

به نام خدا



## درس مبانی برنامه سازی

تمرین ۳

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال اول ۹۹-۰۰

---

استاد:

رضا فکوری

مهلت ارسال:

۲۸ آذر - ساعت ۲۳:۵۹:۵۹

مسئول تمرین ها:

امیرمهدی نامجو، پرهام صارمی

مسئول تمرین ۳:

نیما فتاحی

طراحان تمرین ۳:

نازنین آذریان - پویا اسمعیلی - مسیح بیگی - پرهام چاوشیان - متین داغیانی - مازیار شمسی پور

# فهرست

## سوالات

۲	سوال ۱. سوال شانس
۲	سوال ۲. تغییر جمعیت
۴	سوال ۳. استونکس
۷	سوال ۴. مسافرخانه‌ی جک زرد
۹	سوال ۵. رقم بازی
۱۱	سوال ۶. کمی هم گوگل کنیم!
۱۳	



## سوالات

### سوال ۱. سوال شانسی

یکی از دستیاران آموزشی درس مبانی برنامه‌نویسی می‌خواهد برای این درس تمرین طرح کند اما با توجه به این‌که او به شدت به این کار علاقه‌مند است، هیجان‌زده شده و تعداد بسیار زیادی سوال طرح کرده است. این موضوع باعث شده که کار او برای انتخاب تعدادی از سوالات جهت ارائه به دانشجویان سخت شود. بنابراین تصمیم می‌گیرد که ابتدا سوالات را شماره‌گذاری کند و سپس از بین آن‌ها سؤالاتی را که شماره‌شان اعداد تقریباً شانسی هستند انتخاب کنند. از دید TA اعداد شانسی اعدادی هستند که تنها از ارقام ۴ یا ۷ تشکیل شده‌اند. اعداد تقریباً شانسی نیز اعدادی هستند که حداقل یکی از شمارنده‌هایشان عدد شانسی باشد. با توجه به این‌که TA بسیار هیجان‌زده است در تعیین سوالات با شماره تقریباً شانسی به مشکل خورده است، بنابراین به سراغ شما آمده تا در تعیین آن‌ها به او کمک کنید.

### ورودی

در خط اول عدد  $n$  می‌آید و سپس در هرکدام از  $n$  خط بعدی یک عدد می‌آید که باید خروجی متناظر با آن را چاپ کنید.

### خروجی

به ازای هر خط اگر عدد ورودی تقریباً شانسی است در خروجی “YES” و در غیر این صورت “NO” را چاپ کنید.



## ورودی نمونه

3  
47  
16  
78

## خروجی نمونه

YES  
YES  
NO

## ورودی نمونه

5  
753  
272  
539  
710  
518

## خروجی نمونه

NO  
YES  
YES  
NO  
YES



## سوال ۲. تغییر جمعیت

توی ده شلمرود حسنی تک و تنها روی چارپایه نشسته بود و به رفت و آمد افراد روستا توجه می‌کرد. او خیلی زود متوجه شد که جمعیت این ده به نحو خاصی تغییر می‌کند، به طوری که هر سال جمعیت به دلیل مرگ و میر یا تولد زیاد یا کم می‌شود. هم‌چنین او متوجه شد هنگامی که جمعیت روستا عددی فرد باشد میزان تولد آن سال به صفر میرسد. حسنی بالاخره بعد از محاسبات فراوان الگوی زیر را برای جمعیت ارائه داد:

۱. اگر جمعیت عددی زوج باشد می‌تواند تا پایان آن سال دقیقاً یک و نیم برابر شود.

۲. اگر جمعیت از یک بزرگ‌تر باشد ممکن است تا پایان سال یک نفر کم شود.

هم‌چنین این الگو به صورت اتفاقی تکرار می‌شود به این معنا که اگر ۲ نفر در ده وجود داشته باشد، جمعیت در پایان سال هم می‌تواند ۳ باشد هم ۱. اما اگر ۳ نفر در ده باشند در پایان سال حتماً جمعیت ۲ خواهد بود.

حال حسنی می‌خواهد بداند که اگر در آغاز سال  $x$  نفر در روستا وجود داشته باشند بعد از گذشت سال‌های زیاد آیا ممکن است که جمعیت  $y$  شود.

توجه کنید در این سوال شما باید از تکه کد زیر استفاده کنید و مجاز به تغییر دادن تابع `main` نیستید (در صورت تغییر نمره‌ی این سوال برای شما ۰ لحاظ می‌شود).

```
1 #include <stdio.h>
2
3 int main(){
4     int x,y,t;
5     scanf("%d",&t);
6     while(t--){
7         scanf("%d%d",&x,&y);
8         if (check(x,y))
9             printf("YES\n");
10        else
11            printf("NO\n");
12    }
13    return 0;
14 }
15
```



نحوه‌ی ورودی و خروجی گرفتن در این سوال با توجه به این‌که تابع `main` از قبل پیاده‌سازی شده است مشخص بوده و شما تنها باید تابع `check(x, y)` را به گونه‌ای پیاده‌سازی کنید که عبارت‌های YES و NO در خروجی با توجه به صورت مسئله درست چاپ شود.

### ورودی

همانطور که در تابع `main` مشخص است، در خط اول به شما ورودی `t` داده می‌شود که تعداد سوال هاست. در `t` خط بعد در هر خط دو عدد `x` و `y` وارد می‌شوند.

### خروجی

به ازای هر `x`، `y` اگر `x` می‌تواند پس از گذشت چندسال تبدیل به `y` شود عبارت "YES" در خروجی ظاهر می‌شود و در غیر این صورت عبارت "NO".

### ورودی نمونه

```
5
1 2
2 5
4 3
500 750
100 99
```

### خروجی نمونه

```
NO
NO
YES
```



YES

YES



## سوال ۳. استونکس

محید برای حفظ سرمایه‌اش قصد دارد وارد بازار بورس شود اما کارگزاری‌اش قوانین عجیبی دارد. شما قرار است یک سری عدد به عنوان داده سهام‌های متفاوت بگیرید و سوددهی آنها را بر اساس قوانین این کارگزاری حساب کنید. بدین صورت که:

- اگر عدد حجم معاملات عددی اول حلقوی باشد سهم به اندازه مجموع ارقام درصد سودده خواهد بود. عدد اول حلقوی به عددی گفته میشود که با شیف‌ت دادن ارقام آن اول بماند. به طور مثال عدد ۱۱۹۳ عدد اول حلقوی است زیرا تمام اعداد ۱۱۹۳ و ۳۱۱۹ و ۹۳۱۱ و ۱۹۳۱ اول هستند.  
مثلاً: حجم معاملات = ۱۱۹۳  
پس ۱۴ درصد سود می‌کند. و در غیر این صورت به تعداد مقسوم‌علیه‌های اولش زیان‌ده خواهد بود.  
مثلاً حجم معاملات: ۶، ۲ درصد ضرر ده خواهد بود (۲ - درصد سود)
- اختلاف حجم تقاضا (t) و حجم عرضه (a) را x می‌نامیم:

$$x = t - a$$

اگر  $|x|$  بر تعداد مقسوم‌علیه‌های خودش بخش‌پذیر بود سهم به اندازه ضرب ارقامش سود یا ضرر می‌دهد.  
مثلاً ۱۲: ۱، ۲، ۳، ۴، ۶، ۱۲ در نتیجه ۲ درصد سود یا ضرر می‌دهد.  
در غیر این صورت به اندازه مجموع مقسوم‌علیه‌های اولش سود یا ضرر می‌دهد.  
مثلاً ۱۴: ۱، ۲، ۷، ۱۴ در نتیجه ۹ درصد سود یا ضرر می‌دهد.  
اگر x عددی مثبت باشد سود ده و اگر عددی منفی باشد زیان‌ده خواهد بود.  
اگر x یا حجم معاملات ۰ باشد سود آن بخش نیز ۰ است.

سود نهایی از جمع سود‌های بدست آمده از حجم معاملات و حجم عرضه و تقاضا بدست می‌آید.





## ورودی

عدد  $n$  به عنوان تعداد سهم‌هایی که می‌خریم داده می‌شود.  
سپس در  $n$  خط بعدی در هر خط ۵ عدد صحیح با یک فاصله به صورت زیر داده می‌شوند:

عدد اول: کد شناسایی سهام

عدد دوم: حجم معاملات

عدد سوم: قیمت

عدد چهارم: حجم تقاضا

عدد پنجم: حجم عرضه

## خروجی

بعد از هر خط ورودی درصد نهایی سود یا زیان هر سهم نمایش داده شود و در آخر پرسودترین سهم (بیشترین درصد سود تقسیم بر قیمت) مشخص شود. در صورتی که دو سهم یک مقدار درصد سود تقسیم بر قیمت داشتند سهمی که زودتر آمده انتخاب می‌شود.

## ورودی نمونه

```
2
2302 20 1000 18 30
1102 20 10 18 30
```

## خروجی نمونه

```
-4%
-4%
Best option: 2302
```



## سوال ۴. مسافر خانه‌ی جک زرد

دیوید جونز قصد دارد تا یک بازی جهان‌باز (open-world) بسازد و در قسمتی از این بازی شخصیت اصلی قرار است دارت بازی کند. او از شما می‌خواهد که منطق این بازی را به صورت زیر پیاده‌سازی کنید.

۳ دایره با شعاع‌های مختلف و هم‌مرکز داریم (مرکز مبدا مختصات است) که نسبت شعاع‌ها یکسان نیست (اگر یکسان بود برنامه باید عبارت error را چاپ کند و خارج شود). هر شعاع یک امتیاز خاص دارد به طوری که شعاع کوچک‌تر امتیاز بیشتر و شعاع بزرگ‌تر امتیاز کمتری دارد.

قرار است تا مختصات برخورد دارت‌ها به شما داده شود و با توجه به شعاع کوچک‌ترین دایره و نسبت شعاع‌ها و امتیاز هر شعاع، امتیاز هر سری بازی توسط برنامه شما محاسبه شود.

## ورودی

خط اول: ۳ عدد که بعد از هر کدام حرف R آمده با یک فاصله وارد می‌شود.

خط دوم: ۳ عدد که بیانگر امتیازهاست.

خط سوم: شعاع کوچک‌ترین دایره.

خط چهارم: تعداد پرتاب‌ها  $(n)$ .

$n$  خط بعدی: مختصات برخورد.

همه اعداد ورودی صحیح هستند.

## خروجی

عددی صحیح که بیانگر امتیاز کسب شده است.



## ورودی نمونه

100R 200R 900R  
100 20 50  
10  
3  
0 0  
5 12  
100 100

## خروجی نمونه

150

پرتاب اول ۱۰۰ امتیاز، پرتاب دوم ۵۰ امتیاز و پرتاب سوم (به دلیل خارج از محدوده بودن) صفر امتیاز دارند. همان‌طور که دیدید نسبت شعاع‌های داده شده و R بیانگر کوچکترین شعاع نیست و به عبارت دیگر ۱ و ۲ و ۹ با ۵ و ۱۰ و ۴۵ فرقی ندارد و کوچک‌ترین شعاع تعیین کننده اندازه دایره هاست.



## سوال ۵. رقم بازی

فاطمه و محمد در حال انجام یک مسابقه هستند. مسابقه به این صورت است که ابتدا یک عدد به آن‌ها داده می‌شود. سپس آن‌ها ارقام این عدد را از سمت چپ (یعنی رقم با بیشترین ارزش) شماره‌گذاری می‌کنند. فاطمه حق دارد اعدادی که در خانه‌های فرد هستند را بردارد و محمد اجازه برداشتن اعداد موجود در خانه‌های زوج را دارد. فاطمه مسابقه را شروع می‌کند و یک عدد دل‌خواه را برمی‌دارد. سپس محمد عددی برمی‌دارد و آن‌قدر این کار را به نوبت انجام می‌دهند تا فقط یک عدد باقی بماند. اگر عدد باقی‌مانده زوج باشد محمد و اگر فرد باشد فاطمه برنده‌ی مسابقه خواهد بود (یعنی آخرین عدد زوج باشد و زوج یا فرد بودن مکان آن عدد مهم نیست).

برنامه‌ای بنویسید که بتواند برنده این مسابقه را از قبل پیش‌بینی کند. این برنامه ابتدا یک عدد از ورودی می‌گیرد و به آن تعداد مسابقه انجام خواهد شد. در هر مسابقه یک عدد داده می‌شود و شما به ازای هر مسابقه باید برنده را اعلام کنید.

اگر فاطمه برنده شد عدد یک و اگر محمد برنده شد عدد دو را در خروجی چاپ کنید. توجه داشته باشید که جایگاه اعداد فقط در مرحله اول تعیین شده و ثابت می‌ماند و در هر مرحله تغییر نمی‌کند.

## ورودی

ابتدا عدد صحیح  $t$ ، تعداد تست‌ها داده می‌شود. سپس در هر یک از  $t$  خط بعدی عدد صحیح حداکثر ۱۸ رقمی  $n$  داده می‌شود که نشان‌دهنده‌ی یک مسابقه است.

## خروجی

به ازای هر تست ورودی یک‌بار یکی از دو عدد ۱ یا ۲ چاپ خواهد شد.



## ورودی نمونه

4  
2  
3  
102  
2069

## خروجی نمونه

2  
1  
1  
2

- در تست اول عدد ۲ داده شده است. چون فقط یک عدد داده شده هیچ‌کس نمی‌تواند عددی بردارد و عدد ۲ باقی می‌ماند و چون زوج است محمد برنده خواهد شد.
- به صورت مشابه با دادن عدد ۳ برنده فاطمه خواهد بود.
- در عدد ۱۰۲ فاطمه می‌تواند یکی از دو خانه‌ی فرد را انتخاب و حذف کند اما محمد فقط امکان حذف عدد ۰ را دارد (چون تنها عددی است که شماره خانه‌اش زوج است). بنابراین فاطمه برای آن‌که برنده شود در مرحله اول عدد ۲ را می‌سوزاند. سپس محمد در نوبت خود تنها انتخابش یعنی عدد ۰ را می‌سوزاند و چون عدد ۱ که باقی مانده فرد است فاطمه برنده خواهد شد.
- در عدد ۲۰۶۹ فاطمه می‌تواند اعداد ۲ و ۶ را بسوزاند اما محمد فقط یکی از دو عدد موجود در خانه‌های زوج را می‌تواند بسوزاند تا عدد باقی‌مانده برنده را تعیین کند. بنابراین محمد عدد ۹ را می‌سوزاند تا در مرحله‌ی آخر عدد ۰ باقی‌مانده و خودش برنده شود.



## سوال ۶. کمی هم گوگل کنیم!

شما باید تابع `findDivisorSum` را به صورتی پیاده‌سازی کنید که تعدادی عدد از ورودی گرفته و مجموع شمارنده‌های آن‌ها را محاسبه کند. سپس به صورت زیر خروجی را برگرداند. (متغیر `s` خارج از برنامه شما به صورت گلوبال تعریف شده است و شما می‌توانید در هر جای برنامه از آن استفاده کنید):

- اگر مقدار `s` برابر ۰ بود، تابع شما ۴ ورودی گرفته که ورودی اول آن یک کاراکتر و ورودی‌های بعدی اعداد شما هستند. اگر مقدار کاراکتر `M` بود، بیشترین و اگر `m` بود کمترین مقدار را از بین مجموع عامل‌های اول ورودی‌ها برگردانید.
  - اگر مقدار `s` برابر ۰ نبود، ورودی اول تابع `n` بوده و پس از آن `n` ورودی می‌آید. این بار نیز مجموع عامل‌های اول را محاسبه کرده و کمترین آن‌ها را برگردانید.
- اعداد ورودی تابع شما به صورت `calc(num1, num2, +)` یا هر عملیات ریاضی که جایگزین `+` شده است، می‌باشند. عملیات ریاضی به صورت کاراکتر نبوده و شما باید کدی بنویسید که دقیقاً همین فرمت را پردازش کند. (در صورتی که عملیات ریاضی تقسیم باشد، عدد اول حتماً بر عدد دوم بخش‌پذیر است).

## مثال

حالت اول (متغیر `s` با مقدار ۰ در برنامه وجود دارد):

```
findDivisorSum('M', calc(5, 10, +), calc(17, 13, *), calc(45, 5, /))
```

حالت دوم (متغیر `s` با مقدار ۱ در برنامه وجود دارد):

```
findDivisorSum(2, calc(12, 2, *), calc(50, 17, %))
```

شما تنها باید تابع‌های خواسته شده را پیاده‌سازی کنید و کد شما نباید تابع `main` داشته باشد. همچنین باید کد زیر در ابتدای برنامه‌ی شما قرار بگیرد.



```
1 #include "grader.h"
```

به علاوه تابع زیر باید در برنامه ی شما وجود داشته باشد:

```
1 long long run(char type, int num1, int num2, int num3, int num4) {  
2     return findDivisorSum(type,  
3     calc(num1, num2, +),  
4     calc(num1, num2, -),  
5     calc(num3, num4, *));  
6 }  
7
```

نکات ارسال پاسخ:

۱. پسوند فایل نهایی خود را به `cpp`. تغییر داده و به عنوان کد `C++` ارسال کنید. (کد شما باید به زبان `C` باشد ولی ارسال به عنوان `C++` اشکالی ندارد).
۲. برای قسمت اول و دوم سوال کد یکسان ارسال کنید. توجه کنید که در صورت مغایرت کد های ارسالی نمره ای از سوال دریافت نخواهید کرد.