

A- ای وای حمله!

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



۲۱ سال پیش هنگامی که سیاره‌ی کریپتون در شرف نابودی بود، *Jor-El* و همسرش *Kara*، موفق شدند پسرشان که تازه به دنیا آمده بود را در یک موشک بگذارند تا از مرگ نجات پیدا کند. این موشک نهایتاً راه خود را به زمین پیدا می‌کند و روبه‌روی دانشکده‌ی فنی قدیم سقوط می‌کند؛ جایی که سرنوشت این پسر تا آخر عمرش به آن گره می‌خورد؛ همان جایی که خانواده‌ی *Kent* پسر بچه را می‌یابند و تصمیم می‌گیرند او را بزرگ کنند و نام او را متین می‌گذارند.

اما متین یک بچه‌ی متفاوت بود. او در ۴ سالگی اولین *Hello world!* خود را چاپ کرد و ۶ سالگی می‌توانست به راحتی از یک *segment tree* استفاده کند. خانواده‌ی *Kent* نیز توانایی‌های او را دیدند. آن‌ها متین را با عشق و محبت بزرگ کردند و چهارچوب اخلاقی او را شکل دادند. آن‌ها به او یاد دادند که چطور از توانایی‌هایش در راستای بشریت استفاده کند.

بنابراین او ۱۰ سال پیش وارد شرکت پیام‌پرداز شد تا محصولی طراحی کند که پایش مستمر و پیوسته کلیه حمله‌های موجود در شبکه سازمان را فراهم کند *HOW COOL IS THAT!* دی

این محصول به این شکل کار می‌کند که هرچند وقت یک‌بار پیغامی از سمت محصول برای ما ارسال می‌شود و تعداد بارهایی که حروف آن پیغام در کنار هم رشته‌ی EYVAYHAMLE (ای وای حمله) را تشکیل می‌دهند تعداد حمله‌هایی است که در آن بازه‌ی زمانی به شبکه‌ی سازمان صورت گرفته است.

برای مثال اگر پیغام YEHHMWLEAYVM را دریافت کنیم، با توجه به این‌که حرف E دو مرتبه، حرف Y دو مرتبه، حرف A دو مرتبه، حرف H یک مرتبه، حرف V یک مرتبه، حرف L یک مرتبه و حرف M دو مرتبه تکرار شده است، می‌توان یک رشته‌ی کامل EYVAYHAMLE را ساخت و یک M و یک W هم اضافه آورد و بنابراین می‌فهمیم که یک حمله به شبکه‌ی سازمان صورت گرفته است.

یا اگر پیغام AAAAHHEEEYVVLLMMA را دریافت کنیم، می‌توانیم با حروف این پیغام ۲ رشته‌ی EYVAYHAMLE را بسازیم و یک A هم اضافه بیاوریم. بنابراین می‌فهمیم که دو حمله به شبکه‌ی سازمان صورت گرفته است.

متین که قسمت رمزنگاری و پایش شبکه را پیاده‌سازی کرده است، اکنون از شما می‌خواهد تا برنامه‌ای بنویسید تا با بررسی پیغام ارسال شده، تعداد حملات انجام شده به شبکه را خروجی دهد.

ورودی

ورودی شامل دنباله‌ای از حروف بزرگ (*capital letter*) از A تا Z است که ترتیب به خصوصی ندارند و می‌تواند چندین بار تکرار شوند یا هرگز استفاده نشوند. هر خط ماکسیمم از ۶۰۰ کاراکتر تشکیل شده است.

خروجی

تعداد حملاتی که به شبکه‌ی سازمان شده است را با توجه به حروف داخل پیغام در خروجی چاپ کنید.

دقت کنید که در هر پیغام ممکن است حروف بلااستفاده نیز وجود داشته باشند.

مثال

ورودی نمونه ۱

EMLAVHMLYHVYEA

خروجی نمونه ۱

1

ورودی نمونه 2

AAAAHHEEEYVLLMMA

خروجی نمونه 2

2

ورودی نمونه 3

AABBEEVMLZZ

خروجی نمونه 3

0

B- بحران انقراض ترمکی‌های سالم

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

در سال ۱۴۰۳ یک بیماری مسری در بین ترمکی‌های دانشکده‌ی فنی قدیم شیوع پیدا کرد. این بیماری به این صورت بود که دانشجویان را از شرکت در مسابقات *ACM* باز می‌داشت. دانشجویان فکر می‌کردند که پروژه‌های برنامه‌نویسی پیشرفته یا پاس کردن ریاضی ۲ و فیزیک ۲ (و فقط همین چون دانشجوی ترم ۲ ای که کار مهم دیگری ندارد) مهم‌تر است. این درحالی است که آن‌ها نمی‌دانند مسابقات *ACM* چقدر می‌تواند در ادامه‌ی زندگی‌شان تاثیر بگذارد و دارند دستی دستی آینده‌شان را با خاک یکسان می‌کنند.

بنابراین انجمن *ACM* دانشکده دور هم جمع شدند تا راه حلی برای مشکل ایجاد شده پیدا کنند. آن‌ها ترمکی‌ها را در یک جدول با n سطر و m ستون قرار دادند. در خانه‌ی سطر i ام و ستون j ام ازین جدول مقدار a_{ij} نوشته شده است عددی بین -100 و 100 است و میزان آلوده بودن دانشجویان به بیماری را نشان می‌دهد. هرچه مقدار a_{ij} کمتر باشد، مقدار آلودگی دانشجو بیشتر است و هرچقدر مقدار آن بیشتر باشد دانشجو سالم‌تر است.

پس از مرحله‌ی تشخیص بیماری، اعضای انجمن عملیات زیر را به هرتعداد که دلشان بخواهد (که می‌تواند ۰ هم باشد) روی دانشجویان انجام می‌دهند:

- دو خانه‌ی مجاور دل‌خواه را انتخاب می‌کنند و به دانشجویان آن دوخانه واکسنی تزریق می‌کنند که مقدار آلوده بودنشان را در $1 -$ ضرب می‌کند. دو خانه مجاور هستند اگر یک ضلع مشترک داشته باشند.

دقت کنید که آن‌ها می‌توانند بیشتر از یک‌بار واکسن را به هردانشجو تزریق کنند.

هدف انجمن *ACM* این است که میزان سلامتی ترمکی‌ها را ماکسیمم کند. به عبارت دیگر عملیات بالا را به نحوی انجام دهد که جمع مقادیر داخل خانه‌ها یا X ، بیشترین مقدار ممکن شود.

با توجه به میزان آلودگی افراد (مقادیر داخل خانه‌های جدول)، بیشترین مقدار ممکن X را پیدا کرده و در خروجی چاپ کنید.

ورودی

هر تست شامل چند سناریو متفاوت است.

خط اول ورودی شامل عدد t است که نشان‌دهنده تعداد سناریوهای متفاوت است.

$$1 \leq t \leq 100$$

ورودی برای هر سناریو به صورت زیر است:

اولین خط شامل دو عدد n و m است که به ترتیب نشان‌دهنده تعداد سطر و ستون جدول است.

$$2 \leq n, m \leq 10$$

در n خط بعد هر خط شامل m عدد است که نشان‌دهنده میزان آلوده بودن بیماران است.

$$-100 \leq a_i j \leq 100$$

خروجی

برای هر سناریو بیشترین میزان سلامتی مجموع همه ترمکی‌ها را چاپ کنید.

مثال

ورودی نمونه ۱

```
2
2 2
-1 1
1 1
3 4
0 -1 -2 -3
```

-1 -2 -3 -4
-2 -3 -4 -5

خروجی نمونه ۱

2
30

ورودی نمونه ۲

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

خروجی نمونه ۲

7
7

C - ABCDEF ummmm

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

در جنگ بین متین و پیام‌پردازان، پیام‌پردازان برای ارتباط با یکدیگر از یک روش رمزگذاری استفاده می‌کردند. آن‌ها برای فرستادن پیام عددی ۵ رقمی $abcde$ آن‌را بر یک کلید عددی ۵ رقمی $fghij$ تقسیم می‌کردند (که هرکدام از حروف نشان‌دهنده‌ی یک رقم است) و نتیجه‌ی تقسیم که عدد N است را برای یکدیگر ارسال می‌کردند.

متین قصد دارد که پیام‌های آن‌ها را رمزگشایی کند. به همین دلیل خطوط ارتباط آن‌ها را شنود می‌کند. در اثر این شنودها متین به حقایق زیر دست پیدا می‌کند:

- مقدار N همواره بین ۲ تا ۷۹ است:

$$2 \leq N \leq 79$$

- پیام و کلید ۵ رقمی هردو از ارقام بین ۰ تا ۹ تشکیل شده‌اند و تمام این ۱۰ رقم در پیام و کلید ما دقیقاً یک‌بار استفاده شده‌اند.

$$abscd/fghij = N$$

متین برای رمزگشایی و پیدا کردن پیام $abcde$ باید با توجه به مقدار N تمام پیام‌های $abcde$ و کلیدهای $fghij$ که در معادله‌ی بالا صدق می‌کنند را پیدا کند تا بتواند پیام‌پردازان را شکست دهد.

برنامه‌ای بنویسید که با دریافت مقدار N ، تمام جفت‌های معتبر $abcde$ و $fghij$ را پیدا کند.

ورودی

هر تست شامل چند سناریو متفاوت است.

خط اول ورودی شامل عدد t است که نشان‌دهنده تعداد سناریوهای متفاوت است.

$$1 \leq t \leq 78$$

ورودی برای هر سناریو شامل عدد طبیعی N است.

$$2 \leq N \leq 79$$

خروجی

برنامه‌ی شما باید تمام جفت‌های معتبر را به ترتیب کوچک به بزرگ (بر اساس صورت کسر) نمایش دهد. شکل خروجی شما به طور کلی باید شبیه حالت زیر باشد:

$$xxxx/xxxx = N$$

$$xxxx/xxxx = N$$

...

در صورتی که هیچ جفت عددی‌ای با شروط مطرح شده وجود نداشته باشد، عبارت There are no solutions for N. را در خروجی چاپ کنید.

مثال

ورودی نمونه ۱

2
61
62

خروجی نمونه ۱

There are no solutions for 61.

$$79546 / 01283 = 62$$

$$94736 / 01528 = 62$$

برای $N = 62$ دو جواب متفاوت وجود داشت اما دقت کنید که باید به ترتیب صعودی (بر اساس صورت کسر) چاپ شوند.

ورودی نمونه ۲

3

23

24

25

خروجی نمونه ۲

$$36294 / 01578 = 23$$

$$81627 / 03549 = 23$$

$$81972 / 03564 = 23$$

$$39528 / 01647 = 24$$

$$46872 / 01953 = 24$$

There are no solutions for 25.

D-عه! 🤔

و: 0

ورودی

ورودی شامل عدد طبیعی n است.

$$1 \leq n \leq 10^9$$

خروجی

خروجی شامل یک زوج مرتب از اعداد طبیعی مانند a, b است.

$$0 \leq a, b \leq 10^9$$

مثال

ورودی نمونه ۱

6

خروجی نمونه ۱

2 4

ورودی نمونه ۲

7

خروجی نمونه ۲

5 2

ورودی نمونه ۳

13

خروجی نمونه ۳

9 4

ورودی نمونه ۴

33

خروجی نمونه ۴

1 32

ورودی نمونه ۵

64

خروجی نمونه ۵

64 0

ورودی نمونه ۶

15

خروجی نمونه ۶

5 10

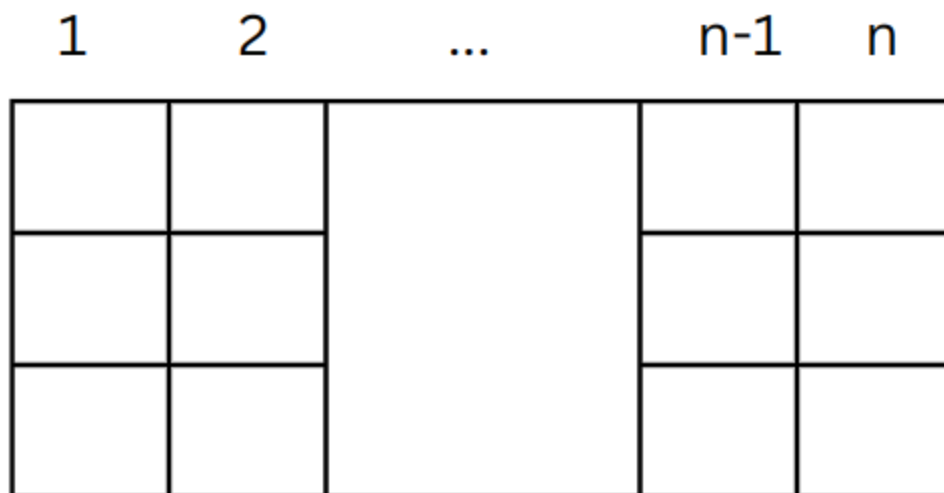
E - مقابله با تنهایی

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

تنهایی یکی از آن مسائلی است که چند صد سال است توجه فیلسوفان و نویسندگان را به خود جلب کرده است. جالب است بدانید که بیشتر رفتارهای روزمره‌ی ما نیز در راستای این است که در جامعه‌ی مورد علاقه‌مان پذیرفته شویم و تنها نمانیم.

اما مجید تنهاست. آن‌قدر تنهاست که حتی جامعه‌ای وجود ندارد که بتواند دلش بخواهد مورد قبول آن واقع شود. و تراپیستش به او گفته است که برای مقابله با تنهاییش سعی کند خودش را سرگرم کند.

بنابراین به اون یک مستطیل $3 \times n$ داد تا برای مدتی سرگرمش کند تا از افکار منفی‌اش دور شود. او از مجید خواست که به ازای عدد n بگوید به چند حالت می‌تواند دومینوهای 1×2 را در این مستطیل $3 \times n$ بچیند.



حالا شما قرار است مجید را تنها نگذارید و برنامه‌ای بنویسید که به او کمک کند این مسئله را حل کند.

ورودی

ورودی شامل عدد صحیح n است که باید پاسخ را به ازای آن محاسبه کنیم.

$$0 \leq n \leq 30$$

خروجی

یک عدد صحیح خروجی دهید که تعداد حالت‌هایی است که می‌توان دومینوهای $1 * 2$ را داخل مستطیل $n * 3$ چید.

مثال

ورودی نمونه ۱

2

خروجی نمونه ۱

3

ورودی نمونه ۲

8

خروجی نمونه ۲

153

ورودی نمونه ۳

12

خروجی نمونه ۳

2131

ورودی نمونه ۴

15

خروجی نمونه ۴

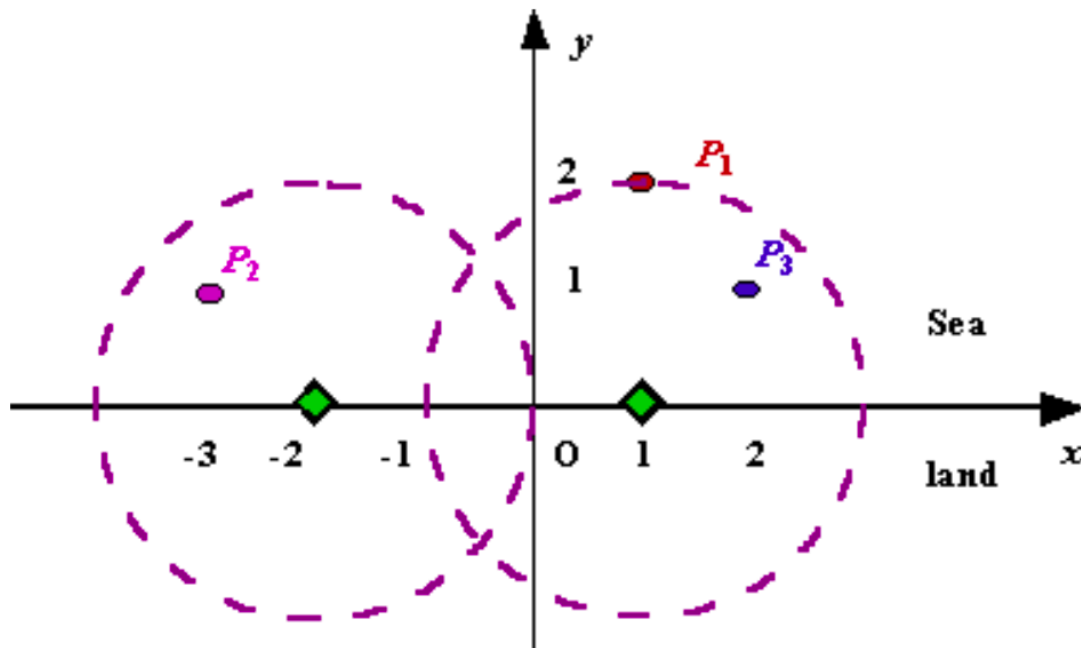
0

F - نصب یه مشت رادار

فرض کنید که خطی که ساحل را از دریا جدا می‌کند، یک خط بی‌نهایت و صاف است. یک طرف خط ساحل، خشکی و یک طرف آن دریاست. همچنین جزیره‌های کوچکی در قسمت‌های مختلف دریا قرار دارند.

ما قصد داریم که بر روی خط ساحل، رادارهایی نصب کنیم. هر کدام از این رادارها می‌توانند تا شعاع d را پوشش دهند. بنابراین یک جزیره در صورتی تحت پوشش یک رادار قرار می‌گیرد که در فاصله‌ی حداکثر d از آن رادار باشد.

اگر شرایط ایجاد شده را روی دستگاه مختصات در نظر بگیریم، خط ساحل همان محور x هاست. دریا، بالای محور x و خشکی، پایین آن قرار دارد. با توجه به مختصات هر کدام از جزیره‌ها و شعاعی که رادارها پوشش می‌دهند، شما باید برنامه‌ای بنویسید که کمترین تعداد رادارهای مورد نیاز برای تحت پوشش قرار دادن تمام جزیره‌ها را خروجی دهد.



لازم به ذکر است که مختصات هر جزیره به صورت طول و عرض در دستگاه مختصات نمایش داده می‌شود.

ورودی

هر تست شامل چند سناریو متفاوت است.

خط اول ورودی شامل عدد t است که نشان‌دهنده تعداد سناریوهای متفاوت است.

$$1 \leq t \leq 5000$$

خط اول هر تست شامل دو عدد صحیح است. مقدار n که تعداد جزیره‌هاست و مقدار d که شعاعی است که رادارها پوشش می‌دهند.

$$1 \leq n \leq 1000$$

$$1 \leq d \leq 100000$$

در ادامه‌ی تست n خط دریافت می‌شود که هر کدام شامل دو عدد صحیح x, y است که نشان‌دهنده‌ی مختصات هر جزیره هستند.

$$-100000 \leq x \leq 100000$$

$$0 \leq y \leq 100000$$

خروجی

به ازای هر تست، خروجی یک خط شامل شماره‌ی تست و سپس پاسخ سوال (کمترین تعداد رادارهای مورد نیاز) می‌باشد. در صورتی که تست هیچ راه حلی نداشت، مقدار -1 در خروجی چاپ می‌شود.

ورودی نمونه ۱

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

خروجی نمونه ۱

Case 1: 1

ورودی نمونه ۲

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

خروجی نمونه ۲

Case 1: 2

Case 2: 1