

## ارجاع

برنامه‌ای که جلسه گذشته برای سوال دوم نوشته بودید را با استفاده از Reference بازنویسی کنید.

## بدون شرط

برنامه‌ای بنویسید که یک عدد صحیح از کاربر دریافت کند و seed تابع rand() را با آن مقداردهی کند. سپس با استفاده از این تابع، یک عدد تصادفی بین 1 تا 6 تولید کرده و بر اساس عدد حاصل، به صورت زیر عمل کند:

اگر عدد حاصل 1 یا 4 باشد، یک واحد به عدد واردشده توسط کاربر اضافه کرده و آن را به همراه عبارت "Go to right" چاپ کند.

اگر عدد حاصل 2 یا 5 باشد، یک واحد از عدد واردشده توسط کاربر کم کرده و آن را به همراه عبارت "Go to left" چاپ کند.

اگر عدد حاصل 3 یا 6 باشد، عدد واردشده توسط کاربر را به همراه عبارت "Do not move" چاپ کند.

توجه کنید که در این برنامه مجاز به استفاده از هرگونه دستورات شرط و حلقه از قبیل if, switch, ?, for, while و ... نیستید.

### ورودی نمونه ۱

15

### خروجی نمونه ۱

Go to right 16

### ورودی نمونه ۲

6

### خروجی نمونه ۲

Go to left 5

## حساب بی حساب

بانک مانی بسیار عجیب است، به طوری که هیچگاه به مشتریانش موجودی حسابشان را نمی گوید و تنها لیستی از تراکنش ها به آن ها می دهد. مانی که موجودی اولیه حساب خودش را فراموش کرده، به بانک می رود و لیست تراکنش های خود را دریافت می کند. هر تراکنش به یکی از صورت های زیر است:

- به حساب مانی پول ریخته شده است، به این معنی است که در ساعت HH:MM موجودی حساب مانی  $x$  تومان افزایش یافته است:

DEP  $x$  HH:MM

- مانی از حسابش پول دریافت کرده اما موجودی کافی نبوده است، به این معنی است که در ساعت HH:MM مانی قصد کرده  $x$  تومان از حسابش بردارد. اما موجودی حسابش کافی نبوده و تراکنش انجام نشده است:

WIT  $x$  HH:MM FAIL

- مانی از حسابش پول دریافت کرده و موجودی کافی بوده است، به این معنی است که در ساعت HH:MM مانی  $x$  تومان از حسابش برداشت کرده و موجودی حسابش  $x$  تومان کاهش یافته است:

WIT  $x$  HH:MM OK

به او کمک کنید حداقل موجودی اولیه ی حسابش را بیابد. دقت کنید همه ی ساعت های داده شده برای یک روز هستند و شما باید حداقل موجودی اولیه حساب مانی را در ابتدای روز بدست آورید.

## ورودی

در خط اول ورودی عدد  $n$  آمده که نشان دهنده ی تعداد تراکنش ها است.

$$1 \leq n \leq 1000$$

در هر خط از  $n$  خط بعدی یک تراکنش آمده که به صورت یکی از ۳ حالت توضیح داده شده در صورت سوال است.

$$1 \leq x \leq 2000$$

$$0 \leq HH \leq 59$$

$$0 \leq MM \leq 59$$

توجه کنید زمان تراکنش‌ها لزوماً به ترتیب نیست.

تضمین می‌شود هیچ دو تراکنشی در یک زمان انجام نمی‌گیرند.

## خروجی

در تنها خط خروجی حداقل موجودی اولیه حساب مانی را چاپ کنید. همچنین اگر لیست تراکنش‌های داده شده معتبر نبود عبارت DOROGHE را چاپ کنید.

## مثال

### ورودی نمونه ۱

```
3
DEP 100 08:57
WIT 500 18:00 FAIL
WIT 150 13:23 OK
```

### خروجی نمونه ۱

50

در این صورت می‌توان ثابت کرد مانی در ابتدا حداقل ۵۰ تومان در حسابش داشته است و موجودی اولیه‌ای کمتر از این مقدار ممکن نیست.

## ورودی نمونه ۲

5

DEP 100 08:57  
WIT 250 18:00 OK  
WIT 150 13:23 OK  
DEP 1000 18:01  
WIT 600 20:07 OK

## خروجی نمونه ۲

300

در این صورت می‌توان ثابت کرد مانی در ابتدا حداقل ۳۰۰ تومان در حسابش داشته است.

## ورودی نمونه ۳

8

DEP 210 00:07  
WIT 170 03:23 OK  
WIT 130 10:00 OK  
DEP 1000 12:00  
WIT 600 16:30 OK  
WIT 1400 18:00 FAIL  
DEP 100 19:00  
WIT 800 19:45 OK

## خروجی نمونه ۳

390

در این صورت می‌توان ثابت کرد مانی در ابتدا حداقل ۳۹۰ تومان در حسابش داشته است.

## ورودی نمونه ۴

2

DEP 100 12:00

WIT 50 13:00 OK

## خروجی نمونه ۴

0

در این مثال حساب مانی می‌تواند در ابتدا خالی باشد.

## ورودی نمونه ۵

2

WIT 100 10:08 FAIL

DEP 200 09:00

## خروجی نمونه ۵

DOROGHE

در ساعت 09:00 ۲۰۰ تومان به حساب مانی ریخته شده است و سپس در ساعت 10:08 ، موجودی او برای برداشت ۱۰۰ تومان کافی نبوده که تناقض است. زیرا در ساعت 10:08 حداقل ۲۰۰ تومان در حسابش وجود داشته و ۱۰۰ تومان قابل برداشت بوده است.

## ویرایشگر متنی

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

ویرایشگر متنی جدید آرپا به نام «new word» اجازه‌ی ساخت و ویرایش تنها یک کلمه را می‌دهد. این ویرایشگر قادر به پردازش حروف a-z و همچنین R (به سمت راست) و L (به سمت چپ) است. هنگامی که آرپا ویرایشگر خود را باز می‌کند، این ویرایشگر یک کلمه‌ی خالی ساخته و نشانگر (cursor) را روی چپ‌ترین (left-most) جایگاه قرار می‌دهد. اگر یکی از کلیدهای a-z فشرده شود، ویرایشگر آن را بلافاصله پس از نشانگر قرار می‌دهد. پس از آن، نشانگر یک جایگاه به سمت راست حرکت می‌کند به طوری که دقیقاً بلافاصله بعد از حرف جدید قرار گرفته باشد. اگر کلید L فشرده شود، نشانگر یک جایگاه به سمت چپ و در صورت فشرده شدن کلید R، نشانگر یک جایگاه به سمت راست حرکت می‌کند. در صورتی که نشانگر توانایی حرکت نداشته باشد، دستور مورد توجه قرار نمی‌گیرد.

شما باید برنامه‌ای بنویسید که بتواند ترتیبی از فشرده شدن کلیدها را در این ویرایشگر پردازش کرده و نتیجه را به صورت یک رشته چاپ کند.

## ورودی

ورودی شامل یک رشته است که از حروف a-z و L و R تشکیل شده است. طول این رشته از 1 کمتر نبوده و از 1000000 بیشتر نیست.

## خروجی

رشته‌ی مورد نظر را در خروجی کنید.

## مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

## ورودی نمونه ۱



abLcd

خروجی نمونه ۱

acdb

ورودی نمونه ۲

icpcLLLLLacmRRRRRRRRRRRc

خروجی نمونه ۲

acmicpcc

## کد سوئیسی

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

بیاین برگردیم به تقریباً صدسال پیش، اون موقع که هیتلر و چرچیل و استالین داشتن دورهم میجنگیدن(!)، که الان به اسم جنگ جهانی دوم ازش یاد میکنیم!

اون موقع ارتش سوئیس، یه چاقو ی باحال داشتن که بهش میگفتن Offiziersmesser (تونستی بخونی اسمشو؟) تلفظ این کلمه خیلی سخت بود برای همین آمریکایی ها یه اسم خیلی ساده روش گذاشتن، بهش میگفتن "Swiss Army Knife" (:).

"چاقو ارتش سوئیس" یک چاقو جیبی همه کاره اس که یه عالمه وسیله های مختلف مثل تیغ، اهر چوب، قیچی، سیم بر، قوطی باز کن و ... تو خودش جا داده.



حالا بیاین برگردیم به زمان حال، که جنگ جهانی سوم شده و اصغر، سرباز شجاع و فداکار، میخواد از کشورش محافظت کنه و برای این کار به کمک ما نیاز داره. اون باید یه برنامه ای بنویسه که مثل چاقو ی همه کاره برای رشته ها (Strings) عمل کنه.

یعنی باید یه سری رشته رو از کاربر بگیره و با توجه به اینکه کاربر چه آپشنی رو انتخاب کرده یه سری کارا رو براش بکنه.

آپشن هایی که چاقو ی ما داره چیا هستن؟

#### ۱. Palindrome (-p, --palindrome) :

ما به رشته ای میگیم palindrome که قرینه باشه، یعنی از هر طرفی بخونیش همون بشه!

مثلا رشته های "madam" و "racecar" و "bcb" پالیندروم هستن ولی رشته هایی مثل "milk" و "a bh" "hba" پالیندروم نیستند.

#### ۲. Find (-f, --find):

دو تا رشته بگیره ، ببینه رشته اول تو دومی هست یا نه؟ (توضیح بیشتر در مثال های پایین هست)

#### ۳. Reverse (-r, --reverse):

یه رشته بگیره و برعکسش کنه . مثلا وقتی رشته "reverse123" رو میگیره ، رشته "321esrever" رو خروجی بده.

#### ۴. Sort (-s, --sort) :

چنتا رشته بگیره و اونا رو برحسب الفبا مرتب کنه. مثلا وقتی رشته های "amain" و "fabric" و "bill" و "amnesia" رو میگیره باید این ترتیب رو خروجی بده: amain , amnesia , bill , fabric

#### ۵. Concatenate (-c, --concatenate):

تمام رشته های ورودی را به هم می چسباند و یک رشته واحد درست میکند.

#### ۶. Help (-h, --help):

اگر به برنامه این دستور داده شود برنامه تمام آپشن هایی (palindrome, find, ...) که دارد را برای کاربر

چاپ میکند.

## ورودی و خروجی

کاربر یکی از آپشن های بالا را قبل از اجرای برنامه انتخاب کرده و رشته های خود را به صورت زیر به برنامه میدهد. دقت کنید که آپشن هایی که کد ما داره همون موقع اجرای برنامه بهش داده میشن (نه در زمان اجرا !)

دقت کنید که هر کدام از آپشن ها بجز find میتوانند تعداد مختلفی رشته (string) ورودی داشته باشند.

مثال از help:

ورودی:

```
>>./code.out -h
OR
>>./code.out --help
```

خروجی:

```
-h, --help      prints all commands in terminal.
-p, --palindrome checks if input string is palindrome.
-c, --concatenate concatenates all the input strings.
-s, --sort      sorts all of the input strings in alphabetic order.
-f, --find      find a substring in a string.
-r, --reverse   reverse all the input strings.
```

مثال از palindrome:

ورودی:

```
>>./code.out -p "aba" "madam" "car"  
OR  
>>./code.out --palindrome "aba" "madam" "car"
```

خروجی:

```
"aba" is palindrome.  
"madam" is palindrome.  
"car" is not palindrome.
```

مثال از sort:

ورودی:

```
>>./code.out -s "aba" "madam" "car"  
OR  
>>./code.out --sort "aba" "madam" "car"
```

خروجی:

```
aba  
car  
madam
```

مثال از Reverse:

ورودی:

```
>>./code.out -r "quera" "racecar" "car"  
OR  
>>./code.out --reverse "quera" "racecar" "car"
```

خروجی:

```
areuq  
racecar  
rac
```

مثال از find:

ورودی:

```
>>./code.out -f "happy" "I am happy"  
OR  
>>./code.out --find "happy" "I am happy"
```

خروجی:

```
"happy" was found in "I am happy"
```

مثال از concatenate:

ورودی:

```
>>./code.out -c "I " "like " "cats."  
OR  
>>./code.out --concatenate "I " "like " "cats."
```

خروجی:

```
I like cats.
```