
حل مسائل جاوا

(حل ۱۰۰۰ برنامه – مرجع کامل)

تألیف:

دکتر رمضان عباس نژادورزی
دکتر فاطمه عبدی سقاواز



فن آوری نوین

شماره کتابشناسی ملی	:	ایران ۷۶۳۵۸۹۶
شابک	:	۹۷۸-۶۲۲-۷۳۹۳-۳۷-۸
سرشناسه	:	عباس نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	:	حل مسائل جاوا (حل ۱۰۰۰ برنامه - مرجع کامل) (منابع الکترونیکی: کتاب) / تألیف رمضان عباس نژاد ورزی، فاطمه عبدی سقاواز.
مشخصات نشر	:	بابل: فناوری نوین، ۱۴۰۰.
مشخصات ظاهری	:	۱ منبع برخط (۷۳۷ ص.).: مصور، جدول.
وضعیت فهرست نویسی	:	فیبا
نوع منبع الکترونیکی	:	فایل pdf.
یادداشت	:	دسترسی از طریق وب.
شناسه افزوده	:	عبدی سقاواز، فاطمه، ۱۳۵۶ -
موضوع	:	جاوا (زبان برنامه نویسی کامپیوتر) -- مسائل، تمرین ها و غیره.
موضوع	:	Java (Computer program language) -- Problems, exercises, etc
رده بندی کنگره	:	۷۶/۷۳QA
رده بندی دیویی	:	۰۰۵/۱۳۳
دسترسی و محل الکترونیکی	:	آدرس الکترونیکی منبع

@fanavarienovinpub

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

حل مسائل جاوا (حل ۱۰۰۰ برنامه - مرجع کامل)

تألیف: رمضان عباس نژاد ورزی، فاطمه عبدی سقاواز

نوبت چاپ: چاپ اول

سال چاپ: زمستان ۱۴۰۰

شمارگان: ۲۰۰

قیمت: ۱۹۹۰۰۰ تومان

نام چاپخانه و صحافی: دفتر فنی سورنا

شابک: ۹۷۸-۶۲۲-۷۳۹۳-۳۷-۸

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

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

فروشگاه و پخش کتاب چاپی: تهران، تلفن ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

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

فهرست مطالب

فصل اول: آشنایی با جاوا (دستورات ورودی و خروجی) ۵

۱-۱. مسائل تحلیلی ۵

۱-۲. مسائل برنامه‌نویسی ۲۰

فصل دوم: ساختار تصمیم و حلقه تکرار ۲۹

۱-۱. مسائل تحلیلی ۴۹

۲-۲. مسائل برنامه‌نویسی ۵۶

فصل سوم: توابع و متدها ۵۳

۳-۱. مسائل تحلیلی ۱۵۳

۳-۲. مسائل برنامه‌نویسی ۱۵۹

فصل چهارم: آرایه و رشته‌ها ۶۳

۴-۱. مسائل تحلیلی ۲۴۳

۴-۲. مسائل برنامه‌نویسی ۲۶۱

فصل پنجم: کلاس‌ها و وراثت ۹۱

۵-۱. مسائل تحلیلی ۴۹۱

۵-۲. مسائل برنامه‌نویسی ۴۵۵

فصل ششم: فایل‌های ورودی و خروجی ۱۱

۶-۱. مسائل تحلیلی ۵۱۱

۶-۲. مسائل برنامه‌نویسی ۵۱۷

پیوست الکترونیکی: ۵۴۴

منابع: ۷۰۷

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 ای 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 ۱۱۰ برنامه است که مفاهیمی از قبیل آشنایی با جاوا، عملگرها و دستورات ورودی و خروجی را آموزش می دهد.

فصل دوم، شامل ۲۸۰ برنامه است که مفاهیمی از قبیل ساختارهای تصمیم (if, switch) و ساختارهای تکرار (for, do while, while, break, continue) را آموزش می دهد. فصل سوم، شامل ۱۳۰ 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 ۱۶۰ برنامه است که مفاهیم از قبیل آرایه یک بعدی، دوبعدی و رشته را آموزش می دهد. فصل پنجم، شامل ۲۰۰ 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، هر گونه 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

فصل

۱

آشنایی با جاوا (دستورات ورودی و خروجی)

۱-۱. مسائل تحلیلی

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 $p1$ 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:

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

ابتدا $x * 255$ 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 $x * 18 -$ 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:

```
y = x << 8;
y1 = y - x;
y2 = x << 4;
y2 += (x + x);
y = y1 - y2 + 15;
```

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 اضافه‌شده، 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. آیا همه کلاس‌ها به یک متد main() نیاز دارند؟

خیر. کلاس با متد `main()` نشان‌دهنده نقطه شروع برنامه است. بنابراین، کلاس‌های دیگر نیازی به متد `main()` ندارند.

۹. چرا double بر float ترجیح داده می‌شود؟

دقت double دو برابر float است. بنابراین، برای حفظ دقت در محاسبات، double نسبت به float انتخاب بهتری است.

۱۰. تفاوت کاراکتر در جاوا با کاراکتر در C/C++ چیست؟

در C/C++، کاراکتر، نوع عددی صحیح (۸ بیتی یا کد اسکی) است. اما جاوا از یونیکد (UTF-16) برای نمایش کاراکتر استفاده می‌کند. در جاوا، کاراکتر یک نوع ۱۶ بیتی است.

۱۱. یونیکد چیست؟

یونیکد مجموعه‌ای کامل از کاراکترهای بین‌المللی را که در بیش‌تر زبان‌های بشری / سیستم‌های نوشتاری دنیا یافت می‌شود، در برمی‌گیرد. به عبارت دیگر، یونیکد اجتماعی از همه آن مجموعه کاراکترها است. با استفاده از این مجموعه می‌توان تمامی متن‌های موجود در آن زبان‌ها را به شیوه‌ای استاندارد کدگذاری کرد، نمایش داد و اداره نمود.

۱۲. منظور از تبدیل خودکار نوع چیست؟

دو اصل ضروری در تبدیل ضمنی (خودکار) باید رعایت شود که عبارت‌اند از:

➤ نوع مقصد باید بزرگ‌تر از نوع مبدأ باشد

➤ انواع باید سازگار باشند.

به عنوان مثال تبدیل نوع زیر، خودکار است:

```
int i1 = 15;
double d = i1; //ok
```

۷ آشنایی با جاوا (دستورات ورودی و خروجی)

```
System.out.println("d=" + d); //15.0
```

اما تبدیل نوع زیر مجاز نیست:

```
boolean b = true;  
int i2 = b; //error
```

۱۳. چرا انواع اولیه را به اشیاء تبدیل نمی‌کند؟

به علت آن که این نوع از تبدیل‌ها، سربار غیرضروری ایجاد می‌کند و ممکن است کارایی برنامه‌های جاوا کاهش یابد.

۱۴. چرا انواع اولیه جاوا دامنه ثابتی دارند؟

چون می‌خواهند از ویژگی قابل حمل بودن^۱ پشتیبانی کنند.

۱۵. منظور از قابل حمل بودن چیست؟

به زبان ساده، فرض کنید برنامه‌ای را در یک ماشین توسعه داده‌اید. اکنون قصد دارید آن برنامه را در محیط دیگر (به عنوان مثال، در بسترهای سخت‌افزاری / نسخه‌های نرم‌افزاری متفاوت یا سیستم‌عامل‌های مختلف و غیره) بدون دوباره‌کاری قابل توجه (به صورت ایده آل: بدون هیچ دوباره‌کاری) اجرا کنید. اگر موفق به انجام این کار شدید، می‌توانید ادعا کنید که برنامه‌تان قابل حمل است. باز هم یادآوری می‌کنیم که JVM و bytecode با هم جاوا را قابل حمل می‌سازند.

۱۶. اگر برنامه را به صورت `java CommandLineEx1 Book C# Java` اجرا کنیم، خروجی دستورات زیر چیست؟

```
package fundamentals;  
public class CommandLineEx1 {  
    public static void main(String args[])  
    {  
        System.out.println(args[0]);  
        System.out.println(args[1]);  
        System.out.println(args[2]);  
    }  
}
```

Book
C#
Java

توضیح: وقتی که از آرگومان‌های خط فرمان به صورت زیر استفاده می‌شود:

```
java <programName> a0 a1 a2 ...
```

مقادیر ارسالی در آرایه رشته‌ای `arg` به صورت `a0 = arg[0]`، `a1 = arg[1]`، `a2 = arg[2]` و غیره

قرار می‌گیرند.

۱۷. خروجی دستورات زیر چیست؟

```
package fundamentals;  
public class FundamentalEx1 {  
    public static void main(String args[])  
    {  
        int a = 5; //ok  
        int b = 07; //ok
```

¹ Portability

```

    System.out.println("a = " + a );
    System.out.println("b = " + b);
}
}
a = 5
b = 7

```

۱۸. خروجی دستورات زیر چیست؟

```

package fundamentals;
public class FundamentalEx2 {
    public static void main(String args[])
    {
        int a = 5;    //ok
        int b = 07;   //ok
        int c = 09;   //Error
        System.out.println("a = " + a );
        System.out.println("b = " + b);
        System.out.println("c = " + c);
    }
}

```

Compilation error: The literal 09 is out of range.

توضیح: هنگامی که از پیشوند 0 برای اعداد استفاده می‌شود، جاوا آن عدد را به صورت مبنای ۸ در نظر می‌گیرد. این در حالی است که دامنه مبنای ۸ بین ارقام ۰ تا ۷ است. در این مثال، در دستور `int c = 09;` مقدار ۹ از محدوده مبنای ۸ خارج شده است. پس، برنامه با خطای کامپایلری مواجه خواهد شد. برای چاپ عدد ۸ باید از کد زیر استفاده کرد:

```

int d = 010; //ok. It will print 8
System.out.println("d = " + d );

```

۱۹. خروجی دستورات زیر چیست؟

```

package fundamentals;
public class FundamentalEx3 {
    public static void main(String args[])
    {
        int c = 0x12; //ok, will print 18
        int d = 0x1E; //ok, will print 30
        int e = 0x1F; //ok, will print 31
        System.out.println("c = " + c); //18
        System.out.println("d = " + d); //30
        System.out.println("e = " + e); //31
    }
}

```

```

c = 18
d = 30
e = 31

```

توضیح: وقتی که از پیشوند 0x یا 0X در هنگام تعریف اعداد استفاده می‌شود، جاوا آن‌ها را به عنوان یک عدد صحیح مبنای ۱۶ (هگزا دسیمال) در نظر می‌گیرد. دامنه ارقام مبنای ۱۶ بین ۰ تا ۱۵ است. در این مبنی، برای نمایش اعداد ۱۰ تا ۱۵ از حروف A تا F استفاده می‌شود.

۲۰. خروجی دستورات زیر چیست؟

```

package fundamentals;
public class FundamentalEx4 {
    public static void main(String args[])
    {

```



```
int a = 5;
double const = 3.14; //Error
System.out.println("const value is = " + const);
//Error
}
```

Compilation error.

توضیح: در جاوا از کلمه کلیدی `const` برای تعریف ثوابت استفاده می‌شود. اما در این مثال از آن به عنوان نام متغیر استفاده شده است که نادرست است. چون کلمات کلیدی نمی‌توانند به عنوان نام متغیر استفاده شوند، بنابراین، این برنامه با پیام خطا مواجه خواهد شد.

۲۱. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx5 {
    public static void main(String args[])
    {
        byte b1 = 127; //ok
        byte b2 = 128; //Error
        System.out.println("b1 = " + b1);
        System.out.println("b2 = " + b2);
    }
}
```

Compilation error.

توضیح: دامنه اعداد نوع بایت (`byte`) بین ۱۲۸- تا ۱۲۷ می‌باشد. پس، ۱۲۸ نمی‌تواند در متغیر نوع `byte` قرار گیرد و دستور `byte b2 = 128;` با خطای کامپایلری مواجه می‌گردد.

۲۲. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx6 {
    public static void main(String args[])
    {
        byte b1 = 127; //ok
        int i1 = b1; //ok
        System.out.println("b1 = " + b1);
        System.out.println("i1 = " + i1);
    }
}
```

b1 = 127

i1 = 127

توضیح: دو نوع `int` و `byte` با یکدیگر سازگار هستند. قرار دادن نوع `byte` در `int` صحیح است. چون طول نوع مقصد (یعنی، متغیر `i1`) بزرگ‌تر از نوع مبدأ (یعنی، متغیر `b1`) بوده، کامپایلر با این تبدیل مشکلی ندارد.

۲۳. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx7 {
    public static void main(String args[])
    {
        byte b1 = 127;
        int i1 = b1; //ok: small to big
        b1=i1; //Error: big to small
        System.out.println("b1 = " + b1);
        System.out.println("i1 = " + i1);
    }
}
```

}
Compilation error.

توضیح: دو نوع `int` و `byte` با یکدیگر سازگار هستند. قرار دادن نوع `int` در `byte` صحیح نیست. چون طول نوع مقصد (یعنی، متغیر `b1`) کوچکتر از نوع مبدأ (یعنی، متغیر `i1`) بوده، کامپایلر با خطا مواجه می‌شود.

۲۴. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx9
{
    public static void main(String args[])
    {
        int i = 5;
        int j = i++;           //j becomes 5, i becomes 6
        System.out.println("j now = " + j);
        System.out.println("i now = " + i);
        int k = ++j;           //j and k both becomes 6
        System.out.println("j = " + j);
        System.out.println("k = " + k);
    }
}
```

j now = 5
i now = 6
j = 6
k = 6

۲۵. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx10 {
    public static void main(String args[])
    {
        int i = 260;
        byte b = (byte) i;
        System.out.println("b = " + b);
    }
}
```

b = 4

توضیح: این مثال می‌خواهد یک نوع بزرگ‌تر (`int`) را به یک نوع کوچک‌تر (`byte`) تغییر نوع دهد (type casting). در این وضعیت، جاوا، باقی‌مانده تقسیم صحیح مقدار متغیر بزرگ‌تر را بر دامنه متغیر کوچک‌تر محاسبه می‌کند. دامنه نوع `byte` بین ۱۲۸- تا ۱۲۷+ است. بنابراین، نتیجه نهایی ۲۵۶ % ۲۶۰ برابر ۴ خواهد بود.

۲۶. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx11 {
    public static void main(String args[])
    {
        int i = 65550;
        short s = (short) i;
        System.out.println("s = " + s);
    }
}
```

s = 14

توضیح: دامنه نوع `short` از ۳۲۷۶۸- تا ۳۲۷۶۷+ می‌باشد که فاصله کلی آن برابر ۶۵۵۳۶ است. بنابراین، مقدار متغیر `s` برابر خواهد بود با:

$$65550 \% 65536 = 14$$

۲۷. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx12 {
    public static void main(String args[])
    {
        char c1 = 65;
        char c2 = 'a'+ 3;
        System.out.println("c1 = " + c1);
        System.out.println("c2 = " + c2);
    }
}
```

c1 = A
c2 = d

توضیح: کد اسکی A، ۶۵ و کد اسکی a، ۹۷ است. ۹۷ به علاوه ۳ برابر است با ۱۰۰ که کد اسکی d است.

۲۸. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx13 {
    public static void main(String args[])
    {
        int x = 10;
        int result = ++ x * 5;
        System.out.println(" The result is : " + result);
    }
}
```

The result is : 55

توضیح: اولویت عملگر ++ از عملگر * بالاتر است. بنابراین، ابتدا ++x ارزیابی شده، سپس نتیجه آن ضرب در عدد ۵ می شود.

۲۹. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx14 {
    public static void main(String args[])
    {
        int x = 5;
        System.out.println(" ~x is : " + ~x);
    }
}
```

~x is : -6

توضیح: عدد ۵ در مبنای ۲ به صورت ۰۰۰۰۱۰۱ نمایش داده می شود. ۵ ~ باعث می شود که نمایش قبل به ۱۱۱۱۰۱۰ تبدیل شود (یعنی، تمام یک ها به صفر و همه صفرها به یک تبدیل می شوند) که نتیجه برابر ۶- خواهد بود. برای درک بهتر این مطلب، ۶- را به صورت باینری مکمل ۲ عدد ۶ (یعنی: ۰۰۰۰۱۱۰) است. مکمل ۲ عدد ۶ برابر ۱۱۱۱۰۱۰ است. چون، در مکمل ۲ همه بیت ها را معکوس کرده و سپس ۱ به آن اضافه می شود).

۳۰. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx15 {
    public static void main(String args[])
    {
        int x = 21;
```

```

    int y = 15;
    int z = x ^ y;
    System.out.println(" z is : " + z);
}

```

z is : 26

توضیح: معادل دودویی عدد ۲۱ برابر ۰۰۰۱۰۱۰۱ است، معادل دودویی عدد ۱۵ برابر ۰۰۰۰۱۱۱۱ می‌باشد. عملگر XOR (^) به این صورت عمل می‌کند که اگر فقط یک عملوند ۱ باشد، آنگاه نتیجه ۱ است. بنابراین، نتیجه به صورت ۰۰۰۱۱۰۱۰ می‌شود که برابر عدد ۲۶ است.

۳.۱. خروجی دستورات زیر چیست؟

```

package fundamentals;
public class FundamentalEx16 {
    public static void main(String args[])
    {
        int x = 24;
        int y = 11;
        int result = ++ x * y --;
        System.out.println("Result is : " + result);
        System.out.println("y now : " + y);
    }
}

```

Result is : 275
y now : 10

توضیح:

$++ x = 25, 25 * y -- = 275$

بعد از این عمل، y کاهش می‌یابد.

۳.۲. خروجی دستورات زیر چیست؟

```

package fundamentals;
public class FundamentalEx17 {
    public static void main(String args[])
    {
        int x = 24;
        int y = 11;
        int z = 100;
        //int result= ++ x * y --; //275
        int result= ++ x * -- y % z;
        System.out.println(" Result is : " + result);
        System.out.println(" y now : " + y);
    }
}

```

Result is: 50
y now : 10

توضیح: قبل از ضرب x در y، برای x عملگر ++ و برای y، عملگر -- انجام می‌شود که پس از ضرب، باقی‌مانده نتیجه آن بر z برابر ۵۰ می‌شود. $(250 \% 100 = 50)$.

۳.۳. خروجی دستورات زیر چیست؟

```

package fundamentals;
public class FundamentalEx18 {
    public static void main(String args[])
    {
        int x = 10, y = 4;
        double result = ++ y * x / y;
    }
}

```

۱۳ آشنایی با جاوا (دستورات ورودی و خروجی)

```
        System.out.println(" Result is : " + result);
        System.out.println(" y now : " + y);
    }
}
```

```
Result is :10.0
y now: 5
```

توضیح: توجه داشته باشید که ابتدا y افزایش یافته و مقدارش ۵ می شود و به علت آن که نتیجه را در یک متغیر از نوع double ذخیره شده است، مقدار 10.0 شده است.

۳۴. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx19 {
    public static void main(String args[])
    {
        int a = 7, b = 12;
        System.out.println(a + b)           ;//19
        System.out.println("a + b = " + a + b); //a+b=712
        System.out.println(a + b + " = a + b = " + a + b);
    }
}
```

```
19
a + b = 712
19 = a + b = 712
```

توضیح: عملگر + هنگامی که بین دو رشته قرار می گیرد، عمل اتصال رشته ها را انجام می دهد. به همین دلیل، در a + b، نتیجه ۱۹ می شود، ولی در مواجهه با "a + b" که رشته است، عمل اتصال صورت می گیرد.

۳۵. خروجی برنامه زیر چیست؟

```
class output {
    public static void main(String args[])
    {
        double a, b,c;
        a = 3.0 / 0;
        b = 0 / 4.0;
        c = 0 / 0.0;
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```

```
Infinity
0.0
NaN
```

توضیح: در نوع اعشاری، مقدار ثابتی برای (0.0 / 0.0) بی نهایت مثبت یا منفی داریم و همچنین NaN داریم (عددی برای مقدار تعریف نشده مانند 0.0 / 0 نیست)، اما برای نوع صحیح، ثابت بی نهایت نداریم. به همین دلیل استثنا به حساب می آید.

۳۶. خروجی دستورات زیر چیست؟

```
class increment {
    public static void main(String args[])
    {
        int g = 3;
        System.out.print(++ g * 8);
    }
}
```

```

    }
}
32

```

توضیح: در این عبارت عملگر ++ اولویت بیش تری نسبت به * دارد. بنابراین، مقدار g برابر 4 می شود و وقتی در 8 ضرب می گردد 32 به دست می آید.

۳۷. خروجی دستورات زیر چیست؟

```

class area {
    public static void main(String args[])
    {
        double r, pi, a;
        r = 9.8;
        pi = 3.14;
        a = pi * r * r;
        System.out.println(a);
    }
}
301.5656

```

۳۸. خروجی دستورات زیر چیست؟

```

class mainclass {
    public static void main(String args[])
    {
        char a = 'A';
        a ++;
        System.out.print((int) a);
    }
}
66

```

توضیح: کد اسکی کاراکتر 'A' برابر 65 است، با استفاده از ++ مقدار کاراکتر یکی افزایش می یابد. پس، int متغیر a برابر 66 خواهد شد.

۳۹. خروجی دستورات زیر چیست؟

```

class booloperators {
    public static void main(String args[])
    {
        boolean var1 = true, var2 = false;
        System.out.println((var1 & var2));
    }
}
false

```

توضیح: عملگر منطقی "&" مقدار true (وقتی هر دو عملوند true باشند) یا false (وقتی حداقل یکی از آن ها false باشد) را برمی گرداند. var1 با مقدار true تعریف شده است و var2 با مقدار false تعریف گردید. بنابراین، نتیجه عملگر "&" آن ها برابر false است.

۴۰. خروجی دستورات زیر چیست؟

```

class asciiCodes {
    public static void main(String args[])
    {
        char var1 = 'A', var2 = 'a';
        System.out.println((int) var1 + " " + (int) var2);
    }
}

```

آشنایی با جاوا (دستورات ورودی و خروجی) ۱۵

65 97

توضیح: کد اسکی کاراکتر 'A' برابر 65 و کد اسکی کاراکتر 'a' برابر 97 است.

۴۱. خروجی دستورات زیر چیست؟

```
class increment
{
    public static void main(String args[])
    {
        double var1 = 1 + 5;
        double var2 = var1 / 4;
        int var3 = 1 + 5;
        int var4 = var3 / 4;
        System.out.print(var2 + " " + var4);
    }
}
```

1.5 1

۴۲. خروجی دستورات زیر چیست؟

```
class Modulus
{
    public static void main(String args[])
    {
        double a = 25.64;
        int b = 25;
        a = a % 10;
        b = b % 10;
        System.out.println(a + " " + b);
    }
}
```

5.6400000000000001 5

توضیح: عملگر %، باقی مانده عملیات تقسیم را بر روی عملوندها برمی گرداند. نتیجه عبارت 10 % a = a (10 % 25.64) برابر 5.6400000000000001 است. به طور مشابه، b = b % 10 مقدار 5 را برمی گرداند.

۴۳. خروجی دستورات زیر چیست؟

```
class Output
{
    public static void main(String args[])
    {
        int a = 1, b = 2, c, d;
        c = ++ b;
        d = a ++;
        c ++;
        b ++;
        ++ a;
        System.out.println(a + " " + b + " " + c);
    }
}
```

3 4 4

۴۴. خروجی دستورات زیر چیست؟

```
class bitwise_operator
{
    public static void main(String args[])
    {
        int var1 = 42;
```

```

        int var2 = ~var1;
        System.out.print(var1 + " " + var2);
    }
}
42 -43

```

توضیح: عملگر not یا ~، همه بیت‌های عملوند خود را معکوس می‌کند. مقدار 42 در باینری 00101010 است. با استفاده از دستور ~var1 به var2 تخصیص می‌یابد که معکوس 42 مقدار 11010101 به دست می‌آید که 43-دهه‌ای است.

۴۵. خروجی دستورات زیر چیست؟

```

class bitwise_operator
{
    public static void main(String args[])
    {
        int a = 3, b = 6;
        int c = a | b;
        int d = a & b;
        System.out.println(c + " " + d);
    }
}
7 2

```

توضیح: نتیجه عملگر & بیتی، اگر هر دو عملوند 1 باشند، بیت 1 است. اما عملگر | بیتی، اگر حداقل یک عملوند 1 باشد، بیت 1 را تولید خواهد کرد.

۴۶. خروجی دستورات زیر چیست؟

```

class leftshift_operator
{
    public static void main(String args[])
    {
        byte x = 64, y;
        int i;
        i = x << 2;
        y = (byte) (x << 2);
        System.out.print(i + " " + y);
    }
}
256 0

```

۴۷. خروجی دستورات زیر چیست؟

```

class rightshift_operator
{
    public static void main(String args[])
    {
        int x;
        x = 10;
        x = x >> 1;
        System.out.println(x);
    }
}
2

```

توضیح: عملگر >> (شیفت به راست)، مقدار را بر 2 تقسیم می‌کند.

۴۸. خروجی دستورات زیر چیست؟


```
class Output
{
    public static void main(String args[])
    {
        int a = 1, b = 2, c = 3;
        a |= 4;
        b >>= 1;
        c <<= 1;
        a ^= c;
        System.out.println(a + " " + b + " " + c);
    }
}
```

3 1 6

۴۹. خروجی دستورات زیر چیست؟

```
class Relational_operator
{
    public static void main(String args[])
    {
        int var1 = 5, var2 = 6;
        System.out.print(var1 > var2);
    }
}
```

false

توضیح: عملگر > یک مقدار منطقی را برمی گرداند. چون ۵ بیش تر از ۶ نیست، بنابراین، false برگشت داده می شود.

۵۰. خروجی دستورات زیر چیست؟

```
class bool_operator
{
    public static void main(String args[])
    {
        boolean a = true, b = !true;
        boolean c = a | b;
        boolean d = a & b;
        boolean e = d ? b : c;
        System.out.println(d + " " + e);
    }
}
```

false true

توضیح: عملگر |، اگر یکی از عملوندها true باشد، true را برمی گرداند، بنابراین c = true | false برابر true است. اگر هر دو عملوند true باشند، این عملگر true را برمی گرداند. پس d برابر false است. عملگر سه گانه؟: اگر شرط true باشد، نتیجه عبارت قبل از ":" و اگر شرط false باشد، نتیجه عبارت بعد از ":" را به e اختصاص می دهد. d برابر false است، بنابراین، در دستور e = d ? b : c، e را به e اختصاص می دهد. پس e برابر true خواهد شد.

۵۱. خروجی دستورات زیر چیست؟

```
class ternary_operator
{
    public static void main(String args[])
    {
        int x = 3;
        int y = ~ x;
```

```

        int z = x > y ? x : y;
        System.out.print(z);
    }
}
3

```

۵۲. خروجی دستورات زیر چیست؟

```

class Output
{
    public static void main(String args[])
    {
        boolean a = true, b = false;
        boolean c = a ^ b;
        System.out.println(!c);
    }
}
false

```

۵۳. خروجی دستورات زیر چیست؟

```

class operators
{
    public static void main(String args[])
    {
        int var1 = 5, var2 = 6, var3;
        var3 = ++ var2 * var1 / var2 + var2;
        System.out.print(var3);
    }
}
12

```

توضیح: در این عبارت عملگر ++ دارای بالاترین اولویت نسبت به عملگرهای / ، * و + است. پس var2 به 7 افزایش یافته و سپس در عبارت استفاده می‌شود، $var3 = 7 * 5 / 7 + 7$ ، 12 را می‌دهد.

۵۴. خروجی دستورات زیر چیست؟

```

class operators
{
    public static void main(String args[])
    {
        int x = 8;
        System.out.println(++x * 3 + " " + x);
    }
}
27

```

توضیح: در این عبارت عملگر ++ نسبت به عملگر * اولویت بیش‌تری دارد. پس، x به 9 افزایش می‌یابد تا در 3 ضرب شود و 27 به دست آید.

۵۵. خروجی دستورات زیر چیست؟

```

class Output
{
    public static void main(String args[])
    {
        int x = y = z = 20;
    }
}
compile time error

```

۵۶. خروجی دستورات زیر چیست؟

```
class Output
{
    public static void main(String args[])
    {
        int a, b, c, d;
        a = b = c = d = 20
        a += b -= c *= d /= 20
        System.out.println(a + " " + b + " " + c + " " + d);
    }
}
```

20 0 20 1

توضیح: در این دستور عبارت از راست به چپ ارزیابی خواهد شد.

۵۷. خروجی دستورات زیر چیست؟

```
class conversion
{
    public static void main(String args[])
    {
        double a = 295.04;
        int b = 300;
        byte c = (byte) a;
        byte d = (byte) b;
        System.out.println(c + " " + d);
    }
}
```

39 44

توضیح: در هنگام قرار دادن یک متغیر با اندازه بزرگتر در یک متغیر با اندازه کوچکتر، ن باقی مانده تقسیم متغیر با اندازه بزرگتر بر دامنه متغیر با اندازه کوچکتر در متغیر با اندازه کوچکتر قرار خواهد گرفت. متغیر b حاوی ۳۰۰ است که بزرگتر از محدوده بایت یعنی از -128 تا 127 است. از این رو، d شامل 300 باقی مانده تقسیم آن بر 256 یعنی 44 است.

۵۸. خروجی دستورات زیر چیست؟

```
class string_demo
{
    public static void main(String args[])
    {
        String obj = "I" + "like" + "Java";
        System.out.println(obj);
    }
}
```

IlikeJava

توضیح: چنانچه عملگر + بین دو رشته قرار گیرد، آن‌ها را به یکدیگر متصل خواهد کرد.

۲-۱. مسائل برنامه نویسی

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

متغیر	هدف
m	وزن یک مولکول

1	وزن یک لیتر آب
w	وزن آب به لیتر از رودی خوانده می شود.
tedad	تعداد مولکول های آب

```
package ch1_1;
import java.util.*;
public class Ch1_1 {
    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);
    }
}
```

```
Enter w:123
Tedad = 3.895E27
```

۲. هر سال برابر با 3.156×10^7 ثانیه است. برنامه ای که سن شما را به سال دریافت کرده، به ثانیه تبدیل می کند.

```
package ch1_2;
import java.util.*;
public class Ch1_2 {
    public static void main(String[] args) {
        double ss = 3.156e7, second;
        int age;
        System.out.print("Enter age:");
        age = (new Scanner(System.in)).nextInt();
        second = age * ss;
        System.out.printf("Second = %1$s" + "\n", second);
    }
}
```

```
Enter age:48
Second = 1.51488E9
```

متغیر	هدف
ss	هر سال بر حسب ثانیه
age	سن به سال
second	سن بر حسب ثانیه

۳. برنامه ای که حقوق کارمندی را خوانده، بیمه، مالیات و دریافتی او را محاسبه کرده، نمایش می دهد. بیمه و مالیات به ترتیب ۷ و ۱۰ درصد حقوق می باشند.

```
package ch1_3;
import java.util.*;
public class Ch1_3 {
    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);
    }
}
```

```
Enter salary:1000000
s = 1000000 b = 70000 m = 100000 p = 830000
```

متغیر	هدف
s	حقوق
b	بیمه
m	مالیات
p	دریافتی

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

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

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

```
package ch1_4;
import java.util.*;
public class Ch1_4 {
    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;
        y3 = y2 + y2 * t;
        System.out.printf("Extera = %1$s\t\t Price next
            year = %2$s", t, y3);
    }
}
```

```
Enter price for year1:100000
Enter price for year2:110000
Price next year = 121000.0          Extera = 0.1
```

متغیر	هدف
T	نرخ تورم
y1	قیمت کالا در سال قبل
y2	قیمت کالا در سال جاری
y3	قیمت تخمینی کالا در سال بعد

۵. مدیر شرکتی به هر متخصص خود ماهانه ۷۵۰۰۰۰ تومان حقوق پرداخت می‌کند. او می‌خواهد بداند که اگر ۱۳.۵ درصد به حقوق هر متخصص اضافه کند، سالانه چقدر به هزینه شرکت اضافه می‌شود. برنامه‌ای که تعداد متخصصین شرکت را خوانده، این کار را انجام می‌دهد.

```
package ch1_5;
import java.util.*;
public class Ch1_5 {
    public static void main(String[] args) {
        int pay = 750000, tedad;
        double extera;
        System.out.print("Enter tedad:");
        tedad = (new Scanner(System.in)).nextInt();
        extera = (double)pay * 12 * tedad * 13.5 / 100;
        System.out.printf("Extera is %1$s" + "\n", extera);
    }
}
```

```
Enter tedad:5
Extera is 6075000.0
```

متغیر	هدف
pay	۷۵۰۰۰۰ (حقوق هر متخصص)
tedad	تعداد متخصصین شرکت
extera	هزینه‌ای که اضافه می‌شود

۶. برنامه‌ای که عددی دورقمی را خوانده، مجموع ارقام و مغلوب آن را چاپ می‌کند.

```
package ch1_6;
import java.util.*;
public class Ch1_6 {
```

```
public static void main(String[] args) {
    int num, n1, n2;
    System.out.print("Enter a number:");
    num = (new Scanner(System.in)).nextInt();
    n1 = num % 10;

    n2 = num / 10;

    System.out.printf("Reverse is %1$s%2$s\tSum is %3$s"
        + "\n", n1, n2, n1 + n2);
}
```

متغیر	هدف
num	عدد دریافتی
n1	رقم یکان
n2	رقم دهگان

```
Enter a number:12
Reverse is 21 Sum is 3
```

۷. برنامه‌ای که x را خوانده، بدون استفاده از عملگر ضرب، عبارت زیر را محاسبه می‌کند:

$y = 31 * x - 17 * x + 5;$

(توضیح: در این برنامه، برای انجام ضرب از عملگر << (شیفت به چپ) استفاده می‌کنیم).

```
package ch1_7;
import java.util.*;
public class Ch1_7 {
    public static void main(String[] args)
    {
        int x, y, m, n;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextInt();
        m = (x << 5) - x;
        n = -((x << 4) + x);
        y = m + n + 5;
        System.out.printf("y = %1$s" + "\n", y);
    }
}
```

متغیر	هدف
m	مقدار $31 * x$
n	مقدار $-17 * x$
x	عدد دریافتی
y	نتیجه عبارت

```
Enter x:14
y = 201
```

۸. برنامه‌ای که ۳ عدد را خوانده، بزرگ‌ترین عدد را نمایش می‌دهد. برای حل این مسئله، عملگر ؟ مورد استفاده قرار گرفت.

```
package ch1_8;
import java.util.*;
public class Ch1_8 {
    public static void main(String[] args)
    {
        int a, b, c, max;
        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();
        max = (a > b) ? a : b;
        max = (max > c) ? max : c;
        System.out.printf("Max is = %1$s" + "\n", max);
    }
}
```

متغیر	هدف
a	اولین عدد
b	دومین عدد
c	سومین عدد
max	بزرگ‌ترین عدد

```
Enter a:10
Enter b:14
Enter c:6
Max is = 14
```

۹. برنامه‌ای که عددی را خوانده، اگر عدد زوج باشد، عبارت Even وگرنه عبارت Odd را چاپ می‌کند. برای حل این مسئله، عملگرهای % و مورد استفاده قرار می‌گیرند. در این برنامه، عدد دریافتی در متغیر num قرار می‌گیرد.

```
package ch1_9;
import java.util.*;
public class Ch1_9 {
    public static void main(String[] args) {
        int num;
        System.out.print("Enter a number::");
        num = (new Scanner(System.in)).nextInt();
        String s = (num % 2 == 0) ? String.valueOf(num) +
            " is Even" : String.valueOf(num) + " is odd";
        System.out.println(s);
    }
}
```

Enter a number::12
12 is Even

۱۰. برنامه‌ای که قد، وزن و سن فردی را خوانده، اندازه لباس‌های او را به صورت زیر محاسبه می‌کند:

- ❖ ضریب افزایش دور کمر برای سن‌های بیش از ۲۸ سال، هر دو سال یک ضریب.
- ❖ ضریب افزایش کت برای سن‌های بیش از ۳۰ سال، هر دو سال یک ضریب.
- ❖ اندازه دور کمر برابر با وزن تقسیم بر ۵.۷، به علاوه ۰.۱ ضریب در افزایش کت.
- ❖ اندازه دور کت برابر با وزن * ۲، به علاوه ۱ تقسیم بر ۸ ضریب در افزایش کت.
- ❖ اندازه کلاه برابر با وزن * ۲.۹، تقسیم بر قد.

متغیر	هدف
s	قد
w	وزن
e	سن
size_c	اندازه کت
size_d	اندازه دور کمر
size_k	اندازه کلاه
y1	ضریب افزایش دور کمر برای هر ۲ سال
y2	ضریب افزایش کت برای هر ۱۰ سال

```
package ch1_10;
import java.util.*;
public class Ch1_10 {
    public static void main(String[] args) {
        double size_c, size_d, size_k;
        int s, w, e;
        System.out.print("Enter s:");
        s = (new Scanner(System.in)).nextInt();
        System.out.print("Enter w:");
        w = (new Scanner(System.in)).nextInt();
        System.out.print("Enter e:");
        e = (new Scanner(System.in)).nextInt();
        size_k = 2.9 * w / s;
        int y1 = (e > 28) ? (int)((e - 28) / 2) : 0;
        int y2 = (e > 30) ? (int)((e - 30) / 10) : 0;
        size_d = w / 5.7 + 1.0 / 10 * y1;
        size_c = w * s + 1.0 / 8 * y2;
        System.out.printf("Size_c = %1s\tSize_k = %2s\tSize_d\n", size_c, size_k, size_d);
    }
}
```

Enter s:26
Enter w:17

```
Enter e:18
Size_c = 442.0      Size_k = 1.89615384615
Size_d = 2.982456140350877
```

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

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

```
package ch1_11;
import java.util.*;
public class Ch1_11 {
    public static void main(String[] args){
        double x, y;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextDoble();
        y = 1.0 / (x * x + x + 1);
        System.out.printf("y is %1$s", y);
    }
}
```

متغیر	هدف
x	ورودی
y	نتیجه عبارت

```
Enter x:10
y is 0.009000900090009009
```

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

جریان * مقاومت = ولتاژ

```
package ch1_12;
import java.util.*;
public class Ch1_12 {
    public static void main(String[] args) {
        int r, a, v;
        System.out.print("Enter r:");
        r = (new Scanner(System.in)).nextInt();
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextInt();
        v = r * a;
        System.out.printf("v is %1$s", v);
    }
}
```

متغیر	هدف
r	مقاومت
a	جریان
v	ولتاژ

این برنامه، ابتدا متغیرهای موردنیاز را از نوع int تعریف کرده، با یک پیام مقاومت (r) و جریان (a) را دریافت می‌نماید. در ادامه، ولتاژ را محاسبه نموده، نمایش می‌دهد.

```
Enter R:10
Enter A:20
V is 200
```

۱۳. مجموع مقاومت‌های R1، R2 و R3 که به‌طور موازی به هم متصل‌اند، با فرمول زیر محاسبه می‌شود:

$$\frac{1}{R} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} = \frac{1}{R} = \frac{R2 * R3 + R1 * R3 + R1 * R2}{R1 * R2 * R3}$$

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

```
package ch1_13;
import java.util.*;
public class Ch1_13 {
    public static void main(String[] args) {
        float r, r1, r2, r3;
        System.out.print("Enter r1:");
        r1 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter r2:");
        r2 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter r3:");
        r3 = (new Scanner(System.in)).nextInt();
        r = (r1 * r2 * r3) / (r1 * r2 + r2 * r3 + r1 * r3);
    }
}
```

متغیر	هدف
r1	مقاومت ۱
r2	مقاومت ۲
r3	مقاومت ۳
r	مقاومت کل

۲۵ آشنایی با جاوا (دستورات ورودی و خروجی)

```
System.out.printf("r is %1$s", r);  
}
```

این برنامه متغیرهای r، r1، r2 و r3 را به صورت float تعریف کرده، با یک پیام آن‌ها را می‌خواند. سپس مقاومت r را با فرمول بیان شده محاسبه کرده، نمایش می‌دهد.

```
Enter R1:10  
Enter R2:20  
Enter R3:30  
R is 5.4545455
```

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

```
package ch1_14;  
import java.util.*;  
public class Ch1_14 {  
    public static void main(String[] args) {  
        long kg, g;  
        System.out.print("Enter weight(kg):");  
        Kg = (new Scanner(System.in)).nextInt();  
        g = kg * 1000;  
        System.out.printf("Weight(g) is %1$s ", g);  
    }  
}
```

متغیر	هدف
kg	کیلوگرم (وزن)
g	وزن به گرم

```
Enter weight(kg):20  
Weight(g) is 20000
```

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

```
package ch1_15;  
import java.util.*;  
public class Ch1_15 {  
    public static void main(String[] args) {  
        int salary, reward;  
        System.out.print("Enter salary:");  
        Salary = (new Scanner(System.in)).nextInt();  
        reward = salary * 15 / 100;  
        System.out.printf("Reward is %1$s", reward);  
    }  
}
```

متغیر	هدف
salary	حقوق
reward	پاداش

این برنامه، حقوق (salary) و پاداش (reward) را با نوع float تعریف کرده، با یک پیام حقوق را خوانده، پاداش را محاسبه کرده، نمایش می‌دهد.

```
Enter salary:10000000  
Reward is 1500000
```

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

قاعده * ارتفاع × 0.5 = مساحت مثلث

```
package ch1_16;  
import java.util.*;  
public class Ch1_16 {  
    public static void main(String[] args) {  
        float x, y, s;  
        System.out.print("Enter x:");  
        x = (new Scanner(System.in)).nextFloat();  
        System.out.print("Enter y:");  
        y = (new Scanner(System.in)).nextFloat();  
        s = (float)1 / 2 * x * y;
```

متغیر	هدف
x	ارتفاع
y	قاعده
s	مساحت

```

        System.out.printf("Area is %1$s", s);
    }
}
Enter x:10
Enter y:20
Area is 100.0
    
```

۱۷. در شرکتی، سالانه ۱۵۰ خودکار، ۵۰ بسته کاغذ A4 مصرف می‌شود. در پایان سال این شرکت می‌خواهد بداند 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 اضافه‌شده 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.

```

package ch1_17;
import java.util.*;
public class Ch1_17 {
    public static void main(String[] args) {
        float a4, pen, t, coste;
        System.out.print("Enter a4 price:");
        a4 = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter pen price:");
        pen = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter t:");
        t = (new Scanner(System.in)).nextFloat();
        coste = 50 * pen * t / 100 + 150 * a4 * t / 100;
        System.out.printf("Extra cost is %1$s", coste);
    }
}
    
```

متغیر	هدف
a4	قیمت یک بسته A4
pen	قیمت خودکار
t	تورم
coste	هزینه اضافه‌شده

```

Enter a4 price:10000
Enter pen price:1000
Enter t:12
Extra cost is 186000.0
    
```

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

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

```

package ch1_18;
import java.util.*;
public class Ch1_18 {
    public static void main(String[] args) {
        float k, x, n, acceleration;
        System.out.print("Enter k:");
        k = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextFloat();
        acceleration = (k - x) * 60 / n;
        System.out.printf("acceleration is %1$s", acceleration);
    }
}
    
```

متغیر	هدف
k	سرعت نهایی
x	سرعت اولیه
n	زمان
acceleration	شتاب

```
}  
}
```

این برنامه، متغیرها را تعریف کرده، با یک پیام سرعت نهایی، سرعت اولیه و زمان را خوانده، شتاب را محاسبه می‌کند و نمایش می‌دهد.

```
Enter k:10  
Enter x:16  
Enter n:20  
acceleration is -18.0
```

۱۹. 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.

```
package ch1_19;  
import java.util.*;  
public class Ch1_19 {  
    public static void main(String[] args) {  
        System.out.print("Enter Consumption:");  
        float capacity=(new Scanner(System.in)).nextFloat();  
        System.out.print("Enter Consumption:");  
        float consumption=(new Scanner(System.in)).nextFloat();  
        float distance = capacity / consumption * 100;  
        System.out.printf("Distance is %1$s", distance);  
    }  
}
```

```
Enter Consumption:100  
Enter Consumption:200  
Distance is 50.0
```

متغیر	هدف
capacity	ظرفیت
distance	فاصله
consumption	مصرف

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

```
package ch1_20;  
import java.util.*;  
public class Ch1_20 {  
    public static void main(String[] args) {  
        int a, b;  
        System.out.print("Enter a:");  
        a = (new Scanner(System.in)).nextInt();  
        System.out.print("Enter b:");  
        b = (new Scanner(System.in)).nextInt();  
        System.out.printf("%1$s + %2$s = %3$s" + "\n", a, b, a + b);  
        System.out.printf("%1$s - %2$s = %3$s" + "\n", a, b, a - b);  
        System.out.printf("%1$s * %2$s = %3$s" + "\n", a, b, a * b);  
        System.out.printf("%1$s / %2$s = %3$s" + "\n", a, b, (float)a/b);  
        System.out.printf("%1$s %% %2$s = %3$s" + "\n", a, b, a % b);  
    }  
}
```

```
Enter a:10  
Enter b:20  
10 + 20 = 30  
10 - 20 = -10  
10 * 20 = 200  
10 / 20 = 0.5  
10 % 20 = 10
```

متغیر	هدف
a	عدد اول
b	عدد دوم

۲۱. 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.

```
package ch1_21;
import java.util.*;
public class Ch1_21 {
    public static void main(String[] args) {
        int a, b;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b:");
        String s = "";
        b = (new Scanner(System.in)).nextInt();
        s = (a > b) ? String.valueOf(a) + " is larger than " +
            String.valueOf(b) : s;
        s = (a < b) ? String.valueOf(a) + " is less than " +
            String.valueOf(b) : s;
        s = (a == b) ? "These number are equal. " : s;
        System.out.println(s);
    }
}
```

متغیر	هدف
a	عدد اول
b	عدد دوم

```
Enter a:10
Enter b:20
10 is less than 20
```

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

```
package ch1_22;
import java.util.*;
public class Ch1_22 {
    public static void main(String[] args) {
        int a, b, c, max;
        int min, sum, product;
        float average;
        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();
        min = max = a;
        max = (max < b) ? b : max;
        max = (max < c) ? c : max;
        min = (min > b) ? b : min;
        min = (min > c) ? c : min;
        sum = a + b + c;
        average = (float)(a + b + c) / 3;
```

متغیر	هدف
a	عدد اول
b	عدد دوم
c	عدد سوم
sum	مجموع سه عدد
average	میانگین سه عدد
product	حاصل ضرب سه عدد
min	کوچک‌ترین عدد
max	بزرگ‌ترین عدد

```

        product = a * b * c;
        System.out.printf("sum is %1$s " + "\n", sum);
        System.out.printf("Average is %1$s " + "\n", average);
        System.out.printf("Product is %1$s" + "\n", product);
        System.out.printf("Smallest is %1$s " + "\n", min);
        System.out.printf("Largest is %1$s" + "\n", max);
    }
}
Enter a:10
Enter b:5
Enter c:20
sum is 35
Average is 11.666667
Product is 1000
Smallest is 5
Largest is 20

```

۲۳. برنامه‌ای که دو عدد را از صفحه کلید می‌گیرد و تعیین می‌کند که آیا عدد اول بر عدد دوم بخش‌پذیر است یا نه؟

```

package ch1_23;
import java.util.*;
public class Ch1_23 {
    public static void main(String[] args) {
        int a, b;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b:");
        b = (new Scanner(System.in)).nextInt();
        String s = (a % b == 0) ? String.valueOf(a) + " mod " +
            String.valueOf(b) + " is zero." : String.valueOf(a) + "mod "
            + String.valueOf(b) + " is not zero.";
        System.out.println(s);
    }
}
Enter a:10
Enter b:2
10 mod 2 is zero.

```

منغیر	هدف
a	عدد اول
b	عدد دوم

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

```

package ch1_24;
import java.util.*;
public class Ch1_24 {
    public static void main(String[] args) {
        int a, b, c, d, e, max, min;
        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();
        System.out.print("Enter d:");
        d = (new Scanner(System.in)).nextInt();
        System.out.print("Enter e:");

```

منغیر	هدف
a	اولین عدد
b	دومین عدد
c	سومین عدد
d	چهارمین عدد
e	پنجمین عدد
max	بزرگ‌ترین عدد
min	کوچک‌ترین عدد

```

e = (new Scanner(System.in)).nextInt();
min = max = a;
max = (max < b) ? b : max;
max = (max < c) ? c : max;
max = (max < d) ? d : max;
max = (max < e) ? e : max;
min = (min > b) ? b : min;
min = (min > c) ? c : min;
min = (min > d) ? d : min;
min = (min > e) ? e : min;
System.out.printf("Smallest is %1$s" + "\n", min);
System.out.printf("Largest is %1$s", max);
}
}
Enter a:10
Enter b:9
Enter c:11
Enter d:12
Enter e:7
Smallest is 7
Largest is 12

```

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

```

package ch1_25;
import java.util.*;
public class Ch1_25 {
    public static void main(String[] args) {
        System.out.println("Number\tSquare\tCube");
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 0, 0 * 0, 0 * 0 * 0);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 1, 1 * 1, 1 * 1 * 1);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 2, 2 * 2, 2 * 2 * 2);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 3, 3 * 3, 3 * 3 * 3);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 4, 4 * 4, 4 * 4 * 4);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 5, 5 * 5, 5 * 5 * 5);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 6, 6 * 6, 6 * 6 * 6);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 7, 7 * 7, 7 * 7 * 7);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 8, 8 * 8, 8 * 8 * 8);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 9, 9 * 9, 9 * 9 * 9);
        System.out.printf("%1$s\t%2$s\t%3$s" + "\n", 10, 10 * 10, 10 * 10 * 10);
    }
}

```

Number	Square	Cube
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

۲۶. برنامه‌ای که یک عدد صحیح پنج رقمی 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

متغیر	هدف
n	عدد پنج رقمی
temp	متغیر کمکی
r1	رقم اول
r2	رقم دوم
r3	رقم سوم
r4	رقم چهارم
r5	رقم پنجم

```
package ch1_26;
import java.util.*;
public class Ch1_26 {
    public static void main(String[] args) {
        int n, temp, r1, r2, r3, r4, r5;
        System.out.print("Enter a number between 10000 to 99999:");
        n = (new Scanner(System.in)).nextInt();
        temp = n;
        r5 = temp % 10;
        temp = temp / 10;
        r4 = temp % 10;
        temp = temp / 10;
        r3 = temp % 10;
        temp = temp / 10;
        r2 = temp % 10;
        temp = temp / 10;
        r1 = temp % 10;
        temp = temp / 10;
        System.out.printf("Result is %1$s %2$s %3$s %4$s %5$s" + "\n", r1, r2, r3, r4, r5);
    }
}
```

Enter a number between 10000 to 99999:67432
Result is 6 7 4 3 2

۲۷. هر تن 35273.92 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.

متغیر	هدف
ton	تبدیل شده به تن
ons	وزن به اونس
rate	واحد تبدیل تن به اونس

```
package ch1_27;
import java.util.*;
public class Ch1_27 {
    public static void main(String[] args) {
        final float rate = 35273.92f;
        float ton, ons;
        System.out.print("Enter ons:");
        ons = (new Scanner(System.in)).nextFloat();
        ton = ons / rate;
        System.out.printf("Ton is %1$s" + "\n", ton);
    }
}
```

Enter ons:100
Ton is 0.0028349555

۲۸. هر لیتر معادل 0.264179 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.

متغیر	هدف
liter	مصرف به لیتر

مسافت به مایل	mile
مصرف به گالن	gallon
نسبت مصرف مایل به گالن	result

```
package ch1_28;
import java.util.*;
public class Ch1_28 {
    public static void main(String[] args) {
        float liter, gallon, mile;
        double result;
        System.out.print("Enter liter:");
        liter = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter mile:");
        mile = (new Scanner(System.in)).nextFloat();
        result = (double)(mile / liter) * 0.264179;
        System.out.printf("miles / gallons is %1$s" + "\n",result);
    }
}
```

```
Enter liter:10
Enter mile:150
miles / gallons is 3.962685
```

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

متغیر	هدف
x	عدد خوانده شده

```
package ch1_29;
import java.util.*;
public class Ch1_29 {
    public static void main(String[] args) {
        float x;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextFloat();
        System.out.printf("%1$s ^ 2 = %2$s\n%1$s ^ 3 = %3$s" +
            "\n", x, x * x, x * x * x);
    }
}
```

```
Enter x:10
10.0 ^ 2 = 100.0
10.0 ^ 3 = 1000.0
```

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

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

متغیر	هدف
a	ضلع اول
b	ضلع دوم
h	ارتفاع
s	مساحت

```
package ch1_30;
import java.util.*;
public class Ch1_30 {
    public static void main(String[] args) {
        float a, b, h, s;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter b:");
        b = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter h:");
        h = (new Scanner(System.in)).nextFloat();
        s = h * (a + b) / 2;
        System.out.printf("S is %1$s" + "\n", s);
    }
}
```

```
Enter a:10
Enter b:15
```


Enter h:17
S is 212.5

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

$$\begin{aligned} \text{شعاع} &= 2 * \text{قطر} \\ \text{محیط دایره} &= 2 * \text{شعاع} * \pi \\ \text{مساحت دایره} &= \pi * (\text{شعاع})^2 \end{aligned}$$

```
package ch1_31;
import java.util.*;
public class Ch1_31 {
    public static void main(String[] args) {
        final double PI = Math.PI;
        double r, a, p, n;
        System.out.print("Enter r:");
        r = (new Scanner(System.in)).nextDouble();
        n = 2 * r;
        p = n * PI;
        a = PI * r * r;
        System.out.printf(" n = %1$s\n p = %2$s\n a = %3$s" + "\n", n,
            p, a);
    }
}
```

متغیر	هدف
pi ثابت	مقدار π یعنی 3.14159 را نگهداری می‌کند.
r	شعاع خوانده شده را نگهداری می‌کند
a	مساحت دایره را نگهداری می‌کند.
p	محیط دایره را نگهداری می‌کند.
n	قطر دایره را نگهداری می‌کند.

Enter r:10
n = 20.0
p = 62.83185307179586
a = 314.1592653589793

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

$$\begin{aligned} \text{سال} &= 365.25 * \text{روز} \\ \text{ماه} &= \text{روز} / 30 \\ \text{ثانیه} &= \text{روز} * 24 * 60 * 60 \end{aligned}$$

```
package ch1_32;
import java.util.*;
public class Ch1_32 {
    public static void main(String[] args) {
        int year;
        float day, second, month;
        System.out.print("Enter year:");
        year=(new Scanner(System.in)).nextInt();
        day = 365.25f * year;
        System.out.printf("Day is %1$s" + "\n", day);
        month = day / 30;
        System.out.printf("Month is %1$s" + "\n", month);
        second = day * 24 * 60 * 60;
        System.out.printf("Second is %1$s" + "\n", second);
    }
}
```

متغیر	هدف
year	عدد ورودی به سال
day	عدد تبدیل شده به روز
month	عدد تبدیل شده به ماه
second	عدد تبدیل شده به ثانیه

```

    }
Enter year:48
Day is 17532.0
Month is 584.4
Second is 1.5147648E9

```

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

```

a = m2 - n2
b = 2 * m * n
c = m2 + n2

package ch1_33;
import java.util.*;
public class Ch1_33 {
    public static void main(String[] args) {
        float m, n, a, b, c;
        System.out.print("Enter m:");
        m = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter n:");
        n = (new Scanner(System.in)).nextFloat();
        a = m * m - n * n;
        b = 2 * m * n;
        c = m * m + n * n;
        System.out.printf("a is %1$s" + "\n", a);
        System.out.printf("b is %1$s" + "\n", b);
        System.out.printf("c is %1$s" + "\n", c);
    }
}
Enter m:10
Enter n:29
a is -741.0
b is 580.0
c is 941.0

```

۳۴. 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.

```

package ch1_41;
package ch1_34;
import java.util.*;
public class Ch1_34 {
    public static void main(String[] args) {
        float a, b;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter b:");
        b = (new Scanner(System.in)).nextFloat();
        System.out.printf(" a + b = %1$s" + "\n", a + b);
        System.out.printf(" a - b = %1$s" + "\n", a - b);
        System.out.printf(" a * b = %1$s" + "\n", a * b);
        System.out.printf(" a / b = %1$s" + "\n", (float)a / b);
        System.out.printf(" a ^ 2 + b ^ 2 = %1$s" + "\n", a * a
            + b * b);
        System.out.printf(" a ^ 3 + b ^ 3 = %1$s" + "\n", a * a
            * a + b * b * b);
    }
}
Enter a:10
Enter b:20
a + b = 30.0
a - b = -10.0
a * b = 200.0
a / b = 0.5

```

آشنایی با جاوا (دستورات ورودی و خروجی) ۳۵

```
a ^ 2 + b ^ 2 = 500.0
a ^ 3 + b ^ 3 = 9000.0
```

۳۵. برنامه‌ای که یک عدد سه‌رقمی را خوانده، اگر مجموع رقم‌های اول و سوم برابر رقم دوم باشد، "Yes"، وگرنه "No" را چاپ می‌کند.

```
package ch1_35;
import java.util.*;
public class Ch1_35 {
    public static void main(String[] args) {
        int a, r1, r2, r3;
        System.out.print("Enter a number[100 to 999]:");
        a = (new Scanner(System.in)).nextInt();
        r1 = a % 10;
        r3 = a / 100;
        r2 = a / 10 % 10;
        String s = (r1 + r3 == r2) ? "Yes " : "No";
        System.out.println(s);
    }
}
```

متغیر	هدف
a	عدد خوانده شده
r1	رقم یکان
r2	رقم دهگان
r3	رقم صدگان

```
Enter a number[100 to 999]:342
No
```

۳۶. 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).

```
package ch1_36;
import java.util.*;
public class Ch1_36 {
    public static void main(String[] args) {
        int pages, Gb;
        float number;
        long bookBytes;
        System.out.print("Enter pages:");
        pages=(new Scanner(System.in)).nextInt();
        System.out.print("Enter size of memory:");
        Gb = (new Scanner(System.in)).nextInt();
        bookBytes = 80 * 30 * pages;
        number = (float) 1024 * 1024 * 1024 * Gb / bookBytes;
        System.out.printf("Number of book is %1$s"+"\\n",number);
    }
}
```

متغیر	هدف
pages	تعداد صفحات کتاب که از ورودی خوانده می‌شود.
Gb	ظرفیت فلش به گیگابایت که از ورودی خوانده می‌شود.
Number	تعداد کتاب‌هایی که این فلش می‌تواند ذخیره کند.
bookbytes	تعداد بایت‌هایی که این کتاب نیاز دارد.

```
Enter pages:1000
Enter size of memory:16
Number of book is 7158.279
```

۳۷. 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.

متغیر	هدف
num	عدد چهاررقمی خوانده شده
n1	رقم اول (یکان)
n2	رقم دوم (دهگان)
n3	رقم سوم (صدگان)
n4	رقم چهارم (هزارگان)

```
package ch1_37;
import java.util.*;
public class Ch1_37 {
    public static void main(String[] args) {
        int num, n1, n2, n3, n4;
        System.out.print("Enter a number[1000 to 9999] :");
        num = (new Scanner(System.in)).nextInt();
        n1 = num % 10;
        n2 = num % 100 / 10;
        n3 = num / 100 % 10;
        n4 = num / 1000;
        String s = ((n2 + n3) == (n1 * n4)) ? "Yes" : "NO";
        System.out.println(s);
    }
}
```

```
Enter a number[1000 to 9999] :1001
NO
```

۳۸. برنامه‌ای که تعداد کالا و قیمت هر کالا را خوانده، مبلغ فروش را نمایش می‌دهد (مبلغ فروش برابر با تعداد کالا * قیمت کالا است).

متغیر	هدف
count	تعداد کالا
price	قیمت کالا
sell	مبلغ کالا

```
package ch1_38;
import java.util.*;
public class Ch1_38 {
    public static void main(String[] args) {
        int count, price;
        long sell;
        System.out.print("Enter count:");
        count = (new Scanner(System.in)).nextInt();
        System.out.print("Enter price:");
        price = (new Scanner(System.in)).nextInt();
        sell = count * price;
        System.out.printf("Sells is %1$s" + "\n", sell);
    }
}
```

```
Enter count:10
Enter price:60000
Sells is 600000
```

۳۹. برنامه‌ای که سن‌تان را به سال خوانده، مشخص کند چند دقیقه زندگی کرده‌اید (هرسال 365.25 روز است و هرروز (60×24) دقیقه می‌باشد).

متغیر	هدف
year	سن‌تان به سال که از ورودی خوانده می‌شود.
minute	سن‌تان که به دقیقه تبدیل شده است.

```
package ch1_39;
import java.util.*;
public class Ch1_39 {
```

```
public static void main(String[] args) {
    double year, minute;
    System.out.print("Enter year:");
    year = (new Scanner(System.in)).nextDouble();
    minute = year * 365.25 * 24 * 60 * 60;
    System.out.printf("Minute is %1$s", minute);
}
```

```
Enter year:10
Minute is 3.15576E8
```

۴۰. 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).

متغیر	هدف
ثابت M-PER-MILE	ثابت نرخ تبدیل مایل به متر
ثابت M-PER-FOOT	ثابت نرخ تبدیل فوت به متر
miles	مایل که از ورودی خوانده می‌شود.
feet	فوت که از ورودی خوانده می‌شود.
meters	مقدار فوت و مایل که به متر تبدیل شده را نگه‌داری می‌کند.
kilometers	مقادیر فوت و مایل که به کیلومتر تبدیل شده را نگه‌داری می‌کند.

```
package ch1_40;
import java.util.*;
public class Ch1_40 {
    public static void main(String[] args) {
        final double M_PER_MILE = 1609.35;
        final double M_PER_FOOT = 0.30480;
        float miles, feet, kilometers, meters;
        double total_meters, total_kilometers;
        System.out.print("Enter the number of miles: ");
        miles = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter the number of feet:: ");
        feet = (new Scanner(System.in)).nextFloat();
        // Convert everything to meters
        total_meters = miles * M_PER_MILE + feet * M_PER_FOOT;
        // Calculate the number of kilometers
        total_kilometers = total_meters / 1000;
        kilometers = (int) total_kilometers; // truncates to integer
        // Convert decimal part of total_kilometers to meters
        meters = (float)(total_kilometers - kilometers) * 1000;
        // Display results
        System.out.printf("The distance is %1$s kilometers, %2$s meters." + "\n", kilometers, meters);
    }
}
```

```
Enter the number of miles: 100
Enter the number of feet:: 105
The distance is 160.0 kilometers, 967.004 meters
```

۴۱. برنامه‌ای که x و y را از ورودی خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$Z = x++ * y / x++;$

```
package ch1_41;
import java.util.*;
public class Ch1_41 {
    public static void main(String[] args) {
```

```

int x, y, z;
System.out.print("Enter x: ");
x = (new Scanner(System.in)).nextInt();
System.out.print("Enter y: ");
y = (new Scanner(System.in)).nextInt();
z = x++ * y / x++;
System.out.printf("z is %1$s" + "\n", z);
}

```

```

Enter x: 20
Enter y: 30
z is 28

```

۴۲. برنامه‌ای که یک حرف انگلیسی کوچک را خوانده، به حرف بزرگ تبدیل کرده و نمایش می‌دهد.

```

package ch1_42;
import java.util.*;
public class Ch1_42 {
    public static void main(String[] args) {
        String ch;
        System.out.print("Enter a char:");
        ch = new Scanner(System.in).nextLine();
        ch=ch.toUpperCase();
        System.out.printf("char is %1$s", ch );
    }
}

```

```

Enter a char:s
char is S

```

۴۳. برنامه‌ای که x و y را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

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

```

package ch1_43;
import java.util.*;
public class Ch1_43 {
    public static void main(String[] args) {
        int x, y, z;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextInt();
        System.out.print("Enter y:");
        y = (new Scanner(System.in)).nextInt();
        z = x * x * x + 3 * x * x + 3 * y - 5;
        System.out.printf("z = %1$s", z);
    }
}

```

```

Enter x:10
Enter y:20
z = 18995

```

۴۴. 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).

```

package ch1_44;
import java.util.*;
public class Ch1_44 {
    public static void main(String[] args) {
        float grade;
        System.out.print("Enter a grade:");
        grade = (new Scanner(System.in)).nextFloat();
        String s = (grade >= 10) ? "Passed" : "Failed";
        System.out.print(s);
    }
}

```

```

Enter a grade:12.5

```

Passed

۴۵. برنامه‌ای که دو عدد را خوانده، بدون استفاده از متغیر کمکی محتوی آن‌ها را تعویض می‌کند.

```
package ch1_45;
import java.util.*;
public class Ch1_45 {
    public static void main(String[] args) {
        float x, y;
        System.out.print("Enter x:");
        x = (new Scanner(System.in)).nextFloat();
        System.out.print("Enter y:");
        y = (new Scanner(System.in)).nextFloat();
        x = x + y;
        y = x - y;
        x = x - y;
        System.out.printf("X = %1$s\tY = %2$s ", x, y);
    }
}
```

```
Enter x:10
Enter y:20
Y =10.0      X = 20.0
```

۴۶. برنامه‌ای که قاعده و ارتفاع مثلث را خوانده، مساحت مثلث را محاسبه می‌کند (مساحت مثلث برابر با ارتفاع ضربدر نصف قاعده است).

```
package ch1_46;
import java.util.*;
public class Ch_46 {
    public static void main(String[] args) {
        int p, h;
        float s;
        System.out.print("Enter p:");
        p = (new Scanner(System.in)).nextInt();
        System.out.print("Enter h:");
        h = (new Scanner(System.in)).nextInt();
        s = (float)(p * h) / 2;
        System.out.printf("S is %1$s", s);
    }
}
```

```
Enter p:10
Enter h:20
S is 100.0
```

متغیر	هدف
h	ارتفاع
p	قاعده
s	مساحت

۴۷. برنامه‌ای که نرخ حقوق به ازای هر ساعت کارکرد و تعداد ساعت کارکرد یک کارمند را خوانده، میزان حقوق او را محاسبه می‌کند و نمایش می‌دهد (از حقوق ۱۰ درصد به عنوان مالیات کسر می‌شود. حقوق خالص برابر با حقوق منهای مالیات است).

```
package ch1_47;
import java.util.*;
public class Ch1_47 {
    public static void main(String[] args) {
```

متغیر	هدف
h	ساعت کارکرد کارمند
r	نرخ هر ساعت کارکرد
tp	مبلغ حقوق کارمند (مبلغ کل حقوق)
t	مبلغ مالیات
pp	حقوق خالص

```

float r, h, tp, t, pp;
System.out.print("Enter hours:");
h = (new Scanner(System.in)).nextFloat();
System.out.print("Enter rate:");
r = (new Scanner(System.in)).nextFloat();
tp = h * r;
t = (float) tp / 10;
pp = (float) tp - t;
System.out.printf("Total payment:%1$s" + "\n", tp);
System.out.printf("Tax:%1$s" + "\n", t);
System.out.printf("Payment:%1$s" + "\n", pp);
}
}
Enter hours:10
Enter rate:100000
Total payment:1000000.0
Tax:100000.0
Payment:900000.0

```

۴۸. پروژه برنامه‌نویسی ۱: برنامه‌ای که ۵ عدد را خوانده، بزرگ‌ترین و کوچک‌ترین عدد را نمایش می‌دهد.

```

package project1;
import java.util.*;
public class Project1 {
    public static void main(String[] args) {
        int a, b, c, d, e, max, min;
        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();
        System.out.print("Enter d:");
        d = (new Scanner(System.in)).nextInt();
        System.out.print("Enter e:");
        e = (new Scanner(System.in)).nextInt();
        max = min = a;
        max = (b > max) ? b : max;
        max = (c > max) ? c : max;
        max = (d > max) ? d : max;
        max = (e > max) ? e : max;
        min = (b < min) ? b : min;
        min = (c < min) ? c : min;
        min = (d < min) ? d : min;
        min = (e < min) ? e : min;
        System.out.printf("Max is %1$s" + "\n", max);
        System.out.printf("Min is %1$s" + "\n", min);
    }
}

```

متغیر	هدف
a	اولین عدد
b	دومین عدد
c	سومین عدد
d	چهارمین عدد
e	پنجمین عدد
min	کوچک‌ترین عدد
max	بزرگ‌ترین عدد

```

Enter a:1
Enter b:5
Enter c:3
Enter d:7
Enter e:2
Max is 7
Min is 1

```

۴۹. پروژه برنامه‌نویسی ۲: برنامه‌ای که یک عدد ۵ رقمی را خوانده، ارقام عدد را مقلوب نموده و با فاصله نمایش می‌دهد.

هدف

متغیر


```
package project2;
import java.util.*;
public class Project2 {
    public static void main(String[] args) {
        int a1, a2, a3, a4, a5, num, temp;
        System.out.print("Enter a number:");
        num = (new Scanner(System.in)).nextInt();
        a1 = num % 10;
        temp = num / 10;
        a2 = temp % 10;
        temp /= 10;
        a3 = temp % 10;
        temp /= 10;
        a4 = temp % 10;
        temp /= 10;
        a5 = temp % 10;
        temp /= 10;
        System.out.printf("%1$s %2$s %3$s %4$s %5$s"+"\\n", a1, a2, a3, a4, a5);
    }
}
```

```
Enter a number:23537
7 3 5 3 2
```

num	عدد پنج رقمی خوانده شده
temp	متغیر کمکی که پس از حذف هر رقم num را نگه‌داری می‌کند.
a1	رقم یکان
a2	رقم دهگان
a3	رقم صدگان
a4	رقم هزارگان
a5	رقم ده هزارگان

۵۰. برنامه‌ای که مختصات دو نقطه را دریافت کرده، فاصله بین آن دو نقطه را محاسبه می‌نماید و نمایش می‌دهد.

$$\text{فاصله بین دو نقطه} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
package ch1_50;
import java.util.*;
public class Ch1_50 {
    public static void main(String[] args) {
        int x1, x2, y1, y2;
        double d;
        System.out.print("Enter x1:");
        x1 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter x2:");
        x2 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter y1:");
        y1 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter y2:");
        y2 = (new Scanner(System.in)).nextInt();
        d = Math.sqrt(((double)(x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1)));
        System.out.printf("Distance is %1$s" + "\\n", d);
    }
}
```

```
Enter x1:10
Enter x2:10
Enter y1:10
Enter y2:20
Distance is 10.0
```

متغیر	هدف
x ₁	مختصات x نقطه اول
x ₂	مختصات x نقطه دوم
y ₁	مختصات y نقطه اول
y ₂	مختصات y نقطه دوم

۵۱. زمینی به ابعاد 13×16 وجود دارد. موزاییک‌هایی به ابعاد 2×3 ، برای فرش کردن زمین نیاز است. اولاً تعیین کنید چند موزاییک نیاز است. آیا قسمتی از زمین می‌ماند که با موزاییک فرش نشود؟

متغیر	هدف
s_1	کل مساحت زمین
s_2	مساحت یک موزاییک
b	تعداد موزاییک‌های مورد نیاز
a	باقی‌مانده زمین که موزاییک نشده است.

```
package ch1_51;
import java.util.*;
public class Ch1_51 {
    public static void main(String[] args) {
        int s1, s2;
        float b, a;
        s1 = 13 * 16;
        System.out.printf("s1 = %1$s" + "\r\n", s1);
        s2 = 2 * 3;
        System.out.printf("s2 = %1$s" + "\r\n", s2);
        b = (float)s1 / s2;
        a = s1 % s2;
        System.out.printf("b = %1$s" + "\r\n", b);
        System.out.printf("a = %1$s" + "\r\n", a);
    }
}
```

```
s1 = 208
s2 = 6
b = 34.666668
a = 4.0
```

۵۲. برنامه‌ای که a و b را خوانده، مقدار $y = b * \sin(a)$ را نمایش می‌دهد.

```
package ch1_52;
import java.util.*;
public class Ch1_52 {
    public static void main(String[] args) {
        int a, b;
        double y;
        System.out.print("Enter a:");
        a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b:");
        b = (new Scanner(System.in)).nextInt();
        y = a * Math.sin((b * 3.14 / 180));
        System.out.printf("y is %1$s", y);
    }
}
```

```
Enter a:40
Enter b:60
y is 34.630393579693774
```

۵۳. برنامه‌ای که تعداد بایت‌هایی را نمایش می‌دهد که متغیرهای با انواع `double` و `float`، `long`، `int`، `char` اشغال می‌کنند.

```
package ch1_53;
import java.util.*;
public class Ch1_53 {
    public static void main(String[] args) {
        System.out.printf("Size of char is %1$s" + "\n", (
```

۴۳ آشنایی با جاوا (دستورات ورودی و خروجی)

```
        Character.SIZE / Byte.SIZE));
    System.out.printf("Size of int is %1$s" + "\n",
        (Integer.SIZE / Byte.SIZE));
    System.out.printf("Size of long is %1$s" + "\n",
        (Long.SIZE / Byte.SIZE));
    System.out.printf("Size of float is %1$s" + "\n",
        (Float.SIZE / Byte.SIZE));
    System.out.printf("Size of double is %1$s" + "\n",
        (Double.SIZE / Byte.SIZE));
    }
}
```

Size of char is 2
Size of int is 4
Size of long is 8
Size of float is 4
Size of double is 8
...

۵۴. برنامه‌ای که عددی را خوانده، با استفاده از عملگر یا انحصاری تشخیص می‌دهد، عدد خوانده‌شده زوج است یا فرد؟

```
package ch1_54;
import java.util.*;
public class Ch1_54 {
    public static void main(String[] args) {
        int num, n = 1;
        System.out.print("Enter a number:");
        num = (new Scanner(System.in)).nextInt();
        int x = num & n;
        String s = (x != 0) ? String.valueOf(num) + " is odd." :
            String.valueOf(num) + " is even.";
        System.out.println(s);
    }
}
```

Enter a number:100
100 is even.

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

```
package ch1_55;
import java.util.*;
public class Ch1_55 {
    public static void main(String[] args) {
        int num, n, bitStatus;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        System.out.print("Enter nth bit to check (0-31): ");
        n = (new Scanner(System.in)).nextInt();
        bitStatus = (num >> n) & 1;
        System.out.printf("The %1$sth bit is set to %2$s" +
            "\n", n, bitStatus);
    }
}
```

در این برنامه num (عدد ورودی)، n (بیت موردنظر) و bitState (وضعیت یا مقدار n امین بیت num) را داریم.

این برنامه برای خواندن بیت n ام یک عدد، ابتدا عدد موردنظر را n بیت با سمت راست شیفت داده و با ۱ و بیتی می‌نماید.

```
Enter a number: 121
Enter nth bit to check (0-31): 10
```

The 10th bit is set to 0

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

```
package ch1_56;
import java.util.*;
public class Ch1_56 {
    public static void main(String[] args) {
        int num, n, newNum;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        /* Input bit position you want to check */
        System.out.print("Enter nth bit to check (0-31): ");
        n = (new Scanner(System.in)).nextInt();
        /* Left shift 1, n times and perform bitwise OR with num */
        newNum = (1 << n) | num;
        System.out.println("Bit set successfully.");
        System.out.printf("Number before setting %1$s bit:%2$s" +
            "\n", n, num);
        System.out.printf("Number after setting bit:%2$s decimal" +
            "\n", n, newNum);
        int bitStatus = (newNum >> n) & 1;
        System.out.printf("The %1$sth bit is set to %2$s" + "\n", n,
            bitStatus);
    }
}
```

در این برنامه متغیرهای num (عدد ورودی)، n (بیتی که باید یک شود) و newNum (عدد جدید پس از یک شدن بیت n ام) را داریم. این برنامه برای یک کردن بیت n ام عدد num، یک را n بیت به سمت چپ شیفت داده با num یا بیتی می‌کند.

```
Enter a number: 121
Enter nth bit to check (0-31): 10
Bit set successfully.
Number before setting 10 bit:121
Number after setting bit:1145 decimal
The 10th bit is set to 1
```

۵۷. برنامه‌ای که ابتدا عددی را خواند، سپس n را می‌خواند و n امین بیت عدد خوانده‌شده را صفر می‌نماید. به‌عنوان 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 << n
11101111	←	~(1 << n)
00110111	←	num(in binary)
00100111	←	Num & (~ (1 << n))

```
package ch1_57;
import java.util.*;
public class Ch1_57 {
    public static void main(String[] args) {
        int num, n, newNum;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        /* Input bit position you want to check */
        System.out.print("Enter nth bit to check (0-31): ");
```

۴۵ آشنایی با جاوا (دستورات ورودی و خروجی)

```
n = (new Scanner(System.in)).nextInt();
newNum = num & ~(1 << n);
System.out.println("Bit cleared successfully.");
System.out.printf("Number before clearing %1$s bit:%2$s"
    + "\n", n, num);
System.out.printf("Number after clearing bit:%1$s
    decimal" + "\n", newNum);
int bitStatus = (newNum >> n) & 1;
System.out.printf("The %1$sth bit is set to %2$s" +
    "\n", n, bitStatus);
}
```

در این برنامه num (عدد ورودی)، n (عددی که شماره بیت num آن باید صفر شود) و newNum (عدد جدید) را داریم.

```
Enter a number: 121
Enter nth bit to check (0-31): 4
Bit cleared successfully.
Number before clearing 4 bit:121
Number after clearing bit:105 decimal
The 4th bit is set to 0
```

۵۸. 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.

```
package ch1_58;
import java.util.*;
public class Ch1_58 {
    public static void main(String[] args) {
        int num, n, newNum;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        System.out.print("Enter nth bit to check (0-31): ");
        n = (new Scanner(System.in)).nextInt();
        newNum = num ^ (1 << n);
        System.out.println("Bit toggled successfully.");
        System.out.printf("Number before toggling %1$s bi
            t:%2$s" + "\n", n, num);
        System.out.printf("Number after toggling :%1$s decimal"
            + "\n", newNum);
    }
}
```

در این برنامه num (عدد ورودی)، n (شماره بیتی از عدد num است که باید معکوس گردد) و newNum (عدد جدید) را داریم.

این برنامه ابتدا عدد یک (۱) را n بیت به سمت چپ شیفت می‌دهد و با num یا بیتی می‌نماید.

```
Enter a number: 100
Enter nth bit to check (0-31): 11
Bit toggled successfully.
Number before toggling 11 bit:100
Number after toggling :2148 decimal
```

۵۹. 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

```
package ch1_59;
import java.util.*;
public class Ch1_59 {
    public static void main(String[] args) {
        int num1, num2;
        /* Input two numbers from user */
        /* Swap two numbers */
        System.out.print("Enter number1: ");
        num1 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter number2: ");
        num2 = (new Scanner(System.in)).nextInt();
        num1 ^= num2;
        num2 ^= num1;
        num1 ^= num2;
        System.out.printf("Num1 = %1$s" + "\n", num1);
        System.out.printf("Num2 = %1$s" + "\n", num2);
    }
}
```

در این برنامه num₁ (عدد اول) و num₂ (عدد دوم) هستند.

```
Enter number1: 101
Enter number2: 10
Num1 = 10
Num2 = 101
```

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

$$\text{مساحت دوزنقه} = \frac{(\text{قاعده بزرگ} + \text{قاعده کوچک}) \times \text{ارتفاع}}{2}$$

در این برنامه a، b، c به ترتیب قاعده کوچک، قاعده بزرگ و ارتفاع دوزنقه هستند.

```
package ch1_60;
import java.util.*;
public class Ch1_60 {
    public static void main(String[] args) {
        System.out.print("Enter a: ");
        int a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b: ");
        int b = (new Scanner(System.in)).nextInt();
        System.out.print("Enter c: ");
        int c = (new Scanner(System.in)).nextInt();
        System.out.printf("Area is %1$s", (float)1/2 * (a+ b) * c);
    }
}
```

```
Enter a: 10
Enter b: 20
Enter c: 15
Area is 225.0
```

۶۱. 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.

```
package ch1_61;
import java.util.*;
public class Ch1_61 {
    public static void main(String[] args) {
        int num, flippedNumber;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        flippedNumber = ~num;
        System.out.printf("Original number = %1$s (in decimal)" + "\n", num);
    }
}
```

۴۷ آشنایی با جاوا (دستورات ورودی و خروجی)

```
System.out.printf("Number after bits are flipped=%1$s(in
decimal)+"\n", flippedNumber);
}
```

در این برنامه num (عدد ورودی) و flippedNumber (عددی است که باید بیت‌های آن معکوس گردد) را داریم.

```
Enter a number: 100
Original number = 100 (in decimal)
Number after bits are flipped = -101 (in decimal)
```

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

```
package ch1_62;
import java.util.*;
public class Ch1_62 {
    public static void main(String[] args) {
        int num, n, newNum;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        /* Input bit position you want to check */
        System.out.print("Enter nth bit to check (0-31): ");
        n = (new Scanner(System.in)).nextInt();
        newNum = num ^ (1 << n);
        System.out.println("Bit toggled successfully.");
        System.out.printf("Number before toggling %1$s bit:%2$s"
            + "\n", n, num);
        System.out.printf("Number after toggling :%1$s decimal" +
            "\n", newNum);
    }
}
```

این برنامه num (عدد ورودی)، n (شماره بیتی از عدد num است که باید معکوس گردد) و newNum (عدد جدید) را داریم. این برنامه ابتدا عدد یک (۱) را n بیت به سمت چپ شیف می‌دهد و با num یا بیتی می‌نماید.

```
Enter a number: 124
Enter nth bit to check (0-31): 5
Bit toggled successfully.
Number before toggling 5 bit:124
Number after toggling :92 decimal
```

۶۳. 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
<hr/>		
x 01011011	a 00011010	b 01000001

```
package ch1_63;
import java.util.*;
public class Ch1_63 {
    public static void main(String[] args) {
        int num1, num2;
        /* Input two numbers from user */
        /* Swap two numbers */
        System.out.print("Enter number1: ");
        num1 = (new Scanner(System.in)).nextInt();
        System.out.print("Enter number2: ");
        num2 = (new Scanner(System.in)).nextInt();
```

```

        num1 ^= num2;
        num2 ^= num1;
        num1 ^= num2;
        System.out.printf("Num1 = %1$s" + "\n", num1);
        System.out.printf("Num2 = %1$s" + "\n", num2);
    }
}
Enter number1: 10
Enter number2: 20
Num1 = 20
Num2 = 10

```

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

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

در این برنامه a، b، c به ترتیب قاعده کوچک، قاعده بزرگ و ارتفاع دوزنقه هستند.

```

package ch1_64;
import java.util.*;
public class Ch1_64 {
    public static void main(String[] args) {
        System.out.print("Enter a: ");
        int a = (new Scanner(System.in)).nextInt();
        System.out.print("Enter b: ");
        int b = (new Scanner(System.in)).nextInt();
        System.out.print("Enter c: ");
        int c = (new Scanner(System.in)).nextInt();
        System.out.printf("Area is %1$s", (float)1 / 2 * (a + b) * c);
    }
}
Enter a: 10
Enter b: 20
Enter c: 15
Area is 225.0

```

۴۵. 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.

```

package ch1_65;
import java.util.*;
public class Ch1_65 {
    public static void main(String[] args) {
        int num, flippedNumber;
        System.out.print("Enter a number: ");
        num = (new Scanner(System.in)).nextInt();
        flippedNumber = ~num;
        System.out.printf("Original number = %1$s(in decimal)+" + "\n", num);
        System.out.printf("Number after bits are flipped=%1$s(in decimal)"+ "\n", flippedNumber);
    }
}

```

در این برنامه num (عدد ورودی) و flippedNumber (عددی است که باید بیت‌های آن معکوس گردد) را داریم.

```

Enter a number: 112
Original number = 112 (in decimal)
Number after bits are flipped = -113 (in decimal)

```


ساختار تصمیم و حلقه تکرار

۲-۱. مسائل تحلیلی

۱. چه زمانی از switch به جای if-else استفاده می‌شود؟

قانون کلی در این مورد وجود ندارد. استفاده از هر یک از آن‌ها به شرایط و نیاز برنامه وابسته است. اما باید به خاطر داشته باشید که با if-else می‌توان شرایط مختلف و بازه‌های از مقادیر را ارزیابی کرد، حال آن‌که switch بر روی یک مقدار عددی صحیح، یک داده شمارشی یا یک شیء رشته‌ای و عملگر == کار می‌کند. البته switch خوانایی برنامه را افزایش می‌دهد.

۲. انواع حلقه‌های مختلف در جاوا را نام ببرید.

- While loop
- do ... while loop
- for loop
- for-each loop

۳. تفاوت اصلی بین while loop و do ... while چیست؟

در while ... do به دلیل این که شرط در انتهای حلقه بررسی می‌شود، حتی با وجود نادرست بودن شرط، حلقه، حداقل یک‌بار اجرا می‌شود. ولی، در while...loop دستورات بدنه حلقه ممکن است اصلاً اجرا نشوند.

۴. خروجی دستورات زیر چیست؟

```
package fundamentals;
public class FundamentalEx20 {
    public static void main(String args[])
    {
        System.out.println("****break vs continue****");
        System.out.println("***Example : break***");
        for(int i = 0; i < 5; i++)
        {
            System.out.print("At entry i is : " + i);
            if( i == 3)
                break;
            System.out.print("\t At Exit i is : " + i);
            System.out.println();
        }
        System.out.println();
        System.out.print("***Example : continue***\n");
        for(int i = 0; i < 5; i++)
        {
            System.out.print("At entry i is : " + i);
            if( i == 3)
                continue;
            System.out.print("\t At Exit i is : " + i);
            System.out.println();
        }
    }
}

***break vs continue***
```