

لیوان‌ها (Glass)

زمان مجاز برای هر مورد: ۴۰۰ میلی‌ثانیه^۱
حافظه‌ی مصرفی مجاز: ۶۴ مگابایت^۲

پنتاگون تصمیم به گسترش یکی از مهم‌ترین بخش‌های خود گرفته است: آشپزخانه! در قدم اول در این راستا، برای آشپزخانه تعداد زیادی لیوان شیشه‌ای خریده است. در قدم بعد تصمیم دارد تعداد سالن‌های غذاخوری را هم افزایش دهد.

آقای ب.، رئیس کل امور تغذیه‌ی پنتاگون، خوب می‌داند که کیفیت بد غذاهای پنتاگون (که حتا با فراخ‌ترین گسترش‌ها هم خوب نخواهد شد) موجب خواهد شد کارکنان سازمان از فرط عصبانیت لیوان‌های جدید شیشه‌ای را از پنجره به بیرون پرتاب کنند. اگر لیوان‌ها از طبقات بالا پرتاب شوند می‌شکنند و موضوع تابلوتر از حد مجاز می‌شود. برای همین وی در نامه‌ای^۳ از پنتاگون خواست که به دلیل کم‌رودش سالن‌های غذاخوری جدید را در طبقات بالا قرار ندهند. آن چه وی در پاسخ به نامه‌اش دریافت کرد به شرح زیر است:

با سلام،

لطفاً منظور خود را از عبارت «طبقات بالا» به صورت واضح‌تر بیان نمایید.

این جا بود که او تصمیم گرفت با انجام آزمایشی به‌تنهایی بفهمد پرتاب لیوان‌ها از چه طبقه‌ای به بالا موجب شکستن‌شان می‌شود. آزمایش او شامل حرکت (از طریق پله‌ها) بین طبقات مختلف پنتاگون و پرتاب بعضی از این لیوان‌ها از پنجره‌ی بعضی از طبقات ساختمان پنتاگون است. او از اندک لیوان‌هایی برای آزمایش خود استفاده می‌کند که بر اثر جوب‌های اداری شماره نشده بودند. بر همگان واضح و مبرهن است که:

- همه‌ی لیوان‌ها کاملاً یکسان‌اند و در برابر شرایط برابر پاسخ‌های یکسان دارند؛
- اگر پرتاب لیوان از یک طبقه موجب شکسته شدن آن شود، پرتاب لیوان از طبقات بالاتر هم موجب شکسته شدن آن می‌شود؛
- با تجهیزات پیش‌رفته‌ی موجود، شکسته شدن یا نشدن لیوان پرتاب شده را از هر طبقه‌ای می‌توان دید؛
- پرتاب لیوان‌ها از طبقه‌ی ۱ (همکف) موجب شکسته شدن آن‌ها نمی‌شود؛
- پرتاب لیوان‌ها از طبقه‌ی آخر حتماً موجب شکسته شدن آن‌ها می‌شود؛
- از لیوان‌های شکسته‌شده نمی‌توان در ادامه‌ی آزمایش استفاده کرد؛ و
- از لیوان‌های شکسته نشده می‌توان در ادامه‌ی آزمایش استفاده کرد.

برای برداشتن لیوان‌های پرتاب شده‌ی شکسته نشده، آقای ب. باید ابتدا به طبقه‌ی ۱ برود و بعد آن‌ها را در جیب خود جمع کند. آقای ب. کم‌رود دارد و می‌خواهد در بدترین حالت، کم‌ترین میزان ممکن بالا رفتن از طبقات را داشته باشد؛ دقت کنید که میزان پایین رفتن از پله‌ها و مکان نهایی او اهمیتی ندارد. او در ابتدا در طبقه‌ی ۱ قرار دارد و همه‌ی لیوان‌های قابل آزمایش در جیبش هستند.

^۱ زمان مجاز در واقع ۵۰۰ میلی‌ثانیه می‌باشد و کتاب‌خانه حداکثر ۱۰۰ میلی‌ثانیه از آن را استفاده می‌کند.

^۲ حافظه‌ی مصرفی مجاز در واقع ۶۵ مگابایت می‌باشد و کتاب‌خانه حداکثر ۱ مگابایت از آن را استفاده می‌کند.

^۳ نامه‌نگاری جزء لاینفک کارهای اداری است.

مسئله

برنامه‌ای بنویسید که

- تعداد طبقات ساختمان پنتاگون (همان f) و تعداد لیوان‌های موجود (یعنی g) را از کتاب‌خانه‌ی پیوندی بخواند؛
- به کمک توابع کتاب‌خانه‌ی پیوندی آقای ب. را در انجام آزمایش یاری نماید؛
- کم‌ترین شماره‌ی طبقه‌ای را به کتاب‌خانه‌ی پیوندی اعلام کند که پرتاب لیوان از آن موجب شکسته شدنش می‌شود.

کتاب‌خانه

توابع زیر در کتاب‌خانه یافت می‌شود:

- `int getFloors()` این تابع f (تعداد طبقات ساختمان پنتاگون) را باز می‌گرداند ($2 \leq f \leq 50$).
- `int getGlasses()` این تابع g (تعداد لیوان‌های قابل استفاده‌ی موجود) را باز می‌گرداند ($1 \leq g \leq 10$).
- `void go(int floorNumber)` این تابع به آقای ب. می‌گوید که به چه طبقه‌ای برود. بدیهی است که `floorNumber` باید عددی بین ۱ و f باشد.
- `int drop()` با فراخوانی این تابع، آقای ب.، یکی از لیوان‌های موجود در جیبش را از پنجره‌ی طبقه‌ای که در آن است به بیرون پرتاب می‌کند. بدیهی است فراخوانی این تابع زمانی معنی دارد که لیوانی در جیب آقای ب. موجود باشد. خروجی تابع یکی از دو مقدار `BROKEN` یا `NOTBROKEN` است که در فایل `glass.h` تعریف شده‌اند و به ترتیب نشان می‌دهند که لیوان پرتاب شده شکسته شده یا نه.
- `void collect()` با فراخوانی این تابع، آقای ب.، لیوان‌های پرتاب شده‌ی شکسته نشده در طبقه‌ی ۱ را جمع می‌کند و در جیبش می‌گذارد. فراخوانی این تابع زمانی معنی دارد که او در طبقه‌ی ۱ باشد.
- `void report(int floorNumber)` با فراخوانی این تابع می‌توانید پاسخ خود را به کتاب‌خانه تحویل دهید. این تابع باز نمی‌گردد و پس از فراخوانی‌اش برنامه‌ی‌تان پایان می‌یابد. مقدار `floorNumber` باید عددی بین ۱ و f و شماره‌ی پایین‌ترین طبقه‌ای باشد که پرتاب لیوان از آن‌جا موجب شکسته شدن آن می‌شود.

یکی از دو تابع `getFloors()` یا `getGlasses()` باید حتماً قبل از دیگر توابع کتاب‌خانه فراخوانی شوند. نقض شرایط گفته‌شده به منزله‌ی نقض استفاده‌ی درست از کتاب‌خانه است. برنامه‌ی‌تان نباید با هیچ فایلی کار کرده، از ورودی استاندارد بخواند یا به خروجی استاندارد بنویسد. در ضمن برنامه‌ی‌تان نباید سعی کند که به فضایی خارج از برنامه دسترسی داشته باشد. بدیهی است که نقض هر یک از این حدود می‌تواند منجر به حذف شدن از گردونه‌ی رقابت‌ها گردد.

دو فایل `glasslib.cpp` و `glass.h` به شما داده شده است. بالای برنامه‌ی‌تان باید عبارت `#include "glass.h"` را قرار دهید. توابع و تعاریفی که توسط کتاب‌خانه در اختیار شما قرار گرفته، به صورت زیر است:

```
#define BROKEN 1
#define NOTBROKEN 0
int getFloors();
int getGlasses();
void go(int floorNumber);
int drop();
void collect();
void report(int floorNumber);
```

برای هم‌گردانی برنامه‌ی تان فایل‌های فوق‌الذکر را پیش برنامه‌ی تان (در این جا با نام `glass.cpp`) کپی کرده و از دستور

```
g++ -O2 -static glass.cpp glasslib.cpp -lm
```

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

(۱) برنامه‌ی نمونه‌ی داده شده، یعنی `sglass.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 `glass.h` را حمایت می‌کند.

آزمایش

کتاب‌خانه‌ی آزمایش را می‌توانید از `http://contest/` دریافت کنید. کتاب‌خانه‌ی آزمایشی داده شده، داده‌های مورد نیاز را از ورودی استاندارد می‌خواند. در تنها سطر ورودی، باید به ترتیب ۳ عدد f (تعداد طبقات ساختمان پنتاگون)، g (تعداد لیوان‌ها)، و a (شماره‌ی پایین‌ترین طبقه‌ای که لیوان پرتاب‌شده از آن می‌شکند) نوشته شده باشند.

هنگام استفاده از واسطه آزمایش (TEST) برنامه‌ی تان در برابر همین کتاب‌خانه کار می‌کند و فایلی که به‌عنوان ورودی ارسال می‌کنید، باید دارای ساختار مذکور باشد.

شما آزادید برنامه‌هایی که در اختیار تان قرار داده شده را تغییر دهید ولی توصیه می‌شود که فایل `glass.h` را تغییر ندهید.

شیوه‌ی ارزیابی

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

اگر در پایان کار، آقای ب. از پله بیش از آن بالا رفته باشد که در بدترین حالت لازم بوده، یا با آزمایش‌های انجام شده، پاسخ مسئله به‌طور قطع معلوم نباشد، نمره‌ی آن مورد آزمون به برنامه داده نمی‌شود.

تعامل نمونه

فراخوانی	رویداد	
	floors=getFloors();	مقدار floors برابر خروجی تابع یعنی 4 می شود.
	glasses=getGlasses();	مقدار glasses برابر خروجی تابع یعنی 1 می شود.
	collect();	آقای ب. لیوان های نشکسته ی روی زمین را جمع می کند.
	go(2);	آقای ب. به طبقه ی 2 می رود.
	int result=drop();	مقدار result برابر خروجی تابع یعنی NOTBROKEN می شود.
	report(3)	اعلام می کنیم که جواب طبقه ی 3 است.