

## سوال ۱

پید اکردن بهینه‌ترین شهر برای افتتاح مرکز پست

شرکت پست می‌خواهد در یکی از شهرهای کشور یک مرکز پست افتتاح کند. اگر فرض کنیم نقشه کشور به صورت یک خط راست با طول  $n - 1$  است که  $n$  شهر با فاصله‌ی مساوی روی آن قرار گرفته‌اند، شرکت پست برای انتخاب محل برای این مرکز، دنبال شهری است که مجموع فاصله‌ی آن تا مردم کشور کمترین مقدار باشد (مردم برای مراجعه به این مرکز در مجموع فاصله‌ی کمتری را باید طی کنند). حالا شما قرار است به شرکت پست در انتخاب شهر کمک کنید. برنامه‌ای بنویسید که با گرفتن تعداد شهر به عنوان ورودی اول و جمعیت هر شهر به ترتیب به عنوان ورودی‌های بعدی، شماره شهری که باید مرکز پست در آن افتتاح شود را پیدا کرده و به عنوان خروجی چاپ کند. در صورتی که چند شهر فاصله‌ی یکسانی داشته باشند، شهر با شماره کوچکتر در اولویت است. ورودی‌ها را با `scanf` گرفته و خروجی را با `printf` چاپ کنید.

## ورودی‌ها

عدد صحیح  $n$  که تعداد شهرها را نشان می‌دهد. حداکثر تعداد شهرها ۱۰۰۰ شهر است.

لیستی از  $n$  عدد صحیح که جمعیت هر شهر را به ترتیب نشان می‌دهد. حداکثر جمعیت یک شهر ۱۵ میلیون نفر است.

## خروجی‌ها

یک عدد صحیح که شماره شهر انتخاب شده در لیست خوانده شده را نشان می‌دهد. دقت کنید شماره از یک شروع می‌شود و نه صفر.

## مثال

در اینجا چند نمونه ورودی و خروجی برای فهم بهتر صورت سوال و قالب ورودی و خروجی آورده شده است.

ورودی نمونه ۱

6

3 1 0 0 2 2

خروجی نمونه ۱

2

ورودی نمونه ۲

12

12 10 8 1 2 3 4 5 6 7 8 1

خروجی نمونه ۲

6

## سوال ۲

بدست آوردن اجتماع دو مجموعه با استفاده از اشتراک آنها

تابع زیر را تکمیل کنید. این تابع اشتراک دو مجموعه  $a$  و  $b$  را محاسبه می‌کند. تعداد اعضای نتیجه با  $size\_result$  مشخص می‌شود و اشاره‌گر به مجموعه‌ی نتیجه  $return$  می‌شود. دقت کنید که نمی‌توانید آدرس شروع یک آرایه را  $return$  کنید چون آرایه یک متغیر  $automatic$  است که در  $stack$  قرار می‌گیرد و با تمام شدن تابع از بین می‌رود. یک برنامه بنویسید که اطلاعات دو مجموعه را از کاربر بگیرد و با این تابع اجتماع آنها را محاسبه کرده و نتیجه را چاپ کند. توجه کنید که حتماً باید تابع اشتراک را پیاده‌سازی کرده و اجتماع را با استفاده از این تابع بدست آورید و نه از روش‌های دیگر. عدم انجام پیاده‌سازی‌های خواسته شده حتی اگر تمام تست کیس‌ها را به درستی پاس کرده باشید باعث از دست رفتن بیشتر نمره‌ی شما خواهد شد.

```
int * intersection(int *a, int size_a, int *b, int size_b, int *size_result)
```

## ورودی‌ها

دو عدد صحیح  $m$  و  $n$  که به ترتیب نشان‌دهنده‌ی تعداد اعداد در مجموعه‌ی اول و مجموعه‌ی دوم هستند. در ادامه ابتدا  $m$  عدد صحیح مربوط به مجموعه‌ی اول و سپس  $n$  عدد صحیح مربوط به مجموعه‌ی دوم وارد خواهد شد.

## خروجی‌ها

یک دنباله از اعداد صحیح که نشان‌دهنده‌ی اجتماع دو مجموعه است. در این دنباله اعداد پشت سر هم در یک خط چاپ خواهند شد و بین هر دو عدد فقط یک فاصله وجود دارد. در خروجی ابتدا باید اعداد مربوط به مجموعه‌ی اول چاپ شوند و سپس اعداد انتخاب شده از مجموعه‌ی دوم.

## مثال

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

## ورودی نمونه ۱

7 6  
1 2 3 4 5 6 7  
6 9 3 100 5 1000

## خروجی نمونه ۱

1 2 3 4 5 6 7 9 100 1000

## ورودی نمونه ۲

8 10  
70 12 9 45 17 26 87 14  
32 12 70 9 410 15 27 13 16 18

## خروجی نمونه ۲

70 12 9 45 17 26 87 14 32 410 15 27 13 16 18

## سوال ۳

شمارش تعداد تکرار کلمات یکتا در لیستی از رشته‌ها

تابع زیر را تکمیل کنید. این تابع یک آرایه از رشته (تعداد رشته‌ها برابر *rows* و حداکثر طول هر رشته برابر *len* است و همه اعضای آن حروف کوچک هستند) و یک آرایه از *struct count* را به عنوان ورودی می‌گیرد، سپس این رشته‌ها را بررسی کرده و تعداد دفعات تکرار هر کلمه را پیدا کرده و آنها را در آرایه *result* ذخیره کرده و در نهایت تعداد کلمات غیر تکراری را برمی‌گرداند. حداکثر تعداد کلمات یکتا در رشته‌ها ۱۰۰ تا در نظر گرفته شود. برای سادگی کار رشته‌ها فقط شامل کلمات هستند و هیچ کاراکتر اضافه‌ای از جمله علائم نگارشی مثل نقطه و کاما در آنها وجود ندارد.

```
struct count{char word[20]; int cnt;};  
int frequency(int rows, int len, char strings[rows][len], struct count result []);
```

برای مثال اگر رشته‌ها به شکل زیر باشد:

```
this is a test string  
test string is for test  
test is test
```

این تابع عدد 6 را برمی‌گرداند و آرایه *result* به شکل زیر خواهد بود:

```
{this 1} {is 3} {a 1} {test 5} {string 2} {for 1}
```

برنامه‌ای بنویسید که در ابتدا *rows* و *len* را از کاربر بگیرد، بعد از آن، به تعداد *rows* رشته را از کاربر گرفته و کلمات غیر تکراری و دفعات تکرار آنها را بنویسید. ترتیب کلمات در خروجی به ترتیب رخداد آنها در رشته‌ها است و هر کلمه و تعدادش در یک خط باید چاپ شود که با یک کاراکتر فاصله از هم جدا شده‌اند.

ورودی‌ها

دو عدد صحیح *rows* و *len* که به ترتیب نشان‌دهنده‌ی تعداد خطوط و حداکثر طول یک خط هستند.

در ادامه به تعداد *rows* خط از ورودی دریافت خواهد شد که طول هر کدام حتماً از *len* کوچکتر است.

## خروجی‌ها

یک دنباله از کلمات و تعداد تکرار آنها. هر کلمه باید در یک خط چاپ شود و بین کلمه و تعداد تکرارش یک فاصله باید وجود داشته باشد. ترتیب کلمات در خروجی به ترتیب رخداد آنها در رشته‌ها است. حداکثر تعداد کلمات یکتا نیز ۱۰۰ عدد در نظر گرفته شود.

## مثال

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

### ورودی نمونه ۱

```
3 30
this is a test string
test string is for test
test is test
```

### خروجی نمونه ۱

```
this 1
is 3
a 1
test 5
string 2
for 1
```

### ورودی نمونه ۲

3 70

biden signs executive orders to expand health care access

biden uses executive authority to address health care climate change

how the biden administration should stand up for abortion rights

خروجی نمونه ۲

biden 3

signs 1

executive 2

orders 1

to 2

expand 1

health 2

care 2

access 1

uses 1

authority 1

address 1

climate 1

change 1

how 1

the 1

administration 1

should 1

stand 1

up 1

for 1

abortion 1

rights 1