به نام او

دوره‌ی پایانی المپیاد کامپیوتر ایران
فرودین ۱۳۸۷
آزمون دوم

جزیره‌ی گنج

Treasure Island

زمان مجاز برای هر مورد: ۲۵۰۰ میلی‌ثانیه[۱]
حافظه‌ی مصرفی مجاز: ۱۲۰ مگابایت[۲]

لانگ جان سیلور[۳] در طی سفر دور و دراز خود به دور دنیا، سرانجام به جزیره‌ی گنج پا گذاشت. همه می‌دانستند که مدت‌ها پیش، آن‌هنگام که کاپیتان فلینت[۴] به این جزیره آمده بود، سکه‌های طلا و صندوق‌چه‌های پر از مروارید خود را در آن مخفی کرد. پس لانگ جان سیلور درصدد یافتن این گنج‌ها برآمد. او به دنبال اطلاعاتی در مورد نقشه‌ی جزیره و مکان گنج‌ها می‌گشت، تا این‌که کاملاً اتفاقی با بن گان[۵] که از زمان کاپیتان فلینت در آن جزیره تنها گذاشته شده بود، برخورد کرد. سیلور از بن گان خواست که جای گنجینه‌های فلینت را به او نشان دهد، تا در مقابل او را از این جزیره‌ی دورافتاده نجات دهد. بن گان، در عوض این پیشنهاد، تنها گفت: «نقشه‌ی جزیره به صورت یک مستطیل (جدول) $n \times m$ است، و در هر یک از این $n \times m$ خانه، ممکن است گنجی نهفته باشد.»

از آنجا که سیلور حوصله‌ی جستجو کردن همه‌ی این $n \times m$ خانه را نداشت، به بن گان گفت: «خوب، جای گنج‌ها رو هم بگو!» بن گان که مدّت‌ها بود پنیر تُست‌شده[۶] نخورده بود، به سیلور گفت: «پس به من پنیر تست‌شده بده». لانگ جان سیلور هم که دیگه از دست ادا و اطفارهای بن گان خسته شده بود به بن گان گفت: « برو بینینم بابا!».

بن گان: عمراً دیگه جای گنج‌ها رو بگم!

لانگ جان: خوب حالا! نمی‌خواد قاطی کنی، بهت پنیر تست‌شده هم می‌دم!

---

[۱] زمان مجاز در واقع ۳۰۰۰ میلی‌ثانیه می‌باشد و کتاب‌خانه حداکثر ۵۰۰ میلی‌ثانیه از آن را استفاده می‌کند.
[۲] حافظه‌ی مصرفی مجاز در واقع ۱۳۰ مگابایت می‌باشد و کتاب‌خانه حداکثر ۱۰ مگابایت از آن را استفاده می‌کند.
[۳] Long John Silver
[۴] Captain Flint
[۵] Ben Gunn
[۶] toasted cheese

بن گان: دونقطه دی[۷]. توی هر مربع ۲ × ۲ از جزیره، تعداد خونه‌هایی که گنج دارن زوجه! من بهت اجازه می‌دم از من یه سری سؤال به این شکل بپرسی که آیا تو فلان خونه‌ی جزیره گنج هست یا نه. به ازای هر سؤالی که از من بپرسی باید بهم یه پنیر تست شده بدی. درضمن برای این که تنبیهت کنم، تو رو از پرسیدن سؤال راجع به بعضی از خونه‌ها محروم می‌کنم. حالا برای این که ارفاق کرده باشم، قبل از این که سؤالاتو شروع کنی خونه‌هایی که ازشون محروم شدی رو بهت می‌گم.

حالا شما به لانگ جان سیلور کمک کنید که مجموعه‌ی خانه‌هایی را که در آن‌ها گنجی نهفته است پیدا کند، البته باید با کم‌خرج‌ترین روش (با پرسیدن کم‌ترین تعداد سؤال) این کار را انجام دهید. هم‌چنین در اولین زمانی که متوجه شدید نمی‌توان به کمک راهنمایی‌های بن گان، خانه‌های شامل گنج را به طور یکتا شناسایی کرد، این موضوع را به لانگ جان سیلور اطلاع دهید تا بی‌خیال شود! می‌توانید فرض کنید بن گان همیشه حقیقت را می‌گوید. البته باید توجه کنید که شکم بن گان تنها برای ۳۰۰۰۰ پنیر تست‌شده جا دارد و زحمت پاسخ‌گویی به پرسش‌های بعدی را به خود نخواهد داد.

مسئله

برنامه‌ای بنویسید که

- ابعاد جزیره (m و n) و خانه‌های ممنوع‌شده توسط بن گان را از کتاب‌خانه‌ی پیوندی بخواند؛
- به کمک کتاب‌خانه‌ی پیوندی سؤالات مورد نظر را از بن گان بپرسد؛
- با کم‌ترین پرسش به کتاب‌خانه‌ی پیوندی گزارش دهد که دقیقاً چه خانه‌هایی شامل گنج هستند یا این که نمی‌توان این موضوع را به طور کامل و یکتا مشخص کرد.

کتاب‌خانه

توابع و تعاریفی که توسط کتاب‌خانه در اختیار شما قرار گرفته، به صورت زیر است:

```
#define MAX_M 1000
#define MAX_N 1000
#define MAX_Q 30000
typedef int Table[MAX_M][MAX_N];
```

این تعاریف ثابت‌های مسئله، حداکثر ابعاد جدول و نوع داده‌ساختار جدول (`Table`) را تعریف می‌کنند. منظور از خانه‌ی `[i][j]` در جدول، خانه‌ی سطر iاُم ($0 \le i < m$) و ستون jاُم ($0 \le j < n$) است.

```
void init(int *m, int *n, Table *forbidden);
```

این تابع متغیرهای صحیحی که با `m` و `n` به آن‌ها اشاره شده را به ترتیب با تعداد سطرها و تعداد ستون‌های جدول پر می‌کند. هم‌چنین داده‌ساختار جدولی که توسط `forbidden` به آن اشاره شده را به این شکل پُر می‌کند که اگر خانه‌ی `[i][j]` ($0 \le i < m, 0 \le j < n$) ممنوع شده باشد، مقدار `(*forbidden)[i][j]` برابر 1، و در غیر این صورت، مقدار آن برابر 0 می‌شود. حافظه‌ی مربوط به این اشاره‌گرها را باید قبلاً در محیط اجرای خود گرفته باشید؛ تخصیص حافظه در محیط کتاب‌خانه انجام نمی‌گیرد.

[۷] :D

این تابع باید دقیقاً یک بار و پیش از دیگر توابع کتاب‌خانه فراخوانی شود.

```
int ask(int i, int j);
```

این تابع یک سؤال از بن گان می‌پرسد و پاسخ او را باز می‌گرداند. این سؤال مربوط به خانه‌ی [i][j] جزیره می‌باشد ($0 \le i < m, 0 \le j < n$). اگر در زیر این خانه گنجی پنهان شده باشد، خروجی تابع برابر 1، و در غیر این صورت، خروجی تابع برابر 0 خواهد بود. دقت کنید که نباید درباره‌ی خانه‌ای پرسش کنید که ممنوع شده است. در غیر این صورت، برنامه به علت استفاده‌ی نامناسب از کتاب‌خانه پایان می‌یابد.

```
void reportNo();
```

```
void reportYes(Table *value);
```

با فراخوانی یکی از این دو تابع می‌توانید پاسخ نهایی خود را به کتاب‌خانه تحویل دهید. این تابع‌ها باز نمی‌گردند و پس از فراخوانی‌شان برنامه‌ی‌تان پایان می‌یابد. تابع reportNo اعلام می‌کند که با این شرایط، به هر روشی که پرسش‌ها را ادامه دهیم نمی‌توانیم به طور یکتا تعیین کنیم که هر خانه‌ی جدول گنج دارد یا نه. در مقابل، تابع reportYes اعلام می‌کند که توانسته است این موضوع را برای هر خانه‌ی جدول به صورت یکتا مشخص کند. در این حالت، باید از طریق اشاره‌گر به داده‌ساختار جدولی که به عنوان پارامتر به این تابع می‌دهید (value)، مجموعه‌ی خانه‌هایی که گنج دارند را مشخص کنید. اگر خانه‌ی [i][j] در زیر خود گنجی دارد ($0 \le i < m, 0 \le j < n$)، مقدار (*value)[i][j] را برابر 1، و در غیر این صورت، مقدار آن را برابر 0 قرار دهید.

نقض شرایط گفته‌شده به منزله‌ی نقض استفاده‌ی درست از کتاب‌خانه است. برنامه‌ی‌تان نباید با هیچ فایلی کار کرده، از ورودی استانده بخواند یا به ورودی استانده بنویسد. در ضمن برنامه‌ی شما نباید سعی کند به فضایی خارج از محدوده‌ی خود[۸] دست‌رسی داشته باشد. بدیهی است که نقض هریک از این حدود می‌تواند منجر به حذف شدن از گردونه‌ی رقابت‌ها گردد.

دو فایل tisland.h و tislandlib.cpp به شما داده شده است. بالای برنامه‌ی‌تان باید عبارت «#include "tisland.h"» را قرار دهید. برای هم‌گردانی[۹] برنامه‌ی‌تان فایل‌های فوق‌الذکر را در کنار برنامه‌ی خودتان (در این جا با نام tisland.cpp) کپی کرده و از دستور

```
g++ -O2 -static tisland.cpp tislandlib.cpp -lm
```

استفاده کنید. فایل stisland.cpp نیز یک نمونه از کار کردن با این کتاب‌خانه است. توجه به دو نکته ضروری است:

۱) برنامه‌ی نمونه‌ی داده شده، یعنی stisland.cpp کار مفیدی انجام نمی‌دهد و تنها برای نمایش نحوه‌ی کار کردن با کتاب‌خانه فراهم شده است.

۲) در هنگام ارزیابی برنامه‌ها از کتاب‌خانه‌ی دیگری استفاده خواهد شد که توابع و تعاریف درون tisland.h را حمایت می‌کند. مادامی که مطابق دستورالعمل‌ها از کتاب‌خانه استفاده می‌کنید، این موضوع بر برنامه‌ی شما تأثیری نخواهد داشت.

آزمایش

کتاب‌خانه‌ی آزمایش را می‌توانید از واسط مسابقه دریافت[۱۰] کنید. کتاب‌خانه‌ی آزمایشی داده شده، داده‌های مورد نیاز را از ورودی استانده می‌خواند. در سطر اول ورودی، باید m (تعداد سطرها) و n (تعداد ستون‌ها) با

[۸] به خصوص فضای مربوط به کتاب‌خانه
[۹] compile
[۱۰] Download

فاصله از هم نوشته شده باشند. در هر یک از m سطر بعد، باید n عددِ 0 یا 1 با فاصله از هم نوشته شده باشد که عدد jاُم ($0 \le j < n$) در سطر iاُم ($0 \le i < m$) از این m سطر 1 است اگر و تنها اگر پرسیدن خانه‌ی [j][i] ممنوع شده باشد. در m سطر دیگر نیز به طور مشابه یک جدول $m \times n$ از 0 و 1 آمده است که عدد jاُم ($0 \le j < n$) در سطر iاُم ($0 \le i < m$) از این m سطر 1 است اگر و تنها اگر در زیرِ خانه‌ی [j][i] گنجی نهفته باشد.

هنگام استفاده از واسط آزمایش[11] برنامه‌ی‌تان در برابر همین کتاب‌خانه کار می‌کند و فایلی که به‌عنوان ورودی ارسال می‌کنید، باید دارای ساختار مذکور باشد.

شما آزادید برنامه‌هایی که دراختیارتان قرار داده شده را تغییر دهید ولی توصیه می‌شود که فایل tisland.h را تغییر ندهید.

شیوه‌ی ارزیابی

اگر برنامه‌ی‌تان از محدودیت‌های اعمال شده تجاوز کند، یا از کتاب‌خانه به‌صورت شایسته استفاده ننماید، به‌ازای آن موردِ آزمون، نمره‌ای به برنامه تعلق نمی‌گیرد.

برنامه‌تان حداکثر اجازه‌ی انجام ۳۰۰۰۰ پرسش را دارد. اگر بیش از این تعداد پرسش انجام دهید، برنامه‌تان متوقف شده و نمره‌ای نمی‌گیرید. اگر در پایان کار، گزارش داده‌شده نادرست باشد نیز نمره‌ای نمی‌گیرید. هم‌چنین اگر وضعیت گنج‌ها را به درستی (در reportYes) گزارش دهید ولی با توجه به پرسش‌هایی که کرده‌اید تعیین وضعیت گنج‌ها به طور یکتا غیرممکن باشد، نمره‌ای برای آن مورد آزمون به برنامه داده نمی‌شود. تنها زمانی به برنامه نمره تعلق می‌گیرد که تعداد پرسش‌های انجام‌شده تا رسیدن به نتیجه‌ی درست کمینه باشد. دقت کنید که اگر در حالتی، بتوان گزارش مناسب را بدون پرسشی از کتاب‌خانه به آن اعلام کرد، در صورت انجام حتی یک پرسش، تعداد پرسش‌های انجام‌شده کمینه نبوده و در نتیجه نمره‌ی آن مورد آزمون به برنامه داده نمی‌شود.

در تمامی تست‌ها $1 \le m, n \le 1000$ است، ولی در ۶۰٪ تست‌ها $1 \le m, n \le 100$ می‌باشد.

تعامل نمونه

در زیر یک تعامل ساده با کتاب‌خانه آمده است.

[11] Test Run

فراخوانی	توضیحات
`int m, n;`	تعریف متغیرهای m و n
`Table forbidden, value;`	تعریف forbidden و value از نوع داده‌ساختار جدول
`init(&m, &n, &forbidden);`	آغاز تعامل با مقداردهی m و n و forbidden
`int a=-1, b=-1;`	تعریف دو متغیر موقتی
`if (!forbidden[0][0])`	پرسش در موردِ
`a=ask(0,0);`	خانه‌ی گوشه‌ی بالا سمت چپ جدول.
`if (!forbidden[m-1][n-1])`	پرسش در موردِ
`b=ask(m-1,n-1);`	خانه‌ی گوشه‌ی پایین سمت راست جدول.
`if (a<0 || b<0) {`	
`reportNo();`	گزارش می‌دهیم که نمی‌توان جواب را پیدا کرد.
`} else {`	
`value[0][0]=a;`	مقداردهی خانه‌ی گوشه بالا سمت چپ جدول value.
`value[m-1][n-1]=b;`	مقداردهی خانه‌ی گوشه پایین سمت راست جدول value.
`reportYes(&value);`	جواب خود را در جدول value گزارش می‌دهیم.
`}`	

یک نمونه از فایل ورودی برای `tislandlib.cpp` را در زیر مشاهده می‌کنید.

```
4 5
0 0 1 0 0
0 0 0 1 0
1 0 0 0 0
1 1 1 0 1
1 1 0 0 1
0 0 1 1 0
1 1 0 0 1
1 1 0 0 1
```