سکه های سزار

- محدودیت زمان پایتون: ۳ ثانیه
- محدودیت زمان جاوا : ۶ ثانیه
- محدودیت زمان سی و سی پلاس پلاس: ۰.۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

گایوس ژولیوس سزار برای حمله به گالیا به کمک شما نیاز دارد. او قصد دارد تا با خرج کردن کم‌ترین میزان پول جنگ را پیروز شود. او که می‌داند شما خیلی از ساز و کار جنگ سر در نمی‌آورید مسئله را برای شما اینگونه ساده کرده است.

او در کل n سرباز دارد که قدرت هر کدام را با a_i نشان می‌دهد. همچنین m دشمن وجود دارد که باید از شکست داده شوند. قدرت حمله و قدرت دفاع هر یک از m دشمن را با ea_i و ed_i نشان می‌دهد. شما باید از بین سرباز های سزار، سربازی را برای شکست دادن سرباز حریف انتخاب کنید. شرط اینکه سرباز انتخاب شده بتواند دشمن را شکست دهد این است که قدرت حمله‌اش از قدرت دفاع سرباز حریف بیشتر و یا مساوی باشد. همچنین برای احتیاط لازم است مجموع قدرت سربازهای باقی مانده، بیشتر یا مساوی قدرت حمله‌ی دشمن باشد. شما می‌توانید با خرج کردن یک سکه، قدرت هر کدام از سربازهایتان را یک واحد افزایش دهید سزار از شما می‌خواهد که کم‌ترین میزان سکه‌ای که برای شکست دادن هر دشمن باید خرج کند را به او بگویید.

نکات مهم

پیشنهاد می‌شود که برای تمارین این درس، از زبان‌های C و C++ استفاده کنید. اما ممانتعی برای استفاده از سایر زبان‌ها وجود ندارد.

برای این که به مشکل `Time Limit Exceeded` بر نخورید، لازم است از چیزی تحت عنوان **Fast I/O** استفاده کنید! این موضوع را برای هر کدام از زبان‌های `C++` و پایتون در ادامه شرح می‌دهیم.

C++

در این زبان، برای ورودی گرفتن و خروجی دادن سریع، می‌توانید از `scanf` و `printf` استفاده کنید. اما اگر خواستید که از `cin` و `cout` استفاده کنید، لازم است در اول تابع `main` خود، کد زیر را استفاده کنید:

```
ios_base::sync_with_stdio(false);
cin.tie(0);
```

همچنین، حواستان باشد که برای خروجی دادن از `endl` استفاده نکنید و به جای آن از `"\n"` استفاده کنید.

جاوا

در این زبان، برای ورودی گرفتن و خروجی دادن سریع، می‌توانید از `BufferedReader` استفاده کنید. برای استفاده از `BufferedReader` ، ابتدا باید کتابخانه‌ی `java.io.BufferedReader` را به برنامه‌ی خود اضافه کنید. سپس، با استفاده از دستور زیر یک `BufferedReader` تعریف کنید که ورودی را بخواند.

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

در ادامه، برای خواندن یک خط از ورودی، از دستور `br.readLine()` استفاده کنید. برای اطلاعات بیشتر در رابطه با روش‌های دیگر **Fast I/O** در جاوا، به این لینک مراجعه کنید.

پایتون

زمانی که کد خود را تست می‌کنید، مثل حالت عادی از تابع `input()` برای ورودی گرفتن استفاده کنید. اما زمانی که می‌خواهید کد خود را بفرستید، اول کد خود دو خط زیر را اضافه کنید:

```
import io, os
input = io.BytesIO(os.read(0,os.fstat(0).st_size)).readline
```

توجه کنید که اضافه کردن این دو خط باعث می‌شود که دیگر در کنسول نتوانید به برنامه‌ی خود ورودی دهید، برای همین فقط زمان ارسال فایل خود از این دو خط استفاده کنید.

ورودی

در خط اول ورودی n تعداد سربازان سزار می‌آید. در خط دوم قدرت هر یک از سربازان a_i با فاصله می‌آید. قدرت سربازان از **کوچک به بزرگ** مرتب شده است.
در خط سوم m تعداد دشمنان می‌آید و در m خط بعدی به ترتیب قدرت دفاع ed_i و حمله‌ی ea_i هر یک از m دشمن می‌آید.

$$1 \leq m, n, a_i \leq 2 * 10^5$$

$$1 < n$$

$$1 \leq ea_i, ed_i \leq 10^{12}$$

خروجی

به ترتیب در m خط برای هر دشمن کم‌ترین میزان سکه ای که سزار باید خرج کند را چاپ کنید. توجه کنید که هر دشمن را جدا در نظر می‌گیریم و هزینه‌هایی که دشمن‌های قبلی کرده‌ایم در دشمن‌های بعدی تأثیری نخواهد داشت.

مثال

ورودی نمونه

```
8
2 2 4 5 11 16 17 20
3
12 70
```

```
2 30
23 20
```

خروجی نمونه

```
5
0
3
```

دقت کنید که برای شکست دادن هر دشمن قدرت حمله ی سربازها را همان قدرت داده شده در ورودی اولیه بگیرید. در مثال داده شده مجموع قدرت ۸ سرباز ۷۷ است.

برای شکست دادن دشمن اول، باید سربازی با قدرت بیشتر یا مساوی ۱۲ را انتخاب کنیم. سرباز با قدرت ۱۱ را انتخاب و با خرج ۱ سکه قدرت او را افزایش می دهیم. حال قدرت سربازهای باقی مانده ۶۶ است. پس لازم است ۴ سکه خرج کنیم تا قدرت سرباز های باقی مانده بیشتر یا مساوی قدرت دشمن بشود. پس در کل حداقل ۵ سکه باید خرج کنیم.

برای شکست دادن دشمن دوم کافیست سرباز با قدرت ۴ را بفرستیم و با توجه به اینکه سرباز های باقی مانده قدرتشان از قدرت حمله ی دشمن بیشتر است نیازی به خرج سکه نداریم.

برای شکست دادن دشمن سوم کافیست سرباز با قدرت ۲۰ را بفرستیم و قدرت او را ۳ تا افزایش دهیم. پس حداقل ۳ سکه خرج می کنیم.

مالیات

- محدودیت زمان: ۰.۲ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

بعد از سوال قبل خزانه روم کمی خالی شده است. به همین دلیل سزار مالیات جدیدی را برای تاجران معرفی می‌کند. این نوع مالیات اینگونه است که درصدی از بیشترین سود یک تاجر در یک بازه متوالی به عنوان مالیات از او گرفته می شود. حال سزار از شما می‌خواهد با گرفتن میزان سود یا ضرر یک تاجر بیشترین سود او در یک بازه متوالی گزارش دهید.

توجه کنید که تست‌های این سؤال برای اردرهای زمانی مختلف طرح شده‌اند. برای مثال، راه حلی با اردر $O(n^3)$ فقط سه تست اول سؤال را می‌گیرد و راه حلی با اردر $O(n^2)$ فقط 6 تست اول سؤال را می‌گیرد.

ورودی

در خط اول ورودی تعداد روزهایی که قرار است سود و ضرر در آن داده شود و در ادامه آرایه‌ی درصد سود و ضررها در این روزها گرفته می‌شود.

تضمین می‌شود که بازه‌ای وجود دارد که تاجر در آن سود کرده باشد.

$$1 \leq n \leq 10^5$$

خروجی

در خروجی شما باید میزان بیشترین سود را بیان کنید. به ورودی و خروجی نمونه دقت کنید.

مثال

ورودی نمونه ۱

```
12
7 -1 -2 1 5 -11 9 1 4 -1 3 -10
```

خروجی نمونه ۱

```
16
```

توضیح خروجی: بیشترین سود تاجر در روزهای ۷ تا ۱۱ است که مجموع اعداد شماره ۷ تا ۱۱ برابر ۱۶ است.

باران در روم

- محدودیت زمانی پایتون: ۷ ثانیه
- محدودیت زمانی سی و سی پلاس پلاس: ۲ ثانیه
- محدودیت زمانی جاوا: ۳ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

گایوس ژولیوس سزار برای محاسبه میزان آب ذخیره شده از باران سالانه از شما کمک می‌خواهد تا برای او برنامه‌ای طراحی کنید. یک خیابان در روم باستان قرار دارد که از ساختمان های بلند در یک ردیف ساخته شده است و عرض هر ساختمان دقیقاً 1متر است . شما باید برنامه‌ای طراحی کنید که حساب کند موقع بارش باران، دقیقاً چه مقدار آب روی بام ساختمان‌ها باقی می‌ماند. ساختمان ها از راست به چپ به هم چسبیده‌اند.

ورودی

در سطر اول ورودی عدد طبیعی n (تعداد ساختمان‌ها) آمده است.

در سطر بعد n عدد آمده است که به ترتیب ارتفاع ساختمان‌ها را از راست به چپ مشخص می‌کنند و با فاصله از هم جدا شده‌اند.

ارتفاع هر ساختمان حداکثر ۱۰۰۰ متر خواهد بود.

$$1 \leq n \leq 1\,000\,000$$

خروجی

در تنها خط خروجی حداکثر میزان آب جمع شده روی سقف ساختمان‌ها (بر حسب متر مربع) بنویسید.

مثال

ورودی نمونه

```
7
4 1 3 5 2 3 4
```

خروجی نمونه

```
7
```