

به نام خدا



## درس مبانی برنامه‌سازی

تمرین ۲

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال اول ۹۹-۰۰

---

استاد:

رضا فکوری

مهلت ارسال:

۱۶ آذر - ساعت ۲۳:۵۹:۵۹

مسئول تمرین‌ها:

امیرمهدی نامجو، پرهام صارمی

مسئول تمرین ۲:

فاطمه خاشعی

طراحان تمرین ۲:

آریا جلالی

میلاذ سعادت

محمدآرمان سلیمانی

نوید اسلامی

سایه جارالهی

# فهرست

## نکات قابل توجه

۲

## سوالات

۳

- سوال ۱. جهان های موازی ..... ۳
- سوال ۲. گم شدن پدر بزرگ ریک ..... ۵
- سوال ۳. از این شکلا ..... ۷
- سوال ۴. ریک در سیاره کم زبان ها ..... ۱۰
- سوال ۵. Aulos ..... ۱۳

## پاسخ ها

۱۵

- پاسخ ۱. جهان های موازی ..... ۱۵
- پاسخ ۲. گم شدن پدر بزرگ ریک ..... ۱۶
- پاسخ ۳. از این شکلا ..... ۱۷
- پاسخ ۴. ریک در سیاره کم زبان ها ..... ۱۸
- پاسخ ۵. Aulos ..... ۲۰



## نکات قابل توجه

- توجه داشته باشید که در این تمرین مجاز به استفاده از مباحث بعد از شرط و حلقه نیستید. همچنین حق استفاده از توابع آماده‌ی زبان C (به جز printf و scanf) را ندارید.
- سوالات و ابهامات خود درباره‌ی تمرین را در کوئرا مطرح کنید.



## سوالات

### سوال ۱. جهان های موازی

ریک که بار قبل از متوقف کردن زمان لذت برده بود، دوباره تصمیم گرفته زمان را متوقف کند تا مورتی به دلایل های تمام نشدنی خود برسد. اما این بار بعد از اینکه زمان را فعال می‌کنند، دوباره درگیر حضور در چند جهان موازی می‌شوند. آن‌ها تنها ۴ ساعت زمان دارند تا دکمه تطبیق جهان‌ها را بزنند، وگرنه هیچ‌گاه به دنیای معمول بر نمی‌گردند. اما ریک که در حالت طبیعی قرار ندارد، کاری از دستش بر نمی‌آید و حالا مورتی باید سعی کند تا دکمه تطبیق جهان‌ها را به تنهایی بزند. برای زدن این دکمه ابتدا باید رمزی را وارد کند که یک عدد صحیح است. اما این عدد نیاز به مقدار زیادی محاسبه دارد که مورتی از انجام آن عاجز است. مورتی برای محاسبه این رمز از شما کمک می‌خواهد. مورتی تنها میدانند که دو بازه عددی را با دانستن ابتدا و انتهای آن‌ها دارد. او باید با انتخاب یک **عدد صحیح** از هر بازه و ضرب دو عدد انتخاب شده، عدد رمز را بسازد، اما انتخاب این دو عدد باید به صورتی باشد که عدد رمز **بیشترین مقدار ممکن** باشد. به مورتی کمک کنید و عدد رمز را برای او پیدا کنید.

#### ورودی

ورودی شامل دو خط است. در خط اول به ترتیب دو عدد صحیح  $a$  و  $b$  با یک فاصله از هم آمده‌اند که دو سر بازه اول هستند. در خط دوم نیز دو عدد صحیح  $c$  و  $d$  آمده‌اند که دو سر بازه دوم هستند.

$$-10^9 \leq a \leq b \leq 10^9$$

$$-10^9 \leq c \leq d \leq 10^9$$

#### خروجی

خروجی برنامه‌ی شما باید شامل یک عدد صحیح باشد. این عدد ماکزیمم حاصل ضرب دو عدد از دو بازه است. در حل این سوال مجاز به استفاده از توابع آماده زبان سی نیستید.



## مثال

### ورودی نمونه ۱

1	5	25
2	4	120
3		

### خروجی نمونه ۱

1	3000
2	

### ورودی نمونه ۲

1	25	25
2	4	120
3		

### خروجی نمونه ۲

1	3000
2	



## سوال ۲. گم شدن پدر بزرگ ریک

ریک که تله‌پورترش را گم کرده است، در سیاره‌ی Blaart گیر افتاده است. ساکنان سیاره‌ی Blaart دارای تله‌پورترهایی ابتدایی هستند که فقط می‌توانند یک چیز یا یک کس را بین ۲ مقصد از پیش تعیین شده جابه‌جا کنند. متأسفانه مورتی باز خرابکاری کرده است و باعث به وجود آمدن چند نسخه از خودش و پدر بزرگ خود شده است. ریک‌ها باید برای درست کردن این خرابکاری، تمام مورتی‌ها را پیدا کنند، اما چون ریک‌ها آدم‌های کم حوصله‌ای هستند، می‌خواهند ببینند آیا می‌توانند به مورتی خود برسند یا باید به دنبال یک مورتی جدید باشند. آن‌ها برای جواب این سوال به شما رو آورده‌اند. به آن‌ها کمک کنید.

### ورودی

ورودی ابتدا شامل عدد  $s$  که تعداد ریک‌ها را نشان می‌دهد. سپس در  $s$  تست بعدی عددی  $m$  و  $n$  در یک خط آمده‌اند که  $n$  نشان دهنده‌ی تعداد تله‌پورترهای هر ریک و  $m$  نشان دهنده‌ی موقعیت مورتی ریک  $n$  ام می‌باشد. برای هر تست به تعداد تله‌پورترهای آن تست نقطه‌ی شروع و پایان هر تله‌پورتر در خط جدید داده می‌شود.

$$1 \leq n, m \leq 100$$

دقت کنید هر تله‌پورتر قادر است ریک را در هر نقطه‌ای که بین نقطه‌ی شروع و پایان آن قرار دارد تله‌پورت کند. همچنین دقت داشته باشید که مشخصات تله‌پورترها به ترتیب مبدا داده شده است.

### خروجی

خروجی برنامه‌ی شما باید به تعداد ریک‌های موجود، در صورت وجود راه عبارت "YES" و در غیر این صورت عبارت "NO" را چاپ کند.



## مثال

### ورودی نمونه ۱

1	2
2	3 7
3	0 2
4	3 5
5	5 6
6	2 5
7	0 4
8	4 8
9	

### خروجی نمونه ۱

1	NO
2	YES
3	

### ورودی نمونه ۲

1	1
2	4 10
3	0 3
4	2 6
5	4 8
6	7 10
7	

### خروجی نمونه ۲

1	YES
2	

ریک ابتدا به خانه‌ی دوم سپس به خانه‌ی چهارم و پس از آن به خانه‌ی هفتم و در نهایت به خانه‌ی دهم می‌رسد.



### سوال ۳. از این شکلا

ریک که پس از تلاش‌های بسیار در داستان‌های تمرین نتوانست شما را هوادار واقعی خودش کند، بسیار ناراحت است. او تصمیم می‌گیرد به جای این افراد فن‌نما فن‌های واقعی برود تا کمی احساس دوست داشته شدن کند. او برای این کار تصمیم می‌گیرد فن‌هایی بسازد که به او وفادار باشند. فن‌های او از چهار پره مثلی به صورت زیر تشکیل شده‌اند. او که می‌بیند شما در برنامه‌نویسی ادعا دارید از شما می‌خواهد به عنوان حداقل کار در طراحی فن‌ها به او کمک کنید.

#### ورودی

در تنها سطر ورودی یک عدد که برابر طول هر ضلع شکل است آمده است.

$$2 \leq n \leq 700$$

#### خروجی

شکل مورد نظر را چاپ کنید.





## مثال

### ورودی نمونه ۱

1	11
2	

### خروجی نمونه ۱

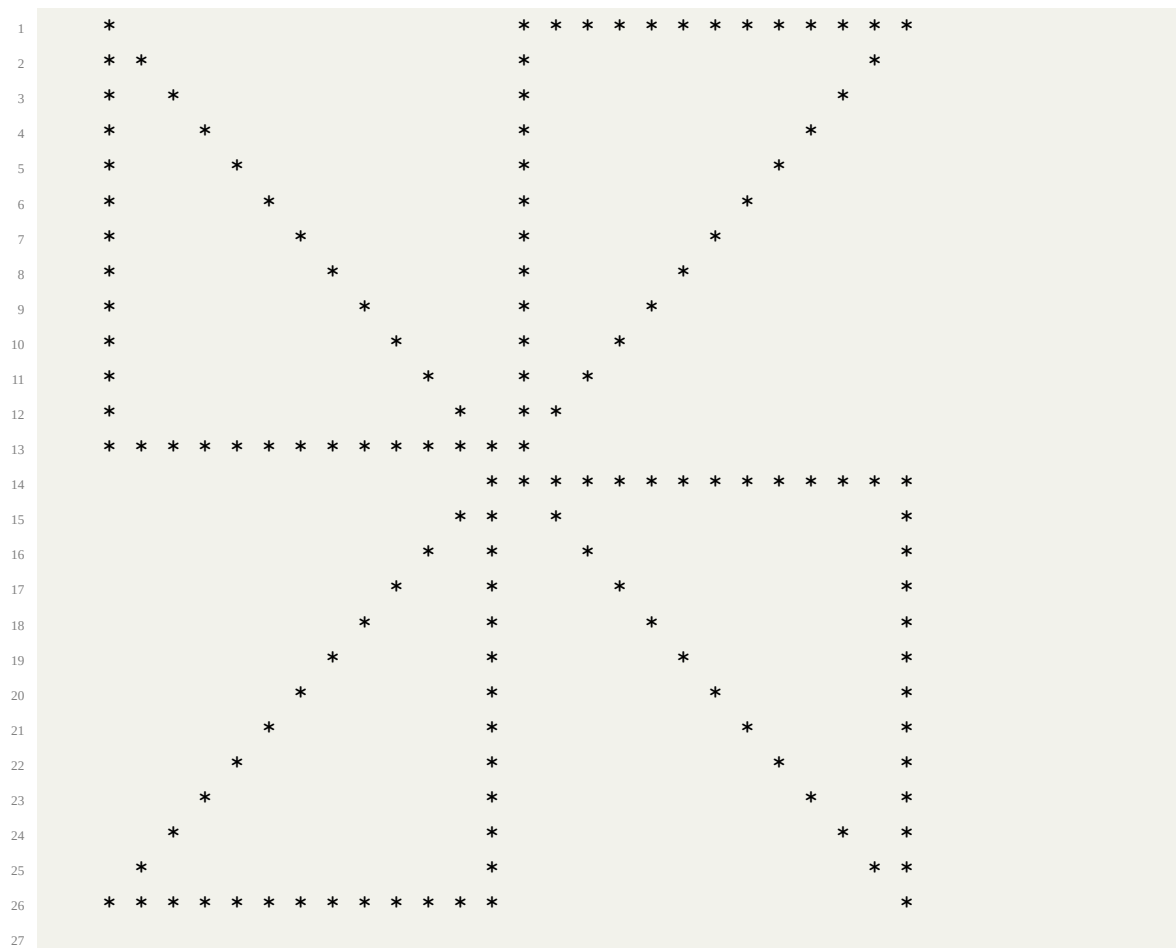
1	*								*	*	*	*	*	*	*	*	*	*	*	*	*
2	*	*							*										*		
3	*		*						*										*		
4	*			*					*									*			
5	*				*				*							*					
6	*					*			*						*						
7	*						*		*					*							
8	*							*	*				*								
9	*								*		*										
10	*								*		*	*									
11	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
12									*	*	*	*	*	*	*	*	*	*	*	*	*
13								*	*	*	*	*	*	*	*	*	*	*	*	*	*
14							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
15						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
16					*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
17				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
18			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
19		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
20	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
21	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
22	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
23																					

### ورودی نمونه ۲

1	13
2	



## خروجی نمونه ۲





## سوال ۴. ریک در سیاره کم‌زبان‌ها

در حین فرار از دشمنان همیشگی خود، سفینه ریک دچار آسیب دیدگی جدی می‌شود و او (که این بار به تنهایی سفر کرده) مجبور به فرود اضطراری به روی سیاره کم‌زبان‌ها می‌گردد. ساکنین این سیاره - کم‌زبان‌ها - دارای زبانی بدوی هستند. کل این زبان از ۹ کاراکتر تشکیل شده است و حتی فاصله نیز در جملات زبان آنها جایگاهی ندارد. ریک پس از مدتی کلنجار رفتن با کم‌زبان‌ها، تصمیم می‌گیرد نرم‌افزاری تولید کند که به او در فهم زبان کم‌زبان‌ها یاری رساند و جایگاه واژگانی که ریک از قبل می‌شناسد را در جمله پیدا کند تا بلکه او بتواند جملات آنها را متوجه شود. از آنجایی که مورتی مشغول یادگیری آنلاین در مدرسه بوده است و نتوانسته همراه ریک شود، ریک برای ساخت این نرم‌افزار به کمک شما احتیاج دارد.

این نرم‌افزار با دریافت یک جمله و یک کلمه، محل قرارگیری کلمه در جمله را پیدا می‌کند و اگر کلمه در جمله یافت نشود عدد ۱- را چاپ می‌کند.

### ورودی

ابتدا در یک خط عدد  $n$  داده می‌شود که نشان دهنده طول جمله‌ای است که قرار است دریافت کند. جمله ورودی فقط متشکل از کاراکترهای  $a, b, c, d, e, f, g, h, i$  است و شامل فاصله، کاراکترهای دیگر نمی‌باشد. سپس در خط دوم ورودی، عدد  $m$  داده می‌شود که نشان دهنده طول کلمه است. قوانین گفته شده درباره جمله درباره کلمه نیز صدق می‌کند.

$$0 < n, m < 10$$

سپس ابتدا در  $n$  خط، کاراکترهای جمله به ترتیب داده می‌شوند (در هر خط یک کاراکتر) و بعد از آن در  $m$  خط، کاراکترهای کلمه به ترتیب داده می‌شوند. (در هر خط یک کاراکتر) همچنین تضمین می‌گردد کلمه حداکثر یک بار در جمله ظاهر می‌گردد.

### خروجی

در تنها خط خروجی باید یک عدد بین ۰ تا  $n - m$  داده شود که نشان دهنده این است



که کلمه در چندمین اندیس جمله آمده است. اگر کلمه در جمله یافت نمی‌شد، باید عدد 1- چاپ گردد. منظور از اینکه کلمه در چندمین اندیس جمله ظاهر شده است این است که چنانچه کلمه در جمله آمده‌است، جایی که کلمه شروع می‌شود چندمین حرف از جمله است (و شمردن اندیس از ۰ آغاز می‌شود نه ۱) (مثلا در test، اگر دنبال es باشیم می‌گوییم از اندیس اول شروع شده، te از اندیس صفرام شروع شده و ...)

### مثال

#### ورودی نمونه ۱

1	5
2	2
3	a
4	b
5	c
6	d
7	e
8	c
9	d
10	

#### خروجی نمونه ۱

1	2
2	

در این مثال جمله داده شده abcde است و کلمه cd است. کلمه ما در جمله ظاهر شده و اندیس دوم جمله است.

#### ورودی نمونه ۲

1	9
2	3
3	a
4	a
5	a
6	b
7	b
8	b
9	i
10	i
11	i
12	b



13  
14  
15

b  
i

### خروجی نمونه ۲

1  
2

4

جمله داده شده aaabbbiii است و کلمه داده شده bbi است. کلمه در جمله ظاهر شده و در اندیس چهارم قرار گرفته است.

### ورودی نمونه ۳

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11

5  
3  
a  
a  
e  
f  
h  
a  
e  
g

### خروجی نمونه ۳

1  
2

-1

جمله داده شده aaefh است و کلمه داده شده aeg است. این کلمه اصلاً در جمله ظاهر نشده است پس خروجی 1- است.



## سوال ۵. Aulos

یک روز مورتی وقتی که داشت از کنار جکوز رد می‌شد، یک فروشنده‌ی بسیار مشکوک دید.

چون که مورتی بیکار بود، تصمیم گرفت که با این شخص صحبت کند. آن شخص هم چون که فروشنده بود، دائما تلاش می‌کرد که به مورتی جنس بفروشد. مورتی بعد از کلی تلاش به طرف فهماند که قصد خرید چیزی از او را ندارد. فروشنده که دید از این وضعیت پولی نمی‌تواند در بیاورد، تصمیم گرفت که یک معامله‌ای با مورتی انجام دهد. به او گفت که اگر سوال زیر را برایش حل بکند، زیباترین ویدئو دنیا را به اون نشان خواهد داد! سوال فروشنده به این صورت است که از مورتی می‌خواهد که برنامه‌ای به صورت فوق برایش بنویسد.  $T$  تست برای مورتی تعریف خواهد شد، که در هر کدام از این تست‌ها دنباله‌ای از اعداد به نام  $A$  به طول  $n$  به او داده می‌شود. به ازای هر کدام از این دنباله‌ها اگر اندیسی مثل  $x$  موجود بود که یکی از روابط زیر برقرار باشد، مورتی باید خروجی YES و در غیر این صورت NO را چاپ کند.

$$A_1 \leq A_2 \leq \dots \leq A_x \geq \dots \geq A_{n-1} \geq A_n$$

$$A_1 \geq A_2 \geq \dots \geq A_x \leq \dots \leq A_{n-1} \leq A_n$$

مورتی که به شدت کنجکاو شده بود، تصمیم گرفت که سوال را برای او حل کند، ولی چون سوال برایش سخت بود از شما کمک خواسته است!  
اگر شما هم کنجکاو شده‌اید، به اینجا بروید :

## ورودی

ورودی ابتدا شامل عدد  $T$  در یک خط است. سپس در  $T$  جفت خطی بعدی، اول عدد  $n$  در یک خط و در خط بعدی دنباله‌ی  $A$  مربوط به آن تست به شما داده خواهد شد.

$$1 \leq n \leq 5000$$

$$1 \leq T \leq 20$$

$$-10^9 \leq A_i \leq 10^9$$



## خروجی

خروجی برنامه‌ی شما باید شامل  $T$  خط باشد که در هر خط، جواب با فرمت YES یا NO مربوط به همان تست چاپ شده باشد. دقت کنید که خروجی را دقیقاً با همین فرمت و حروف بدهید و اشتباهاً از حروف کوچک استفاده نکنید.

## مثال

### ورودی نمونه ۱

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

### خروجی نمونه ۱

```
1 YES
2 YES
3
```

همان طور که در دنباله‌های عددی مشاهده می‌شود، شروط برقرار بودند. در دنباله‌ی اول  $x = 2$  و در دنباله‌ی دوم  $x = 3$  می‌باشد.

### ورودی نمونه ۲

```
1 3
2 4
3 1 2 1 3
4 5
5 8 10 15 17 10000
6 5
7 10 10 2 5 1
8
```

### خروجی نمونه ۲

```
1 NO
2 YES
3 NO
4
```



## پاسخ‌ها

## پاسخ ۱. جهان‌های موازی

نمونه پاسخ مورد قبول:

```
1 #include <stdio.h>
2
3 #include <stdio.h>
4
5 int main()
6 {
7     long long int a,b,c,d;
8     scanf("%lld %lld %lld %lld",&a,&b,&c,&d);
9     long long int max = b*d;
10    if(a<0 || c<0){
11        if(max< a*c)
12            max = a*c;
13        if(max<a*d)
14            max = a*d;
15        if(max<b*c)
16            max = b*c;
17    }
18    printf("%lld",max);
19    return 0;
20 }
21
```

برای حل سوال کافی است ماکزیمم حاصل ضرب هر یک از دو سر بازه‌ها را پیدا کنیم. در بدترین حالت، با حساب کردن چهار حاصل ضرب، به بیشترین مقدار ممکن می‌رسیم. واضح است که نیازی به بررسی کردن داخل بازه‌ها نیست و محاسبه حاصل ضرب‌های  $a * c$  و  $b * c$  و  $b * d$  و  $a * d$  کفایت می‌کند. مقدار حاصل ضرب را  $max$  می‌نامیم و در ابتدا آن را برابر با حاصل ضرب انتهای دو بازه قرار می‌دهیم. پس از آن، این مقدار را با هر یک از سه حاصل ضرب دیگر مقایسه می‌کنیم و در صورتی که حاصل ضرب‌ها بزرگتر از  $max$  بود، مقدار حاصل ضرب را در این متغیر می‌ریزیم. در انتها مقدار  $max$  برابر با پاسخ مسئله است. متغیر  $max$  را از نوع `long long` تعریف کردیم، زیرا هر یک از اعداد در بازه  $[-10^9, 10^9]$  هستند و در صورتی که دو عدد در یکدیگر ضرب شوند و متغیر از `int` باشد، ممکن است سرریز رخ دهد.





## پاسخ ۲. گم شدن پدر بزرگ ریک

نمونه پاسخ مورد قبول:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int numberOfTeleports, destination, numberOfRicks;
6     scanf("%d", &numberOfRicks);
7     for (int k = 0; k < numberOfRicks; k++)
8     {
9         int temporaryMax = 0;
10        scanf("%d %d", &numberOfTeleports, &destination);
11        for (int i = 0; i < numberOfTeleports; i++)
12        {
13            int startPoint, endPoint;
14            scanf("%d %d", &startPoint, &endPoint);
15            if (temporaryMax >= startPoint && temporaryMax <= endPoint)
16                temporaryMax = endPoint;
17        }
18
19        if (temporaryMax >= destination)
20            printf("YES\n");
21        else
22            printf("NO\n");
23    }
24 }
25
```

در هر تست یک متغیر به نام *temporaryMax* را برابر با ۰ قرار می‌دهیم. این متغیر پس از گرفتن نقطه‌ی شروع و پایان هر تلیپورتر بررسی می‌کند که آیا در بازه‌ی نقطه‌ی شروع و پایان تلیپورتر جدید قرار می‌گیرد یا نه. در صورت قرار گرفتن مقدار نقطه‌ی پایانی تلیپورتر جدید را در متغیر *temporaryMax* قرار می‌دهیم. دقت کنید این متغیر مقدار بیشترین خانه‌ای را که تا آن نقطه از تست به آن دسترسی داریم را در خود نگه می‌دارد. در پایان هر تست بررسی می‌شود که آیا بیشترین خانه‌ای که می‌توانیم به آن دسترسی داشته باشیم از مقصد بزرگتر است یا خیر و پاسخ متناسب با آن چاپ می‌شود و مقدار ۰ در متغیر *temporaryMax* برای تست بعد ذخیره می‌شود



## پاسخ ۳. از این شکلا

نمونه پاسخ مورد قبول:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int n;
6     scanf("%d" , &n);
7     for(int i=0 ; i<2*n ; i++){
8         for(int j=0 ; j<2*n ; j++){
9             if(i == j || i == 2*n-1-j) // Diagonal lines
10                printf("* ");
11             else if( (i==n-1 && j<=n) || (i==n && j>=n)) // Middle
12                horizontal lines
13                printf("* ");
14             else if( (j==n && i<=n) || (j==n-1 && i>=n)) // Middle
15                vertical lines
16                printf("* ");
17             else if( (i==0 && j>n) || (i==2*n-1 && j<n)) // Horizontal
18                lines of the first and last rows
19                printf("* ");
20             else if( (j==0 && i<n) || (j==2*n-1 && i>n)) // Left and
21                right vertical lines
22                printf("* ");
23             else
24                printf(" ");
25         }
26         printf("\n");
27     }
28 }
```

یک جدول  $2n$  در  $2n$  در نظر بگیرید که شکل درون آن کشیده شده باشد. شکل خواسته شده را می‌توان به ۳ دسته از خط‌ها تقسیم کرد:

- خطوط قطری: قطر اصلی و فرعی مربع  $2n$  در  $2n$
- خطوط افقی: ۴ پاره خط یکی در بالا دوتا در وسط و یکی در پایین
- خطوط عمودی: ۴ پاره خط یکی در چپ دوتا در وسط و یکی در سمت راست



با به دست آوردن معادله خط هر یک از این خطوط اگر نقطه ای در یکی از این معادلات صدق می کرد به ازای آن ستاره و در غیر آن صورت فاصله چاپ میکنیم.

#### پاسخ ۴. ریک در سیاره کم‌زبان ها

برای حل سوال های این چینی، در زمانی که محدودیت عدم استفاده از رشته و آرایه ها برقرار باشد، یک ایده این است که به نحوی واژه و جمله را در متغیرهای ساده تر ذخیره کنیم تا بتوان آنها را جلوتر استفاده کرد.

متغیرهایی که تا مبحث این تمرین بررسی شده اند، ماهیتا یک عدد هستند. پس لازم است واژه را به یک عدد map کنیم، یعنی در واقع یک راهی پیدا کنیم که بتواند چندین کاراکتر را به یک عدد تبدیل کند.

توجه کنید که راه ما باید یک به یک باشد (از دیدگاه تابعی که map می کند) یعنی به ازای هر عدد، فقط یک واژه وجود داشته باشد که آن عدد را بسازد (و از هر واژه یک عدد یکتا بدست آید)

می توان هر کاراکتر را به یک رقم نسبت داد. چون کم زبان ها فقط ۹ کاراکتر دارند، مبنای ۱۰ که با آن آشنا هستیم کفایت می کند (و رقم صفر را برای کاهش پیچیدگی نادیده می گیریم)

یعنی می گوئیم a متناظر با ۱ است و ... و i متناظر با ۹ است. پس مثلا babaei متناظر با ۲۱۲۱۵۹ است و یا eg متناظر با ۵۷ است.

حال با استفاده از این ایده، ابتدا عدد متناظر با واژه را معادل صفر قرار می دهیم. هر کاراکتر که دریافت شد، عدد را به اضافه آن کرده و ضرب در ۱۰ می کنیم. اینگونه عدد معادل با واژه ما به دست می آید.

با این روش، هم جمله هم کلمه را در دو متغیر عددی ذخیره می کنیم. سپس کافی ست با عملیات ریاضی (تقسیم و باقی مانده) بررسی کنیم که آیا عدد متناظر با کلمه، درون عدد متناظر با جمله وجود دارد یا خیر.

```
1 #include <stdio.h>
2 int n;
3 int m;
4 int string=0;
5 int substring=0;
```



```
6 char input;
7 /*
8 a = 1 / b = 2 / c = 3 / d = 4 / e = 5 / f = 6 / g = 7 / h = 8 / i =
9
10
11
12
13 baad str baad substr
14 */
15
16 int main() {
17     scanf("%d",&n);
18     scanf("%d",&m);
19     //daryafte jomle
20     for(int i=0;i<n;i++){
21         scanf(" %c",&input);
22         string += (input-96);
23         string *= 10;
24     }
25     string /= 10;
26     //daryafte kalame
27     for(int i=0;i<m;i++){
28         scanf(" %c",&input);
29         substring += (input-96);
30         substring *= 10;
31     }
32     substring /= 10;
33
34     // ta inja dota vorodi ro darim. hala bayad check konim in tu oon
    has ya na
35
36     for(int i=0;i<n+1;i++){
37         int i_p = 1;
38         int j_p = 1;
39         for(int k=0;k<i;k++){
40             i_p *= 10;
41         }
42         for(int l=0;l<m;l++){
43             j_p *= 10;
44         }
45         if((string/i_p)%j_p == substring){
46             printf("%d",n-m-i);
47             return 0;
```



```
48     }
49 }
50 printf("%d",-1);
51
52 return 0;
53 }
54
55
```

## پاسخ ۵. Aulos

نمونه پاسخ مورد قبول:

```
1 #include <stdio.h>
2
3 int main() {
4     int T, n;
5     scanf("%d", &T);
6
7     for (int T_T = 0; T_T < T; T_T++) {
8         scanf("%d", &n);
9         int state1 = 0, state2 = 0, val = 0, last = -1;
10
11         for (int i = 0; i < n; i++) {
12             scanf("%d", &val);
13             if (i) {
14                 if (state1 == 0) {
15                     if (val < last) {
16                         state1++;
17                     }
18                 }
19                 else if (state1 == 1) {
20                     if (val > last) {
21                         state1++;
22                     }
23                 }
24
25                 if (state2 == 0) {
26                     if (val > last) {
27                         state2++;
28                     }
29                 }
30             }
31         }
32     }
33 }
```



```
30     else if (state2 == 1) {
31         if (val < last) {
32             state2++;
33         }
34     }
35     last = val;
36 }
37 if (state1 != 2 || state2 != 2) {
38     printf("YES\n");
39 }
40 else {
41     printf("NO\n");
42 }
43 }
44 }
45 }
46 }
```

سعی کنید اول نسخه‌ی ساده‌تر سوال را حل کنید. ما باید چک کنیم که دنباله‌ی ورودی حداقل یکی از دو شرط زیر را برای یک  $x$  ای داشته باشد:

$$A_1 \leq A_2 \leq \dots \leq A_x \geq \dots \geq A_{n-1} \geq A_n$$

$$A_1 \geq A_2 \geq \dots \geq A_x \leq \dots \leq A_{n-1} \leq A_n$$

شرط اول را در نظر بگیرید. فرض کنید می‌خواهیم فقط وجود یا عدم وجود این شرط را چک کنیم. برای این کار، یک متغیر  $state$  تعریف می‌کنیم که با آن مسئله را حل کنیم. این متغیر را در هر مرحله که سراغ عنصر جدید می‌رویم، مطابق با زیر مقداردهی می‌کنیم:

۱. حالت  $state = 0$ : مقدار اولیه متغیر ما در واقع صفر است و به این معنا است که تا این مرحله، تمام عناصری که ورودی گرفته بودیم به صورت صعودی بوده‌اند و هیچ‌گاه نزول مشاهده نشده است. پس اگر وقتی در این حالت باشیم و صعود یا برابری در عناصر متوالی ببینیم، در همین حالت می‌مانیم.

۲. حالت  $state = 1$ : می‌خواهیم مفهوم این حالت این باشد که ورودی ما تا یک جایی نزول نکرده است و از آن‌جا به بعد صعود نکرده است. در واقع اگر در حالت  $state = 0$  بودیم و یک نزول دیدیم، باید به این حالت بیایم. همچنین اگر در این حالت بودیم و یک نزول یا برابری در دو عنصر متوالی دیدیم، باید در همین حالت باقی بمانیم.



۳. حالت  $state = 2$ : این حالت را نیز به این شکل تعریف می‌کنیم که اگر وقتی  $state = 1$  بود و دوباره یک صعود دیدیم، به این حالت بیاییم. در واقع اگر به این حالت بیاییم، ورودی ما حتما شرط سوال را نداشته است و ورودی نادرستی بوده است. پس در ادامه نیز باید در همین حالت بمانیم، چون جواب مسئله همچنان منفی خواهد بود.

این سه حالت را با چند شرط می‌توان چک کرد و تغییرات متناظر با آن‌ها را انجام داد. همچنین می‌دانیم که اگر در آخر کار، مقدار  $state$  برابر با ۰ یا ۱ بود، جواب مسئله YES است. چون در حالت ۰ فقط برابری و صعود داشتیم و حالت ۱ هم متناظر با خود شرط تعریف شده است! اما اگر در حالت ۲ بودیم، می‌فهمیم که ورودی شرط مطلوب را نداشته است و جواب مسئله NO است.

حال فرض کنید فقط می‌خواهیم شرط دوم را برای دنباله‌ی ورودی چک کنیم. به طور مشابه با استفاده یک متغیر  $state$  دیگر و حالت‌بندی‌های با همین منطق، می‌توانیم سوال را حل کنیم.

حال اگر دقت کنیم، می‌بینیم که دو راه حل داده شده مستقل از هم هستند و ادغام پذیر هستند! به عبارتی برای شرط اول می‌توان یک متغیر  $state1$  گرفت و برای شرط دوم یک متغیر  $state2$  و با همان الگوریتم‌ها، در یک حلقه‌ی کلی، این دو متغیر را به روز رسانی کرد. در نهایت هم جواب نهایی YES خواهد بود اگر حداقل یکی از این دو مقدار ۲ نداشته باشند. پس مسئله مطابق با کد بالا حل می‌شود.



همین مسئله را با استفاده از گرفتن اختلاف دو عنصر متوالی نیز می‌توان حل کرد. در واقع سوال به این تبدیل می‌شود که آیا دنباله‌ی اختلاف‌ها حداکثر یک باز تغییر علامت داده است یا خیر. با استفاده از این دنباله، الگوریتمی مشابه با الگوریتم بالا با استفاده از متغیرهای *state* می‌توان ارائه داد و عملاً معادل الگوریتم بالا است. با استفاده از ضرب دو اختلاف متوالی و علامت آن نیز می‌توان تعداد تغییر علامت‌های ذکر شده را محاسبه نمود، که به راه حل زیر تبدیل می‌شود. فقط باید کمی به حالتی که ضرب برابر با صفر می‌شود دقت کنید و عناصری که بعداً چک می‌کنید را عوض کنید. با تشکر از شایان صالحی (:)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int t = 0, moves = 0, i = 0;
4 long int n = 0, e = 0;
5 long long int a = 0, a1 = 0, a2 = 0;
6 int main(){
7     scanf("%d", &t);
8     for(i = t; i > 0; i--){
9         scanf("%ld", &n);
10        moves = 0;
11        scanf("%lld%c", &a);
12        if(n != 1){
13            a2 = a;
14            scanf("%lld%c", &a);
15            a1 = a;
16            for(e = n; e > 2; e--){
17                scanf("%lld%c", &a);
18                if((a1 - a2) * (a - a1) < 0){
19                    moves++;
20                    a2 = a1;
21                    a1 = a;
22                } else if((a1 - a2) * (a - a1) == 0){
23                    a1 = a;
24                    a2 = a2;
25                } else {
26                    a2 = a1;
27                    a1 = a;
28                }
29            }
30            if(moves < 2){
31                printf("YES\n");
32            } else {
33                printf("NO\n");
```





```
34         }  
35     } else {  
36         printf("YES\n");  
37     }  
38 }  
39 return 0;  
40 }  
41
```