# رشته چکر

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

---

در ورودی دو رشته داده می‌شود و از شما خواسته شده است تا بررسی کنید این دو رشته در کنار هم زیبا هستند یا نه.

دو رشته در کنار هم زیبا هستند، اگر حرف اول رشته‌ی اولی، با حرف آخر رشته‌ی دومی برابر باشد.

## ورودی

ورودی شامل دو خط است که در هر خط یک رشته شامل حروف کوچک انگلیسی به طول حداکثر ۵۰ آمده است.

## خروجی

در صورتی که دو رشته‌ی داده در کنار هم زیبا هستند عبارت `YES` و در غیر این صورت عبارت `NO` را چاپ کنید.

### ورودی نمونه ۱

```
salam
khodafes
```

### خروجی نمونه ۱

```
YES
```

از آنجا که حرف اول رشته‌ی `salam` برابر با حرف آخر رشته‌ی `khodafes` ، یعنی `s` است، پس این دو رشته در کنار هم زیبا هستند و باید عبارت `YES` را چاپ کرد.

### ورودی نمونه ۲

```
salam
salam
```

### خروجی نمونه ۲

```
NO
```

از آنجا که حرف اول رشته‌ی `salam` ، یعنی `s` برابر با حرف آخر رشته‌ی `salam` ، یعنی `m` نیست، پس این دو رشته در کنار هم زیبا نیستند و باید عبارت `NO` را چاپ کرد.

### ورودی نمونه ۳

```
snapp
box
```

### خروجی نمونه ۳

```
NO
```

از آنجا که حرف اول رشته‌ی `snapp` ، یعنی `s` برابر با حرف آخر رشته‌ی `box` ، یعنی `x` نیست، پس این دو رشته در کنار هم زیبا نیستند و باید عبارت `NO` را چاپ کرد.

### ورودی نمونه ۴

```
software
engineers
```

### خروجی نمونه ۴

```
YES
```

از آنجا که حرف اول رشته‌ی `software` برابر با حرف آخر رشته‌ی `engineers` ، یعنی `s` است، پس این دو رشته در کنار هم زیبا هستند و باید عبارت `YES` را چاپ کرد.

# پکمن

در این تمرین می‌خواهیم یک انیمیشن بسیار ساده طراحی کنیم به این صورت که پکمن در طول صفحه حرکت می‌کند. حرکت پکمن به این صورت است که از ابتدای محور افقی شروع می‌کند و تا انتهای آن می‌رود سپس همین حرکت را به صورت برعکس انجام می‌دهد. مختصات پکمن در صفحه عمودی باید در وسط صفحه باشد. توجه کنید هنگامی که پکمن از چپ به راست می‌رود دهن آن در سمت راست قرار دارد و وقتی از سمت راست به سمت چپ حرکت می‌کند دهن آن در سمت چپ قرار دارد.

می‌توانید قسمتی از ویدیو بازی را در اینجا تماشا کنید.

## کلاس DrawingPanel

برای ساخت بازی باید از کلاس آماده DrawingPanel استفاده کنید.

## کلاس Main

کد شما باید داخل کلاس Main قرار گیرد.

▼ راهنمایی

ساختار کلی برنامه شما می‌تواند به صورت زیر باشد:

```
// pre-requirements
while (true) {
    for (int i = 0; i < 950; i++) {
    // move to right
    }

    for (int i = 900; i > 0; i--) {
    // move to left
    }
}
```

در کد فوق فرض شده است که اندازه پکمن ۵۰ واحد است.

## نکات

- نیازی به بارگذاری کلاس DrawingPanel **نیست**.
- توجه کنید که در فایل خود کلاسی را import نکنید.
- اگر به بیش از یک کلاس احتیاج دارید آنها را در همان فایل Main.java قرار دهید اما توجه کنید که این کلاس‌ها نباید public باشند.

# نقطه درون مثلث

در این مسئله می‌خواهیم مختصات یک مثلث و یک نقطه در صفحه را از کاربر بگیریم و مشخص کنیم که آیا نقطه درون مثلث قرار دارد یا خیر.

برای مشخص کردن یک مثلث در صفحه کافی است مختصات سه راس آن را داشته باشیم. البته این سه نقطه می‌توانند به صورت ساعتگرد و یا پادساعتگرد از کاربر گرفته شوند که در اینجا ما نقاط را به صورت ساعتگرد از کاربر دریافت می‌کنیم:

برای ذخیره‌سازی مختصات یک نقطه از کلاس OrderedPair استفاده کنید.

بعد از اینکه مختصات سه راس مثلث را از کاربر دریافت کردیم باید با استفاده از آن مثلث را در صفحه بکشیم برای این کار از تابع زیر استفاده کنید:

```
Graphics.drawLine(int x1, int y1, int x2, int y2);
```

حال نوبت به آن می‌رسد که بررسی کنیم تا نقطه درون این مثلث قرار داشته باشد. می‌دانیم که یک نقطه درون یک چندضلعی محدب قرار دارد اگر این نقطه سمت راست همه اضلاع قرار گرفته باشد. با توجه به اینکه هر مثلثی محدب است تنها کافی است که همین نکته را بررسی کنیم. توجه کنید برای اینکه مشخص کنیم یک نقطه سمت راست یک ضلع قرار دارد یا نه ابتدا باید به پاره خط جهت بدهیم:
توجه کنید این جهت‌دهی باید در جهت ساعتگرد باشد. اگر شست خود را در جهت پاره خط بگیریم سایر انگشتان در جهت راست قرار می‌گیرند. بعد از این کار باید معادله خط گذرنده از پاره‌خط را پیدا کنیم. برای این کار کافی است شیب پاره‌خط را محاسبه و سپس با استفاده از آن و یک نقطه روی پاره خط عرض از مبدا خط را پیدا کنیم:

حال اگر معادله خط را بنویسیم به صورت زیر خواهد بود:

$$y = ax - b$$

اگر از روی معادله این خط یک تابع دو بعدی بسازیم بصورت زیر خواهد شد:

$$P(x, y) = y - ax + b$$

حال اگر یک نقطه مانند$(x_0, y_0)$ را در این تابع بگذاریم و مقدار آن منفی شود این نقطه سمت راست خط قرار دارد. به عنوان مثال خط زیر را در نظر بگیرید:

$$-0.5x = y$$

نقطه $(110, -65)$ را در تابع دو بعدی متناظر قرار می‌دهیم و نتیجه $-10$ خواهد شد و در نتیجه این نقطه سمت راست این خط قرار دارد.

تابع زیر را برای بررسی این که یک نقطه سمت راست یک خط قرار دارد در نظر می‌گیریم:

```
public static boolean isPointOnRight(OrderedPair p1, OrderedPair p2, OrderedPair point) {
    // calculate a and b

    if (a == Double.POSITIVE_INFINITY) {
        // special case
    } else if (a == Double.NEGATIVE_INFINITY) {
        // special case
    } else if (a == 0) {
        // special case
    } else {
        // calculate P(point)
        // check if P(point) <= 0.0000000001
    }
}
```

در این تابع ابتدا نقطه ابتدایی و انتهایی پاره‌خط و سپس نقطه مورد نظر را می‌گیریم. با استفاده از اطلاعات فوق $a$ و $b$ را محاسبه می‌کنیم و سپس سمت راست بودن را در حالات مختلف تشخیص می‌دهیم. هنگامی که $a$ را محاسبه می‌کنیم باید توجه کنید که ابتدا مقادیر را به double تغییر دهید (cast کنید). هنگام محاسبه $a$ با توجه به تقسیمی که انجام می‌شود ممکن است خطای گرد کردن اتفاق افتد که باید این خطا را با ۱۰ رقم اعشار کنترل کنیم.

## ورودی

ورودی شامل چهار خط است که در هر خط آن دو عدد طبیعی $n$ و $m$ با فاصله از هم آمده است.

$$1 \leq n \leq 400$$

$$-400 \leq m \leq -1$$

## خروجی

خروجی برنامه‌ی شما یک پنجره گرافیکی است که در آن سه نقطه اول به صورت یک مثلث رسم شده است و نقطه چهارم به صورت یک دایره کوچک، اگر این دایره داخل مثلث باشد رنگ آن سبز خواهد بود و اگر بیرون مثلث باشد رنگ آن به صورت قرمز خواهد بود.

## مثال

### ورودی نمونه ۱

```
100 -100
100 -50
200 -100
110 -65
```

### خروجی نمونه ۱

### ورودی نمونه ۲

```
100 -100
100 -50
200 -100
110 -150
```

### خروجی نمونه ۲

### ورودی نمونه ۳

```
150 -150
180 -50
210 -150
100 -100
```

### خروجی نمونه ۳

### ورودی نمونه ۴

```
150 -150
180 -50
210 -150
165 -100
```

### خروجی نمونه ۴

به خطای گرد کردن در این مثال توجه کنید!

## نکات

- برای پیاده سازی گرافیک از کلاس DrawingPanel استفاده کنید.
- نیازی به قرار دادن کلاس DrawingPanel در کد خود ندارید.
- کد خود را در یک فایل Main.java قرار دهید و اگر می‌خواهید از کلاس اضافه استفاده کنید آن‌ها را در همین فایل قرار داده و کلمه کلیدی public را از ابتدای آن‌ها حذف کنید.

# ماشین حساب

در این تمرین، یک ماشین حساب ساده با *Java Swing* پیاده‌سازی می‌کنید.


ماشین حساب ساده شامل اپراتورهای **جمع**، **تفریق**، **ضرب** و **تقسیم** است و از اعداد *صحیح* و *اعشاری* مثبت و منفی پشتیبانی می‌کند. اعداد منفی با درج یک علامت `-` پشت عدد مشخص می‌شوند.

- ترتیب چینش دکمه های ماشین حساب حائز اهمیت نمی‌باشد.
- دکمه‌ی `clear` برای پاک کردن نوار بالا ماشین حساب مورد استفاده قرار می‌گیرد.
- کاربر نباید بتواند *تقسیم بر صفر* انجام دهد، در واقع اگر کاربر مخرج تقسیم را ۰ وارد کرد، باید پیام **not valid** چاپ شود.

## توضیحات

نحوه‌ی کار ماشین حساب به این صورت می‌باشد:

ابتدا کاربر عددی را وارد می‌کند سپس یکی از چهار عملگر (جمع، تفریق، ضرب یا تقسیم) را انتخاب میکند که با این کار عدد روی صفحه نمایش از روی صفحه پاک می‌شود و به حافظه انتقال پیدا می‌کند. سپس کاربر عدد بعدی را وارد میکند و با فشار دادن تساوی حاصل عبارت نمایش داده می‌شود و عدد ذخیره شده در حافظه پاک می‌شود. توجه کنید در این حالت کاربر باز می‌تواند یک عملگر دیگر انتخاب کنید و محاسبات را مانند قبل ادامه دهد.

در واقع برای پیاده سازی این ماشین‌حساب نیاز به سه متغیر داریم:

- متغیر مربوط به عدد ذخیره شده در حافظه
- متغیر مربوط به عملگر انتخاب شده
- متغیر مربوط به دومین عدد وارد شده

توجه کنید هر زمان که کاربر دکمه‌ی CLEAR را فشار دهد صفحه نمایش و حافظه به طور کلی پاک می‌شوند.

## توابع عملگری

ماشین حساب باید شامل توابع جمع، تفریق، ضرب و تقسیم به صورت زیر باشد که ورودی آن دو عدد اعشاری میباشد.

```java
public static double add(double numOne, double numTwo) {
    // sum two number
}

public static double subtract(double numOne, double numTwo) {
    // minus two number
}

public static double divide(double numOne, double numTwo) {
    // divide two number
}

public static double multiply(double numOne, double numTwo) {
    // multiply two number
}
```

## تذکر

- برنامه خود را در یک فایل با نام Main.java بارگزاری کنید.
- برای این سوال لازم به ارائه‌ی شفاهی می‌باشد.
- به یاد داشته باشید فقط مجاز به استفاده از مفاهیمی می‌باشید که در کارگاه و کلاس درس بیان شده باشد.