

## روز برنامه‌نویس

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

روز برنامه‌نویس در ۲۵۶مین روز سال جشن گرفته می‌شود.

۲۵۶ بزرگ‌ترین توانی از ۲ است که از طول سال کمتر است. تابعی بنویسید که با دریافت طول سال در یک سیاره، روز برنامه‌نویس آن سیاره را محاسبه کند. عده‌ای خرابکار قصد سو استفاده از برنامه‌ی شما را دارند... پس تابع خود را به گونه‌ای امن کنید که فقط دو بار قابل استفاده باشد و بعد از بار دوم بدون توجه به ورودی، مقدار منفی یک را برگرداند.

راهنمایی: جهت امن کردن تابع از متغیر استاتیک استفاده کنید. (حق استفاده از متغیرهای گلوبال را ندارید)

## ورودی

در یک خط، تعدادی عدد طبیعی وارد می‌شود که هرکدام نشان‌دهنده‌ی طول سال در یک سیاره است.

$$1 \leq a_i \leq 1000000000$$

آخرین عددی که وارد می‌شود صفر است.

## خروجی

در یک خط از خروجی استاندارد، خروجی‌های تابع را با کاراکتر فاصله جدا کرده و چاپ کنید.

## ورودی نمونه ۱

600 128 42 0

## خروجی نمونه ۱

512 128 -1

ورودی نمونه ۲

200 42 127 132 0

خروجی نمونه ۲

128 32 -1 -1

ورودی نمونه ۳

0

خروجی نمونه ۳

## مشکل تارا

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

تارا به کشور جدیدی برای تحصیل سفر کرده است و تصمیم دارد در کنار درس در شرکت پست *Golden* کار کند. نواحی تحت نظر این شرکت ویژگی‌های زیر را دارند:

- همگی روی یک خط مستقیم واقع و به ترتیب از شماره‌ی 1 تا  $n$  کدگذاری شده‌اند.
- از هر ناحیه به ناحیه‌ی بعد تنها یک مسیر مستقیم وجود دارد که پیمایش آن به یک لیتر سوخت نیاز دارد.
- در هر ناحیه یک پمپ بنزین قرار دارد و قیمت بنزین در شهر  $k$ ام،  $k$  دلار است.

در ناحیه‌ی اول مخزن سوخت موتور تارا خالی است و قرار است به ناحیه‌ی  $n$ ام برود و هر بار که سه ناحیه‌ی متوالی را طی می‌کند یک بطری نوشابه به قیمت یک دلار می‌خرد. برنامه‌ای بنویسید که کمترین هزینه‌ی خرج شده توسط تارا را محاسبه کند.

## ورودی

در یک خط، دو عدد  $n$  و  $v$  نوشته می‌شوند که به ترتیب تعداد شهرها و گنجایش مخزن سوخت را نشان می‌دهند.

$$2 \leq n, v \leq 100$$

## خروجی

در یک خط از خروجی استاندارد، حداقل پولی که تارا باید خرج کند را چاپ کنید.

## مثال

ورودی نمونه ۱

4 2

خروجی نمونه ۱

5

ورودی نمونه ۲

6 7

خروجی نمونه ۲

7

## مثلث‌های تمیز

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

توابع زیر را پیاده‌سازی کرده و از آن‌ها در برنامه‌ای جهت تشخیص نوع یک مثلث استفاده کنید.

- تابعی که قائم الزاویه بودن یک مثلث را مشخص می‌کند.

```
1 | bool is_right_tri(int x1, int y1, int x2, int y2, int x3, int y3);
```

- تابعی که متساوی الساقین بودن یک مثلث را مشخص می‌کند.

```
1 | bool is_Isosceles_tri(int x1, int y1, int x2, int y2, int x3, int y3);
```

- تابعی که فاصله‌ی اقلیدسی دو نقطه یا توان دوم آن را برمی‌گرداند.

```
1 | double oq_distance_2(int x1, int y1, int x2, int y2);
```

گرچه نیازی ندارید، اما می‌توانید از کتابخانه‌های ریاضی زبان C یا C++ استفاده کنید.

## ورودی

در یک خط، شش عدد صحیح  $x_1, y_1, x_2, y_2, x_3, y_3$  و  $y_3$  نوشته می‌شوند که هر جفت، مختصات یکی از رئوس مثلث را نشان می‌دهد.

$$-20 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 20$$

تضمین می‌شود که نقاط ورودی روی یک خط واقع نشده‌اند.

## خروجی

با توجه به مثال‌ها، در یک خط از خروجی استاندارد، دو بیت چاپ کنید که بیت اول برای قائم الزاویه بودن و بیت دوم برای متساوی الساقین بودن است.

## مثال

### ورودی نمونه ۱

2 2 0 0 2 0

### خروجی نمونه ۱

1 1

مثلت هم قائم الزاویه و هم متساوی الساقین است.

### ورودی نمونه ۲

2 0 3 3 4 0

### خروجی نمونه ۲

0 1

مثلت فقط متساوی الساقین است.

### ورودی نمونه ۳

15 0 0 0 0 4

## خروجی نمونه ۳

1 0

مثلت فقط قائم الزاویه است.

## عقربه‌های بهشتی

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

مدتهاست مهاجرت از ساعت‌های عقربه‌ای به ساعت‌های دیجیتال شروع شده است.... اما در این سوال باید تابعی را پیاده‌سازی کنید که با دریافت مقادیر ساعت و دقیقه و با فرض صفر بودن ثانیه، زاویه‌ی حاده‌ی بین عقربه‌های ساعت‌شمار و دقیقه‌شمار را (در یک ساعت عقربه‌ای!!!) برحسب درجه محاسبه کند.

کاربری که از برنامه استفاده می‌کند ممکن است دقیقه‌ها را اشتباه وارد کند. مثلاً به جای ساعت دوازده بگوید ده و صد و بیست دقیقه. بنابراین در برنامه‌ی خود این موضوع را در نظر داشته باشید.

## ورودی

در یک خط، دو عدد  $h$  و  $m$  نوشته می‌شوند که به ترتیب ساعت و دقیقه را نشان می‌دهند.

$$0 \leq h \leq 12$$

$$0 \leq m \leq 500$$

## خروجی

در یک خط از خروجی استاندارد، زاویه‌ی بین عقربه‌های ساعت‌شمار و دقیقه‌شمار را با دقت دو رقم اعشار چاپ کنید.

## مثال

## ورودی نمونه ۱



9 0

خروجی نمونه ۱

90.00

ورودی نمونه ۲

10 120

خروجی نمونه ۲

0.00

ورودی نمونه ۳

11 95

خروجی نمونه ۳

167.50

## الگو یابی

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

تابعی بنویسید که با استفاده از عملگرهای بیتی و بدون تبدیل مبنا، عددی طبیعی را دریافت کرده و تعداد الگوهای 101 موجود در ساختار باینری آن را پیدا کند. دقت کنید حالت‌های همپوشانی باید در نظر گرفته شوند. مثلاً در هر دو عبارت 10101 و 101101، دو بار الگوی 101 تکرار شده است.

## ورودی

در یک خط، عدد طبیعی  $n$  نوشته می‌شود.

$$1 \leq n \leq 1000$$

## خروجی

در یک خط از خروجی استاندارد، تعداد الگوهای موردنظر را در ساختار باینری عدد چاپ کنید.

## مثال

### ورودی نمونه ۱

5

### خروجی نمونه ۱

1

ورودی نمونه ۲

21

خروجی نمونه ۲

2

## تقریب مرتب

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

حافظه یکی از منابع با ارزش کامپیوتر است. با رعایت این موضوع، برنامه‌ای بنویسید که با دریافت سه عدد اعشاری، آن‌ها را مرتب کرده و تقریب بزند. در برنامه‌ی خود باید توابع زیر را پیاده‌سازی کرده باشید:

- تابعی که مقدار دو متغیر را بدون متغیر اضافی جابه‌جا کند.

```
1 | void swap(double &a, double &b)
```

- تابعی که سه متغیر را با استفاده از تابع `swap` از بزرگ به کوچک مرتب کند.

```
1 | void sort(double &a, double &b, double &c)
```

- تابعی که یک عدد اعشاری را گرد کند. طوری که مقادیر بالای 0.3 به بالا و باقی مقادیر به پایین گرد می‌شوند.

```
1 | void round(double &a)
```

به نکات زیر توجه کنید:

- با توجه به کارایی، می‌توانید تابع یا توابعی را `inline` تعریف کنید.
- در کل برنامه، تنها حق تعریف سه متغیر اعشاری را دارید.

## ورودی

در یک خط، سه عدد اعشاری  $a$ ،  $b$  و  $c$  با حداکثر پنج رقم اعشار نوشته می‌شوند.

$$0 \leq a, b, c \leq 1000$$

## خروجی

با توجه به مثال‌ها در سه خط از خروجی استاندارد، اعداد مرتب و گرد شده را به ترتیب چاپ کنید.

## مثال

### ورودی نمونه ۱

1.1 2.4 5

### خروجی نمونه ۱

5

3

1

### ورودی نمونه ۲

3.14159 1.78 1.2142

### خروجی نمونه ۲

3

2

1

### ورودی نمونه ۳

3.1 3.2 3.3

خروجی نمونه ۳

3  
3  
3

## تاس‌های ارسلان (امتیازی)

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

ارسلان علاقه دارد بداند با داشتن تعداد مشخصی تاس معمولی، با چه حالت‌هایی ممکن است به یک مجموع خاص برسد. مثلاً دریافته است که با دو تاس، در حالت‌های ۱-۲ و ۲-۱ می‌توان به مجموع سه رسید. همواره پیدا کردن تعداد حالت‌ها ساده نیست. برنامه‌ای بنویسید که به ارسلان در به دست آوردن تعداد حالات کمک کند.

### ورودی

در یک خط دو عدد  $n$  و  $sum$  نوشته می‌شوند که به ترتیب، تعداد تاس‌ها و مجموع مورد نظر را نشان می‌دهند.

$$1 \leq n < 10$$

$$1 \leq sum \leq 10000$$

### خروجی

در یک خط از خروجی استاندارد، تعداد حالات مورد نظر چاپ کنید.

### مثال

#### ورودی نمونه ۱

2 4

خروجی نمونه ۱

3

ورودی نمونه ۲

3 19

خروجی نمونه ۲

0

ورودی نمونه ۳

3 7

خروجی نمونه ۳

15