
    public static void main(String[] args) {
        int i, n;
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextInt();
        System.out.print("Result is:");
        for (i = 1; i <= n; i++)
        {
            System.out.printf("%1$s    ", fibo(i));
        }
    }
}
```

خروجی برنامه:

Enter n:۱۰

Result is:۱ ۱ ۲ ۳ ۵ ۸ ۱۳ ۲۱ ۳۴ ۵۵

۲. برنامه‌ای که فاصله‌ای را برحسب فوت و اینچ دریافت کرده، معادل آن را برحسب متر و سانتی‌متر بیان می‌کند. هر فوت ۰/۳۰۴۸ متر، یک متر ۱۰۰ سانتی‌متر و هر فوت ۱۲ اینچ است. حداقل از سه متد استفاده کنید. یکی برای ورودی، یکی برای انجام محاسبات و یکی برای خروجی. وظایف متدها:

+ متد **main**: این متد، متدها **read**، **write** و **calculate** را فراخوانی می‌کند.
 + متد **read**: این متد فاصله را به فوت و اینچ می‌خواند.
 + متد **calculate**: این متد، هر یک از فواصل برحسب فوت و اینچ را به متر و سانتی‌متر تبدیل می‌کند.
 + متد **write**: این متد برای نمایش فواصل تبدیل‌شده به کار می‌رود.

متد	متغیر	هدف
	foot	ورودی به فوت
	Inch	ورودی به اینچ
	foot_to_meter	فوت به متر
	inch_to_meter	اینچ به متر

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۴۵

سانتیمتر	centi_meter	عمومی
فوت به سانتیمتر	foot_to_centimeter	
اینچ به سانتیمتر	inch_to_centimeter	

```

package ch3_3;
import java.util.*;
public class Ch3_3 {
    private static int foot;
    private static int inch;
    private static float foot_to_meter;
    private static float foot_to_centimeter;
    private static float inch_to_centimeter;
    private static float inch_to_meter;
    private static void read()
    {
        System.out.print("Foot?");
        foot = (new Scanner(System.in)).nextInt();
        System.out.print("Inch?");
        inch = (new Scanner(System.in)).nextInt();
    }
    private static void calculate()
    {
        foot_to_meter = 0.3048f * foot;
        foot_to_centimeter = 100 * foot_to_meter;
        inch_to_meter = (float)(1.0f / 12) * 0.3048f * inch;
        inch_to_centimeter = 100 * inch_to_meter;
    }
    private static void write()
    {
        System.out.printf("The %1$s foot is %2$s meter" + "\r\n", foot,
            foot_to_meter);
        System.out.printf("The %1$s foot is %2$s centiMeter" + "\r\n",
            foot, foot_to_centimeter);
        System.out.printf("The %1$s inch is %2$s meter" + "\r\n", inch,
            inch_to_meter);
        System.out.printf("The %1$s inch is %1$s centiMeter" + "\r\n",
            inch, inch_to_centimeter);
    }
    public static void main(String[] args) {
        read();
        calculate();
        write();
    }
}

```

خروجی برنامه:

```

Foot?۱۰۰
Inch?۲۰۰
The ۱۰۰ foot is ۳۰,۴۸ meter
The ۱۰۰ foot is ۳۰۴۸,۰ centiMeter
The ۲۰۰ , inch is ۰ meter
The ۲۰۰ , inch is ۰۰۸,۰ centiMeter

```

۳. نیروی جاذبه بین دو جسم به صورت زیر محاسبه می‌گردد:

$$f = \frac{g * m1 * m2}{d^2}$$

m_1 جرم جسم اول، m_2 جرم جسم دوم، d فاصله بین دو جسم و g ثابت جهانی جاذبه با مقدار $cm(g,sec) 6,673 \times 10^{-8}$ است. برنامه‌ای که با استفاده از دو متد مقدار نیروی جاذبه محاسبه می‌کند.

متد	متغیر	هدف
سراسری	g	نیروی جاذبه جهانی
main	m_1	جرم جسم اول
	m_2	جرم جسم دوم
	d	فاصله بین دو جسم

```
package ch3_3;
import java.util.*;
public class Ch3_3 {
    private static final double g =
6.673e-8;
    private static double calc_f(double m1, double m2, double d)
    {
        return (g * m1 * m2 / pow(d));
    }
    private static double pow(double d)
    {
        return (d * d);
    }
    public static void main(String[] args) {
        double m1,m2,d;
        System.out.print("Enter m1:");
        m1 = (new Scanner(System.in)).nextDouble();
        System.out.print("Enter m2:");
        m2 = (new Scanner(System.in)).nextDouble();
        System.out.print("Enter d:");
        d = (new Scanner(System.in)).nextDouble();
        System.out.printf("F is %1$s", calc_f(m1, m2, d));
    }
}
```

خروجی برنامه:

```
Enter m1:۷
Enter m2:۸
Enter d:۸
F is ۰,۸۲۸۸۷۰E-۸
```

۴. یکی از مشکلات هواشناسی محاسبه شاخص سردی باد است. برای این منظور، فرمول زیر وجود دارد:

$$w = 33 - \frac{(10\sqrt{V} - V + 10.5)(33 - t)}{23 - 1}$$

V ، سرعت باید برحسب متر بر ثانیه، t دما برحسب درجه سانتی‌گراد، ($t \leq 10$) و W شاخص سردی باد است. برنامه‌ای که با استفاده از یک متد، سرعت باد و دما را می‌خواند، شاخص سردی باد را نمایش می‌دهد.

متد	متغیر	هدف
main	v	سرعت باد
	t	دما

```
package ch3_4;
import java.util.*;
public class Ch3_4 {
    private static double calc_cold(double v,
double t)
    {
        return (33 - ((10 * Math.sqrt(v) - v + 10.5) * (33 - t)) / 23.1);
    }
    public static void main(String[] args) {
        double v,t;
        System.out.print("Enter v:");
        v = (new Scanner(System.in)).nextDouble();
        System.out.print("Enter t:");
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۴۷

```
t = (new Scanner(System.in)).nextDouble();
System.out.printf("W is %1$s ", calc_cold(v, t));
}
```

خروجی برنامه:

Enter v: ۱۰۰

Enter t: ۲۰

W is ۲۷,۰۹۰۹۰۹۰۹۰۹۰۹۰۹۳

۵. برنامه‌ای که یک عدد و یک رقم را خوانده، با استفاده از متدی تعداد تکرار رقم در عدد را برمی‌گرداند و چاپ می‌کند.

متد	متغیر	هدف
main	n	عدد مورد نظر
	m	رقمی که باید در عدد جستجو گردد
repeat	count	تعداد تکرار رقم در عدد

```
package ch3_5;
import java.util.*;
public class Ch3_5 {
    private static int repeat(long n, int m)
    {
        int count = 0;
        while (n > 0)
        {
            if ((n % 10) == m)
            {
                count++;
            }
            n /= 10;
        }
        return (count);
    }
    public static void main(String[] args) {
        long n;
        int m;
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextLong();
        System.out.print("Enter m:");
        m = (new Scanner(System.in)).nextInt();
        System.out.printf("%1$s Repeat %2$s times", m, repeat(n, m));
    }
}
```

خروجی برنامه:

Enter n: ۵۶۵۴۳۵

Enter m: ۵

Repeat ۳ times

۶. برنامه‌ای که x و n را خوانده، مقدار سینوس زاویه x را محاسبه می‌کند.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots + \frac{x^n}{n!}$$

وظایف متدها:

متد main: این متد n و x را خوانده، با فراخوانی متدها **fact** و **pow** حاصل عبارت $\sin(x)$ را چاپ می‌کند.

متد fact: این فاکتوریل عددی را برمی‌گرداند.

متد pow: این متد توان n عددی را برمی‌گرداند.

متد	متغیر	هدف
	x	ورودی
	n	ورودی

مجموع	sum	main
شمارنده	i	
علامت (یکی در میان مثبت و منفی می شود)	sign	

```

package ch3_6;
import java.util.*;
public class Ch3_6 {
    private static double fact(int n)
    {
        if (n == 1)
        {
            return 1.0;
        }
        else
        {
            return (n * fact(n - 1));
        }
    }
    private static double pow(double x, int n)
    {
        if (n == 0)
        {
            return 1.0;
        }
        else
        {
            return (x * pow(x, n - 1));
        }
    }
    public static void main(String[] args) {
        int n, i, sign = 1;
        double x, sum = 0;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextDouble();
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextInt();
        for (i = 1; i <= n; i += 2)
        {
            sum += sign * pow(x, i) / fact(i);
            sign = -sign;
        }
        System.out.printf("Sin(%1$s) = %2$s", x, sum);
    }
}

```

خروجی برنامه:

```

Enter x:۲
Enter n:۱۸
Sin(۲,۰) = ۰,۹۰۹۲۹۷۴۲۶۸۲۹۹۵

```

۷. خروجی قطعه برنامه زیر چیست؟ (توضیح برنامه: دو متغیر x و y به صورت عمومی تعریف می‌شوند و مقدار آنها صفر است (متغیرهای x و y قبل از متد `main` تعریف شده‌اند). با اجرای متد `cout` مقدار $x = 0$ چاپ می‌گردد. سپس متد `f` فراخوانی می‌گردد. با فراخوانی متد `f` متغیر y به صورت محلی تعریف گردیده، مقدار اولیه 10 به آن تخصیص می‌یابد. سپس با اجرای `cout`، مقدار $x = 10$ چاپ خواهد شد و مقدار متغیر عمومی x به 10 تغییر می‌یابد (`x += 10`). با اجرای `cout`، مقدار $y = 10$ چاپ خواهد شد و سپس مقدار 11 در x قرار می‌گیرد (`x++`). پس از برگشت از متد `f` با اجرای `cout`، ابتدا یک واحد

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۴۹

به متغیر عمومی **y** اضافه می‌گردد، مقدار ۱ در **y** قرار می‌گیرد و مقدار **y=۱** چاپ می‌شود. با اجرای آخرین **cout** مقدار **x = ۱۱** چاپ می‌شود. سپس، یک واحد به متغیر **x** اضافه می‌گردد.

```
package ch3_7;
import java.util.*;
public class Ch3_7 {
    private static int x, y;
    private static void f1()
    {
        int y = 10;
        System.out.printf("y = %1$s" + "\r\n", y);
        x += 10;
        System.out.printf("x = %1$s" + "\r\n", x++);
    }
    public static void main(String[] args) {
        System.out.printf("x = %1$s" + "\r\n", x);
        f1();
        System.out.printf("y = %1$s" + "\r\n", ++y);
        System.out.printf("x = %1$s" + "\r\n", x++);
    }
}
```

خروجی برنامه:

```
x = ۰
y = ۱۰
x = ۱۰
y = ۱
x = ۱۱
```

۸. برنامه‌ای که اعداد سه‌رقمی را چاپ می‌کند که مجموع فاکتوریل ارقام آن اعداد برابر با خود عدد باشد. به‌عنوان مثال، اگر n_1 ، n_2 و n_3 به ترتیب رقم یکان، دهگان و صدگان باشند، فرمول زیر برقرار باشد:

$$n_3 n_2 n_1 = n_1! + n_2! + n_3!$$

وظایف متدها:

متد **main**: این متد، کلیه ارقام اعداد سه‌رقمی را جدا کرده، با فراخوانی متد **fact** مجموع ارقام آن را محاسبه می‌کند. چنانچه مجموع فاکتوریل ارقام هر عدد برابر با خود آن عدد باشد، چاپ می‌شود. متد **fact**: فاکتوریل **n** را به روش بازگشتی حساب می‌کند.

متد	متغیر	هدف
main	I	شمارنده
	n_1	رقم اول
	n_2	رقم دوم
	n_3	رقم سوم
	temp	متغیر کمکی
	sum	مجموع فاکتوریل ارقام
fact	n	عددی که باید فاکتوریل آن حساب گردد

```
package ch3_8;
import java.util.*;
public class Ch3_8 {
    private static long
    fact(int n)
    {
        if (n == 0)
        {
            return 1;
        }
        else
        {
            return (n * fact(n - 1));
        }
    }
}
```

```

    }
}
public static void main(String[] args) {
    int n1,n2,n3,temp;
    long sum;
    System.out.print("Result is:");
    for (int i = 100; i <= 999; i++)
    {
        n1 = i % 10;
        temp = i / 10;
        n2 = temp % 10;
        temp /= 10;
        n3 = temp % 10;
        sum = fact(n1) + fact(n2) + fact(n3);
        if (sum == i)
        {
            System.out.printf("    %1$s", i);
        }
    }
}
}
}

```

خروجی برنامه:

Result is: ۱۴۰

۹. برنامه‌ای که کلیه اعداد چهاررقمی چاپ را می‌کند که مجموع رقم اول به توان ۱ و رقم چهارم به توان ۴ برابر با مجموع رقم دوم به توان ۲ و رقم سوم به توان ۳ باشد. به‌عنوان مثال، داریم. $۲۱۴۱ = ۲^۴ + ۱ = ۱۶ + ۱ = ۱۷$

وظایف متدها:

متد **main**: این متد، کلیه ارقام اعداد چهاررقمی را جدا کرده، با فراخوانی **pow** توان ۴ رقم چهارم و توان ۳ رقم سوم را محاسبه می‌کند. چنانچه مجموع توان ۴ رقم چهارم و توان ۳ رقم اول برابر توان ۲ رقم دوم و توان ۳ رقم سوم باشد، عدد را نمایش می‌دهد. متد **pow**: این متد توان **n** عددی را برمی‌گرداند.

متد	متغیر	هدف
main	i	شمارنده
	n1	رقم اول
	n2	رقم دوم
	n3	رقم سوم
	n4	رقم چهارم
	temp	متغیرهای کمکی
pow	x	پایه که باید به توان n برسد
	n	توان

```

package ch3_9;
public class Ch3_9 {
private static long pow(int x,
int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return (x * pow(x, n - 1));
    }
}
}

```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۵۱

```
public static void main(String[] args) {
    int n1, n2, n3, n4, temp;
    System.out.println("Result is:");
    for (int i = 1000; i <= 9999; i++)
    {
        n1 = i % 10;
        temp = i / 10;
        n2 = temp % 10;
        temp /= 10;
        n3 = temp % 10;
        n4 = temp / 10;
        if ((n1 + pow(n4, 4)) == (n2 * n2 + pow(n3, 3)))
        {
            System.out.printf("%1$s\t", i);
        }
    }
}
```

خروجی برنامه:

```
Result is: ۱۰۱۰ ۱۰۲۳ ۱۰۳۸ ۱۱۰۰ ۱۱۱۱ ۱۱۲۴
۱۱۳۹ ۱۲۰۷ ۱۲۱۸ ۲۰۴۰ ۲۰۵۹ ۲۱۴۱ ۲۲۳۱ ۲۲۴۸
۳۰۹۰ ۳۱۹۱ ۳۲۹۸ ۳۴۵۸ ۴۶۷۹
```

۱۰. برنامه ای که دو عدد را خوانده، کوچکترین مضرب مشترک آن‌ها (LCM) را نمایش می‌دهد.

```
package ch3_10;
import java.util.Scanner;
public class Ch3_10 {
    private static int lcm(int x, int y)
    {
        int z, lcm;
        if (x > y)
            z = x;
        else
            z = y;
        while(true){
            if((z % x == 0) && (z % y == 0))
            {
                lcm = z;
                break;
            }
            z += 1;
        }
        return lcm;
    }
    public static void main(String[] args) {
        int x, y;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextInt();
        System.out.print("Enter y:");
        y = (new Scanner(System.in)).nextInt();
        System.out.printf("LCM is %1$s ", lcm(x, y));
    }
}
```

هدف	متغیر	
عدد صحیح اول	x	main
عدد صحیح دوم	y	
پارامتر اول	x	lcm
پارامتر دوم	y	

خروجی برنامه:

```
Enter x:۲۴
Enter y:۲۸
lcm is ۱۶۸
```

۱۱. برنامه‌ای که عدد صحیح مثبتی را خوانده، مجموع ارقام آن را محاسبه می‌کند. اگر حاصل مجموع ارقام، عدد یک‌رقمی نبوده، این روند را ادامه داده تا نهایتاً عدد یک‌رقمی شود (از یک متد بازگشتی استفاده می‌کند).
وظایف متدها:
متد **main**: این متد عددی را خوانده، با فراخوانی متد **sumdigit** حاصل را چاپ می‌کند.
متد **sumdigit**: این متد عددی را به‌عنوان آرگومان می‌پذیرد و عددی را برمی‌گرداند که حاصل مجموع ارقام آن، عدد یک‌رقمی باشد.

متد	متغیر	هدف
main	n	عدد ورودی
sumdigit	sum q	مجموع ارقام خارج‌قسمت عدد به ۱۰

```
package ch3_11;
import java.util.*;
public class Ch3_11 {
    private static long
    sumdigit(long n)
    {
        long sum,q;
        if (n < 10)
        {
            return n;
        }
        else
        {
            q = n;
            sum = 0;
            while (q > 0)
            {
                sum += q % 10;
                q /= 10;
            }
            return (sum);
        }
    }
    public static void main(String[] args) {
        long n;
        System.out.print("Enter a number:");
        n = (new Scanner(System.in)).nextLong();
        System.out.printf("Result is %1$s", sumdigit(n));
    }
}
```

خروجی برنامه:
Enter a number:۱۰۱
Result is ۲

۱۲. برنامه‌ای که اضلاع مثلثی را خوانده، مساحت آن را با متدی محاسبه می‌کند و به خروجی می‌برد. اگر اضلاع مثلثی **a**، **b** و **c** باشد، مساحت آن با فرمول مقابل محاسبه می‌شود.
$$p = (a + b + c) / 2$$

$$\text{مساحت مثلث} = \sqrt{p(p-a)(p-b)(p-c)}$$

وظایف متدها:

متد **main**: این متد **a**، **b** و **c** را خوانده، مساحت مثلث را چاپ می‌کند.
متد **sum**: این متد مجموع اضلاع مثلث (محیط مثلث) را برمی‌گرداند.
متد **area**: این متد مساحت مثلث را با استفاده از فرمول بیان شده برمی‌گرداند.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۵۳

```
package ch2_12;
import java.util.*;
public class Ch2_12 {
    private static float sum(int a, int
b, int c)
    {
        return (a + b + c);
    }
    private static double area(double p, int a, int b, int c)
    {
        return (Math.sqrt(p * (p - a) * (p - b) * (p - c)));
    }
    public static void main(String[] args) {
        int a, b, c;
        double p;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b:");
        b = (new Scanner(System.in)).nextInt();
        System.out.print("Enter c:");
        c = (new Scanner(System.in)).nextInt();
        p = sum(a, b, c) / 2.0f;
        System.out.printf("Area is %1$s", area(p, a, b, c));
    }
}
```

متد	متغیر	هدف
main	a	ضلع اول
	b	ضلع دوم
	c	ضلع سوم
	p	محیط تقسیم بر ۲

خروجی برنامه:

```
Enter a:۱۰
Enter b:۱۲
Enter c:۱۱
Area is ۵۱,۵۲۱۲۳۳۴۸۱۶۷۸۱۶۷۸۴
```

۱۳. برنامه‌ای که زمان مصرف شده بخشی از اجرای برنامه را به ثانیه نمایش می‌دهد.

متد	متغیر	هدف
n	sum	عدد ورودی
sum_of_n_numbers	n	پارامتر
	s	مجموع زمان
	i	شمارنده
	start_time	زمان شروع
	end_time	زمان پایان

```
import time
def sum_of_n_numbers(n):
    start_time = time.time()
    s = 0
    for i in range(1,n+1):
        s = s + i
    end_time = time.time()
    return s,end_time-start_time
def main():
    n = int(input("Enter n:"))
    print("\nTime to sum of 1 to ",n," and required time to calculate
```

```
is
main()
خروجی برنامه:
Enter n: ۱۰۰۰۰
":sum_of_n_numbers(n))
```

Time to sum of ۱ to ۱۰۰۰۰ and required time to calculate is :
(۰۰۰۰۰۰۰۰, ۰,۰۰۰۹۹۹۹۲۷۰۲۰۷۵۱۹۵۳۱)

➤ برنامه‌ای که اعداد سه‌رقمی را چاپ می‌کند که مجموع فاکتوریل ارقام آن اعداد برابر با خود عدد باشد. به‌عنوان مثال، اگر n_1 ، n_2 و n_3 به ترتیب رقم یکان، دهگان و صدگان باشند، فرمول زیر برقرار باشد:

$$n^3 n^2 n^1 = n_1! + n_2! + n_3!$$

➤ برنامه‌ای که کلیه اعداد چهاررقمی چاپ را می‌کند که مجموع رقم اول به توان ۱ و رقم چهارم به توان ۴ برابر با مجموع رقم دوم به توان ۲ و رقم سوم به توان ۳ باشد. به‌عنوان مثال، داریم:

$$۲۱۴۱ = ۲^۴ + ۱^۱ = ۱۳ + ۴^۲ = ۱۶ + ۱ = ۱ + ۱۶ = ۱۷$$

➤ برنامه‌ای که دو عدد را خوانده، به متدی ارسال کند. متد، تفاضل حاصل ضرب و حاصل تقسیم آن‌ها را محاسبه کرده، به برنامه ارسال کند. متدها را به‌صورت قالب‌های متد پیاده‌سازی کنید و دو بار آن را برای اعداد صحیح و اعشاری فراخوانی نمایید. این متد با استفاده از Template پیاده‌سازی شود.

➤ برنامه‌ای که عدد صحیح مثبتی را خوانده، مجموع ارقام آن را محاسبه می‌کند. اگر حاصل مجموع ارقام، عدد یک‌رقمی نبود، این روند را ادامه داده تا نهایتاً عدد یک‌رقمی شود (از یک متد بازگشتی استفاده می‌کند).

➤ برنامه‌ای که اضلاع مثلثی را خوانده، مساحت آن را با متدی محاسبه می‌کند و به خروجی می‌برد. اگر اضلاع مثلثی a ، b و c باشد، مساحت آن با فرمول مقابل محاسبه می‌شود.

$$p = (a + b + c) / ۲$$

$$\sqrt{p(p-a)(p-b)(p-c)} = \text{مساحت مثلث}$$

➤ برنامه‌ای که یک عدد بزرگ (حتی ۱۰۰۰ رقمی) را گرفته، تعیین می‌کند بر ۱۵ بخش پذیر است یا خیر. عددی بر ۱۵ بخش پذیر است که بر ۳ و ۵ بخش پذیر باشد. عددی بر سه بخش پذیر است که مجموع ارقام آن بر سه بخش پذیر باشد و عددی بر ۵ بخش پذیر است که رقم یکان آن صفر یا ۵ باشد (رقم یکان بر ۵ بخش پذیر است).

➤ برنامه‌ای که یک رقم را از ورودی خوانده، تمام اعداد صحیح بین ۱ و ۱۰۰ را چاپ می‌کند، به‌طوری که رقم دریافتی در اعداد ۱ تا ۱۰۰، مجذور و مکعب این اعداد وجود داشته باشد. به‌عنوان مثال، رقم (۱) را از ورودی بخوانیم، عدد ۱۳ یکی از اعدادی است

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۵۵

- که رقم یک در آن شرط صدق می‌کند، زیرا، در ۱۳، ۱۶۹ و ۲۱۹۷ وجود دارد (برای تشخیص وجود رقم در عدد، مجذور و مکعب عدد از یک متد استفاده کنید).
- برنامه‌ای که سه عدد را از ورودی می‌خواند و توسط متدی میانگین آن‌ها را محاسبه می‌کند و نمایش می‌دهد.
 - برنامه‌ای که یک عدد را از ورودی می‌خواند و توسط یک متد بازگشتی، شمارش معکوس از آن عدد به یک را انجام دهد (شماره‌ها را در خروجی چاپ می‌کند).
 - متدی که دو آرگومان را می‌پذیرد و آرگومان اول را به توان آرگومان دوم می‌رساند. آرگومان اول یک مقدار double و آرگومان دوم یک مقدار صحیح مثبت یا منفی است. برنامه‌ای که از این متد استفاده می‌کند.
 - برنامه‌ای که نمایش زمان را از حالت ۲۴ ساعت به ۱۲ ساعت تبدیل می‌کند. مثلاً باید ۱۴:۲۵ را به صورت ۲:۲۵pm تبدیل نماید. این برنامه طوری طراحی شده است که دارای سه متد زیر می‌باشد.
 - قیمت سهام معمولاً به صورت کسری بیان می‌شود، مثلاً $2\frac{7}{8}$ یا $8\frac{1}{2}$. برنامه‌ای که ارزش سهام را به صورت دو عدد صحیح و کسری از کاربر می‌گیرد. قسمت کسری را نیز به صورت دو عدد صحیح (عدد اول صورت و عدد دوم مخرج) دریافت می‌نماید (بر اساس تبدیل سهام از یک متد استفاده می‌کند که سه پارامتر را گرفته، ارزش سهام را به صورت double برمی‌گرداند).
 - پارکینگی برای توقف تا سه ساعت، حداقل ۲ دلار و برای هر ساعت اضافه یا بخشی از ساعت اضافه بر ۳ ساعت ۰/۵ دلار می‌گیرد. حداکثر مبلغ قابل پرداخت برای هر دوره ۲۴ ساعتی ۱۰ دلار است. فرض کنید که هیچ خودرویی بیش از ۲۴ ساعت توقف نمی‌کند. برنامه‌ای که زمان استفاده از پارکینگ را برای مشتریان دریافت می‌کند و مبلغ پرداختی مشتری، مبلغ کل پرداختی را چاپ نماید. کاربر برای هر مشتری ساعات و دقیقه توقف را وارد می‌کند. متدی مبلغ پرداختی را محاسبه و برمی‌گرداند.
 - برنامه‌ای که با استفاده از چند متد اعمال زیر را انجام دهید:
 - تمام اعداد دورقمی که رقم اول و دوم آن‌ها برابر باشند را چاپ کند.
 - تمام اعداد سه‌رقمی که رقم وسط آن‌ها صفر باشد را چاپ کند.
 - تمام اعداد سه‌رقمی که مجموع رقم اول و دوم کوچک‌تر از رقم سوم باشد.
 - تمام اعداد چهاررقمی که قرینه یکدیگرند (مثلاً ۲۳۳۲). یعنی رقم اول برابر رقم چهارم و رقم دوم برابر رقم سوم باشد.

➤ برنامه‌ای که با استفاده از یک متد بازگشتی بزرگ‌ترین مقسوم‌علیه مشترک دو عدد را محاسبه و چاپ می‌کند.

➤ برنامه‌ای که اولین روز سال را می‌گیرد و تقویم سالانه را نمایش می‌دهد.

➤ برنامه‌ای که یک عدد زوج را گرفته، تمام زوج‌های اولی (دو عدد فرد اولی که مجموع آن‌ها برابر با آن عدد زوج باشد) را نمایش می‌دهد. به‌عنوان مثال، اگر کاربر عدد ۲۲ را وارد کند، خروجی به‌صورت زیر نمایش داده شود.

$$۳ + ۱۹ = ۲۲ \quad ۵ + ۱۷ = ۲۲ \quad ۱۱ + ۱۱ = ۲۲$$

➤ یونانیان باستان اعداد را به‌صورت هندسی دسته‌بندی می‌کردند. به‌عنوان مثال، آن‌ها یک عدد را مثلثی می‌نامیدند که می‌توانستند با آن تعدادی ریگ (به‌اندازه عدد ریگ)، در یک تقارن مثلثی بچینند. متدی به نام isTrain که یک عدد را گرفته تشخیص می‌دهد، عدد مثلثی است یا خیر. اگر عدد مثلثی باشد، مقدار ۱، وگرنه مقدار صفر را برمی‌گرداند. دوازده عدد مثلثی در زیر آمده‌اند:

۰ ۱ ۳ ۶ ۱۰ ۱۵ ۲۱ ۲۸ ۳۶ ۴۵ ۵۵ ۶۶

➤ متدی به نام isSquare که تشخیص می‌دهد، یک عدد مربعی است یا خیر؟ چند عدد مربعی عبارت‌اند از:

۰ ۱ ۴ ۹ ۱۶ ۲۵ ۳۶ ۴۹ ۶۴ ۸۱ ۱۰۰ ۱۲۱ ...

➤ برنامه‌ای که x و y را خوانده، حاصل عبارت $\sqrt{|x-y|^{|y|}}$ را محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای که نمره‌ای را به عدد بین صفر تا ۱۰۰ خوانده، توسط متدی مقدار حرفی معادل آن را برمی‌گرداند و نمایش می‌دهد (اگر نمره بین ۸۰ تا ۱۰۰ باشد، حرف A، اگر نمره بین ۶۰ تا ۸۰ باشد، حرف B، چنانچه نمره بین ۵۰ تا ۶۰ باشد، حرف C، وگرنه حرف F را نمایش می‌دهد).

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۵۷

➤ متد main: متغیر با نوع int به نام grade تعریف کرده، سپس با یک پیام مناسب نمره را از ورودی خوانده، در متغیر grade قرار می‌دهد، در ادامه با پیام مناسب و فراخوانی متد selectGrade نمره حرفی را نمایش می‌دهد.

➤ متد selectGrade: نمره عددی را از طریق پارامتر grade دریافت می‌کند و نمره حرفی را برمی‌گرداند. چون نوع متد char انتخاب گردید، با استفاده از if‌های متداخل یکی از حروف A، B، C یا F را برمی‌گرداند (با توجه به مقدار پارامتر grade).

➤ برنامه‌ای که سن فردی را به سال، ماه و روز دریافت کرده، توسط متدهای به‌روز، ساعت، دقیقه و ثانیه تبدیل کرده، برمی‌گرداند و سپس نمایش می‌دهد (هرسال ۳۶۵٫۲۵ روز و هرماه ۳۰ روز است).

➤ برنامه‌ای که حقوق کارمند را خوانده، توسط متدی مالیات بر حقوق را محاسبه کرده، نمایش می‌دهد (مالیات از طریق جدول زیر محاسبه می‌گردد):

درصد مالیات	تا	از
0	483000	0
%10	600000	483001
%15	100000	600001
%20	2000000	1000001
%30	99999999	2000001

➤ برنامه‌ای که عددی را خوانده، با استفاده از متدی حاصل ضرب ارقام بالای ۵ آن را محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای که عددی را خوانده، با استفاده از یک متد حاصل ضرب ارقام زوج آن‌ها را برمی‌گرداند و نمایش می‌دهد.

➤ برنامه‌ای که n را خوانده، با استفاده از یک متد n بار ۱۰ تا x را نمایش می‌دهد.

➤ برنامه‌ای که نمره پایان ترم و میان ترم دانشجویی را گرفته، با استفاده از یک متد نمره نهایی دانشجو را محاسبه و برمی‌گرداند (نمره نهایی دانشجو برابر با نمره میان ترم * ۰٫۴ + نمره پایان ترم * ۰٫۶ است).

➤ برنامه‌ای که استفاده از متغیرهای محلی و سراسری را نمایش می‌دهد.

➤ برنامه‌ای که اطلاعات جعبه‌ای از قبیل طول، عرض و ارتفاع را به‌عنوان پارامتر دریافت کرده، با استفاده از متدی حجم جعبه را محاسبه کرده، نمایش می‌دهد (هدف این برنامه، آشنایی بیش‌تر با متدها با پارامتری با مقادیر پیش‌فرض است). حجم جعبه برابر با ارتفاع \times عرض \times طول است.

➤ برنامه‌ای که قیمت کالا و درصد تخفیف را دریافت کرده، به متدی ارسال می‌کند و این متد میزان تخفیف کالا را برگردانده و در برنامه میزان تخفیف را نمایش می‌دهد.

➤ برنامه‌ای که میزان موجودی حساب بانکی را به همراه درصد بهره سالانه دریافت کرده، تعیین می‌کند سپس از چند سال موجودی حساب (بدون برداشت هیچ مبلغی از حساب) به مبلغ خاصی می‌رسد (موجودی انتهای هر سال توسط متدی محاسبه می‌گردد).

➤ برنامه‌ای که کاراکتری را از ورودی خوانده، توسط متدهای تعیین می‌کند که کاراکتر خوانده‌شده، رقم، حرف بزرگ و حرف کوچک است یا خیر؟

➤ برنامه‌ای که عدد n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:
 $1! + 2! + \dots + n!$

➤ این برنامه متدی برای محاسبه فاکتوریل و متد دیگری برای محاسبه مجموع دارد.

➤ برنامه‌ای که n را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$\frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \dots + \frac{n}{n!}$$

➤ برنامه‌ای که x و n را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

➤ این برنامه متدهای برای محاسبه فاکتوریل، توان (x^n) و مجموع سری دارد.

➤ برنامه‌ای که x و n را خوانده، حاصل جمع n جمله سری زیر را نمایش می‌دهد:

$$\frac{x}{1} + \frac{x^2}{2} + \frac{x^3}{1*3} + \dots + \frac{x^5}{1*3*5}$$

➤ برنامه‌ای که x و n را خوانده، حاصل n جمله سری زیر را نمایش می‌دهد:

$$\frac{x}{1} - \frac{x^2}{2} + \frac{x^3}{1*3} - \frac{x^4}{2*4} + \frac{x^5}{1*3*5} + \dots$$

➤ برنامه‌ای که x و n را خوانده، حاصل $n/2$ جمله عبارت زیر را نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۵۹

$$-\frac{x^2}{2} + \frac{x^4}{2*4} - \frac{x^6}{2*4*6} + \dots$$

➤ برنامه‌ای که دو مقدار عددی را خوانده، سپس توسط متدی به عدد اول یک واحد اضافه نماید و از عدد دوم یک واحد کسر کرده، آن‌ها را در برنامه اصلی نمایش می‌دهد.

➤ برنامه‌ای که x و k را خوانده، لگاریتم x در مبنای k را با یک متد محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای سه عدد را خوانده توسط متدی یکی از مقادیر زیر را برگرداند:
 اگر سه عدد تشکیل مثلث را نمی‌دهند، صفر را برگرداند.
 اگر سه عدد تشکیل مثلث قائم‌الزاویه را می‌دهند، عدد یک را برمی‌گرداند.
 اگر سه عدد تشکیل مثلث متساوی‌الساقین را می‌دهند، عدد ۲ را برمی‌گرداند.
 اگر سه عدد تشکیل مثلث متساوی‌الاضلاع را می‌دهند، عدد ۳ را برمی‌گرداند.
 وگرنه عدد ۱- را برمی‌گرداند.

➤ برنامه‌ای که اعداد n و b را خوانده، عدد n را به روش بازگشتی از مبنای ۱۰ به مبنای b تبدیل می‌کند (دقت داشته باشید که b ، مبنای کوچک‌تر از ۱۰ است).

➤ برنامه‌ای که با روش بازگشتی مقدار مشتق $\frac{d(n)}{dn}(e^{ax})$ را محاسبه می‌کند:

توضیح: اگر از متد e^{ax} مشتق بگیریم، رابطه بازگشتی مقابل به دست می‌آید:
 $(e^{ax})^n = a(e^{ax})^{n-1}$

➤ برنامه‌ای که \sin ، \cos ، \tan زوایای مضرب پنج از ۵ تا ۹۰ را در خروجی با فرمت جدول‌بندی مناسب تا سه رقم اعشار نمایش می‌دهد. برای محاسبه زوایا باید از درجه به رادیان تبدیل شوند. برای این منظور از فرمول زیر استفاده می‌شود:

$$\frac{R}{3.1415} = \frac{D}{180} \Rightarrow R = \frac{3.1415 * D}{180}$$

➤ برنامه‌ای که ۱۵ جمله دنباله زیر را به کمک بازگشتی زیر نمایش می‌دهد:

$$F(1) = 2 \quad n == 1 \text{ اگر}$$

$$F(n) = 2 * F(n-1) + 1 \quad \text{وگرنه}$$

➤ برنامه‌ای که حاصل عبارت $f(x) = 5x^2 - 3x + 4$ را برای مقادیر صحیح و اعشاری x با دو متد هم نام محاسبه کرده و نمایش می‌دهد.

➤ برنامه‌ای که قاعده و ارتفاع مثلث را خوانده، با استفاده دو متد هم نام مساحت مثلث را حساب می‌کند. مساحت مثلث برابر با قاعده ضرب در نصف ارتفاع است. در ضمن ارتفاع و قاعده می‌توانند دو عدد صحیح یا دو عدد اعشاری باشند.

➤ برنامه‌ای که کاراکتری را خوانده، بدون استفاده از عملگر $+$ و if آن را به حروف کوچک تبدیل می‌کند و نمایش می‌دهد. این برنامه برای تبدیل حروف بزرگ و کوچک از یک متد استفاده می‌کند.

➤ برنامه‌ای که یک کاراکتر را خوانده، با استفاده از عملگر XOR (\wedge) اگر حرف کوچک بود، آن را به حروف بزرگ تبدیل کرده، یا اگر حرف بزرگ بود، آن را به حرف کوچک تبدیل نماید. برای این منظور، از یک متد استفاده نماید.

➤ برنامه‌ای که x و n را خوانده، حاصل متد $F(x, n) = x^{n!}$ را نمایش می‌دهد.

➤ برنامه‌ای که x و n را خوانده، حاصل عبارت مقابل را نمایش می‌دهد

$$F(x, n) = \sum_{k=1}^n \frac{x^n}{k!} :$$

➤ برنامه‌ای که دو عدد صفر یا ۱ را خوانده، نتایج عملگرهای XOR، OR، AND، NOR، NAND را نمایش می‌دهد. در این برنامه متدهای برای برگشت نتایج عملگرهای XOR، OR، AND، XOR، NAND پیاده‌سازی شوند.

➤ برنامه‌ای که n را خوانده، تمام اعداد تام (کامل) کوچک‌تر یا مساوی n که مضرب ۳ هم باشند را نمایش می‌دهد. (عددی تام است که مجموع مقسوم‌های کوچک‌تر از خودش برابر خودش باشد).

➤ متد بازگشتی که دو عدد x, y را خوانده، حاصل عبارت زیر را برمی‌گرداند:

$$F(x, y) = \begin{cases} x - y & \text{اگر } x \text{ یا } y \text{ کوچک‌تر از صفر باشد} \\ F(x - 1, y) + F(x, y - 1) & \text{وگرنه} \end{cases}$$

برنامه‌ای که از این متد بازگشتی استفاده می‌کند.

➤ متد بازگشتی n را به‌عنوان پارامتر دریافت کرده، تمام اعداد مضرب ۳ کوچک‌تر از n را نمایش می‌دهد. برنامه که از این متد استفاده می‌کند.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۶۱

- متد بازگشتی که عددی را به عنوان پارامتر دریافت می کند و تمام اعداد فرد بین این دو عدد را نمایش می دهد. برنامه ای که از این متد استفاده می کند.
- موسسه ای قصد دارد x درصد به حقوق کارمندانش اضافه کند. متدی که ضریب افزایش حقوق و حقوق فعلی کارمند را دریافت کرده، حقوق جدید را برمی گرداند. برنامه ای که برای n کارمند از این متد استفاده می کند.
- متدی که دو عدد را به عنوان پارامتر دریافت کرده، تعیین می کند چند عدد اول بین این دو عدد وجود دارند. این برنامه، متدی برای تعیین عدد اول دارد. برنامه ای که از این متدها استفاده می کند. “ عدد اول است که بر هیچ عدد کوچک تر از نصف خودش به جزء یک بخش پذیر نباشد.”
- متدی که تأثیر نیروی جاذبه بر اجسام در حال سقوط را محاسبه می کند. برنامه ای که زمان را دریافت کرده (برحسب ثانیه) و این تأثیر را محاسبه می نماید و چاپ می کند. این برنامه باید ارتفاع شیء سقوط کننده در هر ثانیه را نمایش می دهد. برای این منظور از فرمول های زیر استفاده می نماید:
زمان $t = ۹,۸۰۶۶۵$ ، فاصله $g = ۹,۸۰۶۶۵$ و ارتفاع $t^2 * g * 1/2 =$ فاصله
- برنامه ای که عدد n را خوانده، سپس یک عدد دیگر k را بخواند و حاصل ضرب اعداد کوچک تر یا مساوی عدد n که مضرب آن عدد k باشد را نمایش می دهد.
- اگر دو ضلع (B, C) از مثلث و زاویه بین این دو ضلع Alpha باشد، می توانیم طول ضلع سوم را با فرمول زیر محاسبه کنیم:
$$A^2 = B^2 + C^2 - 2 * B * C * \text{COS}(\text{Alpha})$$
- برنامه ای که B, C و Alpha را خوانده، با استفاده از متدی مقدار A را محاسبه می کند و نمایش می دهد.
- برنامه ای که دو عدد، سپس یک کاراکتر را خوانده، اگر کاراکتر R یا r باشد، مساحت مستطیل را با یک متد محاسبه می کند. چنانچه کاراکتر T یا t را وارد نماید، مساحت مثلث را با فرمول ارتفاع * نصف قاعده محاسبه می نماید. در نهایت مساحت مستطیل یا مثلث را نمایش می دهد.
- فرض کنید اساتید دارای حقوق کم تر از ۴۰۰۰۰۰۰ تومان ۰/۵ درصد اضافه حقوق می گیرند، ولی اساتید بین ۴۰۰۰۰۰۰ تا ۷۰۰۰۰۰۰ تومان ۱۲ درصد و اساتید با حقوق بیش تر ۱۰ درصد اضافه حقوق می گیرند. متدی که اضافه حقوق استاد را محاسبه می کند. برنامه ای که حقوق استادی را خوانده، اضافه حقوق و حقوق جدید استاد را نمایش می دهد.

➤ برنامه‌ای که n و m را خوانده، توسط متد بازگشتی C_n^m را برگرداند، نمایش می‌دهد:

$$C_n^m = \begin{cases} 1 & \text{اگر } m = n \text{ یا } m = 0 \\ n & \text{اگر } m = 1 \\ (n-1) + \binom{n-1}{m-1} & \text{وگرنه} \end{cases}$$

➤ برنامه‌ای که عددی را خوانده با استفاده از متد بازگشتی مغلوب آن را نمایش می‌دهد. به‌عنوان مثال، اگر عدد ۳۲۵۷ باشد، مغلوب آن یعنی ۷۵۲۳ را نمایش می‌دهد.

➤ برنامه‌ای که n و m را خوانده، توسط یک متد بازگشتی $A(m, n)$ را محاسبه می‌کند و نمایش می‌دهد $A(m, n)$ به‌صورت زیر محاسبه می‌شود.

$$A(m, n) = \begin{cases} n+1 & \text{اگر } m = 0 \\ A(m-1, 1) & \text{وگرنه } n = 0 \\ A(m-1, A(m, n-1)) & \text{وگرنه} \end{cases}$$

➤ برنامه‌ای که n و m را خوانده، توسط یک متد بازگشتی $Q(m, n)$ را محاسبه می‌کند و

$$Q(m, n) = \begin{cases} 1 & \text{وگرنه } n = 1 \text{ یا } m = 1 \text{ باشد} \\ Q(m, n) & \text{اگر } m < n \text{ باشد} \\ 1 + Q(m, m-1) & \text{اگر } m = n \text{ باشد} \\ Q(m, n-1) + Q(m-n, n) & \text{اگر } m > n \text{ باشد} \end{cases} \quad \text{نمایش می‌دهد:}$$

➤ برنامه‌ای که مشتق متد $1 + x^2$ را در نقطه x محاسبه می‌کند. در این برنامه به ازای h کوچک می‌توان نوشت:

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

➤ برنامه‌ای که مشتق دوم متد $1 + x \sin x$ در نقطه x با فرمول زیر محاسبه می‌کند.

$$f''(x_0) = \lim_{h \rightarrow 0} \frac{f'(x_0 + h) - f'(x_0)}{h} = \frac{f'(x_0 + h) - f'(x_0)}{h}$$

➤ برنامه‌ای که x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\left(1 + \frac{x^n}{1}\right)^1 + \left(1 + \frac{x^{(n-1)}}{2}\right)^2 + \dots + \left(1 + \frac{x^1}{n}\right)^n$$

➤ برنامه‌ای که سه عدد x ، n و a را دریافت کرده، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۶۳

$$(x+a)^n = \frac{n!}{0!(n-0)!} x^0 a^{n-0} + \frac{n!}{1!(n-1)!} x^1 a^{n-1} + \dots + \frac{n!}{n!(n-n)!} x^n a^{n-n}$$

➤ برنامه‌ای که اعداد صحیح x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\begin{aligned} & n! \frac{x^n}{1} + (n-1)! \frac{x^{(n-1)}}{1 - \frac{1}{x}} + (n+1)! \frac{x^{(n+1)}}{1 + \frac{1}{x}} \\ & + (n-2)! \frac{x^{(n-2)}}{2 - \frac{2}{x}} + (n+2)! \frac{x^{(n+2)}}{2 + \frac{2}{x}} + \dots + \\ & x^{(n-n)} \frac{(n-n)!}{n - \frac{n}{x}} + x^{(n+n)} \frac{(n+n)!}{n + \frac{n}{x}} \end{aligned}$$

➤ **پروژه برنامه‌نویسی ۱.** برنامه‌ای که تمامی جفت‌های اعداد متحابه از یک تا عدد خوانده شده را نمایش می‌دهد. دو عدد صحیح متحابه گویند، اگر مجموع مقسوم‌علیه‌های هر یک به جز خود عدد با عدد دوم برابر باشد. به‌عنوان مثال، مقسوم‌علیه‌های ۲۲۰ که از ۲۲۰ کوچک‌تر هستند، عبارت انداز: ۵۵، ۱۱۰، ۲۲، ۱۰، ۵، ۴، ۲، ۱ که مجموع آن‌ها عدد ۲۴۸ می‌شود و مقسوم‌علیه‌های ۲۴۸ که کوچک‌تر از ۲۴۸ هستند، برابر با ۱۴۲، ۷۱، ۴، ۲، ۱ که مجموع آن‌ها ۲۲۰ است. پس این دو عدد متحابه می‌باشند.

➤ **پروژه برنامه‌نویسی ۲.** برنامه‌ای که مقداری را از یک مبنا دریافت کرده، به مبنا دیگری تبدیل می‌کند (مبناها می‌توانند بین ۲ تا ۱۶ باشند). این پروژه مبنا می‌باشد، عددی در آن مبنا (چنانچه کاربر مقدار را خارج از مبنا وارد کند، جلوی او را می‌گیرد) و مبنا مقصد را از کاربر دریافت می‌کند.

➤ **پروژه برنامه‌نویسی ۳.** برنامه‌ای که ابتدا منوی زیر را نشان می‌دهد:

Calculate one
Calculate two
Calculate three
Exit
SELECT (۱ to ۴):

اگر کاربر عدد ۱ را وارد کند، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$\frac{x^n (n)^5 (n)^{x+2} \sum_{i=1}^n x^i i^x x^n}{x! n! (x+n)! \sum_{i=1}^n x! i! (n+2)!}$$

اگر کاربر عدد ۲ را وارد کند، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \pm \frac{x^n}{n!}$$

اگر کاربر عدد ۳ را وارد کند، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$\frac{((x)^n)^3 (n)^{x+3}}{n! + (x+n+2)!} \sum_{i=2}^x \frac{x^i i^x n^x}{(x+i)! + (n+x)!} \quad i = 2, 4, \dots$$

اگر کاربر عدد ۴ را وارد کند، برنامه خاتمه یابد.

➤ پروژه برنامه نویسی ۴. فرض کنید که سیستم را برای یک بنگاه معاملات ملکی می نویسد. در این سیستم قرار است، منوی با ۵ گزینه نمایش داده شود تا کاربر با انتخاب یکی از گزینه ها شکل زمین فروخته شده را انتخاب کند. زمین ها به شکل های مربع، مستطیل، مثلث و دایره هستند. پس از انتخاب شکل زمین توسط کاربر، قیمت هر متر زمین و درصد حق سهم فروشنده دریافت می گردد و در ادامه با توجه به نوع انتخاب شده کاربر، اطلاعات زمین دریافت می گردد و مبلغی که خریدار باید پرداخت کند را نمایش می دهد. این برنامه تا زمانی ادامه می یابد که کاربر در منوی ظاهر شده گزینه ۵ را انتخاب نماید.

۱. برنامه‌ای که عددی صحیح را از ورودی خوانده، تمام اعداد اول قبل از آن را با استفاده از تعریف زیر تعیین کرده، به خروجی می‌برد. عددی اول است که بر هیچ عدد اول قبل از خودش قابل قسمت نباشد. وظایف متدها

متد **main**: متغیر برنامه را تعریف کرده، متد **prim** را فراخوانی می‌کند.

متد **prim**: تشخیص می‌دهد عددی اول است یا خیر؟ اگر عدد اول باشد، یک را برمی‌گرداند.

متد	متغیر	هدف
main	n	عدد خوانده شده
	p	آرایه‌ای از اعداد اول
	i	شمارنده
	tedad	تعداد اعداد اول
prim	i	شمارنده

```

package ch4_1;
import java.util.*;
public class Ch4_1 {
    private static int prim(int[] a,
int tedad, int num)
    {
        int i;
        for (i = 0; i < tedad; i++)
        {
            if (num % a[i] == 0)
            {
                return (0);
            }
        }
        a[tedad] = num;
        return (1);
    }
    public static void main(String[] args) {
        int n, i, tedad = 0;
        int[] p = new int[300];
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextInt();
        for (i = 2; i <= n; i++)
        {
            if (prim(p, tedad, i) == 1)
            {
                tedad++;
            }
        }
        System.out.print("Primary is ");
        for (i = 0; i < tedad; i++)
        {
            System.out.printf(" %1$s", p[i]);
        }
    }
}

```

خروجی برنامه:

Enter n:۲۰

Primary is ۲ ۳ ۵ ۷ ۱۱ ۱۳ ۱۷ ۱۹ ۲۳ ۲۹

۲. برنامه‌ای که شماره دانشجویی تعدادی از دانشجویان را از ورودی خوانده، در آرایه قرار می‌دهد. سپس عناصر آرایه را به روش انتخابی مرتب می‌کند. مرتب‌سازی آرایه به روش انتخابی به این صورت انجام می‌شود: کوچکترین عنصر آرایه پیدا شده، جای آن با عنصر اول آرایه عوض می‌شود. در مرحله بعد بقیه عناصر آرایه برای یافتن کوچکترین عنصر آرایه جستجو می‌شود و جای آن با عنصر دوم آرایه عوض می‌شود. این روند تا مرتب‌سازی کامل آرایه ادامه می‌یابد. پس از مرتب‌سازی نتیجه را در خروجی چاپ می‌کند. این برنامه سه متد دارد که عبارت‌اند از: متدی برای خواندن عناصر آرایه، متدی برای مرتب‌سازی و متدی برای چاپ عناصر آرایه. وظایف متدها:

متد main: متغیرهای برنامه را تعریف کرده، تعداد دانشجویان را می‌خواند. سپس متدهای readarray، select_sort و writearray را فراخوانی می‌کند.

متد readarray: شماره‌های دانشجویی را می‌خواند و در آرایه قرار می‌دهد.

متد select_sort: شماره‌های دانشجویی موجود در آرایه را مرتب می‌کند.

متد writearray: شماره‌های دانشجویی موجود در آرایه را چاپ می‌کند.

متد	متغیر	هدف
main	id	آرایه‌ای برای نگهداری شماره دانشجویان
	n	تعداد دانشجویان
	i	شمارنده
readarray	i	شمارنده
select_sort	i, j	شمارنده‌ها
	temp	متغیر کمکی
	max	بزرگ‌ترین عدد
writearray	i	شمارنده

```
package ch4_2;
import java.util.*;
public class Ch4_2 {
    private static void
readarray(int[] id, int n)
    {
        int i;
        for (i = 0; i < n; i++)
        {
            System.out.printf("Enter id[%1$s];", i + 1);
            id[i] = (new Scanner(System.in)).nextInt();
        }
    }
    private static void select_sort(int[] id, int n)
    {
        int i, j, max, temp;
        for (i = 0; i < n; i++)
        {
            max = i;
            for (j = i + 1; j < n; j++)
            {
                if (id[max] > id[j])
                {
                    max = j;
                }
            }
            temp = id[i];
            id[i] = id[max];
            id[max] = temp;
        }
    }
}
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۶۷

```
private static void writearray(int[] id, int n)
{
    int i;
    System.out.print("Sorted output ");
    for (i = 0; i < n; i++)
    {
        System.out.printf("\t%1$s", id[i]);
    }
}
public static void main(String[] args) {
    int n;
    int[] id = new int[100];
    System.out.print("Enter n:");
    n = (new Scanner(System.in)).nextInt();
    readarray(id, n);
    select_sort(id, n);
    writearray(id, n);
}
}
```

خروجی برنامه:

```
Enter n:۴
Enter id[۱];۳
Enter id[۲];۲
Enter id[۳];۷
Enter id[۴];۱
Sorted output      ۱      ۲      ۳      ۷
```

۳. برنامه‌ای که تعدادی عدد را از ورودی خوانده، آنها را به‌طور مرتب در آرایه‌ای قرار می‌دهد (دقت داشته باشید که اعداد در موقع گرفتن در آرایه، به‌طور صعودی مرتب شوند). سپس، آرایه مرتب‌شده، را به خروجی می‌برد.

متغیر	هدف
a	آرایه‌ای از اعداد
n	تعداد اعداد
i, j, l	شمارنده‌ها
num	عدد خوانده‌شده

```
package ch4_3;
import java.util.*;
public class Ch4_3 {
    public static void main(String[] args) {
        int i,j,l,n,num;
        int[] a = new int[100];
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextInt();
        for (i = 0; i < n; i++)
        {
            System.out.printf("Enter a[%1$s]:", i + 1);
            num = (new Scanner(System.in)).nextInt();
            for (j = 0; j < i; j++)
            {
                if (a[j] > num)
                {
                    break;
                }
            }
            for (l = i; l > j; l--)
            {
                a[l] = a[l - 1];
            }
            a[j] = num;
        }
        System.out.println("Result is : ");
    }
}
```

```

        for (i = 0; i < n; i++)
        {
            System.out.printf(" %1$s", a[i]);
        }
    }
}

```

خروجی برنامه:

```

Enter n: ۱
Enter a[۱]: ۱
Enter a[۲]: ۷
Enter a[۳]: ۰
Enter a[۴]: ۹
Enter a[۵]: ۹
Enter a[۶]: ۸
Result is :
    ۰ ۱ ۷ ۸ ۹ ۹

```

۴. برنامه‌ای که دو رشته S۱ و S۲ را از ورودی خوانده، رشته S۱ را در رشته S۲ جستجو می‌کند. در این برنامه، خواندن رشته‌ها توسط متد اصلی و وظایف متدها:

متد main: دو رشته را خوانده، با فراخوانی متد search_string، رشته اول را در رشته دوم جستجو می‌کند. سپس، پیغام مناسب چاپ می‌نماید.
 متد search_string: رشته اول را در رشته دوم جستجو می‌کند. اگر رشته اول در رشته دوم باشد، مکان شروع رشته اول در رشته دوم، وگرنه صفر را برمی‌گرداند.

متد	متغیر	هدف
main	s1, s2	رشته اول
	location	عمل وقوع رشته S۲ در رشته S۱
search_string	i, j	شمارنده‌ها
	k	شمارنده از مکان اولین تساوی

```

using System;
package ch4_4;
import java.util.*;
public class Ch4_4 {
    private static int
    search_string(char[] s1,
    char[] s2)
    {
        int i, j, k;
        for (i = 0; i < s2.length; i++)
        {
            k = i;
            j = 0;
            while (j < s1.length && k < s2.length && s1[j] == s2[k])
            {
                j++;
                k++;
            }
            if (j == s1.length)
            {
                return (i + 1);
            }
            if (j == s1.length && k == s2.length)
            {
                return (i + 1);
            }
        }
        return 0;
    }
    public static void main(String[] args) {
        String s1, s2;
        int location;
    }
}

```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۶۹

```

System.out.print("Enter string 1:");
s1 = new Scanner(System.in).nextLine();
System.out.print("Enter string 2:");
s2 = new Scanner(System.in).nextLine();
char[] ss1 = new char[s1.length()];
s1.getChars(0, 0 + ss1.length, ss1, 0);
char[] ss2 = new char[s2.length()];
s2.getChars(0, 0 + ss2.length, ss2, 0);
location = search_string(ss1, ss2);
if (location != 0)
{
    System.out.printf("Found in location %1$s", location);
}
else
{
    System.out.print("Not found ");
}
}
}

```

خروجی برنامه:

```

Enter string ۱:good
Enter string ۲:Java is a good language.
Found in location ۱۱

```

• برنامه‌ای که با خواندن تعدادی عدد از ورودی، آن‌ها را در آرایه‌ای قرار می‌دهد. سپس، کلیه عناصر آرایه را بر عنصر وسط تقسیم می‌کند. اگر عنصر وسط صفر باشد، بر عنصر بعد از عنصر وسط تقسیم می‌نماید. اگر این عنصر صفر باشد، بر عنصر قبل از عنصر وسط تقسیم می‌کند. اگر این عنصر صفر باشد، برای پیدا کردن عنصری غیر از صفر و انجام تقسیم، به روند قبلی ادامه می‌دهد. اگر همه عناصر آرایه صفر باشند، پیام مناسبی صادر می‌کند.

توضیح: پس از قرار دادن عناصر در آرایه، اندیس وسط را محاسبه کرده، عنصر وسط را پیدا می‌نماید. اگر صفر نبود، عمل تقسیم را انجام می‌دهد. این شمارنده به اندیس وسط اضافه می‌شود تا اگر مثبت بود، عنصر بعد از عنصر وسط و اگر منفی بود، عنصر قبل از عنصر وسط مورد بررسی قرار گیرد تا عمل تقسیم انجام شود. اگر عنصر غیر صفری در آرایه پیدا شود، **action** فراخوانی می‌شود تا تمام عناصر آرایه را بر آن عنصر تقسیم کند. اگر همه عناصر صفر باشند، برنامه پیام مناسبی صادر می‌کند.

متد	متغیر	هدف
main	n	تعداد عناصر آرایه
	a	آرایه‌ای با ۱۰۰ عنصر
	mid	عنصر وسط
	i	شمارنده
	sign	علامت (برای تعیین عنصر قبل و بعد به کار می‌رود)
action	i	شمارنده

```

package ch4_5;
import java.util.*;
public class Ch4_5 {
    private static void
    action(int[] a, int loc, int n)
    {
        int i;
        for (i = 0; i < n; i++)
        {
            System.out.printf("\n%1$s / %2$s = %3$s", a[i], a[loc],
                (float)a[i] / a[loc]);
        }
    }
    public static void main(String[] args) {
        int i, mid, n, sign = -1;

```

```

int[] a = new int[100];
System.out.print("Enter n:");
n = (new Scanner(System.in)).nextInt();
mid = n / 2;
for (i = 0; i < n; i++)
{
    System.out.printf("Enter a[%1$s]:", i + 1);
    a[i] = (new Scanner(System.in)).nextInt();
}
for (i = 0; i <= mid; i++)
{
    if (a[mid + (sign * i)] != 0)
    {
        action(a, mid + (sign * i), n);
        break;
    }
    if (sign == 1)
    {
        i--;
    }
    sign = -sign;
}
if (i > mid)
{
    System.out.print("All element are zero");
}
}
}

```

خروجی برنامه:

```

Enter n:۷
Enter a[۱]:۸
Enter a[۲]:۹
Enter a[۳]:۰
Enter a[۴]:۰
Enter a[۵]:۰
Enter a[۶]:۰
Enter a[۷]:۰

```

```

۸ / ۹ = ۰,۸۸۸۸۸۸۸۹
۹ / ۹ = ۱,۰
۰ / ۹ = ۰,۰
۰ / ۹ = ۰,۰
۰ / ۹ = ۰,۰
۰ / ۹ = ۰,۰
۰ / ۹ = ۰,۰

```

۶. کارخانه‌ای دارای ۵ ردیف شغلی است که به کارمندان در مقابل ۴۰ ساعت کار در هفته، ماهانه مطابق جدول زیر حقوق ثابتی پرداخت می‌شود. در صورتی‌که کارمندی بیش از ۱۶۰ ساعت در ماه کار کند به ازای هر ساعت مطابق جدول مشخص‌شده زیر، اضافه‌کاری پرداخت می‌گردد. اگر کارمندی کمتر از ۱۶۰ ساعت در ماه کار کند، به ازای هر ساعت مبلغی مطابق جدول زیر از حقوق وی کسر می‌شود.

برنامه‌ای که شماره کارمندی، ردیف شغلی و ساعت کارکرد در ماه برای هر کارمند را خوانده، دریافتی آن‌ها را چاپ می‌کند. برای خاتمه برنامه به‌جای شماره کارمندی ۹۹۹- وارد می‌شود.

ردیف شغلی	نوع تخصص	حقوق ثابت	هر ساعت اضافه‌کاری	هر ساعت کم‌کاری
۰	مهندس ارشد	۱۰۰۰۰۰	۵۰۰۰	۶۰۰۰
۱	مهندس ساده	۸۰۰۰۰	۴۰۰۰	۵۰۰۰
۲	تکنسین	۶۰۰۰۰	۳۰۰۰	۴۰۰۰
۳	کارگر ماهر	۵۰۰۰۰	۲۵۰۰	۳۰۰۰

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۷۱

۲۵۰۰	۲۰۰۰	۴۶۰۰۰۰	کارگر ساده	۴
------	------	--------	------------	---

متغیر	هدف
table	جدولی با ۵ سطر و ۳ ستون
code	شماره کارمندی
radif	ردیف شغلی
hh	ساعات کارکرد
t	ساعات تأخیر
e	ساعات اضافه کار
t_mablag	مبلغ تأخیر
base	پایه
e_mablag	مبلغ اضافه کار
pay	قابل پرداخت

```

package ch4_6;
import java.util.*;
public class Ch4_6 {
    public static void main(String[]
args) {
        long[][] table = new long[][]
        {
            {1000000, 5000, 6000},
            {800000, 4000, 5000},
            {600000, 3000, 4000},
            {500000, 2500, 3000},
            {460000, 2000, 2500}
        };
        int code, radif, hh, t, e;
        long t_mablag, e_mablag, pay, base1;
        for (; ; )
        {
            t = 0;
            e = 0;
            System.out.print("Enter code:");
            code=(new Scanner(System.in)).nextInt();
            if (code == -999)
            {
                break;
            }
            System.out.print("Enter radif:");
            radif=(new Scanner(System.in)).nextInt();
            System.out.print("Enter hour:");
            hh = (new Scanner(System.in)).nextInt();
            if (radif > 4 || radif < 0)
            {
                System.out.print("Enter radif between 0 to 4");
                continue;
            }
            if (hh > 160)
            {
                e = hh - 160;
            }
            else
            {
                t = 160 - hh;
            }
            base1 = table[radif][0];
            e_mablag = table[radif][1] * e;
        }
    }
}

```

```

        t_mablag = table[radif][2] * t;
        pay = base1 + e_mablag - t_mablag;
        System.out.printf("Code is %1$s\thour is %2$s", code, hh);
        System.out.printf("\nbase is %1$s\te_mablag is
            %2$s\tt_mablag is %3$s\tPay%4$s", base1, e_mablag,
                t_mablag, pay);
        System.out.println();
    }
}

```

خروجی برنامه:

```

Enter code:\
Enter radif:\
Enter hour:\۲
Code is \      hour is \۲
base is ۸۰۰۰۰۰      e_mablag is ۰      t_mablag is ۷۴۰۰۰۰      Pay\۱۰۰۰۰
Enter code:۲
Enter radif:۳
Enter hour:\۴
Code is ۲      hour is \۴
base is ۰۰۰۰۰۰      e_mablag is ۰      t_mablag is ۴۳۸۰۰۰      Pay\۱۲۰۰۰
Enter code:۳
Enter radif:۴
Enter hour:\۲۰
Code is ۳      hour is \۲۰
base is ۴۱۰۰۰۰      e_mablag is ۰      t_mablag is ۳۰۰۰۰۰      Pay\۱۱۰۰۰
Enter code:-۹۹۹

```

۷. برنامه‌ای که n عدد را خوانده، در آرایه‌ای قرار می‌دهد و توسط متدی عناصر آن را معکوس (از آخرین عدد به اولین عدد) کرده، به برنامه برگرداند و برنامه آن را نمایش می‌دهد (برنامه، برای دریافت داده، معکوس کردن و چاپ عناصر آرایه از متدهای جداگانه استفاده می‌کند).
وظایف متدها:

متد **main**: ابتدا تعداد عدد را می‌خواند، با فراخوانی متد **input** تعدادی عدد را از ورودی خوانده، متد **convert** آن‌ها را معکوس می‌کند و با فراخوانی **print** اعداد معکوس شده را چاپ می‌کند.
متد **input**: تعدادی عدد را خوانده و در آرایه قرار می‌دهد.
متد **convert**: اعداد موجود در آرایه را معکوس می‌کند.
متد **print**: عناصر موجود در آرایه را چاپ می‌کند.

متد	نام	هدف
main	n	تعداد اعداد
	num	آرایه‌ای از اعداد
input	i	شمارنده
convert	temp	متغیر کمکی
	i	شمارنده
print	i	شمارنده

```

package ch4_7;
import java.util.*;
public class Ch4_7 {
    private static void input(int[] a,
int n)
    {
        int i;
        for (i = 0; i < n; i++)
        {
            System.out.printf("Enter number [%1$s]:", i + 1);
            a[i] = (new Scanner(System.in)).nextInt();
        }
    }
    private static void convert(int[] a, int n)

```


حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۷۳

```

{
    int i, temp;
    for (i = 0; i < n / 2; i++)
    {
        temp = a[i];
        a[i] = a[n - i - 1];
        a[n - i - 1] = temp;
    }
}
private static void print(int[] a, int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        System.out.printf("    %1$s", a[i]);
    }
}
public static void main(String[] args) {
    int n;
    System.out.print("Enter n:");
    n = (new Scanner(System.in)).nextInt();
    int[] num = new int[n];
    input(num, n);
    System.out.print("Input is\t");
    print(num, n);
    convert(num, n);
    System.out.print("\nConvert is\t");
    print(num, n);
}
}

```

خروجی برنامه:

```

Enter n: ۵
Enter number [۱]: ۱
Enter number [۲]: ۵
Enter number [۳]: ۲
Enter number [۴]: ۷
Enter number [۵]: ۶
Input is      ۱      ۵      ۲      ۷      ۶
Convert is    ۶      ۷      ۲      ۵      ۱

```

۸. فرض کنید که در قسمت تخلفات ماشین‌ها در اداره راهنمایی رانندگی، ۱۰ نوع تخلف منظور گردید و کد تخلف از ۰ تا ۹ در نظر گرفته شد و هر تخلف جریمه خاصی دارد. برای هر ماشین اطلاعاتی مثل شماره ماشین، تعداد تخلفات و کد هر تخلف موجود است. برنامه‌ای که مبلغ جریمه را برای هر ماشین محاسبه می‌کند. برای خروج از برنامه کاربر به‌جای شماره ماشین ۹۹۹- را وارد می‌کند.

توضیح: برای حل این برنامه کد تخلف به همراه مبلغ جریمه را در آرایه‌ای به نام **code** قرار دادیم که ۱۰ سطر دارد. در هر سطر مبلغ جریمه و کد تخلف نگه‌داری می‌شود.

متغیر	هدف
table	جدول مبلغ جریمه
number	شماره ماشین
code	کد تخلف
sum	مجموع تخلفات هر ماشین
i	شمارنده
n	تعداد تخلف هر ماشین

```

package ch4_8;
import java.util.*;
public class Ch4_8 {
    public static void main(String[]

```

```

args) {
    long[] table = new long[] {1000, 2000, 3000, 4000, 5000,
        8000, 10000, 20000, 30000, 50000};
    int number, code, n, i;
    double sum = 0.0;
    for (; ; )
    {
        System.out.print("Enter number:");
        number=(new Scanner(System.in)).nextInt();
        if (number == -999)
        {
            break;
        }
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextInt();
        sum = 0.0;
        for (i = 1; i <= n; i++)
        {
            System.out.printf("Enter code %1$s:", i);
            code=(new Scanner(System.in)).nextInt();
            if (code >= 0 && code < 10)
            {
                sum += (float)table[code];
            }
            else
            {
                i--;
                System.out.println("Enter code between 0 to 9");
                continue;
            }
        }
        System.out.printf("*** Number is %1$s\t Sum is %2$s ***"
            + "\n", number, sum);
    }
}

```

خروجي برنامہ:

```

Enter number:۱۲
Enter n:۴
Enter code ۱:۱
Enter code ۲:۳
Enter code ۳:۵
Enter code ۴:۷
*** Number is ۱۲      Sum is ۳۴۰۰۰,۰ ***
Enter number:۷
Enter n:۱
Enter code ۱:۰
*** Number is ۷      Sum is ۱۰۰۰,۰ ***
Enter number:۹
Enter n:۸
Enter code ۱:۷
Enter code ۲:۴
Enter code ۳:۶
Enter code ۴:۵
Enter code ۵:۵
Enter code ۶:۸
Enter code ۷:۲
Enter code ۸:۷
*** Number is ۹      Sum is ۱۰۴۰۰۰,۰ ***
Enter number:-۹۹۹

```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۷۵

۹. برنامه‌ای که یک ماتریس n در n را خوانده، مقادیر قطر اصلی و فرعی آن را عوض می‌کند. وظایف متدها:

- متد `main()`، ابتدا ثابت‌های n ، و آرایه a (با n سطر و n ستون) را تعریف کرده، با فراخوانی متد `readA2D()`، عناصر آرایه دوبعدی را خوانده، سپس با فراخوانی `printA2D()`، عناصر آرایه دوبعدی را نمایش می‌دهد. در پایان، با فراخوانی متد `Change()` عناصر قطر اصلی و فرعی را جابه‌جا می‌کند و با فراخوانی `printA2D()`، عناصر آرایه دوبعدی جدید را نمایش می‌دهد.
- متد `readA2D()`، پارامترهای آرایه دوبعدی a ، تعداد سطرها و ستون‌های آن (n) را دریافت کرده، با استفاده از دو حلقه تودرتو، عناصر آرایه را می‌خواند و برمی‌گرداند.
- متد `printA2D()`، پارامترهای آرایه دوبعدی a ، تعداد سطرها و ستون‌های آن (n) را دریافت کرده با استفاده از دو حلقه تودرتو عناصر آرایه a را نمایش می‌دهد.
- متد `Change()`، ابتدا، پارامترهای آرایه دوبعدی a ، تعداد سطرها و ستون‌های آن (n) را دریافت کرده را دریافت می‌کند و سپس، با استفاده از حلقه‌ی با شمارنده تودرتو چنانچه i برابر j باشد (یعنی روی قطر اصلی باشد). آن عنصر را با عنصر متناظر قطر فرعی جابه‌جا می‌کند.

متد	متغیر	هدف
main	ثابت n	تعداد سطرهاى آرایه (ثابت ۴)
	ثابت m	تعداد ستون‌های آرایه (ثابت ۴)
	a	آرایه‌ای دوبعدی با n سطر و m ستون
readA2D	a	آرایه‌ای با ۴ سطر و ۴ ستون که مقادیر خوانده شده را نگه‌داری می‌کند
	n	تعداد سطرها و ستون‌های آرایه
	i	شمارنده سطر از صفر تا $n-1$
	j	شمارنده ستون از صفر تا $n-1$
printA2D	a	آرایه‌ای با n سطر و ۴ ستون که مقادیر آن باید نمایش داده شود.
	n	تعداد سطرها و ستون‌های آرایه
	i	شمارنده سطر از صفر تا $n-1$
	j	شمارنده ستون از صفر تا $m-1$
Change	a	آرایه‌ای با n سطر و ۴ ستون که باید مقادیر قطر اصلی و فرعی آن‌ها عوض شوند.
	n	تعداد سطرها و ستون‌های آرایه
	temp	متغیر کمکی برای جابه‌جایی
	j	شمارنده ستون از صفر تا $n-1$ می‌شمارد
	i	شمارنده سطر که از صفر تا $n-1$ می‌شمارد

```
package ch4_9;
import java.util.*;
public class Ch4_9 {
private static void readA2D(int[][] a, int n)
{
    for (int i = 0; i < n; i++)
    {
        System.out.printf("Enter %1$s number for row[%2$s]:",n,i);
        String[] s=new Scanner(System.in).nextline().split(" ",-1);
        for (int j = 0; j < n; j++)
        {
```

```

        a[i][j] = Integer.parseInt(s[j]);
    }
}
private static void printA2D(int[][] a, int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            System.out.printf("%1$s\t", a[i][j]);
        }
        System.out.println();
    }
}
private static void Change(int[][] a, int n)
{
    int temp;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if ( i == j)
            {
                temp = a[i][j];
                a[i][j] = a[i][n - j - 1];
                a[i][n-j-1] = temp;
            }
        }
    }
}
public static void main(String[] args) {
    System.out.print("Enter n:");
    int n = (new Scanner(System.in)).nextInt();
    int[][] a = new int[n][n];
    readA2D(a, n);
    printA2D(a, n);
    Change(a, n);
    System.out.printf("=====\n");
    printA2D(a, n);
}
}

```

خروجی برنامه:

```

Enter n:۴
Enter ۴ number for row[۰]:۱ ۳ ۵ ۷
Enter ۴ number for row[۱]:۲ ۴ ۶ ۸
Enter ۴ number for row[۲]:۰ ۹ ۱۰ ۱۱
Enter ۴ number for row[۳]:۱۲ ۱۳ ۱۵ ۱۸

```

```

۱      ۳      ۵      ۷
۲      ۴      ۶      ۸
۰      ۹      ۱۰     ۱۱
۱۲     ۱۳     ۱۵     ۱۸
=====

```

```

۷      ۳      ۵      ۱
۲      ۶      ۴      ۸
۰      ۱۰     ۹      ۱۱
۱۸     ۱۳     ۱۵     ۱۲

```

۱۰. برنامه‌ای که عناصر آرایه‌ای ۴×۴ را خوانده، بزرگترین عنصر هر سطر را پیدا کرده، به همراه شماره سطر در خروجی چاپ می‌کند. توضیح: ستون پنجم آرایه برای نگه‌داری بزرگترین عنصر هر سطر در نظر گرفته شده است.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۷۷

متد	متغیر	هدف
main	a	آرایه‌ای با ۴ سطر و ۵ ستون
readarray	i, j	شمارنده‌ها
findmax	i, j	شمارنده‌ها
writearray	i, j	شمارنده‌ها

```

package ch4_10;
import java.util.*;
public class Ch4_10 {
    private static void readarray(int[][] a)
    {
        int i, j;
        for (i = 0; i < 4; i++)
        {
            System.out.printf("Enter row(%1$s):", i + 1);
            String[] s=new Scanner(System.in).nextLine().split("[ ]",-);
            for (j = 0; j < 4; j++)
            {
                a[i][j] = Integer.parseInt(s[j]);
            }
        }
    }
    private static void findmaxrow(int[][] a)
    {
        int i, j;
        for (i = 0; i < 4; i++)
        {
            a[i][4] = a[i][0];
            for (j = 1; j < 4; j++)
            {
                if (a[i][4] < a[i][j])
                {
                    a[i][4] = a[i][j];
                }
            }
        }
    }
    private static void writearray(int[][] a)
    {
        int i, j;
        for (i = 0; i < 4; i++)
        {
            System.out.println();
            for (j = 0; j < 4; j++)
            {
                System.out.printf("%1$s\t", a[i][j]);
            }
            System.out.printf("\t %1$s", a[i][4]);
        }
    }
    public static void main(String[] args) {
        int[][] a = new int[4][5];
        readarray(a);
        findmaxrow(a);
        System.out.print("*****result*****\n");
        writearray(a);
    }
}

```

خروجی برنامه:
Enter row(۱): ۱ ۲ ۳ ۴
Enter row(۲): ۵ ۶ ۷ ۸

```
Enter row(۳): ۹ ۰ ۱ ۲
Enter row(۴): ۴ ۷ ۹ ۲
*****result*****
۱      ۲      ۳      ۴      ۴
۵      ۶      ۷      ۸      ۸
۹      ۰      ۱      ۲      ۹
۴      ۷      ۹      ۲      ۹
```

۱۱. برنامه‌ای که رشته‌ای را در رشته دیگر کپی می‌کند (بدون استفاده از متد `strcpy`).
 وظایف متدها:
 + متد `main`: دو رشته را خوانده، با فراخوانی `strcpy` رشته دوم را در رشته اول کپی می‌کند. سپس رشته اول و دوم را با پیغام مناسب چاپ می‌کند.
 + متد `strcpy`: یک رشته را در رشته دیگر کپی می‌کند.

```
package ch4_11;
import java.util.*;
public class Ch4_11 {
    public static String strCopy(String des, String sou)
    {
        des = "";
        int i;
        for (i = 0; i < sou.length(); i++)
        {
            des += String.valueOf(sou.charAt(i));
        }
        return des;
    }
    public static void main(String[] args) {
        String str1 = "";
        String str2 = "";
        System.out.print("Enter a string:");
        str1 = new Scanner(System.in).nextLine();
        str2 = strCopy(str2, str1);
        System.out.print("Result is ");
        System.out.print(str2);
    }
}
```

متد	متغیر	هدف
main	str ^۱	رشته مبدأ
	str ^۲	رشته مقصد
strcpy	i	شمارنده

خروجی برنامه:

```
Enter a string:Fanavarienovin.net
Result is Fanavarienovin.net
```

۱۲. ماتریس جادویی یک ماتریس $N \times N$ است که هر عنصر آن یک عدد صحیح ۱ تا $N^۲$ است. حاصل جمع هر سطر، هر ستون و همچنین عناصر قطر همگی با یک دیگر مساوی‌اند. برای تولید این ماتریس به صورت زیر عمل می‌شود:
 الف. عدد یک در ستون میانی اولین سطر قرار می‌گیرد.
 ب. به طور مورب به سمت چپ و بالا حرکت می‌کنیم.
 ج. اگر این خانه پر باشد، یکخانه به طرف پائین حرکت می‌کنیم و عدد بعدی را در آن مکان می‌نویسیم.
 د. اگر در حرکت به صورت مورب، از محدوده ماتریس خارج شویم، باید به آخرین عنصر در آن سطر یا ستون برگردیم.
 هـ. اگر در محدوده سطر و ستون از هر دو خارج شویم، یکخانه به سمت پائین حرکت می‌کنیم.
 برنامه‌ای که ماتریس جادویی را تولید کرده و چاپ می‌نماید (ماتریس ۵×۵ زیر را در نظر بگیرید):

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۷۹

۱۵	۸	۱	۲۴	۱۷
۱۶	۱۴	۷	۵	۳۳
۲۲	۲۰	۱۳	۶	۴
۳	۲۱	۱۹	۱۲	۱۰
۹	۲	۲۵	۱۸	۱۱

متد	متغیر	هدف
main	matrix	ماتریس
	n	تعداد سطر و ستون
fill_matrix	j, i	شماره سطر و ستون
	j', i'	شماره سطر و ستون خانه بعدی
	n	۱ تا شمارنده
disp_matrix	j, i	شمارنده ۱ برای نمایش ماتریس

```

package ch4_12;
import java.util.*;
public class Ch4_12 {
    private static int test_range(int i1, int n)
    {
        if (i1 > n)
        {
            i1 = 0;
        }
        if (i1 < 0)
        {
            i1 = n;
        }
        return i1;
    }
    private static void fill_matrix(int[][] matrix, int n)
    {
        int i, j, i1, j1;
        n--;
        i = 0;
        j = n / 2;
        matrix[i][j] = 1;
        for (int m = 1; m < (n + 1) * (n + 1); m++)
        {
            i1 = i - 1;
            j1 = j - 1;
            i1 = test_range(i1, n);
            j1 = test_range(j1, n);
            if (matrix[i1][j1] != 0)
            {
                i1 = i + 1;
                j1 = j;
                i1 = test_range(i1, n);
                j1 = test_range(j1, n);
            }
            matrix[i1][j1] = m + 1;
            i = i1;
            j = j1;
        }
    }
    private static void disp_matrix(int[][] matrix, int n)
    {
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if (matrix[i][j] < 10)
                {
                    System.out.printf(" %1$s", matrix[i][j]);
                }
            }
        }
    }
}

```

```

        }
        else if (matrix[i][j] < 100)
        {
            System.out.printf(" %1$s", matrix[i][j]);
        }
        else
        {
            System.out.printf(" %1$s", matrix[i][j]);
        }
    }
    System.out.println();
}
}

public static void main(String[] args) {
    int n;
    System.out.print("Enter n:");
    n = (new Scanner(System.in)).nextInt();
    int[][] matrix = new int[n][n];
    fill_matrix(matrix, n);
    disp_matrix(matrix, n);
}
}
}

```

خروجی برنامه:

Enter n: 7

```

۲۸ ۱۹ ۱۰ ۱ ۴۸ ۳۹ ۳۰
۲۹ ۲۷ ۱۸ ۹ ۷ ۴۷ ۳۸
۳۷ ۳۵ ۲۶ ۱۷ ۸ ۶ ۴۶
۴۵ ۳۶ ۳۴ ۲۵ ۱۶ ۱۴ ۵
۴ ۴۴ ۴۲ ۳۳ ۲۴ ۱۵ ۱۳
۱۲ ۳ ۴۳ ۴۱ ۳۲ ۲۳ ۲۱
۲۰ ۱۱ ۲ ۴۹ ۴۰ ۳۱ ۲۲

```

➤ برنامه‌ای که n عدد را خوانده، در آرایه‌ای قرار می‌دهد و توسط متدی عناصر آن را معکوس (از آخرین عدد به اولین عدد) کرده، به برنامه برگرداند و برنامه آن را نمایش می‌دهد (برنامه، برای دریافت داده، معکوس کردن و چاپ عناصر آرایه از متدها جداگانه استفاده می‌کند).

➤ فرض کنید که در قسمت تخلفات ماشین‌ها در اداره راهنمایی رانندگی، ۱۰ نوع تخلف منظور گردید و کد تخلف از ۰ تا ۹ در نظر گرفته شد و هر تخلف جریمه خاصی دارد. برای هر ماشین اطلاعاتی مثل شماره ماشین، تعداد تخلفات و کد هر تخلف موجود است. برنامه‌ای که مبلغ جریمه را برای هر ماشین محاسبه می‌کند. برای خروج از برنامه کاربر به‌جای شماره ماشین ۹۹۹- را وارد می‌کند.

توضیح: برای حل این برنامه کد تخلف به همراه مبلغ جریمه را در آرایه‌ای به نام code قراردادیم که ۱۰ سطر دارد. در هر سطر مبلغ جریمه و کد تخلف نگه‌داری می‌شود.

➤ برنامه‌ای که عناصر آرایه‌ای 4×4 را خوانده، بزرگ‌ترین عنصر هر سطر را پیدا کرده، به همراه شماره سطر در خروجی چاپ می‌کند.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۸۱

توضیح: ستون پنجم آرایه برای نگهداری بزرگ‌ترین عنصر هر سطر در نظر گرفته شده است.

- برنامه‌ای که رشته‌ای را در رشته دیگر کپی می‌کند (بدون استفاده از متد `strcpy`).
 - ماتریس جادویی یک ماتریس $N \times N$ است که هر عنصر آن یک عدد صحیح ۱ تا N^2 است. حاصل جمع هر سطر، هر ستون و همچنین عناصر قطر همگی با یک دیگر مساوی‌اند. برای تولید این ماتریس به صورت زیر عمل می‌شود:
 - الف. عدد یک در ستون میانی اولین سطر قرار می‌گیرد.
 - ب: به طور مورب به سمت چپ و بالا حرکت می‌کنیم.
 - ج: اگر این خانه پر باشد، یک خانه به طرف پائین حرکت می‌کنیم و عدد بعدی را در آن مکان می‌نویسیم.
 - د. اگر در حرکت به صورت مورب، از محدوده ماتریس خارج شویم، باید به آخرین عنصر در آن سطر یا ستون برگردیم.
 - هـ. اگر در محدوده سطر و ستون از هر دو خارج شویم، یک خانه به سمت پائین حرکت می‌کنیم.
- برنامه‌ای که ماتریس جادویی را تولید کرده و چاپ می‌نماید (ماتریس 5×5 زیر را در نظر بگیرید):

- برنامه‌ای که رشته‌ای را به انتهای رشته دیگر الحاق می‌کند (بدون استفاده از متد `Strcat`).
- برنامه‌ای که رشته‌ای را از ورودی خوانده، به متدی ارسال می‌کند و متد آن را به طور معکوس به خروجی می‌برد.
- برنامه‌ای که رشته عددی را که حاوی نقطه اعشار است از ورودی خوانده، آن را به عدد اعشاری تبدیل کند. به عنوان مثال، رشته "۱۲,۳۴۲" را به ۱۲,۳۴۲ تبدیل نماید. این

برنامه، متدی برای خواندن رشته، متدی برای تبدیل و متدی برای نوشتن عدد در خروجی دارد. پارامترها از طریق فراخوانی با ارجاع به متدها ارسال می‌شوند.

➤ برنامه‌ای که بازی دوز را شبیه‌سازی می‌کند. در این بازی یک ماتریس 3×3 داریم که دو بازیکن ۱ و ۲ باهم بازی می‌کنند که هر یک از این بازیکنان سه مهره دارند (هر سه مهره شماره همان بازیکن را دارد). بازیکنی برنده است که بتواند سریع‌تر مهره‌های خود را به صورت سطری یا ستونی و یا قطری ردیف کند. به عنوان مثال، در ماتریس زیر بازیکن شماره ۲ برنده است:

➤ هر عدد زوج بزرگ‌تر از ۶ برابر با مجموع دو عدد اول کوچک‌تر از خودش است. برنامه‌ای که چند عدد زوج بزرگ‌تر از ۶ را می‌خواند. مجموع دو عدد اولی که برابر با اعداد خوانده شده باشد را پیدا کرده چاپ می‌نماید (برنامه تا زمانی که کاربر بخواهد ادامه می‌یابد).

➤ برنامه‌ای که رشته عددی را که حاوی نقطه اعشار است از ورودی خوانده، آن را به صورتی که گفته شد به عدد اعشاری تبدیل می‌کند. به عنوان مثال، رشته '۱۲۴۵,۴۵۲' را به عدد ۴۵۲,۱۲۴۵ تبدیل می‌کند. برنامه، متدی برای خواندن رشته، متدی برای تبدیل و متدی برای نوشتن عدد در خروجی دارد. پارامترها از طریق فراخوانی با ارجاع به متدها ارسال می‌شوند.

➤ متدی که طول رشته را برمی‌گرداند (با استفاده از اشاره‌گر). سپس، برنامه‌ای که از این متد استفاده می‌کند. برنامه، متدی برای خواندن رشته، متدی برای محاسبه طول رشته و متدی برای چاپ طول رشته دارد.

➤ متدی که توان n یک عدد را برمی‌گرداند. سپس، برنامه‌ای که از این متد استفاده می‌کند.

➤ برنامه‌ای که سه رشته را خوانده، رشته دوم و سوم را به انتهای رشته اول متصل می‌کند.

➤ برنامه‌ای که رشته‌ای را خوانده، اعمال زیر را انجام می‌دهد:
 الف. کلیه حروف بزرگ رشته را به حروف کوچک تبدیل می‌کند.
 ب. کلیه ارقام رشته را شمارش می‌کند و مجموع ارقام موجود در رشته را چاپ می‌کند.
 ج. تعداد حروف رشته را شمارش می‌کند.
 برنامه، برای هر یک از این درخواست‌ها متدی دارد و کلیه پارامترهای متد را با ارجاع ارسال می‌نماید.

➤ برنامه‌ای که میزان میانگین بارش باران را در سال قبل می‌گیرد و سپس میزان واقعی بارش باران را در ۱۲ ماه گرفته، در آرایه‌ای قرار می‌دهد (برای خواندن میزان واقعی

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۸۳

بارش باران در ۱۲ ماه از یک متد استفاده می‌کند. سپس، با یک متد دیگر اختلاف بارش باران در هر ماه را با میانگین بارش باران در سال قبل محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای که ۵ عدد را از ورودی خوانده، در آرایه‌ای به نام S قرار می‌دهد. ابتدا میانگین مقادیر را محاسبه می‌کند (با یک متد). سپس، با استفاده از فرمول زیر انحراف معیار را محاسبه کرده، چاپ می‌نماید (برای محاسبه انحراف معیار نیز از یک متد استفاده

$$\text{میانگین} = \frac{\sum_{i=1}^N (S_i - a)^2}{N}$$

می‌کند):

a میانگین اعداد و N تعداد اعداد است.

➤ برنامه‌ای که رشته‌ای را از ورودی می‌خواند. تمام کلمات چهارحرفی آن را با کلمه "Love" جایگزین می‌کند. مثلاً رشته "hate you, you doer" به رشته "love you, you love" تبدیل می‌گردد.

➤ در یک دوره مسابقات ورزشی تعدادی تیم (حداکثر ۱۰۰ تیم) شرکت کرده‌اند، می‌خواهیم در انتهای مسابقه گزارشی از مجموع امتیازات هر تیم را چاپ کنیم. برنامه‌ای که برای هر مسابقه در یک خط ورودی شامل آیتیم‌های برنده و امتیاز کسب‌شده را می‌خواند و نتیجه را نمایش می‌دهد. اگر به جای تیم برنده ۱- وارد شد، برنامه خاتمه می‌یابد.

➤ برنامه‌ای که یک رشته را خوانده، تعداد تکرار هر حرف را نمایش می‌دهد. به‌عنوان مثال، ACCURENCE را در نظر بگیرید که حرف C سه بار تکرار شده است.

➤ برنامه‌ای که یک رشته را خوانده تشخیص می‌دهد آیا متجانس است یا خیر؟ رشته‌ای متجانس است که از دو طرف یکی خوانده شود.

➤ برنامه‌ای که حداکثر ۱۰ عدد را از ورودی خوانده، می‌گوید هر عدد چند بار تکرار شده است. به‌عنوان مثال، اگر ورودی به‌صورت زیر باشد:

۷۰ ۱۴۳ ۱۰۰ ۷۲ ۱۴۳ ۵۲ ۱۰۰ ۱۴۳ ۱۰۰ ۷۰

خروجی زیر را چاپ می‌کند:

۱۴۳ OCCURS ۳ TIMES, AT POSITIONS ۳ ۶ ۹
۷۰ OCCURS ۲ TIMES, AT POSITIONS ۱ ۱۰

➤ برنامه‌ای ماتریس A[۳×۴] را از ورودی می‌خواند و ترانواده آن را در B ذخیره می‌کند. در ترانواده باید برای هر j, i رابطه مقابل برقرار باشد:

$$B_{i,j} = A_{j,i}$$

➤ برنامه‌ای که انداختن دو تاس را شبیه‌سازی می‌نماید. این برنامه برای انداختن تاس‌ها از متد rand استفاده می‌کند که عدد بین ۱ تا ۶ را تولید می‌نماید. سپس مجموع این دو مقدار محاسبه می‌گردد. از آنجایی که تاس‌ها مقادیر بین ۱ تا ۶ را نشان می‌دهند، پس مجموع این دو مقدار بین ۲ تا ۱۲ است که مجموع ۷، بیش‌ترین دفعات و مجموع ۱۲ کم‌ترین دفعات پیش می‌آیند. جدول زیر ترکیبات ممکن برای این دو تاس را نشان می‌دهد. این برنامه، ۳۶۰۰۰ بار تاس می‌ریزد و با استفاده از یک آرایه، تک‌بعدی تعداد دفعاتی را که هر یک از این مجموعه‌های ممکن پیش می‌آیند، را ثبت می‌کند و سپس تحقیق می‌کند که آیا تعداد مجموعه‌های حاصل معقول است (مثلاً به شش طریق، مجموع ۷ حاصل می‌شود. بنابراین تقریباً $1/6$ همه پیشامدها باید ۷ باشد).

	۱	۲	۳	۴	۵	۶
۱	۲	۳	۴	۵	۶	۷
۲	۳	۴	۵	۶	۷	۸
۳	۴	۵	۶	۷	۸	۹
۴	۵	۶	۷	۸	۹	۱۰
۵	۶	۷	۸	۹	۱۰	۱۱
۶	۷	۸	۹	۱۰	۱۱	۱۲

➤ در هر سطر ورودی برای هر کالا دو عدد وجود دارد. عدد اول شماره کالا و عدد دوم موجودی آن در انبار است.
الف. برنامه‌ای که این اطلاعات را می‌خواند و در یک آرایه ذخیره می‌کند. انتهای اطلاعات با دو عدد ۱- مشخص شده است.
ب: پس از ورود اطلاعات اولیه تعدادی ورودی دیگر وجود دارد. این ورودی‌ها عبارت‌اند از: شماره کالا، تعداد و کاراکتری که مشخص‌کننده این است که تعداد ورودی سفارش (کاراکتر s یا S) یا خرید (کاراکتر p یا P) است. برنامه‌ای که اطلاعات را خوانده، مقدار خروجی انبار هر کالا را نمایش می‌دهد. اگر سفارش بیش از موجودی انبار باشد، آن را انجام نمی‌دهد و پیام مناسبی چاپ می‌کند.

➤ برنامه‌ای که تعدادی کلمه را از ورودی خوانده، تعداد کلمات ۱ تا ۵ کاراکتری، ۶ تا ۱۲ کاراکتری و بیش‌تر از ۱۲ کاراکتر را نمایش می‌دهد (تعداد کلمات را از ورودی می‌خواند).

➤ برنامه‌ای که چندجمله‌ای زیر را در یک آرایه نمایش می‌دهد (سپس، x را می‌خواند) و عبارت p را محاسبه کرده، چاپ می‌نماید (n حداکثر ۱۰۰ باشد).

$$p = anx^n + an - 1x^{n-1} + \dots + a_1x^1 + a.$$

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۸۵

➤ برنامه‌ای که یک رشته حداکثر ۱۰۰ کلمه‌ای را خوانده، تمام کلمات آن را وارونه می‌کند و چاپ می‌نماید (بین کلمات کاراکتر blank (فاصله) قرار می‌گیرد). به‌عنوان مثال، Ali and Reza به‌صورت azeR dna ilA تبدیل می‌شود.

➤ برنامه‌ای که یک متن را از ورودی خوانده، اعمال زیر را انجام می‌دهد:
الف: درصد تکرار هر حرف را در متن پیدا می‌کند (حروف بین A تا Z و حروف کوچک را بزرگ در نظر بگیرد).
ب: درصد تکرار هر کلمه را در متن پیدا می‌کند (کلمات بافاصله از یکدیگر جدا می‌شوند (حداکثر ۱۰۰ کلمه داریم)).

➤ برنامه‌ای که تعدادی عدد را خوانده، تشخیص می‌دهد مرتب شده هستند یا خیر؟

➤ برنامه‌ای که n عدد را خوانده، در آرایه قرار می‌دهد. سپس، توسط متدی عنصر n را حذف می‌کند.

➤ برنامه‌ای که دو آرایه m عنصری و n عنصری را خوانده، n عنصر اول آرایه دوم را به انتهای آرایه اول اضافه می‌کند.

➤ برنامه‌ای که عناصر یک آرایه و مقدار x را خوانده، تعداد تکرار عنصر x در آرایه a را می‌شمارد. در این برنامه متدها زیر نوشته شده است:
متدی برای خواندن عناصر آرایه a
متدی برای شمارش تعداد تکرار x در آرایه a

➤ برنامه‌ای که دو آرایه n عنصری یک‌بعدی را خوانده، حاصل ضرب بیرونی n عنصر اول آرایه a با n عنصر اول آرایه b را برمی‌گرداند (حاصل ضرب این دو آرایه، در آرایه دیگری به نام c قرار می‌گیرد). به‌عنوان مثال، فرض کنید آرایه‌های زیر را داشته باشیم.

a	b			
2.2	2.0	4.4	-2.2	0.0
3.3	-1.0	6.6	-3.3	0.0
4.4	0.0	8.8	-4.4	0.0

متدی برای ضرب بیرونی (outer product) داشته باشید.

$$C[i][j] = a[i] \times b[j]$$

متدی برای چاپ حاصل ضرب داشته باشید.

➤ برنامه‌ای که عناصر یک آرایه دوبعدی را ۹۰ درجه چرخش می‌دهد. به‌عنوان مثال، اگر آرایه زیر را داشته باشیم:

۱۱	۲۲	۳۳
۴۴	۵۵	۶۶
۷۷	۸۸	۹۹

با چرخش ۹۰ درجه آرایه به‌صورت زیر تبدیل می‌شود:

۷۷	۴۴	۱۱
۸۸	۵۵	۲۲
۹۹	۶۶	۳۳

➤ برنامه‌ای که آرایه‌ای ۱۰ عنصری تعریف کرده، سپس ۱۰ عدد را خوانده و تعداد تکرار هر عدد را می‌شمارد.

➤ برنامه‌ای که ۵ عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد دیگر خوانده، تعداد تکرار این عدد و مکان‌های وقوع آن را نمایش می‌دهد.

➤ برنامه‌ای که ۱۰ عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد دیگر را می‌خواند و چنانچه آن عدد در آرایه وجود داشته باشد، آن عدد را از آرایه حذف کرده، آرایه را نمایش می‌دهد (دقت کنید که اگر عدد چند بار در آرایه تکرار شده باشد، همه تکرارهای آن عدد را از آرایه حذف می‌کند).

➤ برنامه‌ای که یک آرایه 4×4 تعریف کرده، سپس عناصر آرایه را خوانده و حاصل ضرب عناصر غیر صفر محیطی آرایه را محاسبه می‌کند و نمایش می‌دهد. عناصر محیطی آرایه، عناصری هستند که در سطر اول، سطر آخر، ستون اول و ستون آخر آرایه می‌باشند.

➤ برنامه‌ای که یک آرایه ۵ در ۶ تعریف کرده، عناصر آن را از ورودی می‌خواند و سپس حاصل ضرب عناصر غیر محیطی که صفر نیستند را محاسبه می‌کند و نمایش می‌دهد. عناصر غیر محیطی عناصری هستند که در سطر اول، سطر آخر، ستون اول و ستون آخر نباشند.

➤ برنامه‌ای که یک ماتریس 4×4 را می‌خواند. اگر مجموع عناصر روی قطر اصلی و فرعی برابر مجموع عناصر محیطی ماتریس باشد، برنامه "Yes"، وگرنه "No" را نمایش می‌دهد. عناصری روی قطر اصلی قرار دارند، که $i == j$ باشد (یعنی، شماره سطر برابر شماره ستون باشد) و عناصری روی قطر فرعی قرار دارند که $i + j == n - 1$ باشد (یعنی، مجموع شماره سطر و شماره ستون برابر با تعداد سطرها منتهی یک باشد. اما، عناصر سطر اول، سطر آخر، ستون اول و ستون آخر عناصر محیطی نام دارند).

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۸۷

- برنامه‌ای که یک ماتریس 4×4 را خوانده، اگر حداقل یک عنصر ماتریس تکراری باشد (بیش از یک‌بار تکرار شده باشد)، عبارت "Yes" وگرنه، عبارت "No" را نمایش می‌دهد.
- برنامه‌ای که یک ماتریس 4×4 را خوانده، اگر همه عناصر آرایه برابر باشند، "Yes"، وگرنه "No" را نمایش می‌دهد.
- برنامه‌ای که یک ماتریس 4×4 را خوانده، اگر تمام عناصر غیر از عناصر محیطی آن باهم برابر باشند، "Yes"، وگرنه "No" را نمایش می‌دهد.
- برنامه‌ای که عناصر آرایه 4×4 را خوانده، اگر عناصر هر سطر به صورت صعودی مرتب باشند، "Yes"، وگرنه "No" را برمی‌گرداند.
- برنامه‌ای که یک آرایه 4×4 را می‌خواند. اگر تمام عناصر آن به صورت صعودی مرتب باشند، عبارت "Yes"، وگرنه "No" را نمایش می‌دهد.
- برنامه‌ای که یک ماتریس 4×4 را خوانده، مجموع عناصر قطر اصلی و فرعی را نمایش می‌دهد. (در عناصر قطر اصلی آرایه $z == i$ است و اگر عنصری روی قطر فرعی باشد، $i - 1 = z + 1$ است. i شماره سطر و z شماره ستون آرایه دوبعدی است).
- برنامه‌ای که عناصر یک ماتریس 5×5 را خوانده، بزرگ‌ترین عنصر محیطی آن را برمی‌گرداند.
- برنامه‌ای که یک رشته و عدد k را خوانده، تمام حروف الفبای رشته را به صورت زیر رمز کرده، نمایش می‌دهد.
$$\%26(k + \text{حرف اصلی}) = \text{حرف رمز شده}$$
$$\% \text{، باقی مانده تقسیم صحیح به } 26 \text{ است.}$$
- برنامه‌ای که متن رمز شده و عدد k را خوانده، متن ساده را با روش زیر به دست می‌آورد و نمایش می‌دهد (یعنی، تمام حروف الفبایی را به صورت زیر رمزگشایی می‌کند):
$$\%26(k - \text{حرف رمز شده}) = \text{حرف ساده}$$
- برنامه‌ای که تعدادی عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس مجموع اعداد اول در این اعداد را محاسبه می‌کند و در پایان، اعدادی که بزرگ‌تر از مجموع اعداد اول هستند را چاپ می‌کند.
- برنامه‌ای که ۱۰ عدد تصادفی بین ۱ تا ۴۰ تولید کرده، در آرایه اول قرار می‌دهد. سپس، ۱۰ عدد تصادفی دیگر بین ۱ تا ۵۰ تولید نموده، در آرایه دوم قرار می‌دهد. در

پایان، عناصری که در آرایه اول وجود دارند، ولی در آرایه دوم نیستند را نمایش می‌دهد. یعنی تفاضل دو آرایه را نمایش می‌دهد.

➤ برنامه‌ای که ۱۰ عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد بین ۰ تا ۸ گرفته، عناصر آرایه را به تعداد عدد خوانده شده به سمت چپ چرخش می‌دهد و چاپ می‌کند.

➤ برنامه‌ای که یک رشته را خوانده، تمام کلمات دوحرفی رشته را حذف کرده، آن را نمایش می‌دهد.

➤ برنامه‌ای که یک رشته را خوانده، تعداد کلمات یک حرفی، دوحرفی، سه حرفی، چهار حرفی و غیره را می‌شمارد.

➤ برنامه‌ای که ۱۰ عدد را خوانده، در آرایه‌ای قرار دهد. سپس تعیین می‌کند چه عددی بیش از همه تکرار شده است.

➤ برنامه‌ای که یک رشته را خوانده، کلیه جاهای خالی از قبیل کاراکتر 'blank'، '\n' و '\t' رشته را حذف می‌کند و نمایش می‌دهد.

➤ یک عدد صحیح اول است، وقتی که بر خودش و یک بخش پذیر باشد. غربال اراتوستن روشی را برای تعیین عدد اول دارد. در این روش آرایه‌ای در نظر می‌گیرد و اعمال زیر را بر روی آن انجام می‌دهد تا اعداد اول را تعیین کند:

۱. مقدار اولیه همه عناصر آرایه را برابر یک قرار می‌دهد.
 ۲. با شروع از اندیس ۲ (اندیس یک حتماً اول است). هر بار که عنصری با مقدار یک یافت می‌شود، بقیه عناصری که اندیسشان مضربی از ۲ باشد (اندیس‌های ۴، ۶، ۸، ۱۰ و ...) را برابر صفر قرار می‌دهد. برای اندیس ۳، همه عناصر بعد از ۳ که اندیسشان مضربی از ۳ است (اندیس‌های ۶، ۹، ۱۲، ۱۵ و ...) صفر قرار می‌گیرد و به همین ترتیب این روند ادامه می‌یابد.
 ۳. در پایان عناصری که هنوز یک هستند، شماره اندیس‌هایشان اول اند.
- برنامه‌ای که با استفاده از یک آرایه ۱۰۰۰ تایی، اعداد اول ۱ تا ۹۹۹ را به دست آورده، نمایش می‌دهد.

➤ برنامه‌ای که آرایه‌ای ۱۰ عنصری با مقادیر اولیه را تعریف کرده و به روش بازگشتی مجموع عناصر آرایه را محاسبه و نمایش می‌دهد.

➤ برنامه‌ای که مقادیر آرایه ۱۰ عنصری که مقدار اولیه دارد، m واحد به سمت راست شیفت می‌دهد.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۸۹

- مند بازگشتی که ترکیبات یک مجموعه را برمی‌گرداند و نمایش می‌دهد. این متد دارای سه پارامتر است که عبارت‌اند از: شروع آرایه (مجموعه)، تعداد عناصر مجموعه و ترکیبات مجموعه. سپس، برنامه‌ای که تعداد ترکیبات مجموعه را نمایش می‌دهد. به‌عنوان مثال، مجموعه $\{A, B, C\}$ را در نظر بگیرید. برای این مجموعه حالت‌های زیر را داریم:
 $\{AAA, AAB, AAC, ABA, ABB, ABC, ACB, ACC, \dots\}$
- همان‌طور که می‌دانید فاکتوریل اعداد بزرگ در یک `double` جا نمی‌شود. برنامه‌ای که فاکتوریل اعداد بزرگ را به کمک آرایه محاسبه می‌کند و نمایش می‌دهد.
- برنامه‌ای که دو آرایه 3×4 به نام‌های x و y را خوانده، حاصل عبارت زیر را در آرایه دیگر به نام z قرار می‌دهد:
$$Z = 2x - 3y$$
- برنامه‌ای که بین تعدادی عدد که در آرایه قرار دارند، بزرگ‌ترین، کوچک‌ترین مقدار و مکان‌های آن‌ها را پیدا کرده، نمایش می‌دهد.
- برنامه‌ای که ۲۰ عدد تصادفی (دو سری اعداد تصادفی ۱۰ تایی) ایجاد کرده، در دو آرایه قرار می‌دهد. سپس یک عدد را خوانده، تمام عناصر آرایه دوم را در آرایه اول از مکان واردشده درج می‌نماید.
- برنامه‌ای که یک رشته را خوانده، سپس دو عدد را می‌خواند و از کاراکترهای عدد اول واردشده تا عدد دوم واردشده رشته ورودی را حذف می‌کند (یعنی، بخشی از رشته را حذف می‌نماید).
- برنامه‌ای که تعداد اعداد تصادفی تولید کرده، در آرایه 4×4 قرار می‌دهد. سپس هر سطر آرایه را جداگانه مرتب می‌نماید. در پایان، آرایه‌ای که عناصر سطرهای آن مرتب‌شده‌اند را نمایش می‌دهد.
- برنامه‌ای که آرایه ۵ عنصری را خوانده، به مقدار هر یک از عناصر آرایه توسط علامت ستار (*) میله افقی رسم می‌کند.
- برنامه‌ای که ۷ عدد را خوانده، سپس دو عدد دیگر را می‌خواند و از این دو عدد خوانده‌شده، عدد دوم را جایگزین عدد اول در آن هفت عدد می‌نماید.
- برنامه‌ای که عدد صحیح n (کوچک‌تر از ۹) را از ورودی خوانده، خروجی زیر را با استفاده از آرایه دوبعدی (ماتریس) تولید و چاپ می‌کند:

1	2	3	...	n
2	3	4	...	n+1
...
n+1	n+2	n+3	...	2n

➤ برنامه‌ای که رشته‌ای را خوانده، با استفاده از یک متد فضای خالی سمت چپ (ابتدا رشته) را حذف می‌کند و نمایش می‌دهد (کد اسکی کاراکترهای خالی ۳۲ است).

➤ برنامه‌ای که رشته‌ای را خوانده، با استفاده از اشاره‌گر و بازگشتی، طول رشته را حساب کرده، نمایش می‌دهد.

➤ **پروژه برنامه‌نویسی ۱:** یک پروژه هواپیمایی قصد دارد سیستم ذخیره و فروش بلیط را کامپیوتری کند. برنامه سیستم رزرو بلیط مراحل زیر را انجام می‌دهد (تعداد صندلی ۴۰ می‌باشد که شماره ۱ تا ۲۰ برای سیگاری‌ها و از ۲۱ تا ۴۰ برای غیر سیگاری‌ها در نظر گرفته می‌شود. در ضمن حداکثر ۱۰ شماره پرواز از ۱ تا ۱۰ داریم):

الف: آرایه دوبعدی با ۴۰ سطر و ۱۰ ستون در نظر گرفته شده که ستون‌های صفر تا ۹ شماره پروازهای ۱ تا ۱۰ را تعیین می‌کنند.

ب: شماره مسافر، نوع مسافر (سیگاری (۱)، غیر سیگاری (۰)) و شماره پرواز را دریافت می‌کند، سپس بلیط را رزرو می‌کند.

ج: اگر نوع مسافر سیگاری باشد، با توجه به شماره پرواز، به آن شمار پرواز در بخش سیگاری صندلی تخصیص می‌دهد. در صورتی که پر بودن بخش سیگاری‌ها از مشتری سؤال می‌کند آیا در بخش غیر سیگاری‌ها پذیرفته شود یا خیر و برعکس.

د: شماره مسافر به شماره صندلی در آرایه تخصیص می‌یابد.

سپس کلید n را فشار دهید تا وضعیت صندلی‌های پرواز را به شکل زیر ببینید:

➤ **پروژه برنامه‌نویسی ۲:** برنامه‌ای که منوی نمایش می‌دهد. از این منو کاربر می‌تواند یکی از گزینه‌های زیر را انتخاب کند:

۱ Read Array، اطلاعات آرایه اول را می‌خواند.

۲ Read Array، اطلاعات آرایه دوم را می‌خواند.

Find Share، اشتراک بین آرایه اول و دوم را پیدا می‌کند.

Find Union، اجتماع بین آرایه اول و دوم را پیدا می‌کند.

Find difference a Form b، تفاضل آرایه a را از b حساب می‌کند.

Find difference b Form a، تفاضل آرایه b را از a حساب می‌کند.

➤ **پروژه برنامه‌نویسی ۳:** برنامه‌ای که دو آرایه 3×3 را خوانده، دو آرایه را با هم جمع، تفریق و ضرب می‌نماید. در این برنامه امکاناتی برای نمایش دو آرایه دارد. در ضمن

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۹۱

این برنامه می‌تواند N, M و k را بخواند. سپس حاصل عبارت $N * A - M * B + K$ را محاسبه کنند و نمایش دهد (A و B به ترتیب آرایه‌های اول و دوم هستند).

➤ پروژه برنامه‌نویسی ۴: برنامه‌ای که عناصر یک ماتریس سه در سه را خوانده، اعمال زیر را انجام می‌دهد:

- تعیین می‌کند ماتریس بالا مثلثی است یا نه؟
- تعیین می‌کند ماتریس پایین مثلثی است یا نه؟
- تعیین می‌کند ماتریس بالا و پایین مثلثی است یا نه؟
- تعیین می‌کند که آیا تمام عناصر هر سطر ماتریس برابرند یا نه؟
- تعیین می‌کند که تمام عناصر هر ستون ماتریس برابرند یا نه؟

➤ پروژه برنامه‌نویسی ۵. فرض کنید بخواهیم وضعیت مدال‌های بین‌المللی یک کشور را در رشته ورزشی خاصی در مهر و موم‌های ۱۳۸۵ تا ۱۳۹۵ مطابق جدول زیر دریافت و پردازش کنیم:

سال	تعداد مدال طلا	تعداد مدال نقره	تعداد مدال برنز
۱۳۸۵			
۱۳۷۱			
۱۳۷۲			
۱۳۹۴			
۱۳۹۵			

برنامه‌ای که ضمن دریافت داده‌های لازم (خانه‌های خالی طول) موارد زیر را نمایش می‌دهد.

- تعداد کل مدال‌های کسب‌شده در همه‌ی سال‌ها
- تعداد کل مدال‌های طلای کسب‌شده در همه‌ی سال‌ها
- شماره سال‌هایی که هیچ مدالی کسب نشده
- سالی که بیش‌ترین مدال کسب‌شده
- سالی که بیش‌ترین مدال طلا کسب‌شده است.
- امتیاز تمام سال‌ها به تفکیک سال، با فرض این که هر مدال طلا ۳ امتیاز، هر مدال نقره ۲ امتیاز و هر مدال برنز ۱ امتیاز داشته باشد.
- تعداد سال‌هایی که امتیاز بیش از ۱۰ کسب‌شده است.

➤ پروژه برنامه‌نویسی ۶. در یک تورنمنت ژیمناستیک با شرکت ۱۰ ورزش‌کار و ۵ داور امتیاز منظر خود را در یک جدولی به شکل زیر یادداشت می‌کنند. ملاک انتخاب قهرمان امتیازات داده‌شده توسط داوران به هر ورزشکار است. گزارش گردید یکی از داوران از یک ورزشکار رشوه گرفته تا به قهرمان شدن او کمک کند. اما داور و ورزشکار متخلف شناسایی نشدند. با توجه به این که داور متخلف امتیازات پرت تری نسبت به همکارانش ثبت کرده است، کمیته برگزاری تصمیم گرفته با بررسی داده‌ها او را شناسایی کرده و امتیازات داده‌شده توسط وی را از جدول حذف نموده و در نهایت نام قهرمان واقعی و نام ورزشکار رشوه دهنده را مشخص کند. این برنامه‌ای کار را انجام می‌دهد.

شماره پیراهن	نام ورزشکار	امتیاز داور اول	امتیاز داور دوم	امتیاز داور پنجم
۱					
۲					
۳					
:					
۹					
۱۰					

۱. برنامه‌ای که کلاس **Circle** را طوری پیاده‌سازی می‌کند که دو سازنده دارد. سازنده اول، سازنده پیش‌فرض را پیاده‌سازی می‌کند و سازنده دوم شعاع دایره را به‌عنوان پارامتر دریافت می‌کند و در فیلد شعاع (**r**) قرار می‌دهد.

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاس **Circle** را به پروژه اضافه کرده، دستورات آن را به‌صورت زیر تغییر دهید:

```
using System;
package ch5_1;
public class Circle {
    public static final double PI = Math.PI;
    private int r;
    public final int getR()
    {
        return r;
    }
    public final void setR(int value)
    {
        if (value > 0)
        {
            r = value;
        }
        else
        {
            r = 0;
        }
    }
    public Circle()
    {
        setR(0);
    }
    public Circle(int r1)
    {
        setR(r1);
    }
    public final double Area()
    {
        return PI * r * r;
    }
    public final double Perime()
    {
        return 2 * PI * r;
    }
}
```

۳. به کلاس اصلی برگردید و دستورات آن را به‌صورت زیر تغییر دهید:

```
package ch5_1;
import java.util.Scanner;
public class Ch5_1 {
```

```
public static void main(String[] args) {
    Circle c1 = new Circle();
    System.out.printf("Radis is %1$s\tArea is %2$s\tPerime is
        %3$s" + "\n", c1.getR(), c1.Area(), c1.Perime());
    System.out.print("Enter Radius:");
    Circle c2 = new Circle(Integer.parseInt(new
        Scanner(System.in).nextLine()));
    System.out.printf("Radis is %1$s\tArea is %2$s\tPerime is
        %3$s" + "\n", c2.getR(), c2.Area(), c2.Perime());
}
}
```

سازنده بدون پارامتر فراخوانی می شود c1 ایجاد می شود. برای ایجاد نمونه c1 در این دستورات ابتدا نمونه نمایش می دهد. سپس شعاع را از c1 تا شعاع برابر صفر شود. در ادامه شعاع، محیط و مساحت را برای نمونه c2 را ایجاد می کند که برای ایجاد نمونه c2 کاربر درخواست می کند. با شعاع وارد شده توسط کاربر، نمونه را نمایش می دهد. c2 سازنده با یک پارامتر فراخوانی می کند و در پایان شعاع، محیط و مساحت نمونه ۴. پروژه را ذخیره و اجرا کنید تا خروجی زیر را مشاهده نمایید:

```
Radis is ۰ Area is ۰,۰ Perime is ۰,۰
Enter Radius:۱۰
Radis is ۱۰ Area is ۳۱۴,۱۵۹۲۶۵۳۵۸۹۷۹۳ Perime is ۶۲,۸۳۱۸۵۳۰۷۱۷۹۵۸۶
```

۲. برنامه ای که تعداد اشیاء ایجاد شده یک کلاس را شمارش می کند و نمایش می دهد.

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاسی به نام Counter ایجاد کرده، دستورات آن را به صورت زیر تغییر دهید:

```
package ch5_2;
public class Counter {
    public static int countObject = 0;
    public Counter()
    {
        countObject++;
    }
    protected void finalize() throws Throwable
    {
        --countObject;
    }
}
```

دارای اعضای زیر است: Counter کلاس

✚ **عضو استاتیک countObject**. تعداد نمونه های (اشیا) ایجاد شده کلاس را نگه داری می کند.

✚ **سازنده Counter()**. به countObject یک واحد اضافه می کند تا تعداد نمونه های ایجاد شده از کلاس در این فیلد قرار گیرد.

✚ **مخرب finalize**، از فیلد countObject یک واحد کم می کند تا با تخریب (حذف) شی ای از کلاس از فیلد countObject یک واحد کم شود.

۳. به کلاس اصلی بروید و دستورات آن را به صورت زیر تغییر دهید:

```
package ch5_2;
public class Ch5_2 {
    public static void main(String[] args) {
        System.out.printf("Count Object is %1$s" + "\n",
            Counter.countObject);
    }
}
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۹۵

```
Counter c1 = new Counter();
Counter c2 = new Counter();
System.out.printf("Count Object is %1$s" + "\n",
    Counter.countObject);
}
```

را نمایش می‌دهد که صفر است. Counter، تعداد نمونه‌های ایجاد شده کلاس main دستورات اول داخل متد ایجاد می‌کنند و دستور چهارم، Counter با نوع کلاس c۲ و c۱ دستورات دوم و سوم، دو نمونه به نام‌های (یعنی، ۲) را نمایش خواهد داد. Counter تعداد نمونه‌های ایجاد شده کلاس ۴. پروژه را ذخیره و اجرا کنید تا خروجی زیر را ببینید:

```
Count Object is ۰
Count Object is ۲
```

۳. برنامه‌ای که نام کاربر و کلمه عبور را دریافت می‌کند. اگر نام کاربر fanavarienovin و کلمه عبور ۱۲۳۴۵۵۶ باشد، پیغام Correct، وگرنه پیغام Not Correct را نمایش می‌دهد. و username به نام‌های static استفاده می‌کند که دو فیلد User توضیح: این برنامه از کلاسی به نام دارد تا تعیین کند، نام کاربر و کلمه عبور صحیح وارد شده است یا isCorrect() دارد و متدی به نام Password خیر؟

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاسی User را به برنامه اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
package ch5_3;
public class User {
    private static String UserName = "fanavarienovin";
    private static String Password = "1234556";
    public static boolean isCorrect(String userName, String password)
    {
        return (UserName.equals(userName) &&
            Password.equals(password));
    }
}
```

۳. به کلاس اصلی بروید و دستورات آن را به صورت زیر تغییر دهید:

```
package ch5_3;
import java.util.Scanner;
public class Ch5_3 {
    public static void main(String[] args) {
        System.out.print("Enter user name:");
        String userName = new Scanner(System.in).nextLine();
        System.out.print("Enter password:");
        String password = new Scanner(System.in).nextLine();
        if (User.isCorrect(userName, password) == true)
        {
            System.out.println("Correct");
        }
        else
        {
            System.out.println("Not Correct");
        }
    }
}
```

، با پیغام‌هایی، نام کاربر و کلمه عبور را خوانده، در متغیرهای main دستورات اول تا چهارم درون متد و userName را با دو پارامتر (isCorrect)، متد if قرار می‌دهند و دستور پنجم (userName و Password) استفاده نموده است User فراخوانی می‌کند. دقت کنید در هنگام فراخوانی از نام کلاس (یعنی، Password (نمونه‌ای از آن ایجاد نکرد).

۴. پروژه را ذخیره و اجرا کنید تا نمونه خروجی زیر را مشاهده نمایید:

```
Enter user name:fanavarienovin
Enter password:۱۲۳۴۵۵۶
Correct
```

۴. برنامه‌ای که شعاع کره‌ای را دریافت کرده، مساحت و حجم کره را محاسبه می‌نماید و نمایش می‌دهد. حجم و مساحت کره با فرمول‌های زیر محاسبه می‌شود:

$$= 3,1415 * r^2 * \frac{4}{3} \quad \text{مساحت کره} \quad = 3,1415 * r^2 * \frac{4}{3} \quad \text{حجم کره}$$

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاس Sphere را به پروژه اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
package ch5_4;
public class Sphere {
    public static final double PI = 3.1415;
    public int r;
    public Sphere(int r)
    {
        if (r > 0)
        {
            this.r = r;
        }
        else
        {
            this.r = 0;
        }
    }
    private long Pow(int r, int n)
    {
        long p = 1;
        for (int i = 1; i <= n; i++)
        {
            p *= r;
        }
        return p;
    }
    public final double Volume()
    {
        return (4.0 / 3 * Pow(r, 3) * PI);
    }
    public final double Area()
    {
        return (4.0 * Pow(r, 2) * PI);
    }
}
```

این کلاس دارای اعضای زیر می‌باشد:

ثابت PI، مقدار ۳,۱۴۱۵ را نگهداری می‌کند.

فیلد r، شعاع کره را نگهداری می‌کند.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۹۷

- سازنده Sphere، مقدار r را به عنوان پارامتر دریافت کرده، در فیلد شعاع (r) قرار می‌دهد (چون نام متغیر با نام فیلد یکی می‌باشد از اشاره گر this استفاده شده است).
 - متد Pow()، پارامتر r و n را گرفته، r^n را برمی‌گرداند.
 - متد Volume()، حجم کره را حساب کرده، برمی‌گرداند.
 - متد Area()، مساحت کره را محاسبه کرده، برمی‌گرداند.
۳. به کلاس اصلی بروید و دستورات آن را به صورت زیر تغییر دهید:

```
package ch5_4;
import java.util.Scanner;
public class Ch5_4 {
    public static void main(String[] args) {
        System.out.print("Enter r:");
        Sphere s=new Sphere(Integer.parseInt(new
            Scanner(System.in).nextLine()));
        System.out.printf("Volume is%1$s\tArea is%2$s" + "\n",
            s.Volume(), s.Area());
    }
}
```

یک پیغام نمایش می‌دهد تا کاربر شعاع کره را وارد کند. دستور دوم با شعاع main() دستور اول داخل ایجاد می‌کند و دستور سوم، مساحت و حجم کره را محاسبه Sphere از کلاس s دریافتی یک نمونه به نام کرده، نمایش می‌دهد.

۴. پروژه را ذخیره و اجرا کنید تا نمونه خروجی زیر را مشاهده نمایید:

```
Enter r:۱۰
Volume is۴۱۸۸,۶۶۶۶۶۶۶۶۶۶۶۷      Area is۱۲۵۶,۶۰۰۰۰۰۰۰۰۰۰۱
```

➤ برنامه‌ای که کلاسی طراحی می‌کند که اطلاعات زیر را دارد:

- شماره استادی
 - نام استاد
 - نام خانوادگی استاد
 - ساعات تدریس
 - مبلغ پرداختی به ازای هر ساعت تدریس
- در این کلاس مبلغ حق التدریس استاد را محاسبه می‌کند که به صورت زیر محاسبه می‌شود:
- مبلغ حق التدریس = ساعت تدریس * مبلغ پرداختی به ازای هر ساعت تدریس
- برنامه‌ای که از این کلاس استفاده می‌کند.

➤ برنامه‌ای که دو زمان را مقایسه می‌کند. در این برنامه از کلاسی به نام Time استفاده شده است. این کلاس دارای سه فیلد hour (ساعت)، minute (دقیقه)، second (ثانیه)، دو سازنده و یک متد به نام isEqual() است که دو شی از نوع Time را مقایسه می‌کند.

➤ برنامه‌ای که پیاده‌سازی ایندکسرها را برای ذخیره و بازیابی چند نام نشان می‌دهد (هدف این برنامه آشنایی با Indexer، پیاده‌سازی و تعریف مجدد آن می‌باشد).

➤ برنامه‌ای که پیاده‌سازی و استفاده از delegate را نشان می‌دهد.

➤ برنامه‌ای که کلاس مربع را پیاده‌سازی می‌کند. برای این منظور کلاس Square را ایجاد می‌نماید. متغیر خصوصی side را در این کلاس ایجاد می‌کند که توسط متدهای set و get بتوان به آن دسترسی یافت. کلاس باید شامل دو متد سازنده باشد. یکی از سازنده‌ها آرگومان ندارد و متد سازنده دیگر مقدار side را باید به عنوان آرگومان بپذیرد.

➤ برنامه‌ای که کلاسی به نام SavingAccount ایجاد می‌کند. این کلاس از متغیر استاتیکی به نام annualInterestRate برای ذخیره سازی نرخ سود همه حساب‌ها استفاده می‌کند. هر شی از این کلاس شامل متغیر خصوصی به نام SavingBalance می‌باشد که مقدار پس‌انداز جاری سپرده را در خود نگه می‌دارد. متدی به نام CalculateModifyInterest() دارد که سود ماهانه را با فرمول زیر محاسبه می‌نماید:

$$(SavingBalance * annualInterestRate) / ۱۲$$

اضافه می‌نماید. کلاس متد استاتیکی به نام SavingBalance سپس سود را به یک مقدار جدید annualInterestRate که به مقدار فیلد ModifyInterestRate تخصیص می‌دهد. برنامه‌ای که از این کلاس استفاده می‌کند. در این برنامه دو شیء به نام - ایجاد می‌کند و به هر یک تراز ۲۰۰۰۰۰ دلار و ۳۰۰۰۰۰ دلار نسبت saver۲ و saver۱ می‌دهد. مقدار ۴٪ قرار می‌دهد و سپس سود ماهانه را حساب می‌کند. مقدار محاسبه کرده، نمایش می‌دهد. سپس مقدار saver۲ و saver۱ را برای ۵٪ قرار می‌دهد و سود ماه بعد را حساب می‌نماید و تراز جدید annualInterestRate (است). savingBalance را برای هر یک از حساب‌ها نشان می‌دهد (تراز همان فیلد

➤ کلاسی به نام ChkNum که دو متد به نام‌های IsPrime() و LeastComFactor() دارد. متد IsPrime() پارامتری به نام x را دریافت کرده، تعیین می‌کند اول است یا خیر و متد LeastComFactor()، دو پارامتر به نام‌های a و b را دریافت می‌کند و کوچکترین فاکتور مشترک بین آن‌ها را برمی‌گرداند (کوچکترین فاکتور مشترک، کوچکترین عدد اولی است که به هر دو پارامتر a و b بخش پذیر باشد).

➤ اگر طول دو ضلع (C و B) از مثلث و زاویه بین این دو ضلع Alpha باشد. می‌توان طول ضلع سوم را با فرمول زیر حساب کرد:

$$A^2 = B^2 + C^2 - 2 * B * C * Cos(Alpha)$$

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۹۹

را با مندهای مساحت و A را دریافت می کند. طول ضلع سوم (Alpha و C، B کلاسی که محیط را حساب کرده، نمایش می دهد. مساحت و محیط مثلث به صورت زیر حساب می شوند.

$$\text{محیط} = (a + b + c) \quad p = \frac{\text{مساحت}}{2} = \sqrt{p(p-a)(p-b)(p-c)}$$

- برنامه ای که با استفاده از Delegate و متدهای متعدد، حرف اول، آخر و کلیه حروف رشته را به حروف بزرگ تبدیل می کند.
- برنامه ای که تعدادی (۴ عدد) را خوانده، به روش صعودی، و نزولی مرتب می نماید و نمایش می دهد (روش مرتب سازی را به عنوان یک پارامتر از نوع delegate دریافت می کند).
- برنامه ای که نقطه شروع و پایان خطی را دریافت می کند، طول خط را محاسبه کرده، نمایش می دهد.
- برنامه ای که مجموع مساحت ها و محیط های چند مربع را نمایش می دهد (در این برنامه تعداد مربع ها و اضلاع آنها به صورت تصادفی تولید می گردند).
- برنامه ای که استفاده از delegate را نشان می دهد.
- برنامه ای که اندازه پارچه را به یارد دریافت کرده، به متر تبدیل می کند (هر متر ۱٫۱۹۶ یارد است).
- برنامه ای که قاعده و ارتفاع مثلثی را دریافت کرده، محیط و مساحت مثلث را حساب می کند. محیط و مساحت مثلث به صورت زیر محاسبه می گردد:
$$\text{محیط مثلث} = \text{ارتفاع} + \text{قاعده} + \sqrt{(\text{ارتفاع})^2 + (\text{قاعده})^2}$$
$$\text{مساحت مثلث} = \frac{\text{ارتفاع} \times \text{قاعده}}{2}$$
- برنامه ای که زمان را دریافت کرده و آن را به صورت Am (قبل از ظهر) یا Pm (بعد از ظهر) نمایش می دهد. این برنامه کلاس های زیر را دارد:
- برنامه ای که شماره کارمندی، نام، تاریخ استخدام و حقوق ناخالص تعدادی کارمند را مقدار دهی می کند، مالیات، بیمه و حقوق خالص را محاسبه می کند. در پایان مجموع

حقوق خالص را نمایش می دهد. مالیات، بیمه و حقوق خالص به صورت زیر محاسبه می گردند.

$$۱۰۰ / ۷ * \text{حقوق ناخالص} = \text{بیمه}$$

اگر حقوق بیشتر از ۴۸۵۰۰۰۰ ریال باشد، آنگاه مالیات برابر است با

$$۱۰۰ / ۱۰ * \text{حقوق ناخالص} = \text{مالیات}$$

وگرنه، مالیات صفر در نظر گرفته می شود.

$$\text{بیمه} - \text{مالیات} - \text{حقوق ناخالص} = \text{حقوق خالص}$$

➤ برنامه ای که شعاع بزرگ (طول) و شعاع کوچک (عرضی) بیضی را دریافت کرده، مساحت بیضی را با فرمول زیر محاسبه می کند:

$$\text{مساحت} = \pi \times \text{شعاع کوچک} * \text{شعاع بزرگ}$$

➤ برنامه ای که کلاس های زیر را تعریف کرده و از آن ها استفاده می کند:

➤ برنامه ای که کلاس های مربع (Square) و مستطیل را پیاده سازی می کند. کلاس مستطیل از کلاس مربع مشتق می شود.

➤ برنامه ای که کلاس های زیر را ایجاد کرده، از آن ها استفاده می کند:

- کلاس Shape، کلاسی abstract است که فقط دو متد Area() (برای محاسبه مساحت) و Volume (برای محاسبه حجم) دارد.
- کلاس Point، مشتق کلاس Shape است که دارای اعضای داده ای x و y است.

- کلاس Circle، مشتق کلاس Point است. علاوه بر این که اعضای کلاس Point را به ارث می برد، دارای اعضای زیر می باشد:

- عضو داده ای شعاع (radius)، شعاع دایره را نگهداری می کند.

- متد Area()، برای محاسبه مساحت دایره به کار می رود.

- کلاس Cylinder (استوانه)، مشتق کلاس Circle است که علاوه بر اعضای Circle به ارث می برد، دارای اعضای زیر نیز است:

- عضو داده ای ارتفاع (height) است.

- متد Area()، برای محاسبه مساحت استوانه به کار می رود. مساحت

استوانه به شکل زیر محاسبه می شود:

$$\text{ارتفاع استوانه} * \text{محیط دایره} + ۲ * \text{مساحت دایره}$$

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۰۱

▪ متد (`Volume()`)، حجم استوانه را محاسبه کرده، برمی‌گرداند. حجم استوانه برابر است با:

ارتفاع * مساحت دایره

برنامه‌ای که از این کلاس‌ها استفاده می‌کند.

➤ کلاس استاتیکی که با دو متد درجه سانتی‌گراد را به فارنهایت و درجه فارنهایت را به سانتی‌گراد تبدیل می‌کند. برنامه‌ای که از این کلاس استفاده می‌کند. برای تبدیل درجه‌ها از فرمول‌های زیر استفاده می‌شود.

درجه به فارنهایت = $۳۲ + ۹/۵ * \text{درجه سانتی‌گراد}$

درجه به سانتی‌گراد = $۵ / ۹ * (\text{درجه فارنهایت} - ۳۲)$

در این برنامه منویی نمایش داده می‌شود تا نوع درجه را دریافت کند.

➤ برنامه‌ای که کد کالا، نام کالا، تعداد فروش، قیمت را دریافت می‌کند و قیمت فروش را محاسبه کرده، نمایش می‌دهد. در این برنامه از یک واسط به نام `IProduct` استفاده شده است.

۱. برنامه‌ای که چند جمله را دریافت کرده، در فایلی که نام آن را از ورودی می‌خواند، می‌نویسد.

```
package ch6_1;
import java.io.IOException;
import java.io.FileWriter;
import java.util.Scanner;
public class Ch6_1 {
    public static void main(String[] args) throws IOException {
        String sentence = " ";
        System.out.print("Enter file name:");
        String filename = new Scanner(System.in).nextLine();
        FileWriter myFile = new FileWriter(filename, false);
        do
        {
            System.out.print("Enter a sentence: ");
            sentence = new Scanner(System.in).nextLine();
            if (sentence.length() != 0)
            {
                sentence=sentence+System.lineSeparator();
                myFile.write(sentence );
            }
        } while (sentence.length() != 0);
        myFile.close();
    }
}
```

را با myFile برای دریافت جملات تعریف می‌کند، سپس متغیر sentence این برنامه ابتدا متغیر رشته‌ای تعریف می‌نماید و فایل متنی که نام فایل را از ورودی می‌خواند، ایجاد می‌کند. در ادامه در FileWriter نوع sentence قرار می‌دهد و چنانچه طول رشته sentence بیک حلقه با پیغام مناسب جمله‌ای را خوانده، در متغیر می‌نویسد و این روند را ادامه می‌دهد تا کاربر رشته‌ای با طول صفر وارد کند. در myFile صفر نباشد، آن را در را می‌بندد. myFile پایان فایل

خروجی برنامه:
Enter file name:D:\DATA\Java\.txt
Enter a sentence: Java
Enter a sentence: C++
Enter a sentence: Cobol
Enter a sentence:

۲. برنامه‌ای که چند جمله را می‌خواند به انتهای فایل متنی در فایلی که نام آن را از ورودی می‌خواند، اضافه می‌کند.

```
package ch6_2;
import java.io.IOException;
import java.io.FileWriter;
import java.util.Scanner;
public class Ch6_2 {
    public static void main(String[] args) throws IOException {
        String sentence = " ";
        System.out.print("Enter file name:");
        String filename = new Scanner(System.in).nextLine();
        FileWriter myFile = new FileWriter(filename, true);
        do
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۰۳

```
{
    System.out.print("Enter a sentence: ");
    sentence = new Scanner(System.in).nextLine();
    if (sentence.length() != 0)
    {
        sentence=sentence+System.lineSeparator();
        myFile.write(sentence );
    }
} while (sentence.length() != 0);
myFile.close();
}
```

است تا بتواند به true مقدار FileWriter این برنامه مانند برنامه قبلی است، با این تفاوت که پارامتر دوم انتهای فایل اضافه کند.

خروجی برنامه:

```
Enter file name:D:\DATA\Java\ .txt
Enter a sentence: Python
Enter a sentence: C#
Enter a sentence: Php
Enter a sentence:
```

۳. برنامه ای که نام یک فایل متنی را از کاربر دریافت کرده، اطلاعات آن را بر روی صفحه نمایش نشان می‌دهد.

```
package ch6_3;
import java.io.IOException;
import java.util.*;
public class Ch6_3 {
    public static void main(String[] args) throws IOException {
        System.out.print("Enter name of file: ");
        String fileName = new Scanner(System.in).nextLine();
        java.io.FileReader myfile =new java.io.FileReader(fileName) ;
        java.io.BufferedReader myfileBufferedReader = new
            java.io.BufferedReader(myfile);
        try
        {
            myfile = new java.io.FileReader(fileName);
            String line = " ";
            do
            {
                line = myfileBufferedReader.readLine();
                if (line != null)
                {
                    System.out.println(line);
                }
            } while (line != null);
        }
        catch (RuntimeException e)
        {
            System.out.println("Error in file.");
        }
    }
}
```

قرار می‌دهد. `fileName` در این برنامه ابتدا با یک پیغام مناسب نام فایل را خوانده، در متغیر رشته‌ای تعریف می‌کند تا بتوان اطلاعات آن فایل را خواند. در ادامه `FileReader` را با نوع `myFile` سپس متغیر `line` قرار می‌دهد. در پایان، رشته `myFile` را به صورت خواندنی باز کرده و دستگیره آن را در متغیر می‌خواند و بر روی صفحه نمایش نشان می‌دهد. این کار را تا زمانی `myFile` تعریف کرده و خط به خط از

try و catch می‌شود. یعنی خطی در فایل موجود نباشد. این دستورات را بین null برابر line ادامه می‌دهد که را نمایش دهد.. Error in file عبارت catch قرار می‌دهد تا چنانچه خطای رخ دهد، در بخش

خروجی برنامه:

Enter name of file: D:\data\Java\ .txt

Java
C++
Cobol
Python
C#
Php

۴. برنامه‌ای که نام یک دایرکتوری را خوانده، لیست فایل‌های آن دایرکتوری را نمایش می‌دهد.

```
package ch6_4;
import java.io.File;
import java.util.Date;
import java.util.Scanner;
public class Ch6_4 {
    public static void main(String[] args) {
        String dirName="";
        System.out.print("Enter directory name:");
        dirName = new Scanner(System.in).nextLine();
        File file = new File(dirName);
        String[] fileList = file.listFiles();
        for(String name:fileList){
            System.out.print(name);
            System.out.print(", ");
        }
    }
}
```

fileList لیست فایل‌ها آن مسیر را در متغیر list این برنامه با پیام مناسب نام دایرکتوری را خوانده و با متد را نشان می‌دهد. fileList مقادیر for قرار داده و با

خروجی برنامه:


Enter directory name:D:\Data

```
\ - Copy.txt, \-.txt, \.bmp, \.bmp001, \.bmp002,
\.bmp003, \.bmp004, \.cs, \.csv, \.dat, \.Enc,
\.gif, \.Inv, \.pas, \.png, \.txt, \.txt.dat,
\.Upp, \11.txt, \Copy.txt, \rev.dat, \Rev.txt, \2.bmp,
\2.gif, \2.txt, \3.txt, \4.txt, \5.txt, \6.txt, \c7.txt,
cc\1.txt, copy-\1.txt, Java\1.txt, Kol.txt, list.txt,
log.txt, newFile.txt, out.dat, p4.cpp, Program.cs,
program.java, test.txt, BUILD SUCCESSFUL (total time: 6
seconds)
```

۵. برنامه‌ای که نام یک فایل متنی را خوانده و اطلاعات فایل را بر روی صفحه نمایش نشان می‌دهد. چنانچه تعداد خطوط فایل بیش از ۲۲ سطر باشد، پس از نمایش هر ۲۲ سطر یک مکث موقت می‌کند تا کاربر کلیدی را فشار دهد.

متد `showData()` نام فایلی که می‌خواهد نمایش دهد را با پارامتر `urlFile` دریافت می‌کند و اطلاعات آن فایل را ۲۲ خط ۲۲ خط نمایش می‌دهد. برای این منظور، ابتدا فایل متنی که نام آن در `urlFile` قرار دارد را به صورت خواندنی باز کرده و اشاره‌گر آن در متغیر `myFile` با نوع `FileReader` قرار می‌دهد و متغیر `count` (تعداد خطوط نمایش داده شده) را تعریف می‌کند و مقدار صفر را در آن قرار می‌دهد. در ادامه با استفاده از یک حلقه `do.... while` خط به خط از فایل `myFile` خوانده و بر روی صفحه نمایش نشان می‌دهد. اگر تعداد خطوط نمایش داده شده ۲۲ شود (یعنی ۲۲٪ `count` برابر صفر

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۰۵

گردد، یک پیغام Press any key to continue.... را نمایش می‌دهد و این روند را تا زمانی ادامه می‌دهد که تمام خطوط فایل خوانده شود. در پایان، فایل myFile را می‌بندد.  متد main()، با پیغام مناسبی نام فایل را خوانده در متغیر fileName قرار می‌دهد و با فراخوانی ShowData() اطلاعات فایلی که نام آن در fileName است را به صورت صفحه به صفحه نمایش می‌دهد.

```
package ch6_5;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.*;
import java.io.FileReader;
public class Ch6_5 {
    public static void ShowData(String urlFile) throws
        FileNotFoundException, IOException
    {
        FileReader myFile =new FileReader(urlFile) ;
        BufferedReader myfileBufferedReader = new
            BufferedReader(myFile);
        String line="";
        int count = 0;
        do
        {
            line = myfileBufferedReader.readLine();
            if (line != null)
            {
                if (count != 0 && count % 22 == 0)
                {
                    System.out.println("Perss any key to continue ... ");
                    new Scanner(System.in).nextLine();
                }
                if (line.length() > 79)
                {
                    line = line.substring(0, 79);
                }
                System.out.println(line);
            }
            count++;
        } while (line != null);
    }
    public static void main(String[] args) throws IOException {
        System.out.print("Enter name of file: ");
        String fileName = new Scanner(System.in).nextLine();
        ShowData(fileName);
    }
}
```

خروجی برنامه:

```
Enter name of file: D:\Data\.\.txt
PYTHON IS A HIGH LEVEL LANGUAGE.
C# IS A GOOD LANGUAGE.
CPP IS A VERY GOOD LANGUAGE.
JAVA IS A GOOD HIGH LEVEL LANGUAGE.
PASCAL IS A LANGUAGE.
C is a high level language.
C++ is a good language.
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
C is a high level language.
```

```
C++ is a good language.
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
C is a high level language.
C++ is a good language.
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
C is a high level language.
C++ is a good language.
Perss any key to continue ...
```

```
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
C is a high level language.
C++ is a good language.
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
C is a high level language.
C++ is a good language.
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
C is a high level language.
C++ is a good language.
C# is a very good language.
Delphi is a windows UI language.
Java is a object oriented language.
```

۶. برنامه‌ای که نام یک فایل متنی را از کاربر دریافت کرده، سپس دو رشته را می‌خواند و در فایل متنی به‌جای رشته اول رشته دوم خوانده شده را جایگزین می‌کند و نتیجه را در فایلی به نام ۲.txt در پوشه Data درایو D می‌نویسد.

متد `main()` ابتدا با پیغام مناسب نام یک فایل متنی را خوانده در متغیر `fileName` قرار می‌دهد، سپس با پیغام دیگری رشته اول را می‌خواند و در متغیر `oldString` قرار می‌دهد، در ادامه با پیغام مناسب دیگری رشته دوم را می‌خواند و در متغیر `newString` قرار می‌دهد و در پایان با فراخوانی متد `Replace()` به‌جای رشته `oldString` رشته `newString` در فایل `fileName` جایگزین می‌کند.

متد `Replace()` نام فایل `(urlFile)`، رشته قدیم `(textReplace)` و رشته جدید `(newText)` را به‌عنوان پارامتر دریافت کرده، فایلی که نام آن در `urlFile` قرار دارد را به‌صورت خواندنی باز می‌کند و اشاره‌گر آن را در `myFileRd` قرار می‌دهد و فایل `۲.txt` را در پوشه `Data` درایو `D` به‌صورت نوشتن باز کرده (آن را ایجاد می‌کند) و اشاره‌گر آن را در متغیر `myFileWr` قرار می‌دهد. سپس با استفاده از یک حلقه تکرار اطلاعات فایل `myFileRd` را به‌صورت خط به خط می‌خواند و در متغیر `line` قرار می‌دهد. چون رشته است با متد `Replace` به‌جای مقدار `textReplace` مقدار `newText` را در `line` قرار می‌دهد و اکنون اطلاعات `line` را در فایل `myFileWr` می‌نویسد. در پایان فایل‌های `myFileRd` و `myFileWr` را می‌بندد.

```
package ch6_6;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۰۷

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
public class Ch6_6 {
    public static void ReplaceTextFile(String urlFile, String
    textReplace, String newText) throws FileNotFoundException,
    IOException
    {
        FileReader myfileRd = new FileReader(urlFile);
        BufferedReader myfileRdBufferedReader = new
        BufferedReader(myfileRd);
        FileWriter myfileWr = new FileWriter("D:\\data\\2.txt");
        String line = " ";
        do
        {
            line = myfileRdBufferedReader.readLine();
            if (line != null)
            {
                line = line.replace(textReplace, newText);
                myfileWr.write(line + System.lineSeparator());
            }
        } while (line != null);
        myfileWr.close();
        myfileRd.close();
    }
    public static void main(String[] args) throws IOException {
        System.out.print("Enter name of file: ");
        String fileName = new Scanner(System.in).nextLine();
        System.out.print("Enter old string: ");
        String oldString = new Scanner(System.in).nextLine();
        System.out.print("Enter new string: ");
        String newString = new Scanner(System.in).nextLine();
        ReplaceTextFile(fileName, oldString, newString);
    }
}
```

خروجی برنامه:

```
Enter name of file: D:\data\Java\1.txt
Enter old string: C++
Enter new string: Cpp
```

- برنامه‌ای که نام فایلی را خوانده، تعیین می‌کند که آیا این فایل از نوع GIF است یا خیر؟ فایلی با نوع GIF است که بایت‌های اول تا چهارم آن "GIF۸" باشند.
- برنامه‌ای که نام فایلی را خوانده و اطلاعات آن فایل را به حروف بزرگ تبدیل کرده با پسوند dat با همان نام در فایل دیگری می‌نویسد.
- برنامه‌ای که نام یک فایل باینری را خوانده، حروف کوچک آن را به حروف بزرگ تبدیل کرده، در فایلی با همان نام و پسوند UPP می‌نویسد.

- برنامه‌ای که نام فایل باینری را خوانده و اطلاعات آن را در یک فایل باینری دیگر با همان نام و پسوند Inv می‌نویسد.
- برنامه‌ای که نام یک فایل متنی را خوانده، بایت‌های آن را رمز گذاری می‌کند و در فایلی با پسوند Enc می‌نویسد.
- برنامه‌ای که نام فایل متنی را خوانده، تعداد کلمات آن را شمارش می‌کند. کلمات در فایل با کاراکتر (Blank) از یکدیگر جدا می‌شوند.
- برنامه‌ای که نام یک فایل باینری را خوانده، ابتدا تشخیص می‌دهد این فایل یک فایل BMP است یا خیر (فایلی BMP است که بایت اول و دوم آن به ترتیب 0x42 و 0x4D باشد) و سپس عرض و ابتدا فایل را نمایش می‌دهد (عرض و ارتفاع فایل BMP در بایت ۱۸ به بعد هر یک به صورت یک عدد صحیح قرار دارند).
- برنامه‌ای که نام دو فایل را خوانده و فایل اولی را در فایل دومی کپی می‌کند. در هنگام کپی کردن هر مرحله ۵۱۲ k از فایل اولی می‌خواند و در فایل دومی می‌نویسد.
- برنامه‌ای که نام فایلی را خوانده، سپس اندازه (تعداد بایت) را می‌خواند و فایل اولی را با توجه به اندازه به چند فایل دیگر تقسیم می‌کند (شماره فایل‌ها، همان نام فایل اصلی به همراه شماره‌های ۰۰۱، ۰۰۲، می‌باشد).
- برنامه‌ای که نام یک فایل را خوانده، تشخیص می‌دهد که آیا این فایل BMP است یا خیر (فایلی BMP است که با بیت‌های ۰ و یک آن بترتیب کاراکترهای 'B' و 'M' باشند).
- برنامه‌ای که نام یک فایل CSV را خوانده، کلمات آن را جدا کرده و هر کلمه را در یک سطر چاپ می‌کند. فایل‌های CVS، فایل‌هایی متنی هستند که کلمات آن‌ها با کاما از یکدیگر جدا می‌شوند.
- برنامه‌ای که نام دو فایل را می‌خواند و تعداد کاراکترهایی از فایل اول و دوم که نظیر به نظیر با هم برابرند را نمایش می‌دهد.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۰۹

- برنامه‌ای که نام فایلی را خوانده و کاراکترهای قابل چاپ آن را در فایلی با همان نام به علاوه "۰۱.txt" می‌نویسد.
- برنامه‌ای که نام یک فایل را خوانده، اطلاعات آن را به صورت متنی و مکان آن را به صورت هگزا دیسمال (مبنای ۱۶) به صورت ۲۲ خط ۲۲ خط در هر صفحه نمایش می‌دهد.
- برنامه‌ای که نام فایلی را به صورت آرگومان یا از ورودی می‌خواند و آن فایل را از دیسک حذف می‌کند.
- نامه‌ای که تعدادی رشته را خوانده و سپس n را می‌خواند و n خط را خوانده، خطوطی که رشته اول خوانده شده در آن‌ها نباشد، در فایلی که نام آن را از ورودی می‌خواند، می‌نویسد و در پایان اطلاعات فایل را نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، سپس n را می‌خواند و n سطر آخر فایل را از ابتدا به انتها نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و عدد n را خوانده، سپس n سطر انتهای فایل را از انتها به ابتدا نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و n را خوانده، n سطر ابتدای فایل را از ابتدا به انتها نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و عدد n را خوانده، n سطر ابتدای فایل را از ابتدا به انتها نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، اطلاعات شماره سطرهای فرد آن را نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و عدد n را خوانده و از n امین سطر فایل تا انتهای فایل را نمایش می‌دهد.

- برنامه‌ای که نام فایلی را خوانده، تعداد خطوط فایل را نمایش می‌دهد.
- برنامه‌ای که نام دو فایل و یک کاراکتر را خوانده و از فایل اول کاراکتر خوانده شده را حذف کرده و نتیجه را در فایل دوم می‌نویسد.
- برنامه‌ای که نام چند فایل را از ورودی می‌خواند و اطلاعات آن فایل‌ها را در فایل kol.txt در پوشه Data درایو D ادغام می‌کند (برای خاتمه کاربر باید به جای نام فایل Enter را فشار دهد (یعنی نام فایل را وارد نکند)).
- برنامه‌ای که نام سه فایل را از کاربر می‌خواند و خطوط فایل‌های اول و دوم را یک دو میان در فایل سوم می‌نویسد (یعنی، یک خط از فایل اول و یک خط از فایل دوم و همین روند را ادامه می‌دهد).
- برنامه‌ای که نام یک فایل را خوانده و اطلاعات فایل اول را بطور معکوس (از آخرین خط به اولین خط) در فایل newFile.txt در پوشه Data درایو D می‌نویسد.
- برنامه‌ای که نام دو فایل را خوانده، سپس دو عدد را می‌خواند و اطلاعات فایل اول را در فایل دوم می‌نویسد، بطوریکه خطوط بین بازه بسته بین این دو عدد را در فایل دوم نمی‌نویسد.
- برنامه‌ای که نام فایلی را خوانده، بخش توسعه (پسوند فایل) را نمایش می‌دهد.
- برنامه‌ای که نام دو فایل را خوانده و از فایل اول تمام خطوط جدید (کاراکتر '\n') را حذف کرده و در فایل دوم می‌نویسد.
- برنامه‌ای که نام فایلی را خوانده، تعداد خطوط، تعداد کلمات، تعداد کاراکترها و تعداد کاراکترهای سفید آن را شمارش کرده و نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، تعداد حروف کوچک، تعداد حروف بزرگ، تعداد فضای سفید، تعداد ارقام، تعداد کاراکترهای کنترل و تعداد نمادهای آن را نمایش می‌دهد.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۱۱

- برنامه‌ای که نام فایلی را خوانده، سپس یک کلمه را می‌خواند و تعداد تکرار این کلمه در فایل را شمارش می‌کند.
- برنامه‌ای که نام فایلی را خوانده، حرف اول، وسط و آخر هر سطر را نمایش می‌دهد.
- برنامه‌ای که نام فایل‌ها و تعدادی عدد (n) را خوانده، n کاراکتر ابتدای فایل‌ها را نمایش می‌دهد. این برنامه موقعی خاتمه می‌یابد که n عدد منفی وارد شود یا در نام فایل چیزی وارد نگردد.
- برنامه‌ای که نام فایلی را خوانده، اطلاعات آن فایل از قبیل نام کامل، پسوند فایل، نام دایرکتوری، را نمایش می‌دهد.

- برنامه‌ای که نرخ حقوق به ازای هر ساعت و تعداد ساعت کارکرد یک کارمند را خوانده، میزان حقوق او را محاسبه می‌کند و نمایش می‌دهد (از حقوق ۱۰ درصد به‌عنوان مالیات کسر می‌شود. حقوق خالص برابر با حقوق منهای مالیات است).
- برنامه‌ای که مختصات دو نقطه را دریافت کرده، فاصله بین آن دو نقطه را محاسبه می‌نماید و نمایش می‌دهد.

$$\text{فاصله بین دو نقطه} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
- زمینی به ابعاد 16×13 وجود دارد. موزاییک‌هایی به ابعاد 3×2 ، برای فرش کردن زمین نیاز است. اولاً تعیین کنید چند موزاییک نیاز است. آیا قسمتی از زمین می‌ماند که با موزاییک فرش نشود؟
- برنامه‌ای که نقطه تقاطع دو خط $y = 3x + c$ و $y = ax + b$ را نمایش می‌دهد (برنامه a، b و c را از ورودی می‌خواند). مختصات نقطه تقاطع به‌صورت زیر به دست می‌آید:

$$x = (b - c) / (3 - a) \quad y = 3 * x + c$$
- برنامه‌ای که a و b را خوانده، مقدار $y = b * \sin(a)$ را نمایش می‌دهد.
- برنامه‌ای که Four را نمایش داده، چهار بار بوق کامپیوتر را به صدا در می‌آورد، سپس به خط بعد می‌رود، Five را نمایش داده، ۵ بار بوق کامپیوتر را به صدا در می‌آورد.
- **برنامه‌ای که عدد یک و یک کاراکتر قلب، سپس عدد ۲، دو کاراکتر قلب و عدد ۳، سه کاراکتر قلب، ۴ و کاراکتر قلب را نمایش می‌دهد.**
- برنامه‌ای که تعداد بایت‌هایی را نمایش می‌دهد که متغیرهای با انواع long، int، char، double و float اشغال می‌کنند.
- برنامه‌ای که یک کاراکتر را خوانده، کد اسکی آن را نمایش می‌دهد (در این برنامه متغیر ch کاراکتر خوانده‌شده را نگه‌داری می‌کند).
- برنامه‌ای که عددی را خوانده، با استفاده از عملگر یا انحصاری تشخیص می‌دهد، عدد خوانده‌شده زوج است یا فرد؟

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۱۳

➤ برنامه‌ای که ابتدا عددی را خوانده، سپس n را می‌خواند و مقدار n امین بیت عدد اول خوانده شده را نمایش می‌دهد.

➤ برنامه‌ای که ابتدا عددی را خواند، سپس n را می‌خواند و n امین بیت عدد را برابر ۱ (یک) قرار می‌دهد.

➤ برنامه‌ای که ابتدا عددی را خوانده، سپس n را می‌خواند و n امین بیت عدد خوانده شده را صفر می‌نماید. به عنوان مثال، شکل زیر ۴ امین بیت عدد ۵۵ را صفر می‌نماید.

55	←	num(in decimal)
4	←	n(in decimal)
00000001	←	1(in binary)
00010000	←	$1 \ll n$
11101111	←	$\sim(1 \ll n)$
00110111	←	num(in binary)
00100111	←	Num & (~ (1 << n))

➤ برنامه‌ای که ابتدا عددی را خوانده، سپس عدد n را می‌خواند و n امین بیت عدد خوانده شده را معکوس می‌کند. یعنی، چنانچه این بیت یک باشد، به صفر و اگر صفر باشد، به یک تبدیل می‌نماید.

➤ برنامه‌ای که دو عدد را خوانده، با استفاده از عملگر یا انحصاری بیتی آنها را باهم تعویض می‌نماید و نمایش می‌دهد. مراحل انجام این کار در شکل زیر آمده است:

a 00011010	b 01000001	a 00011010
b ^ 01000001	x ^ 01011011	x ^ 01011011
x 01011011	a 00011010	b 01000001

➤ برنامه‌ای که سه عدد را به عنوان قاعده کوچک، قاعده بزرگ و ارتفاع یک ذوزنقه خوانده، مساحت ذوزنقه را نمایش می‌دهد. مساحت ذوزنقه برابر با:

$$\left(\text{قاعده بزرگ} + \text{قاعده کوچک}\right) * \text{ارتفاع} * \frac{1}{2}$$

➤ برنامه‌ای که عددی را خوانده، تمام بیت‌های عدد خوانده شده را معکوس می‌کند. یعنی، چنانچه بیت یک باشد به صفر و اگر صفر باشد، به یک تبدیل می‌نماید.

➤ برنامه‌ای که عددی را خوانده، تعیین می‌کند که مربع کامل است یا خیر؟ عددی مربع کامل است که مجموع تعدادی اعداد فرد پشت سر هم از ۱ و بعد آن برابر آن عدد شود. به‌عنوان مثال، اعداد زیر مربع کامل هستند:

$$\sqrt{9} = 1 + 3 + 5 \qquad \sqrt{1} = 1$$

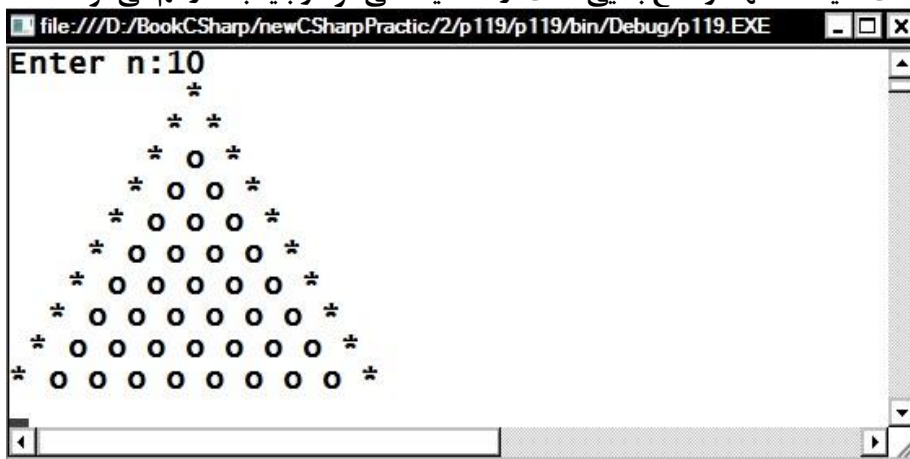
$$\sqrt{16} = 1 + 3 + 5 + 7 \qquad \sqrt{4} = 1 + 3$$

➤ برنامه‌ای که n و x را خوانده، حاصل n جمله سری زیر را نمایش می‌دهد:

$$\frac{x + \frac{x}{2}}{1!} + \frac{x^3 - \frac{x}{4}}{2!} + \frac{x^5 + \frac{x}{8}}{3!} + \dots$$

➤ برنامه‌ای که عددی را خوانده، ارقام زوج آن را نمایش می‌دهد. به‌عنوان مثال، برای عدد ۱۳۵۲۷۴، عدد ۲۴ را نمایش خواهد داد.

➤ برنامه‌ای که n را خوانده، یک مثلث به شکل زیر با n سطر در خروجی نمایش می‌دهد. دقت کنید که تنها دو ضلع بالایی مثلث از * کشیده می‌شود و بقیه با O رسم می‌گردند.



➤ ۱۲۰. برنامه‌ای که یک عدد را خوانده، در خروجی عددی را نمایش دهد که هر رقم آن یک واحد اضافه شده است. به‌عنوان مثال، اگر کاربر عدد ۴۵۹۳۴ را وارد کند، در خروجی ۵۶۰۴۵ چاپ شود (رقم ۹ به صفر تبدیل گردد).

➤ ۱۲۱. برای محاسبه π می‌توان از رابطه زیر استفاده کرد:

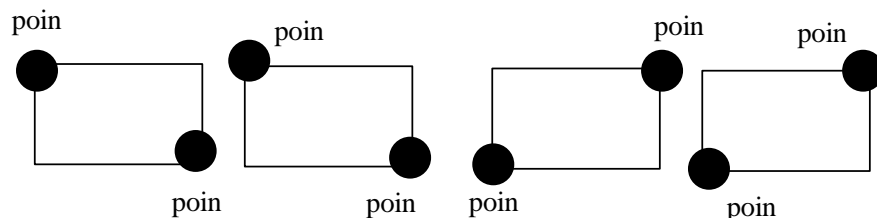
$$\frac{\pi}{2} = \frac{2}{1} + \frac{2}{3} + \frac{4}{3} + \frac{4}{5} + \frac{6}{5} + \frac{6}{7} + \dots$$

➤ برنامه‌ای که تعداد جملات سری فوق (n) را به‌عنوان یک عدد صحیح دریافت کرده و مقدار π را با استفاده از این رابطه محاسبه کرده و نمایش می‌دهد.

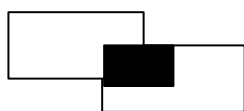
➤ برنامه‌ای که مختصات سه نقطه را از ورودی می‌گیرد و تعیین می‌کند که آن سه نقطه هم خط هستند یا خیر؟

➤ اگر سه نقطه هم خط بودند، خروجی برنامه، یک معادله به‌صورت $ax + by + c = 0$ خواهد بود، وگرنه خروجی "No" خواهد بود. ورودی به‌صورت شش عدد اعشاری خواهد بود که مختصات سه نقطه A و B و C را بیان می‌کند. سه نقطه $A(x_1, y_1)$ ، $B(x_2, y_2)$ و $C(x_3, y_3)$ را هم خط گویند، هرگاه شیب دو خط AB و BC باهم برابر باشند. شیب دو خط AB با X فرمول $m = \frac{y_2 - y_1}{x_2 - x_1}$ قابل محاسبه است. همچنین، معادله خط AB به‌صورت $y - y_1 = m(x - x_1)$ نوشته می‌شود.

➤ برنامه‌ای که مختصات دو مستطیل را از ورودی گرفته و مساحت ناحیه‌ی مشترک بین دو مستطیل را حساب می‌نماید. به‌ازاء هر مستطیل دو نقطه (جمعاً چهار عدد اعشاری) باید گرفته شود. توجه کنید که هر مستطیل با استفاده از دو نقطه می‌تواند به یکی از چهار فرم زیر نمایش داده شود.



➤



➤ به‌عنوان نمونه، هدف این سؤال پیدا کردن ناحیه پررنگ بین دو مستطیل زیر می‌باشد. البته این احتمال وجود دارد که ناحیه مشترک وجود نداشته باشد که در این صورت مساحت ناحیه مشترک صفر خواهد بود.

➤ برای هر کسر به‌صورت $1/k$ (k عدد صحیح بزرگ‌تر از صفر و کوچک‌تر از ۱۰۰۰)، دو

$$\frac{1}{k} = \frac{1}{x} + \frac{1}{y}$$

عدد صحیح x و y وجود دارد که عبارت مقابل برقرار باشد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۱۷

برنامه‌ای که مقدار k را از دریافت می‌کند و تمام x و y هایی که در این عبارت صدق می‌کنند را نمایش می‌دهد. توجه کنید اگر k در بازه درست نباشد، برنامه باید در خروجی هشدار خطا دهد.

نمونه ورودی: ۱۲ خروجی

$$\begin{array}{ccc} \frac{1}{12} = \frac{1}{13} + \frac{1}{156} & \frac{1}{12} = \frac{1}{14} + \frac{1}{84} & \frac{1}{12} = \frac{1}{15} + \frac{1}{60} \\ \frac{1}{12} = \frac{1}{16} + \frac{1}{48} & \frac{1}{12} = \frac{1}{18} + \frac{1}{36} & \frac{1}{12} = \frac{1}{20} + \frac{1}{30} \\ & \frac{1}{12} = \frac{1}{24} + \frac{1}{24} & \frac{1}{12} = \frac{1}{21} + \frac{1}{28} \end{array}$$

➤ برنامه‌ای که به روش تنصیف معادله زیر را تا چهار رقم اعشار حل می‌کند.

$$x + e^x = 5 \quad F(x) = x + e^x - 5 = 0$$

برای حل این معادله دو تخمین اولیه نیاز داریم. یکی عددی که به ازای آن مقدار معادله مثبت باشد (x_p) و دیگری که مقدار منفی به معادله بدهد (x_n). تخمین جدید در هر مرحله (x_m) میانگین دو تخمین قبل است.

➤ برنامه‌ای که n را خوانده، حاصل n جمله سری زیر را نمایش می‌دهد.

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$$

➤ برنامه‌ای که همه حروف الفبای انگلیسی را به ترتیب چاپ می‌کند به طوری که هر حرف بزرگ، مشابه کوچک آن‌هم نمایش داده شود.

➤ برنامه‌ای که تعدادی عدد که به صورت مرتب شده صعودی وارد می‌شوند، را از ورودی خوانده، حاصل جمع آن‌ها را نمایش می‌دهد. اگر عدد بعدی از عدد قبلی کمتر وارد شود، برنامه خاتمه می‌یابد.

○ $y = 3 * t * t + 5;$

$y = -3 * t * t + 5;$

➤ برنامه‌ای که x و n را خوانده، حاصل n جمله عبارت زیر را نمایش می‌دهد:

$$\frac{-x^1}{1+2} + \frac{x^3}{3+4} - \frac{x^5}{5+6} + \dots$$

➤ برنامه‌ای که x و n را خوانده، حاصل n جمله عبارت زیر را حساب می‌کند:

$$-\frac{x^2}{1*3} + \frac{x^4}{3*5} - \frac{x^6}{5*7} + \dots$$

➤ برنامه‌ای که x و n را خوانده، حاصل n جمله سری زیر را محاسبه کرده، نمایش می‌دهد.

$$X / 3 + x^2 / (2 * 3^2) + x^3 / (3 * 3^3) + \dots$$

➤ برنامه‌ای که x و n را خوانده، حاصل n جمله سری زیر را محاسبه کرده و نمایش می‌دهد:

$$\frac{2x^1}{1+2} - \frac{4x^2}{2+4} + \frac{8x^3}{3+8} - \dots$$

➤ برنامه‌ای که x و n را خوانده، حاصل n جمله سری زیر را محاسبه کرده، نمایش می‌دهد:

$$x + \frac{x^2}{5} + \frac{x^3}{25} + \frac{x^4}{125} + \dots$$

➤ برنامه‌ای که ابتدا n را خوانده، سپس به تعداد n نقطه را به صورت اعشاری (x, y) می‌خواند و معادله $y = mx + b$ را به دست آورد. m و b صورت‌های زیر محاسبه می‌شوند:

$$m = \frac{(\sum xy) - \bar{y}(\sum x)}{(\sum xx) - \bar{x}(\sum x)} \quad b = \bar{y} - m\bar{x}$$

\bar{X} و \bar{Y} به ترتیب میانگین X و Y هستند. در این برنامه متغیرهای n (تعداد نقاط)، X و Y (مختصات هر نقطه در هر مرحله)، $\text{sum}X$ (مجموع مقادیر x)، $\text{sum}Y$ (مجموع مقادیر y)، $\text{sum}XY$ (مجموع $x*y$)، $\text{mean}X$ (میانگین x)، $\text{mean}Y$ (میانگین y)، m (ضریب خط)، b (مقدار ثابت خط) و i (شمارنده از ۱ تا n) را داریم.

➤ . برنامه‌ای که تعداد سطرها را خوانده و حاصل خروجی زیر را تولید می‌کند (برای تعداد سرهای ۶ خروجی زیر را نمایش می‌دهد):

```
file:///D:/BookCSharp/newCSharpPractic/2/p136/p136/bin/Debug/p136.EXE
Input number of rows : 6
1
2      3
4      5      6
7      8      9      10
11     12     13     14     15
16     17     18     19     20     21
```

➤ برنامه‌ای که تعداد سطرها را خوانده و خروجی را با الگوی زیر نمایش می‌دهد

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۱۹

```
file:///D:/BookCSharp/newCSharpPractic/2/p137/p137/bin/Debug/p137.EXE
Input number of rows : 7
  1
 2 3
4 5 6
 7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
```

➤ برنامه‌ای که تعداد سطرها را خوانده و الگوی زیر را با ستاره (*) چاپ می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p138/p138/bin/Debug/p138.EXE
Input number of rows : 10
 *
 * *
 * * *
 * * * *
 * * * * *
 * * * * *
 * * * * *
 * * * * *
 * * * * *
 * * * * *
```

➤ برنامه‌ای که تعداد سطرها را خوانده و الگویی به صورت زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p139/p139/bin/Debug/p139.EXE
Input number of rows : 8
  1
 2 2
 3 3 3
 4 4 4 4
 5 5 5 5 5
 6 6 6 6 6 6
 7 7 7 7 7 7 7
 8 8 8 8 8 8 8 8
```

➤ برنامه‌ای که n و k را از ورودی خوانده و سپس n جمله سری زیر را چاپ می‌کند:

➤
$$K + KK + KKK + \dots + KKK \dots K$$

 K عددی بین ۱ تا ۹ است. در پایان، مجموع این سری را نمایش می‌دهد.

➤ برنامه‌ای که x و n را خوانده، حاصل عبارت زیر را محاسبه می‌کند:

➤
$$\frac{x^n n^x x^{(n+n)} n^{(x+x)}}{x! n! (x+n)!} \sum_{i=1}^n \frac{x^n n^x (x+n)^i}{(x+2i)! (n+2i)!}$$

➤ برنامه‌ای که تعیین می‌کند کم‌ارزش‌ترین بیت ۱ است یا ۰ (کم‌ارزش‌ترین بیت، سمت راست‌ترین بیت می‌باشد). به‌عنوان مثال، برای عدد ۱۵ (۰۰۰۰۱۱۱۱)، سمت راست-ترین بیت ۱ است، ولی برای عدد ۱۲ (۰۰۰۰۱۱۰۰) سمت راست‌ترین بیت ۰ است. اگر این اعداد را با ۱ (یک)، و بیتی نماییم، چنانچه نتیجه صفر شود، سمت راست‌ترین بیت صفر، ولی اگر نتیجه یک شود سمت راست‌ترین بیت یک است:

➤ برنامه‌ای که تعیین می‌کند باارزش‌ترین بیت یک عدد یک (۱) است یا صفر (۰)؟ باارزش‌ترین بیت عدد، سمت چپ‌ترین بیت آن است.

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی مانند زیر را نمایش می‌دهد (rows=۹).

```

file:///D:/BookCSharp/newCSharpPractic/2/p144/p144/bin/Debug/p144.EXE
Input rows:9
*****
*****
*****
*****
*****
*****
*****
*****
*****
    
```

➤ ۱۴۵. برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی زیر را نمایش می‌دهد:

```

file:///D:/BookCSharp/newCSharpPractic/2/p145/p145/bin/Debug/p145.EXE
Input rows:8
*****
 *
 *
 *
 *
 *
 *
 *
*****
    
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:

```

file:///D:/BookCSharp/newCSharpPractic/2/p146/p146/bin/Debug/p146.EXE
Input rows:12
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
    
```


حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۲۱

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 147/p 147/bin/Debug/p 147.EXE
Input rows:8
*****
 *
 *
 *
 *
 *
 *
 *
*****
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 148/p 148/bin/Debug/p 148.EXE
Input rows:7
*****
*****
*****
*****
*****
*****
*****
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 149/p 149/bin/Debug/p 149.EXE
Input rows:7
*****
*****
*****
*****
*****
*****
*****
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p150/p150/bin/Debug/p150.EXE
Input rows:8
*****
*****
*****
*****
****
***
**
*
**
***
****
*****
*****
*****
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی مانند زیر را نمایش می‌دهد:

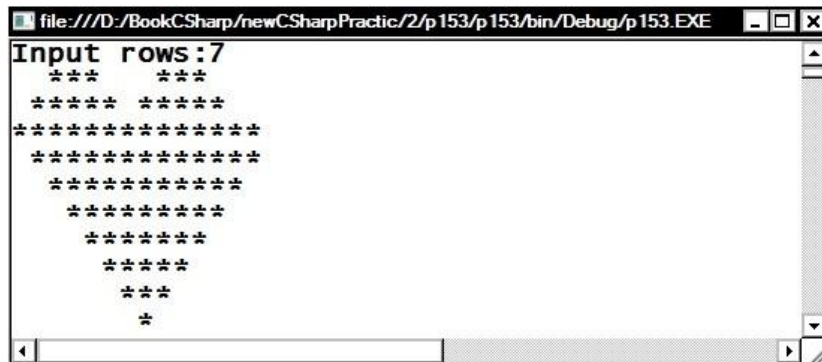
```
file:///D:/BookCSharp/newCSharpPractic/2/p151/p151/bin/Debug/p151.EXE
Input rows:5
+
+
+
+
+++++++
+
+
+
+
```

➤ برنامه‌ای که تعداد سطرها را خوانده (rows) و خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p152/p152/bin/Debug/p152.EXE
Input rows:8
*****
**
**
**
**
**
*****
**
**
**
**
*****
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:

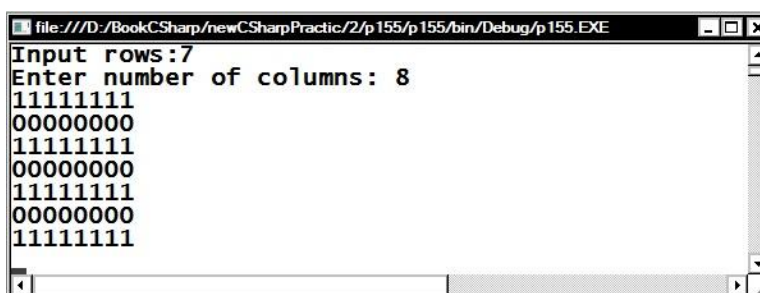
حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۲۳



➤ برنامه‌ای که تعداد سطرها (rows) و یک نام را خوانده، خروجی مانند زیر را نمایش می‌دهد:



➤ برنامه‌ای که تعداد سطرها (rows) و ستون‌ها (cols) را خوانده، خروجی مانند زیر را نمایش می‌دهد:



➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p156/p156/bin/Debug/p156.EXE
Input rows:6
Enter number of columns: 10
0101010101
0101010101
0101010101
0101010101
0101010101
0101010101
0101010101
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را از ورودی خوانده، خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p157/p157/bin/Debug/p157.EXE
Input rows:6
Enter number of columns: 10
1111111111
1000000001
1000000001
1000000001
1000000001
1111111111
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p158/p158/bin/Debug/p158.EXE
Input rows:6
Enter number of columns: 12
101010101010
010101010101
101010101010
010101010101
101010101010
010101010101
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p159/p159/bin/Debug/p159.EXE
Input rows:7
Enter number of columns: 9
111101111
111101111
111101111
000000000
111101111
111101111
111101111
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی نظیر زیر را نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۲۵

```
file:///D:/BookCSharp/newCSharpPractic/2/p160/p160/bin/Debug/p160.EXE
Input rows:7
Enter number of columns: 15
100000000000001
010000000000010
001000000000100
000100000001000
000010000010000
000001000100000
000000101000000
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده، خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p161/p161/bin/Debug/p161.EXE
Input rows:8
Enter number of columns: 9
123456789
234567899
345678999
456789999
567899999
678999999
789999999
899999999
```

➤ برنامه‌ای که با خواندن تعداد سطرها (rows) و تعداد ستون‌ها (cols) خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p162/p162/bin/Debug/p162.EXE
Input rows:5
Input columns: 7
1234567
2345671
3456721
4567321
5674321
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند الگوی زیر را نمایش می‌دهد"

```
file:///D:/BookCSharp/newCSharpPractic/2/p163/p163/bin/Debug/p163.EXE
Input rows:8
8888888888888888
8777777777777778
8766666666666678
87655555555555678
87654444444445678
87654333333345678
8765432222345678
876543212345678
876543222345678
8765433333345678
876544444445678
876555555555678
876666666666678
877777777777778
8888888888888888
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و الگوی نظیر خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p164/p164/bin/Debug/p164.EXE
Input rows:6
1
23
4567
89123456
7891234567891234
56789123456789123456789123456789
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و الگوی مانند خروجی زیر را نمایش می‌دهد" در این

```
file:///D:/BookCSharp/newCSharpPractic/2/p165/p165/bin/Debug/p165.EXE
Input rows:8
1                1
12               21
123              321
1234             4321
12345            54321
123456           654321
1234567          7654321
1234567887654321
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، الگوی نظیر زیر را چاپ می‌نماید:

```
file:///D:/BookCSharp/newCSharpPractic/2/p166/p166/bin/Debug/p166.EXE
Input rows:7
1
2 8
3 9 14
4 10 15 19
5 11 16 20 23
6 12 17 21 24 26
7 13 18 22 25 27 28
```

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۲۷

➤ برنامه‌ای که تعداد سطرها را خوانده (rows) و الگوی مانند خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p167/p167/bin/Debug/p167.EXE
Input rows:8
1
2 4
7 11 16
22 29 37 46
56 67 79 92 106
121 137 154 172 191 211
232 254 277 301 326 352 379
407 436 466 497 529 562 596 631
```

➤ برنامه‌ای که تعداد سطرها را خوانده، الگوی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p168/p168/bin/Debug/p168.EXE
Input rows:7
1
3 2
4 5 6
10 9 8 7
11 12 13 14 15
21 20 19 18 17 16
22 23 24 25 26 27 28
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، الگوی نظیر خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p169/p169/bin/Debug/p169.EXE
Input rows:7
1
12
123
4444
33333
222222
1111111
```

➤ برنامه‌ای که عددی را خوانده و بر اساس عدد خوانده‌شده، خروجی طبق الگوی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p170/p170/bin/Debug/p170.EXE
Enter n:1236754
1236754
123675
12367
1236
123
12
1
```

➤ برنامه‌ای که اگر عدد 24165 را از ورودی بخواند، خروجی با الگوی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p171/p171/bin/Debug/p171.EXE
Enter a number:4525262
4525262
525262
25262
5262
262
62
2
```

➤ برنامه‌ای که ابتدا n را خوانده و سپس الگوی مانند خروجی زیر را نمایش می‌دهد

```
file:///D:/BookCSharp/newCSharpPractic/2/p172/p172/bin/Debug/p172.EXE
Enter n:8
1
12
123
1234
12345
123456
1234567
12345678
1234567
123456
12345
1234
123
12
1
```

➤ برنامه‌ای که n را خوانده و الگوی خروجی نظیر زیر را نمایش می‌دهد

```
file:///D:/BookCSharp/newCSharpPractic/2/p173/p173/bin/Debug/p173.EXE
Enter n:4
1
123
12345
1234567
12345
123
1
```

➤ برنامه‌ای که n را خوانده و الگوی مانند خروجی زیر را نمایش می‌دهد

```
file:///D:/BookCSharp/newCSharpPractic/2/p174/p174/bin/Debug/p174.EXE
Enter n:5
*1*
*123*
*12345*
*1234567*
*123456789*
*1234567*
*12345*
*123*
*1*
```

➤ برنامه‌ای که ابتدا n را خوانده و الگوی نظیر خروجی زیر را نمایش می‌دهد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۲۹

```
file:///D:/BookCSharp/newCSharpPractic/2/p175/p175/bin/Debug/p175.EXE
Enter n:6
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

➤ برنامه‌ای که عددی را خوانده، مکان باارزش‌ترین بیت آن را برمی‌گرداند و نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p176/p176/bin/Debug/p176.EXE
Enter a number:121
Highest order set bit in 121 is 6_
```

➤ برنامه‌ای که عددی را خوانده، مکان سمت راست‌ترین بیتی که ۱ باشد را نمایش می‌دهد. به‌عنوان مثال، اگر عدد ۲۲ در n بیت نمایش داده شود (یعنی ۰۰۰۱۰۱۱۰) مکان سمت راست‌ترین بیت یک آن، ۱ است.

➤ برنامه‌ای که عددی را خوانده، تعداد بیت‌های صفر (۰) و یک (۱) را شمارش می‌کند.

➤ برنامه‌ای که تعداد حروف الفبا را خوانده و الگوی نظیر خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p179/p179/bin/Debug/p179.EXE
Input the number of Letters (less than 26) in the Pyramid : 9
      A
     A B A
    A B C B A
   A B C D C B A
  A B C D E D C B A
 A B C D E F E D C B A
A B C D E F G F E D C B A
A B C D E F G H G F E D C B A
A B C D E F G H I H G F E D C B A
```

➤ برنامه‌ای که عددی را خوانده، تشخیص می‌دهد عددی زشت (ugly) است یا خیر؟ عددی زشت است که عامل‌های اول آن اعداد ۲، ۳ یا ۵ باشند. برخی از اعداد زشت عبارت‌اند از:

▪ ۱ ۲ ۳ ۵ ۶ ۸ ۹ ۱۰ ۱۲.....

➤ برنامه‌ای که ابتدا n را خوانده، سپس n عدد را خوانده و تشخیص می‌دهد که این اعداد تشکیل تصاعد حسابی را می‌دهند یا خیر.

دو سری از اعداد که تشکیل تصاعد حسابی را می‌دهند در زیر آمده‌اند:

➤ -۱ ۳ ۷ ۱۱ ۱۵ ۱۹

➤ ۵ ۱۰ ۱۵ ۲۰ ۲۵ ۳۰

- برنامه‌ای که ابتدا n را خوانده، سپس n عدد را می‌خواند. و تشخیص می‌دهد این اعداد تشکیل تصاعد هندسی را می‌دهند یا خیر. سری‌های زیر تشکیل تصاعد هندسی را می‌دهند:

○ نرخ ۳.....
۶ ۱۸ ۵۴

▪ نرخ $\frac{1}{2}$
۵ ۲,۵ ۱,۲۵

- برنامه‌ای که عددی را خواند، سپس اعدادی بین ۱ تا n را نمایش می‌دهد. که مجموع مکعبات دو یا بیش‌تر عدد برابر با این عدد باشد را نمایش می‌دهد. (نمونه‌ای از این اعداد در شکل زیر آمده است):



$$1729 = 1^3 + 12^3 = 9^3 + 10^3$$

$$4104 = 2^3 + 16^3 = 9^3 + 15^3$$

$$13832 = 2^3 + 24^3 = 18^3 + 20^3$$

$$39312 = 2^3 + 34^3 = 15^3 + 33^3$$

$$46683 = 3^3 + 36^3 = 27^3 + 30^3$$

$$32832 = 4^3 + 32^3 = 18^3 + 30^3$$

$$40033 = 9^3 + 34^3 = 16^3 + 33^3$$

$$20683 = 10^3 + 27^3 = 19^3 + 24^3$$

$$65728 = 12^3 + 40^3 = 31^3 + 33^3$$

$$64232 = 17^3 + 39^3 = 26^3 + 36^3$$

- اگر اضلاع مثلث اعداد تکرریمی باشند، تعداد مثلث‌هایی که می‌توان تشکیل داد را نمایش می‌دهد. سپس اضلاع این مثلث‌ها را نمایش می‌دهد (زمانی اضلاع تشکیل مثلث را می‌دهند که مجموع هر دو ضلع بیش‌تر از ضلع سوم باشد).

- اگر اضلاع مثلث اعداد ۱ تا ۹ (اعداد تکرریمی) باشند، برنامه‌ای که تعداد مثلث متساوی‌الساقین که با این اعداد می‌توان تشکیل داد را نمایش می‌دهد.

- اگر اضلاع مثلث اعداد تکرریمی باشند، تعداد مثلث‌های قائم‌الزاویه که می‌توان تشکیل داد را نمایش می‌دهد. سپس اضلاع این مثلث را نمایش می‌دهد (زمانی که سه ضلع a ، b و c تشکیل مثلث قائم‌الزاویه را می‌دهند که $a^2 + b^2 = c^2$ یا $a^2 + c^2 = b^2$ یا $a^2 = b^2 + c^2$ باشد).

- برنامه‌ای که مقدار سه زاویه را برحسب درجه دریافت کرده، تشخیص می‌دهد که آیا زاویه تشکیل مثلث را می‌دهند یا نه؟

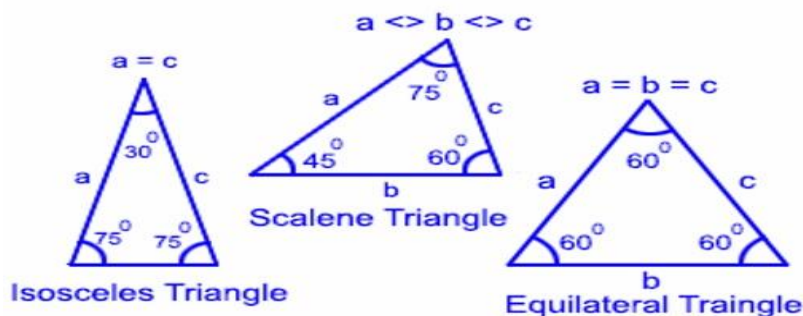
- برنامه‌ای یک عدد در مبنای ۲ را خوانده به مبنای ۱۰ تبدیل می‌کند و نمایش می‌دهد. به‌عنوان مثال عدد زیر از مبنای ۲ به مبنای ۱۰ تبدیل شده است:

$$1100101 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 1 + 0 + 4 + 0 + 0 + 0 + 1 = 6$$

$$= 10^3$$

- برنامه‌ای که اندازه زوایای یک مثلث را گرفته و نوع مثلث را تشخیص می‌دهد. اگر مجموع زوایای مثلث ۱۸۰ درجه نباشد، این زوایا تشکیل مثلث را نمی‌دهند. اما، اگر سه زاویه مثلث برابر باشند، مثلث متساوی‌الاضلاع است، ولی اگر اندازه دو زاویه مثلث برابر باشد، مثلث متساوی‌الساقین است، وگرنه مثلث مختلف‌الاضلاع است (مانند شکل زیر):



- برنامه‌ای که تمام اعداد سه‌رقمی که می‌توان با ارقام ۳ و ۴ و ۵ تولید کرد را نمایش می‌دهد (اعداد را به صورت صعودی مرتب‌شده نمایش می‌دهد).
- برنامه‌ای که تمام اعداد چهاررقمی که با ارقام ۵، ۶، ۷ و ۸ می‌توان تولید کرد را نمایش می‌دهد (در این برنامه در تولید اعداد از ارقام تکراری استفاده نمی‌کند).
- برنامه‌ای که اعداد سه‌رقمی با ارقام فرد می‌توان تولید کرد را نمایش می‌دهد که ارقام آن تکراری نیستند.
- برنامه‌ای که تمام اعداد سه‌رقمی را تولید می‌کند که رقم یکان آن اعداد زوج ۲ تا ۸ رقم دهگان آن اعداد فرد ۵ تا ۹ و رقم یکان آن اعداد مضرب ۳ تا ۹ می‌باشند را نمایش می‌دهد.
- برنامه‌ای که قیمت خرید و فروش یک کالا را خوانده، میزان سود یا زیان را محاسبه کرد، نمایش می‌دهد.
- یک اسب‌سوار به یک پیرزن برخورد کرده، تمام تخم مرغ‌های پیرزن از سبد می‌افتد و می‌شکند. صاحب اسب از پیرزن عذرخواهی می‌کند و به او می‌گوید خسارت تخم

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۳۳

مرغ‌هایتان را می‌دهم. پیرزن می‌گوید، تخم‌مرغ‌ها را دو تا دو تا برداشتم، یکی ماند، سه تا سه برداشتم، نیز یکی ماند، چهار تا چهار تا برداشتم، همچنین یکی ماند، پنج تا پنج تا برداشتم، یکی ماند، شش تا شش تا برداشتم، نیز یکی ماند، ولی هفت تا هفت تا برداشتم، هیچ تخم مرغی نماند. برنامه‌ای که تعیین می‌کند، حداقل چند تخم مرغ در سبد پیرزن وجود داشت.

➤ برنامه‌ای که تا زمانی که کاربر کلید ESC را فشار ندهد، کلید از کاربر می‌گیرد. به محض این که کاربر کلید ESC را فشار دهد، از برنامه خارج می‌شود. این برنامه کلید و کد اسکی کاراکتر فشار داده شده را نمایش می‌دهد. کد اسکی کلید ESC، ۲۷ است.

➤ یک رابطه ریاضی خیلی خوب برای تخمین جذر اعداد به این صورت است که ابتدا یک متغیر t را برابر ۱ قرار می‌دهیم، سپس هر بار t جدید را از روی t قدیم این گونه محاسبه می‌کنیم:

$$\text{قدیم} \frac{x}{t} + \text{قدیم} t * 0,5 = \text{جدید} t$$

به این ترتیب اگر این کار را به صورت متوالی انجام دهیم، به تعمیم خوبی برای \sqrt{x} خواهیم رسید. برنامه‌ای که x و n را خواند و تا n مرحله تخمین \sqrt{x} را پیش می‌برد (تخمین نهایی \sqrt{x} همان جدید t آخرین مرحله خواهد بود).

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه‌های تکرار تو در تو و کاراکتر '*' حرف 'A' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر (عددی زوج) را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' حرف 'B' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تو در تو و کاراکتر '*' حرف 'C' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'D' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'E' را نمایش می‌دهد.

- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و '*' کاراکتر 'F' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'G' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'H' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'I' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'J' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '*' کاراکتر 'K' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تو در تو و کاراکتر '*'، کاراکتر 'L' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، و با استفاده از حلقه تکرار تو در تو و کاراکتر '*'، کاراکتر 'M' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*'، کاراکتر 'N' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*'، کاراکتر 'O' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*'، کاراکتر 'P' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*'، کاراکتر 'Q' را نمایش می‌دهد.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۳۵

- برنامه‌ای که تعداد و سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'R' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'S' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'T' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'U' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'W' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'V' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '*', کاراکتر 'X' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از کاراکتر '*', کاراکتر 'Y' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطرها را خوانده، با استفاده از کاراکتر '*', کاراکتر 'Z' را نمایش می‌دهد.

پیوست

الکترونیکی (فصل سوم)

➤ برنامه‌ای که اعداد x و n را خوانده، حاصل سری زیر را محاسبه کرده و نمایش می‌دهد:

$$\frac{x^n 0^n}{-n} + \frac{x^1 (1)^{n-1}}{-(n-1)} + \dots + \frac{x^{(n-1)} (n-1)^1}{-1} + \frac{x^n n^1}{1} + \frac{x^{(n+1)} (n+1)^2}{2} + \dots + \frac{x^{(2n-1)} (2n-1)^n}{n}$$

➤ برنامه‌ای که n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{1}{1^2+1} + \frac{2}{2^2+2} + \frac{3}{3^2+3} + \dots + \frac{n}{n^2+n}$$

➤ برنامه‌ای که x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{\tan x}{\sin(x^n)} + \frac{\tan^2(x)}{\sin(x^{n-1})} + \frac{\tan^3 x}{\sin(x^{n-2})} + \dots + \frac{\tan^n x}{\sin(x)}$$

➤ برنامه‌ای که x و n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\left\langle \frac{\log(\log(\log \dots n \dots \log(\log(x))))}{n} \right\rangle$$

➤ برنامه‌ای که n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{1!}{1^2} + \frac{2!}{1^2+2^3} + \frac{3!}{1^2+2^3+3^4} + \dots + \frac{n!}{1^2+2^3+\dots+n^{n+1}}$$

➤ برنامه‌ای که x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \pm \frac{(-1)^n x^{2n}}{(2n)!}$$

➤ برنامه‌ای که x (بین -1 تا 1) و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\arcsin x = x + \frac{2!x^3}{4!(1!)^2(3)} + \frac{4!x^5}{4^2(2!)^3(5)} + \dots + \frac{(2^n)x^{2n+1}}{4^n(n!)^2(2n+1)}$$

➤ برنامه‌ای که n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{\sqrt{1}}{1} - \frac{1!}{1^2} + \frac{\sqrt{2}}{2} - \frac{2!}{2^2} + \frac{\sqrt{3}}{3} - \frac{3!}{3^2} + \frac{\sqrt{n}}{n} - \frac{n!}{n^2}$$

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آنها) ۱۳۷

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\frac{\sqrt{3}-\sqrt{2}}{\sqrt{2^2+2}} + \frac{\sqrt{4}-\sqrt{3}}{\sqrt{3^2+3}} + \frac{\sqrt{n+1}-\sqrt{n}}{\sqrt{n^2+n}}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\frac{1}{2^1(\log 2)^2} + \frac{1}{3^2(\log 3)^3} + \dots + \frac{1}{n^{n-1}(\log n)^n}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\log \frac{2}{2^2-1} + \log \frac{3^2}{3^3-1} + \dots + \log \left(\frac{n^{n-1}}{n^n-1} \right)$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\frac{\sqrt{5}}{(2^3+2)^2} + \frac{\sqrt{7}}{(3^3+3)^2} + \dots + \frac{\sqrt{2n+1}}{(n^3+n)^2}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$\frac{\sin(\frac{1}{2})}{\cos(\frac{1}{4}) \cdot \cos(\frac{1}{2})} + \frac{\sin(\frac{1}{6})}{\cos(\frac{1}{2}) \cdot \cos(\frac{1}{3})} + \dots + \frac{\sin\left(\frac{1}{n(n+1)}\right)}{\cos(\frac{1}{n}) \cdot \cos\left(\frac{1}{n+1}\right)}$$

➤ برنامه‌ای که x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{\sqrt{\sin(n) \cdot \log(2)!}}{\tan^n(1)} + \frac{\sqrt{\sin(n-1) \cdot \log(3)!}}{\tan^{n-1}(2)} + \dots + \frac{\sqrt{\sin(o) \cdot \log(n+2)!}}{\tan^n(n+1)}$$

➤ متدی که x و n را به‌عنوان پارامتر دریافت کرده، حاصل سری زیر را برمی‌گرداند.

این برنامه از این متد استفاده می‌کند.

$$\frac{\tan(x^1) + \tan(x^2) + \dots + \tan(x^n)}{\sin^1(x) + \sin^2(x) + \dots + \sin^n(x)}$$

➤ متدی n را به‌عنوان دریافت کرده، سپس حاصل سری زیر را محاسبه و نمایش می‌دهد.

برنامه‌ای که از این متد استفاده می‌کند.

$$(n-1)! \cdot 1^{(n-1)} + (n-2)! \cdot 2^{(n-2)} + \dots + 0! \cdot n^{(n-n)}$$

- متدی که عدد صحیح n را به عنوان پارامتر دریافت کرده، حاصل سری زیر را محاسبه می کند و برنامه اصلی آن را نمایش می دهد:

$$\frac{1}{n-1}x + \frac{n-2}{2}x + \dots + \frac{n-1}{1}x$$

- برنامه ای که n را خوانده با استفاده از متدی حاصل سری زیر را محاسبه و نمایش می دهد:

$$\frac{1 * 2 * 3}{(n+1)(n+2)(n+3)} + \frac{2 * 3 * 4}{(n+2)(n+3)(n+4)} + \dots + \frac{n(n+1)(n+2)}{(n+n)(n+n+1)(n+n+2)}$$

- برنامه ای که x و n را خوانده و توسط متدهای حاصل سری زیر را محاسبه کرده، نمایش می دهد:

$$1 + \frac{nx^1}{1!} + \frac{n(n-1)x^2}{2!} + \dots + \frac{n!x^n}{n!}$$

- **برنامه ای که n را خوانده و با استفاده از متدی حاصل سری زیر را محاسبه کرده و نمایش می دهد:**

$$\phi = \left\{ \begin{array}{l} \frac{1}{1 + \frac{1}{1+1}} \\ \dots \\ 1 + \frac{1}{1+1} \end{array} \right.$$

جمله n

- برنامه ای که عدد n را خوانده، کلیه اعداد مضرب γ زیر n که یکی از ارقام آن γ باشد را نمایش می دهد. در این برنامه برای نمایش اعداد مضرب γ یک متد و برای تعیین این که یک رقم عدد γ است از متد دیگری استفاده می شود.

- برنامه ای که عدد n را خوانده، کلیه اعداد مضرب γ زیر n که هیچ یک از ارقام آن γ نباشد را نمایش می دهد.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۳۹

➤ برنامه‌ای که عددی را خوانده، اگر ارقام از سمت راست به چپ دوتا دوتا کم شدند، Yes، وگرنه No را نمایش می‌دهد. برای تعیین این که اعداد از راست به چپ دوتا دوتا کم شدند از یک متد استفاده می‌کند. به‌عنوان مثال، برای اعداد ۲۴۶، ۳۵۷ و ۱۳۵، "Yes" و برای اعداد ۱۲۴، ۳۵۶، 'No' را نمایش می‌دهد.

- برنامه‌ای که رشته‌ای را خوانده، با استفاده از یک متد فضای خالی سمت چپ (ابتدا رشته) را حذف می‌کند و نمایش می‌دهد (کد اسکی کاراکترهای خالی ۳۲ است).
- برنامه‌ای که رشته‌ای را خوانده، با استفاده از اشاره‌گر و بازگشتی، طول رشته را حساب کرده، نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده، با استفاده از اشاره‌گر و متد بازگشتی معکوس رشته را نمایش می‌دهد.
- برنامه‌ای دو نام را خوانده، تعیین می‌کند که آیا دو نام با هم برابر هستند یا خیر؟
- برنامه‌ای که رشته‌ای را از ورودی خوانده و بین حروف آن فاصله (Blank) ایجاد می‌کند. به‌عنوان مثال، اگر کاربر رشته Test را وارد کند، برنامه رشته Test را نمایش می‌دهد. در این برنامه متغیرهای S_1 (رشته ورودی) و S_2 (رشته خروجی) را داریم.
- برنامه‌ای که ابتدا n را خوانده، سپس n رشته را می‌خواند و طول هر یک را نمایش می‌دهد.
- برنامه‌ای که ابتدا n را خوانده، سپس n رشته را می‌خواند و رشته‌ای که بیش‌ترین طول را دارد، نمایش می‌دهد.
- برنامه‌ای که ابتدا n را خوانده، سپس n رشته را می‌خواند و رشته‌هایی که حرف اول و آخر آن‌ها یکسان باشد را نمایش می‌دهد.
- برنامه‌ای که ابتدا n را خوانده، سپس n رشته را خوانده و در پایان یک کاراکتر را می‌خواند و رشته‌هایی که با این کاراکتر شروع می‌شوند یا پایان می‌یابند را نمایش می‌دهد.
- برنامه‌ای که یک رشته را خوانده، سپس یک کاراکتر را می‌خواند و بین حروف رشته آن کاراکتر را اضافه می‌کند.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۴۱

- برنامه‌ای که رشته‌های S_1 و S_2 را خوانده و کاراکترها را یک‌درمیان در رشته S_3 قرار می‌دهد و رشته S_3 را نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده، کاراکتری از رشته که بیش‌ترین کد اسکی را دارد، نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را از ورودی خوانده و تمام کاراکترهای تکراری آن را حذف می‌نماید.
- برنامه‌ای که دو رشته را خوانده، حروف مشترک بین آن‌ها را نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده، سپس دو عدد را می‌خواند و با توجه به اعداد خوانده‌شده، حروف آن مکان‌ها را جابجا نماید. در این برنامه از متد `Change` به‌صورت زیر استفاده می‌شود:
 - `String Change (String s, int n1, int n2),`
به‌عنوان مثال، اگر متد `Change` به‌صورت زیر فراخوانی شود:
`Change ("Fanavarienovin", ۱, ۳)`
متد، رشته `"naFavarienovin"` را برمی‌گرداند.
- برنامه‌ای که تعداد فاصله اضافی به سمت چپ رشته اضافه می‌کند تا اندازه رشته به n تغییر یابد. اگر اندازه رشته بزرگ‌تر یا مساوی n باشد، رشته هیچ تغییر نمی‌یابد.
- برنامه‌ای که حروف 'a' تا 'z' و 'A' تا 'Z' را به همراه کد اسکی آن‌ها نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده و کاراکتری که بیش‌ترین تکرار را دارد، نمایش می‌دهد.
- برنامه‌ای که دو رشته را خوانده، رشته دوم را در وسط رشته اول اضافه می‌کند.
- برنامه‌ای که رشته‌ای را خوانده و بعد از هر یک از حروف کوچک، حروف بزرگ آن کاراکتر را اضافه می‌کند. به‌عنوان مثال، رشته `"Tree"` به رشته `"TrReEeE"` تبدیل می‌شود.
- برنامه‌ای که دو رشته را دریافت کرده و رشته دوم را بعد از n امین کاراکتر رشته اول درج می‌نماید. در این برنامه متد `insert` استفاده‌شده که دارای اعضای زیر است:
 - `string insert (string s1, string s2, int n);`

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۴۳

ی	۱۶	ی	۱۶	ی	۱۶	ی	۱۶
۰	۰	۰	۱	۰	۱	۰	۱
۴	۰	۰	۵	۰	۱	۰	۱
۸	۰	۰	۹	۰	۱	۰	۱
C	۰	۰	D	۰	۱	۰	۱

به عنوان مثال، عدد باینری ۱۰۱۱۱۱۱۰۱۰۱۰۱ معادل ۱۷۰۵ (۱=۰۰۰۱, ۷=۰۱۱۱, E=۱۱۰۱, ۵=۰۱۰۱) است.

➤ برنامه‌ای که یک آرایه با n عنصر و سه عدد x, y و z را خواند و مقدار ماکزیمم $(x*a[i])+(y*a[j])+(z*a[k])$ را برای $i \leq j \leq k$ پیدا کرده و نمایش می‌دهد.

➤ برنامه‌ای که رشته‌ای را خوانده، کلمات آن را از انتها به ابتدای رشته نمایش می‌دهد. به عنوان مثال، اگر ورودی به صورت زیر باشد:

- I Like Cat Programming

خروجی به صورت زیر خواهد بود.

- Programming Cat Like I

➤ برنامه‌ای که رشته‌ای را خوانده (TXT) و سپس یک الگو (Pat) را می‌خواند و در رشته جست‌وجو می‌نماید. مکان‌های وقوع Pat در TXT را نمایش می‌دهد.

به عنوان مثال، اگر

TXT= "AABAACAADAABAABA"

Pat= "AAB"

باشد، خروجی به صورت زیر نمایش داده شود:

Pattern Found at index ۰

Pattern Found at index ۹

Pattern Found at index ۱۲

➤ برنامه‌ای که آرایه‌ای با n عنصر که مقادیر آن‌ها ۰ یا یک به ترتیب تصادفی هستند را خوانده، ۰های را به سمت چپ و ۱ها را به سمت راست آرایه منتقل کرده، نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

[۰, ۱, ۰, ۰, ۱, ۱, ۰, ۱, ۰, ۰]

خروجی به صورت زیر خواهد بود:

[۱, ۱, ۱, ۱, ۱, ۱, ۱, ۱, ۱, ۱]

➤ برنامه‌ای که آرایه‌ای را خوانده، سپس سه عدد x , y و z را می‌خواند و ماکزیمم عبارت زیر را نمایش می‌دهد:

$$\max((x * a[i]) + (y * a[j]) + (z * a[k]))$$

برای $i \leq j \leq k$

به‌عنوان مثال، اگر آرایه a به‌صورت زیر باشد:

$$a[] = \{-1, -2, -3, -4, -5\}$$

$x=1$, $y=2$, $z=-3$ باشد، آنگاه خروجی ۱۲ خواهد بود. زیرا:

$$(1 * -1) + (2 * -2) + (-3 * -5) = 12$$

➤ برنامه‌ای که یک آرایه را خوانده، دو عنصری را می‌یابد که حاصل جمع آن‌ها نزدیک‌ترین عدد به صفر است. برای انجام این برنامه باید عددی را پیدا کنیم که مینیمم قدر مطلق مجموع دو عنصر آرایه می‌باشد.

➤ برنامه‌ای که آرایه‌ای را خوانده، ماکزیمم مقدار اختلاف بین هر دو عنصر آرایه را نمایش می‌دهد.

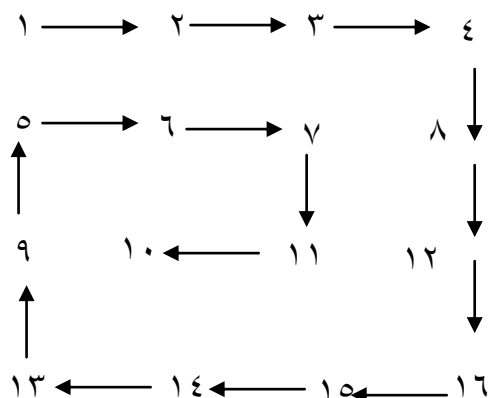
➤ برنامه‌ای که آرایه‌ای را خوانده، اعداد فرد و زوج را از هم جدا می‌کند.

➤ برنامه‌ای که آرایه‌ای را خوانده و ماکزیمم $i-z$ را پیدا می‌کند به‌طوری‌که $arr[j] > arr[i]$ باشد. سپس مقدار ماکزیمم را نمایش می‌دهد. به‌عنوان مثال اگر آرایه به‌صورت زیر باشد:

$$arr[] = \{۳۴, ۸, ۱۰, ۳, ۲, ۸۰, ۳۰, ۳۳, ۱\}$$

آنگاه مقدار ۶ را نمایش می‌دهد وقتی که $(j=7, i=1)$ است.

➤ برنامه‌ای که یک آرایه دوبعدی تعریف کرده و مقداره‌ی اولیه می‌کند. سپس اطلاعات آرایه را به‌صورت مارپیچی نمایش می‌دهد. به‌عنوان مثال، اگر آرایه دوبعدی به‌صورت زیر باشد:



حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۴۵

خروجی به صورت زیر است:

➤ ۱ ۲ ۳ ۴ ۸ ۱۲ ۱۶ ۱۵ ۱۴ ۱۳ ۹ ۵ ۶ ۷ ۱۱ ۱۰

➤ برنامه‌ای که یک آرایه دوبعدی را تعریف کرده، مقداردهی می‌کند (تمام عناصر آرایه (False z) یا (True) هستند)، اگر عنصر $a[i][j]$ برابر یک (۱) باشد، تمام عنصر سطر i ام و ستون j ام را برابر یک قرار می‌دهد. به عنوان مثال، اگر آرایه به صورت زیر باشد:

```
0 1 0
0 0 0
0 0 1
```

خروجی به صورت زیر است:

```
1 1 1
0 1 1
0 1 1
```

➤ برنامه‌ای که مقادیر چند آرایه را مقداردهی اولیه می‌کند و یک نقطه ثابت شده (عنصری از آرایه که مقدار آن عنصر برابر اندیس آن است) را نمایش می‌دهد. چنانچه نقطه ثابت نباشد، -۱ را نمایش خواهد داد. به عنوان مثال، آرایه‌های زیر را در نظر بگیرد:

```
arr1 = {-۱۰, -۵, ۰, ۳, ۷}
arr2 = {۱۰, ۲, ۵, ۸, ۱۷}
```

برای arr_1 خروجی ۳ (چون $arr_1[3]=3$ است) و برای arr_2 خروجی -۱ است.

➤ برنامه‌ای که آرایه‌ای را تعریف کرده و خوانده، به طوری که مقدار عناصر آن ابتدا افزایشی است (یعنی، عنصر قبلی کم‌تر از عنصر بعدی است) و سپس عناصر آن کاهش می‌دهد. این برنامه بزرگ‌ترین مقدار این نوع آرایه را نمایش می‌دهد. به عنوان مثال، اگر عناصر آرایه به صورت زیر باشد:

۷ ۴۱ ۱ ۳ ۵۰ ۱۰

خروجی ۵۰ خواهد بود.

➤ برنامه‌ای که یک آرایه را تعریف کرده، سپس می‌خواند و در پایان، آرایه‌ای را ایجاد می‌کند که هر عنصر آن، تعداد عناصری کوچک‌تر از سمت راست خودش را شمارش می‌کند. به عنوان مثال، اگر آرایه به صورت زیر باشد:

arr={۱۲, ۱, ۲, ۳, ۰, ۱۱, ۴}

خروجی به صورت زیر خواهد بود:

۶, ۱, ۱, ۱, ۰, ۱, ۰

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، و می‌خواند. سپس یک عدد را خوانده و آرایه فرعی را نمایش می‌دهد که مجموع عناصر آن برابر عدد وارد شده می‌باشد. اگر آرایه‌ای وجود نداشته باشد که مجموع عناصر آن برابر عدد وارد شده نباشد، پیغام "No SubArray Found" را نمایش می‌دهد. به عنوان مثال، اگر آرایه‌های ورودی به صورت زیر باشد:

۵ ۱ ۴ ۲۰ ۰ ۳ ۱۰

و اگر $sum=30$ باشد، آنگاه خروجی به صورت زیر است:

➤ Sum found between indexes ۱ and ۴

➤ برنامه‌ای که یک آرایه را تعریف کرده و می‌خواند، سپس یک عدد را می‌خواند، توسط متدی اگر مجموع سه عنصر آرایه برابر با عدد خوانده شده باشد، آن سه عدد را نمایش می‌دهد و True را برمی‌گرداند.

➤ برنامه‌ای که کوچک‌ترین عدد مثبت (غیر منفی) که در آرایه غیر مرتب حذف شده است را می‌یابد و نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

{۲, ۳, -۷, ۶, ۸, ۱, -۱۰, ۱۵}

خروجی عدد ۴ خواهد بود. ولی برای آرایه زیر خروجی ۲ خواهد شد:

{۱, ۱, ۰, -۱, -۲}

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، خوانده، سپس یک عدد را می‌خواند و زوج مقادیر از عناصر آرایه که اختلاف آن‌ها عدد وارد شده باشد، را نمایش می‌دهد. به عنوان مثال، اگر عناصر آرایه به صورت زیر باشد:

{۲, ۲۰, ۳, ۲, ۵۰, ۸۰}

و عدد ورودی ۷۸ باشد، خروجی به صورت زیر است:

Pair Found: (۲, ۸۰)

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، می‌خواند، سپس هر عنصر را با بزرگ‌ترین عنصر بعد از خودش (سمت راست خودش) جایگزین می‌کند، به جای آخرین عدد ۱- قرار می‌گیرد و آرایه جدید را نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۴۷

{ ۱۶, ۱۷, ۴, ۳, ۵, ۲ }

آرایه خروجی به صورت زیر خواهد شد:

{ ۱۷, ۵, ۵, ۵, ۲, - }

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، می‌خواند، سپس یک عدد را خوانده و مجموع ۴ عنصر آرایه که برابر با عدد خوانده شده باشد را نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

{ ۱۰, ۲, ۳, ۴, ۵, ۹, ۷, ۸ }

و عدد ورودی برابر ۲۳ باشد، آنگاه خروجی به صورت زیر است:

➤ ۳ ۵ ۷ ۸

➤ برنامه‌ای که آرایه‌ای را تعریف کرده و عناصر آن را مقداردهی اولیه می‌کند. سپس عناصری از آرایه که مقادیر آن‌ها تشکیل مثلث را می‌دهد را شمارش می‌کند (سه مقدار از آرایه که مجموع هر دو مقدار بیش تر از مقدار سوم باشد، تشکیل مثلث را می‌دهند).

➤ برنامه‌ای که کلاس کسر (Fraction) را پیاده سازی کرده، عملگرهای +، -، *، /، =، !=، >، <، >= و <= را برای آن پیاده سازی می‌کند (کسر دارای صورت (x) و مخرج (y) است).

➤ برنامه‌ایی که دایرکتوری فعلی را نمایش می‌دهد (هدف این برنامه آشنایی با متدهای خارجی است).

➤ برنامه‌ای که اطلاعات کارمندان از قبیل نام، نام خانوادگی، نوع بیمه، مبلغ حقوق ماهانه و تعداد روزهایی که کار کرده را دریافت می‌کند، حقوق دریافتی کارمند را محاسبه کرده، برمی‌گرداند. حقوق دریافتی کارمند به صورت زیر محاسبه می‌شود:

حقوق دریافتی = بیمه - مالیات - تعداد روز * حقوق کارمند

اگر نوع بیمه ۱ باشد، ۲ درصد * حقوق ماهانه

اگر نوع بیمه ۲ باشد، ۹ درصد * حقوق ماهانه

اگر نوع بیمه ۳ باشد، ۱۰ درصد * تعداد روز

اگر حقوق * ماهانه بزرگتر از ۴۸۵۰۰۰ تومان باشد، ۱۰ درصد مازاد بر ۴۸۵۰۰۰ تومان مالیات منظور می‌شود.

توضیح: این برنامه با دو کلاس Person و Employee طراحی شده است. کلاس Person اطلاعاتی از قبیل نام، نام خانوادگی و نوع بیمه را نگهداری می‌کند و کلاس Employee از کلاس Person مشتق شده، علاوه بر اطلاعات نام، نام خانوادگی، و نوع بیمه که از کلاس Person به ارث می‌برد، اعضای داده‌ای salary و days را نیز دارد. در ادامه اعضای این کلاس‌ها را خواهید دید.

➤ برنامه‌ای که کلاس نقطه (Point) را با استفاده از واسط پیاده‌سازی می‌کند (کلاس Point مختصات نقطه را به صورت دو عدد صحیح دریافت کرده، آن را نمایش می‌دهد).

➤ برنامه‌ای که با استفاده از کلاس static اعمال زیر را انجام می‌دهد:

- فیلدی به نام number از نوع static تعریف می‌کند.
- متدی به createRandom() تعریف کرده که دو عدد را به عنوان پارامتر دریافت می‌کند، یک عدد تصادفی بین دو عدد تولید می‌نماید.

حل مسائل جاوا (۶۰۰ برنامه جاوا با حل آن‌ها) ۱۴۹

- متدی به نام `isOdd()` تعریف می‌کند که تعیین می‌کند فیلد داده‌ای `number` فرد است یا خیر؟
 - متدی به نام `isEven()` تعریف می‌کند که تعیین می‌کند فیلد داده‌ای `number` زوج است یا خیر؟
 - متدی به نام `isSquare()` تعریف می‌کند که تعیین می‌کند فیلد داده‌ای `number` مربع کامل است یا خیر؟
 - متدی به نام `isPerfect()` تعریف می‌کند که تعیین می‌کند فیلد `number` عددی کامل (تام) است یا خیر؟
 - متدی به نام `isPrime()` تعریف می‌کند که تعیین می‌کند فیلد `number` عددی اول است یا خیر؟
 - متدی به نام `Fact()` تعریف می‌کند که فاکتوریل فیلد `number` را برمی‌گرداند.
- برنامه‌ای که عملگرهای `+`، `-` و `*` را برای ماتریس‌های دو بعدی تعریف می‌کند.
- برنامه‌ای که مجموع مساحت و محیط چند شکل متفاوت از قبیل مربع، دایره، مستطیل و مثلث را محاسبه می‌کند. این برنامه کاربردهای مختلفی از قبیل محاسبه مساحت و محیط زمین برای ساخت و ساز یا فروش، محاسبه مساحت و محیط ساختمان جهت نقاشی یا لوله کشی دارد.
- امروزه در دانشگاه‌ها دو دسته دانشجو تحصیل می‌کنند: دانشجویان انتفاعی (روزانه) و دانشجویان غیر انتفاعی (مانند شبانه). دانشجویان روزانه هیچ مبلغی را پرداخت نمی‌کنند. اما، دانشجویان شبانه سه نوع مبلغ پرداخت می‌کنند که عبارتند از:
۱. مبلغ شهریه ثابت
 ۲. مبلغ شهریه متغیر برای هر واحد نظری
 ۳. مبلغ شهریه متغیر برای هر واحد عملی
- برنامه‌ای که ابتدا تعداد دانشجو را از ورودی خوانده، سپس به اندازه تعداد دانشجو شماره دانشجویی، نام دانشجو، نوع دانشجو، تعداد واحدهای نظری و عملی دانشجو را دریافت کرده، مبلغ شهریه کل دانشجویان را نمایش می‌دهد (اگر نوع دانشجو صفر وارد شود، دانشجو روزانه وگرنه شبانه است).
- برنامه‌ای که کلاس مربع را از نوع کلاس `static` پیاده سازی می‌کند. این کلاس دارای عضو داده‌ای `side` (ضلع مربع)، عضو متدی `Area()` برای محاسبه مساحت مربع، عضو متدی `Perimeter()`، برای محاسبه محیط مربع است (همان طور که در لیست برنامه می‌بینید، چون کلاس از نوع `static` است، همه این اعضا نیز `static` می‌باشند).

➤ برنامه‌ای که مجموع مساحت و محیط چند شکل متفاوت از قبیله مربع، دایره، مستطیل و مثلث را از طریق واسطه محاسبه می‌کند. این برنامه کاربردهای مختلفی از قبیل محاسبه مساحت و محیط زمین برای ساخت و ساز یا فروش، محاسبه مساحت و محیط ساختمان جهت نقاشی یا لوله کشی دارد.

منابع

۱. جعفر نژاد قمی، عین اله، عباس نژادورزی، رمضان، برنامه نویسی به زبان C# (ویراست سوم)، بابل انتشارات علوم رایانه، ۱۳۸۸.
۲. عباس نژادورزی، رمضان، حل مسائل C++ (مرجع کامل)، بابل فن آوری نوین، ۱۳۸۸.
۳. عباس نژادورزی، رمضان، آموزش گام به گام برنامه نویسی بانک اطلاعاتی با C#، بابل فن آوری نوین، ۱۳۸۸.
۴. عباس نژادورزی، رمضان، رحیم پور کامی، باقر، هاشمیان، ابراهیم، طراحی سیستم‌های شی گرا به زبان C#، بابل فن آوری نوین، ۱۳۹۱.
۵. عباس نژادورزی، رمضان، یونسی، عمران، عباس نژادورزی، یوسف، حل C++ (آزمایشگاه کامپیوتر مرجع کامل)، بابل فن آوری نوین، ۱۳۹۶.
۶. Andrew Troelsen, "Pro C# ۲۰۰۸ and the .NET ۳,۵ Platform, Fourth Edition", Apress, ۲۰۰۷
۷. Daniel Solis, "Illustrated C# ۲۰۰۸", Apress, ۲۰۰۸
۸. JON SKEET, "C# in Depth", Manning, ۲۰۰۸
۹. Jesse Liberty & Alex Horovitz "Programming .NET ۳,۵", O'Reilly, ۲۰۰۸
۱۰. by Joe Duffy, "Professional .NET Framework ۲,۰", Wrox Press, ۲۰۰۶
۱۱. Donis Marshall, "Microsoft Visual C# ۲۰۰۸: The Language", Microsoft Press, ۲۰۰۸
۱۲. Richard Blum, "C# Network Programming", Sybex, ۲۰۰۳
۱۳. <http://barnamenevis.org>
۱۴. <http://www.codeproject.com/>
۱۵. <http://www.srco.ir/>

