

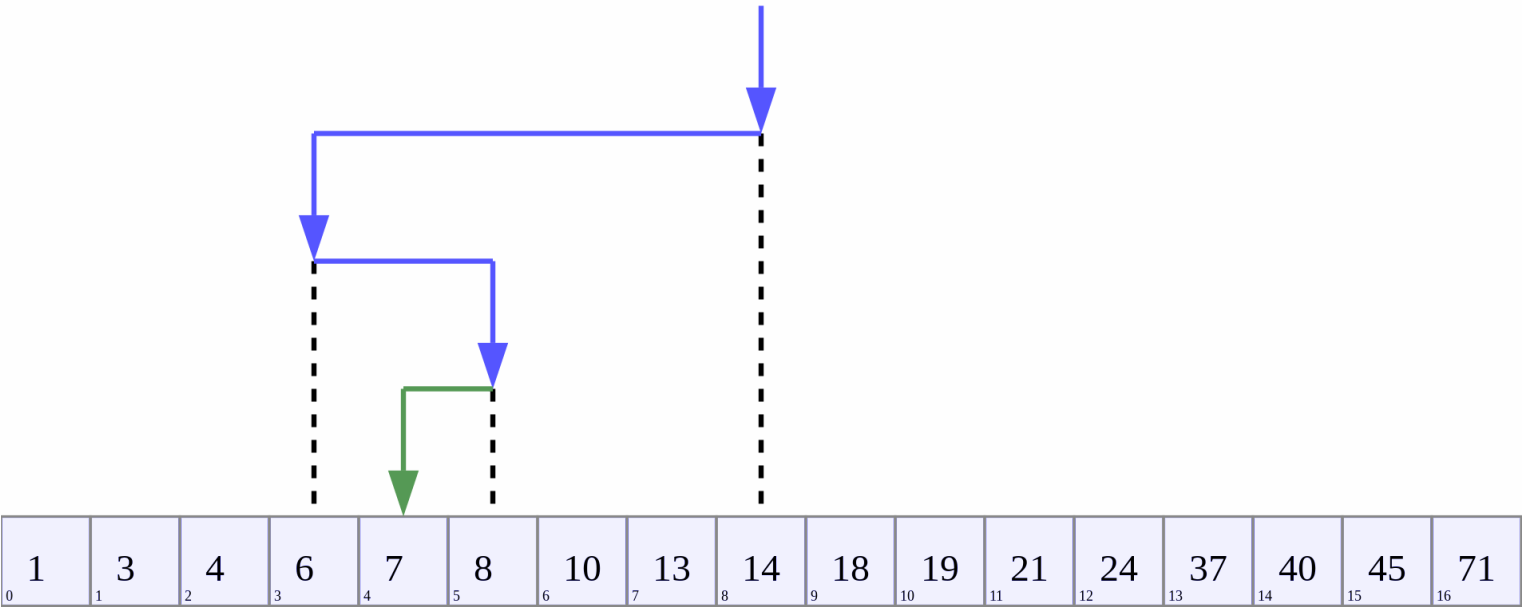
# جستجوی دودویی

برای اولین تمرین باید الگوریتم جستجوی دودویی را در جاوا پیاده‌سازی کنید.

\*محدودیت حافظه: ۵۰ مگابایت\*

\*محدودیت زمان: ۱.۷ ثانیه\*

## الگوریتم جستجوی دودویی



خلاصه‌ی روند کار الگوریتم به این صورت است که اعداد در یک آرایه به صورت مرتب‌شده قرار دارند و ما برای اینکه بفهمیم آیا عددی که از ما پرسیده شده در آرایه قرار دارد یا نه، ابتدا بازه‌ی جستجو را کل آرایه در نظر می‌گیریم، عدد پرسیده شده را با عنصر وسط مقایسه می‌کنیم و اگر مساوی آن بود اندیس عنصر وسط را برمی‌گردانیم، اگر بزرگتر از آن بود بازه‌ی جستجو را به نیمه‌ی سمت راست آرایه و اگر کوچکتر از آن بود بازه‌ی جستجو را به نیمه‌ی سمت چپ آرایه محدود می‌کنیم و همین مراحل را برای آنها تکرار می‌کنیم. اگر در نهایت بازه‌ی ما شامل تنها یک عدد بود و عدد باقیمانده برابر با عدد پرسیده شده نبود، مثلاً 1- برگردانده می‌شود که نشان دهنده‌ی عدم وجود این عدد در آرایه است.

## ورودی

در ورودی ابتدا دو عدد به شما داده می‌شود که اولی  $m$  یا تعداد عناصر آرایه، و دومی  $n$  یا تعداد اعدادی است که قرار است از شما پرسیده شود. سپس در خط دوم  $m$  عدد صحیح که عناصر آرایه به صورت مرتب شده هستند با فاصله از هم نوشته شده است. در  $n$  خط بعدی در هر خط یک عدد نوشته شده که این‌ها اعداد پرسیده شده هستند.

$$n \leq 30000$$

$$m \leq 30000$$

## خروجی

در خروجی شما باید  $n$  عدد را در خطوط مختلف چاپ کنید که هرکدام اندیس عدد پرسیده شده در آرایه است، در صورتی که عدد پرسیده شده در آرایه موجود نبود، عبارت not found چاپ می‌شود.

## ورودی نمونه

```
9 3
-5 -3 0 2 3 12 54 81 511
12
-1
0
```

## خروجی نمونه

5  
not found  
2

## ضرب ماتریس‌ها

تعدادی اطلاعات ماتریس از یک فایل با نام `input.txt` میخواند که عددهای اول و دوم به ترتیب تعداد سطر و ستون ماتریس اول هستند و عددهای دوم و سوم به ترتیب تعداد سطر و ستون ماتریس دوم هستند؛ سپس مقدار هر درایه ماتریس را گرفته و ضرب دو ماتریس را به فایل اضافه میکند.

## ورودی

در خط اول ورودی ۳ عدد آمده که نشانگر تعداد سطر و ستون های ۲ ماتریس است.

در ادامه ورودی درایه‌های ماتریس اول و سپس درایه‌های ماتریس دوم آمده است.

سپس در ادامه اطلاعات ماتریس بعدی با شرایط بالا در فایل موجود میباشد

تمامی اعداد داده شده در ورودی کوچکتر از ۱۰۰ می‌باشند.

## خروجی

در خروجی درایه‌های ماتریس حاصل از ضرب ۲ ماتریس داده شده در فایل اضافه میکند.

## مثال

### ورودی نمونه

محتویات فایل `input.txt` قبل از اجرای برنامه:

```
2 3 2
1 2 3
4 5 6
1 2
3 4
5 6
```

### خروجی نمونه

محتویات فایل `input.txt` بعد از اجرای برنامه:

```
2 3 2
1 2 3
4 5 6
1 2
3 4
5 6
22 28
49 64
```

# تشخیص داده مخدوش

اطلاعات ارسالی از طرف یک سیستم مخابراتی بر روی یک فایل ذخیره می‌شوند. اما بعضی از سیگنال‌های مخابرات به درستی دریافت نمی‌شوند در نتیجه داده به صورت مخدوش شده ذخیره می‌شود. به عنوان مثال فرض کنید که داده‌های زیر از طرف فرستنده دریافت شده است:

```
1 0 1 0 1 1
1 1 0 1 1 1
0 1 1 1 1 0
0 0 1 0
0 1 0 0 0 0
```

با توجه به بسته دریافت شده متوجه می‌شویم که داده یکی مانده به آخر با اختلال مواجه شده است و درنتیجه فاقد اعتبار است.

برای ما مهم است که داده‌های مخدوش شده را به صورت جداگانه بررسی کنیم تا برنامه با مشکل ایجاد نشود. در این تمرین قصد داریم تا داده‌های مخدوش شده را تشخیص دهیم و گزارش کنیم.

## ورودی

ورودی برنامه یک فایل است که اطلاعات دریافتی از فرستنده در آن ذخیره شده است.

## خروجی

خروجی برنامه شامل اندیس تمام داده‌هایی است که مخدوش شده هستند.

### ورودی نمونه ۱

```
1 0 1 0 1 1
1 1 0 1 1 1
0 1 1 1 1 0
0 0 1 0
0 1 0 0 0 0
```

### خروجی برنامه ۱

```
3
```

### ورودی نمونه ۲

```
1 1 0 0 0 0 0 0 0 1
1 1 1 1 1
1 1 0 0 0 1 0 0 0 0
1 0 1 0 0 1 0 1 0 1
1 1 0 0 1 0 1 0 0 0
1 0 1 0 0 1 0 1 0 1 1
1 1 0 0 1 1 1 0 0 0
1 0 1 0 0 0 0 1 0 1
```

### خروجی برنامه ۲

```
1
5
```

## نکات

- تضمین می‌شود که تعداد داده‌های مخدوش شده کمتر از ۱۰ درصد کل داده‌ها است.
- ورودی به صورت یک فایل است اما خروجی بر روی خروجی استاندارد چاپ می‌شود.

## تشخیص خطا

این یک سوال تشریحی است که در آن قصد داریم خطای کدها را مشخص کنیم. در هر یک از موارد زیر خطای کد را مشخص کنید و تصحیح شده کد را ارائه دهید.

### خطای اول

```
1 public class Errors {
2     public static void main(String[] args) {
3         private int a = 3;
4         while (a++ < 10);
5         System.out.println(a);
6     }
7 }
```

### خطای دوم

```
1 public class Errors {
2     public static void main(String[] args) {
3         for (i = 0; ; ) {
4             if (i++ == 10) break;
5         }
6     }
7 }
```

### خطای سوم

```
1 public class Errors {
2     public static void main(String[] args) {
3         String name = new String("Sina");
4         System.out.println(name.charAt(0).toLowerCase());
5     }
6 }
```

### خطای چهارم

```
1 public class Errors {
2     public static void main(String[] args) {
3         int c = 3;
4         if (c = 3) System.out.println("c is three");
5     }
6 }
```

# مدیریت پنجره نمایش

به صورت تشریحی به سوالات زیر پاسخ دهید:

کد زیر را در نظر بگیرید و به سوالات زیر پاسخ دهید:

```
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class JavaSwing {
5
6     public static void main(String[] args) {
7         JFrame jFrame = new JFrame();
8         jFrame.setVisible(true);
9         jFrame.setSize(200, 100);
10        jFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
11
12        FlowLayout layout = new FlowLayout();
13        jFrame.setLayout(layout);
14
15        JButton buttonOne = new JButton("Button one");
16        JButton buttonTwo = new JButton("Button two");
17        JButton buttonThree = new JButton("Button three");
18        JButton buttonFour = new JButton("Button four");
19        JButton buttonFive = new JButton("Button five");
20
21        jFrame.add(buttonOne);
22        jFrame.add(buttonTwo);
23        jFrame.add(buttonThree);
24        jFrame.add(buttonFour);
25        jFrame.add(buttonFive);
26    }
27 }
```

## سوال اول

ابتدا توضیح دهید که به چه منظور از layout استفاده می‌کنیم؟

## سوال دوم

شمایی از خروجی برنامه فوق نشان دهید. می‌توانید آن را روی کاغذ نشان دهید.

## سوال سوم

آیا می‌توانستیم بدون نامگذاری شیئی ساخته شده از کلاس FlowLayout برای jFrame همان layout را قرار دهیم؟ چگونه؟

حال کد مشابه زیر را در نظر بگیرید:

```
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class JavaSwingTwo {
5
6     public static void main(String[] args) {
7         JFrame jFrame = new JFrame();
8         jFrame.setVisible(true);
9         jFrame.setSize(1000, 600);
10        jFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
11
12        BorderLayout layout = new BorderLayout();
13        jFrame.setLayout(layout);
14
15        JButton buttonOne = new JButton("Button one");
16        JButton buttonTwo = new JButton("Button two");
17        JButton buttonThree = new JButton("Button three");
18        JButton buttonFour = new JButton("Button four");
19        JButton buttonFive = new JButton("Button five");
20
21    }
```

```
21  
22 JFrame.add(buttonOne, BorderLayout.CENTER);  
23 JFrame.add(buttonTwo, BorderLayout.SOUTH);  
24 JFrame.add(buttonThree, BorderLayout.NORTH);  
25 JFrame.add(buttonFour, BorderLayout.EAST);  
26 JFrame.add(buttonFive, BorderLayout.WEST);  
27 }
```

## سوال چهارم

شمایی از خروجی برنامه فوق نشان دهید.

## سوال پنجم

آیا می‌توان بیشتر از پنج Component به JFrame اضافه کرد به طوریکه layout آن از نوع BorderLayout باشد؟

## سوال ششم

به نظر شما چه زمانی به‌ترست از BorderLayout استفاده کنیم و چه زمانی بهتر است از Flow Layout استفاده کنیم؟

## سوال هفتم

نام یک نوع layout دیگر را بنویسید و شمایی از آن ارائه دهید.

## توضیحات

لطفا پاسخ خود را به صورت خوانا نوشته و به صورت pdf درآورید سپس آنرا آپلود کنید. خوش خطی و خوانایی مطلب شما در تصحیح تاثیر مثبت خواهد داشت.

## تابع مستقل از کلاس

به سوالات زیر به صورت تشریحی پاسخ دهید:

تابع **static** زیر را در نظر بگیرید:

```
1 public static void test() {  
2     System.out.println("My name is Ali");  
3 }
```

### سوال اول

ابتدا تفاوت تابع **static** و غیر **static** را توضیح دهید.

### سوال دوم

تابع فوق چه نوع خروجی دارد؟

حال کد زیر را در نظر بگیرید:

```
1 public class staticMethod {  
2  
3     int a;  
4     public static void test() {  
5         System.out.println(a);  
6     }  
7 }
```

### سوال سوم

مشکل کد فوق را توضیح دهید.

تابع **test2** را هم به کد اضافه می‌کنیم:

```
1 public class staticMethod {  
2  
3     public static void test() {  
4     }  
5     public void test2() {  
6     }  
7 }
```

### سوال چهارم

آیا می‌توانیم از تابع **test** در تابع **test2** استفاده کنیم؟ برعکس چطور؟ چرا؟