

جمع باستانی

- محدودیت زمان: ۱ ثانیه

- محدودیت حافظه: ۲۵۶ مگابایت

علیرضا با شروع تعطیلات تابستانی ترم جدید قصد دارد ماجراجویی کند. او با تلاش بسیار فراوان، پورتال گان «ریک سانچز» را پیدا می‌کند. او با سفر به دنیاهای مختلف، با سرزمینی جالب آشنا می‌شود. هیولاهای ساکن در این سرزمین، برای نمایش اعداد از انگشتان دست‌های خود کمک می‌گیرند. آن‌ها برای نمایش عدد صفر، تمام انگشت‌های خود را بسته نگه می‌دارند و برای نمایش اعداد بزرگ‌تر از صفر، با شروع از صفر و انگشت‌های بسته، به ازای هر واحد افزایش، یکی از انگشت‌های خود را باز می‌کنند. برای مثال برای نمایش عدد ۳، ابتدا یک انگشت را باز کرده تا عدد یک را نمایش دهند و سپس یک انگشت دیگر را باز کرده تا عدد دو را نمایش دهند و در انتها انگشت سوم را باز کرده تا عدد سه را نمایش دهند. در واقع در انتها سه انگشت خود را باز کرده‌اند.

در هنگام شمارش، اگر به حالتی برسند که تمام انگشت‌هایشان باز بوده و هم چنان باید عدد بزرگ‌تری را بشمارند، به اندازه‌ی تعداد انگشتان خود از عدد کم نموده و انگشتان خود را کامل می‌بندند. سپس از ابتدا شروع به شمارش می‌کنند. برای مثال اگر تعداد کل انگشتان هیولاها سه باشد، عدد سه را با همان سه انگشت نشان می‌دهند، ولی برای نمایش عدد چهار بعد از اینکه سه انگشت خود را باز کرده‌اند، تمام آن‌ها را می‌بندند و سپس یک انگشت را باز می‌کنند. در واقع در صورتی که هیولاها سه انگشت داشته باشند، عدد چهار را با یک انگشت نمایش می‌دهند.

به هنگام جمع نمودن دو عدد نیز، هیولاهای این سرزمین ابتدا با انگشت‌های خود عدد اول را نمایش داده و سپس عدد دوم را به وسیله انگشتان به عدد اول اضافه می‌کنند.

علیرضا که تعداد دست و انگشتانش همانند هیولاها نیست، نمی‌تواند همانند آن‌ها جمع اعداد را محاسبه کند. به او کمک کنید با دانستن تعداد دست‌ها و انگشت‌های یک هیولا، جمع دو عدد را محاسبه کند.

ورودی

در خط اول تعداد انگشتان یک دست ($Fingers$)، در خط دوم تعداد دست‌ها ($Hands$) و در خطوط سوم و چهارم دو عدد a و b که باید با یک‌دیگر جمع شوند، آمده‌اند.

$$1 \leq Fingers, Hands \leq 10000$$

$$0 \leq a, b \leq 10000$$

خروجی

در تنها خط خروجی، عدد نمایش داده شده با دست هیولاها (تعداد انگشت‌های باز) پس از عملیات جمع را چاپ کنید.

مثال

ورودی نمونه ۱

3
4
37
27

خروجی نمونه ۱

4

ورودی نمونه ۲

3

2

3

3

خروجی نمونه ۲

6

ورودی نمونه ۳

4

5

0

0

خروجی نمونه ۳

0

بیت‌وایز

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

دو عدد **مثبت** a و b وجود دارند به طوری که مجموع آنها برابر s و حاصل XOR آنها برابر x می‌باشد.

عدد s در مبنای p و عدد x در مبنای q است.

برنامه‌ای بنویسید که تعداد تمام حالت‌های ممکن برای دوتایی مرتب (a, b) را چاپ کند.

دقت کنید که دوتایی (a, b) دارای **ترتیب** است و با عوض شدن جای a و b ، یک دوتایی جدید ساخته می‌شود. مثلاً $(6, 3)$ و $(3, 6)$ دو حالت متفاوت‌اند.

راهنمایی : جاوا توابعی برای خواندن ورودی در مبناهای مختلف دارد. می‌توانید از آنها استفاده کنید.

ورودی

در خط اول، به ترتیب اعداد p و q و در خط دوم به ترتیب مقادیر s و x می‌آیند که همگی با فاصله از هم جدا شده‌اند.

$$2 \leq p, q \leq 16$$

$$1 \leq a, b \leq s - 1$$

$$0 \leq x \leq 10^9$$

$$2 \leq s \leq 10^9$$

خروجی

در خروجی یک عدد چاپ کنید که تعداد دوتایی‌های (a, b) را نشان می‌دهد. اگر هیچ جوابی وجود نداشت، صفر چاپ کنید.

مثال

ورودی نمونه ۱

8 10
11 5

خروجی نمونه ۱

4

در این نمونه ورودی، می‌توان دوتایی‌های را که مجموع آنها ۹ و XOR شان ۵ است، نوشت:

$$\{(2, 7), (3, 6), (6, 3), (7, 2)\}$$

ورودی نمونه ۲

2 8
10 2

خروجی نمونه ۲

0

حالت بهینه

مورب+ محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۵۰ مگابایت

علیرضا به شما تعداد متغیرها و حداکثر مقداری را که لازم است در خود نگه دارند، می‌دهد و شما باید حساب کنید برنامه او در بهترین حالت چند بایت حافظه مصرف می‌کند. لطفاً به مثال‌ها و توضیح آن‌ها دقت کنید تا مسئله برایتان روشن‌تر شود. راهنمایی: درباره توابعی مثل hasNextInt در کلاس Scanner تحقیق کنید و از آنها استفاده کنید.

ورودی

در خط اول عدد n می‌آید که نشان‌دهنده‌ی تعداد متغیرها است.

در n خط بعدی در هر خط عدد a_i می‌آید که نشان‌دهنده حداکثر مقدار متغیر i ام است. عددها می‌توانند منفی، اعشاری یا صحیح باشند.

خروجی

در خروجی شما باید حداقل تعداد بایت‌های مصرفی ممکن را چاپ کنید.

مثال

ورودی نمونه ۱

3
5
-128
-1.2

خروجی نمونه ۱

6

عدد 5 و 128 – در متغیری از جنس byte جا می‌شوند.

عدد 1.2 – نیز در float جا می‌شود.

هر متغیر byte یک بایت حافظه می‌گیرد و هر float، چهار بایت؛ پس این برنامه حداقل به ۶ بایت حافظه نیاز دارد.

ورودی نمونه ۲

4
30000
1234567890
-654.321
a

خروجی نمونه ۲

12

عدد اول از جنس short، دومی از جنس int، سومی float و چهارمی از جنس char است.

یونیکد

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

برنامه‌ایی بنویسید که یک کاراکتر بگیرد و در ابتدا شماره‌ی unicode آن را چاپ کند؛ سپس، اگر کاراکتر فارسی بود persian و در غیر این صورت، not-persian را چاپ کند.

جدول unicode را میتوانید از [اینجا](#) پیدا کنید.

امتیازی: برنامه‌ی شما بتواند شماره‌ی یونیکد کاراکترهایی که بیشتر از دو بایت حجم می‌گیرند (مثلاً ایموجی‌ها) را هم چاپ کند. به مثال ۳ توجه کنید (۲۵ نمره)

ورودی

یک کاراکتر در ورودی داده می‌شود.

خروجی

در خط اول شماره‌ی یونیکد آن را چاپ کنید.

در خط دوم، اگر کاراکتر فارسی بود persian وگرنه not-persian چاپ کنید.

مثال

ورودی نمونه ۱

A

خروجی نمونه ۱

65
not-persian

ورودی نمونه ۲

پ

خروجی نمونه ۲

1662
persian

ورودی نمونه ۳ (امتیازی)



خروجی نمونه ۳ (امتیازی)

128514

not-persian

عدد فوق اول

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

عدد فوق اول عددی است که هر زیر عدد سمت چپ آن اول باشد. برای مثال ۷۳۳۱ فوق اول است، چون اعداد ۷ و ۷۳ و ۷۳۳ و ۷۳۳۱ **اول** هستند. اولین عدد فوق اول، عدد ۲ است.

برنامه‌ای بنویسید که n امین عدد فوق اول را در خروجی چاپ کند.

ورودی

در تنها خط ورودی عدد n داده می‌شود.

$$1 \leq n \leq 70$$

خروجی

در یک خط، n امین عدد فوق اول را چاپ کنید.

مثال

نمونه ورودی ۱

5

نمونه خروجی ۱

23

نمونه ورودی ۲

20

نمونه خروجی ۲

373

ساعت شنی (امتیازی)

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این سوال یک آرایه $n * n$ می‌سازیم. نحوه‌ی ساخت آرایه در جاوا به صورت زیر است:

```
1 | int[][] twoDimArray = new int[n][n];
```

نحوه‌ی استفاده از آرایه هم دقیقا طبق روال سابق در زبان C++ می‌باشد.

حالا که با آرایه در جاوا آشنا شدید، شما را با ساعت شنی آشنا می‌کنیم. هر الگو از اعداد به شکل زیر یک ساعت شنی می‌باشد:

```
1 | a b c
2 |   d
3 | e f g
```

زمان ساعت شنی را تعریف می‌کنیم: جمع تمامی اعداد موجود در آن ساعت شنی. برای مثال آرایه‌ی زیر را در نظر بگیرید:

```
1 | 1 1 1 0 0 0
2 |
3 |
```



```

0 1 0 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0

```

در این آرایه تعداد زیادی ساعت شنی با زمان صفر وجود دارد و سه تا از آن‌ها که زمانشان صفر نیست، به شکل زیر هستند:

1		1 1 1	1 1 0	1 0 0
2		1	0	0
3		1 1 1	1 1 0	1 0 0

زمان ساعت‌های شنی بالا از چپ به راست به ترتیب 7 و 4 و 2 است.

حال وظیفه‌ی شما این است که یک آرایه در ورودی بگیرید و ساعتی که بیشترین زمان را دارد پیدا کرده و زمان آن را چاپ کنید.

ورودی

در ورودی ابتدا عدد n داده می‌شود. سپس در n خط بعدی، در هر خط دقیقا n عدد با فاصله می‌آیند.

$$3 \leq n \leq 100$$

$$-9 \leq a_{ij} \leq 9$$

خروجی

بیشترین زمان را بین ساعت‌های شنی موجود در آرایه، چاپ کنید.

مثال

ورودی نمونه ۱

6

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
0 0 0 2 0 0
0 0 1 2 4 0
```

خروجی نمونه ۱

19

برای مثال داده‌شده، ساعت شنی با بیشترین زمان به صورت زیر است.

```
1 | 2 4 4
2 |
```

2
1 2 4

چوببری (امتیازی)

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

جواد در یک چوب بری کار می‌کند. کار او این است که هر روز به جنگل می‌رود و یک درخت به طول n را قطع می‌کند و به چوب‌بری می‌آورد، سپس درخت را به قطعات کوچک‌تری تقسیم می‌کند و آن قطعات را می‌فروشد.

هر تکه چوب با توجه به طولش قیمت متفاوتی دارد؛ مثلاً چوبی به طول یک، ۱ تومان است و چوبی به طول ۳، ۸ تومان. جدول قیمت چوب‌ها با توجه به طولش، به شما داده می‌شود.

طول درختی که جواد قطع کرده است به شما داده می‌شود. شما باید بگویید جواد از این درخت **حداکثر** چقدر پول می‌تواند در بیاورد. دقت کنید که ممکن است اصلاً لازم به قطعه‌قطعه کردن چوب نباشد.

ورودی

در ورودی ابتدا عدد m که اندازه‌ی جدول قیمت‌ها است، داده می‌شود. سپس در خط بعدی، m عدد آمده است که عدد i ام نشان‌دهنده‌ی قیمت چوبی به طول i است. در خط بعدی عدد n که طول چوب بریده‌شده توسط جواد است، می‌آید.

$$1 \leq m \leq 500$$

$$1 \leq n \leq m$$

$$1 \leq price(i) \leq 1000$$

خروجی

در خروجی، ابتدا ماکسیمم پولی که جواد می‌تواند به دست بیاورد را چاپ کنید.

در خط بعدی، به هر ترتیبی که دوست دارید، اندازه‌ی قطعاتی که جواد باید چوب را به اندازه‌ی آن‌ها برش دهد، با فاصله از هم چاپ کنید.

مثال

ورودی نمونه ۱

```
10
1 5 8 9 10 17 17 20 24 30
4
```

خروجی نمونه ۱

```
10
2 2
```

در این مثال، جدول قیمت‌ها با توجه به ورودی به صورت زیر است:

طول	قیمت
۱	۱
۲	۵
۳	۸
۴	۹
۵	۱۰
۶	۱۷
۷	۱۷
۸	۲۰
۹	۲۴
۱۰	۳۰

همان‌طور که مشاهده می‌کنید به جای فروختن خود چوب با طول چهار (به قیمت ۹)، می‌توانیم آن را به دو قسمت مساوی به طول دو تقسیم کنیم و به قیمت ۱۰ تومان

بفروشیم.

جمع همه‌ی زیربازه‌ها (امتیازی)

- محدودیت زمان: ۱۰۰ میلی‌ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برنامه‌ای بنویسید که عدد n و سپس یک دنباله n -تایی $a_1, a_2, a_3, \dots, a_n$ را از ورودی بخواند و سپس مقدار زیر را چاپ کند:

$$\sum_{1 \leq l \leq r \leq n} f(l, r)$$

که $f(l, r)$ را این‌گونه تعریف می‌کنیم:

$$f(l, r) = \sum_{i=l}^r a_i$$

ورودی

در سطر اول ورودی یک عدد n آمده است و در سطر دوم n عدد طبیعی آمده است که عدد i -ام نمایان‌گر a_i است.

$$1 \leq n \leq 500\,000$$

$$1 \leq a_i \leq 10$$

خروجی

برنامه‌ی شما باید تنها یک خروجی چاپ کند که برابر مقدار گفته شده است.

زیرمسئله‌ها

محدودیت	نمره	زیرمسئله
$n \leq 100$	۲۰	۱
$n \leq 4\,000$	۳۰	۲
بدون محدودیت اضافی	۵۰	۳

مثال

ورودی نمونه

خروجی نمونه

3
1 2 3

20

$$f(1,1) = 1, f(1,2) = 3, f(1,3) = 6, f(2,2) = 2, f(2,3) = 5, f(3,3) = 3$$

$$\rightarrow ans = 1 + 3 + 6 + 2 + 5 + 3 = 20$$