به‌نام خدا

Nowy Sącz, POLAND 2005
The 17-th International
Olympiad in Informatics

چندضلعی
(Polygon)

زمان مجاز برای هر مورد: ۱۰۰ میلی‌ثانیه[1]
حافظه‌ی مصرفی مجاز: ۱۰ مگابایت

یکی از دوستان به‌تازگی یاد گرفته که برنامه‌های هندسه‌ی محاسباتی بنویسد: مساحت شکل رو حساب می‌کند و تشخیص می‌دهد که آیا نقطه‌ای درون یک چندضلعی محدب هست یا خیر و از این دست.

عمو شایان به این دوست ما می‌گوید: «اگر راست می‌گی، برای چندضلعی‌های غیرمحدب این کار رو بکن!» این دوست ما هم عوض سرخورده شدن، ادعا می‌کند که می‌تواند با زمان کم‌تر از خطی مسئله را حل کند. فقط یک مشکل کوچک وجود دارد: ورودی مسئله n تا نقطه‌ی چندضلعی را شامل می‌شود که خوب خواندنش کار را خراب می‌کند. عمو شایان هم با توجه به سخاوتش، برای این‌که جای پیش‌رفت به این جوان آینده‌دار بدهد، امکاناتی را فراهم می‌کند که دوست ما بتواند ادعای خودش ثابت دهد. دوست ما بایستی از عمو شایان مختصات بعضی رئوس چندضلعی را پرسیده و درنهایت تعیین کند که آیا نقطه‌ی مورد نظر «درون» یا «برون» چندضلعی جای می‌گیرد.

مسئله

برنامه‌ای بنویسید که

- تعداد رئوس و مختصات رأس اصلی را از **کتاب‌خانه‌ی پیوندی** بخواند؛
- مختصات تعداد کمی از رئوس را از **کتاب‌خانه‌ی پیوندی** بگیرد؛ و
- پاسخ را به **کتاب‌خانه‌ی پیوندی** اعلام کند.

کتاب‌خانه

توابع زیر در کتاب‌خانه یافت می‌شود:

`init()` این تابع باید دقیقاً یک بار و پیش از هرگونه تعاملی با کتاب‌خانه فراخوانده شود. تعداد رئوس چندضلعی، $3 \le n \le 1000$، را باز می‌گرداند.

`get_query()` مختصات رأس اصلی (همان رأسی که باید وضعیتش را مشخص کنید) را به شما می‌دهد.

[1] زمان مجاز درواقع ۱۵۰ میلی‌ثانیه می‌باشد و کتاب‌خانه حداکثر ۵۰ میلی‌ثانیه از آن را استفاده می‌کند.

`get_vertex(index)` مختصات رأس شماره‌ی `index` از چندضلعی را در اختیارتان می‌گذارد. رئوس چندضلعی به‌ترتیب ساعت‌گرد و با شماره‌های ۱ تا n مشخص می‌شوند. دقت کنید که قدرمطلق مختصه‌های رئوس چندضلعی و نیز رأس اصلی از 10^8 بیش‌تر نمی‌باشد.

`report(status)` با فراخوانی این تابع می‌توانید پاسخ خود را به کتاب‌خانه تحویل دهید. این تابع به برنامه‌ی شما بازنمی‌گردد و پس از فراخوانی‌اش برنامه‌ی‌تان پایان می‌یابد. مقدار `status` یکی از مقادیر `INSIDE` یا `OUTSIDE` است که در فایل `polygon.h` تعریف شده‌اند.

برنامه‌ی‌تان نباید با هیچ فایلی کار کرده، از ورودی استانده بخواند یا به خروجی استانده بنویسد. در ضمن برنامه‌ی‌تان نباید سعی کند که به فضایی خارج از برنامه دست‌رسی داشته باشد. بدیهی است که نقض هریک از این حدود می‌تواند منجر به حذف شدن از گردونه‌ی رقابت‌ها گردد.

سه فایل `polygon.h` و `polygonlib.cpp` به شما داده شده است. بالای برنامه‌ی‌تان باید عبارت `#include "polygon.h"` را قرار دهید. توابع و تعاریفی که توسط کتاب‌خانه دراختیار شما قرار گرفته، به‌صورت زیر است:

```
#define INSIDE 0
#define OUTSIDE 1
struct Point {
  int x, y;
};
int init();
Point get_query();
Point get_vertex(int index);
void report(int status);
```

برای هم‌گردانی برنامه‌ی‌تان فایل‌های فوق‌الذکر را پیش برنامه‌ی‌تان کپی کرده و از دستور

```
g++ -O2 -static polygon.cpp polygonlib.cpp -lm
```

استفاده کنید. فایل `spolygon.cpp` نیز به شما داده شده، یک نمونه از کار کردن با این کتاب‌خانه است. توجه به دو نکته ضروری است:

۱. برنامه‌ی نمونه‌ی داده شده، یعنی `spolygon.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 `polygon.h` را حمایت می‌کند.

آزمایش

کتاب‌خانه‌ی آزمایش را می‌توانید از `http://contest/` دریافت کنید. کتاب‌خانه‌ی آزمایشی داده شده، داده‌های مورد نیاز را از ورودی استانده می‌خواند. در سطر نخست ورودی باید n را بنویسید. در

سطر بعدی مختصات رأس اصلی و n سطر دیگر مختصات رئوس چندضلعی را نشان می‌دهد.
هنگام استفاده از واسط آزمایش (TEST) 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 `polygon.h` را تغییر ندهید.

شیوه ارزیابی

اگر برنامه‌ی‌تان از کتاب‌خانه به‌صورت شایسته استفاده ننماید، به‌ازای آن مورد آزمون نمره‌ای به شما تعلق نمی‌گیرد؛ همین طور اگر از محدودیت‌های اعمال شده بر برنامه تجاوز کنید.
برنامه‌ی‌تان حداکثر اجازه‌ی انجام ۲۰ پرسش درمورد رئوس چندضلعی را دارد. اگر بیش از این تعداد پرسش انجام دهید، برنامه‌ی‌تان متوقف شده و نمره‌ای نمی‌گیرید. اگر با پرسش‌هایی که انجام داده‌اید، پاسخ مسئله به‌طور قطع معلوم نباشد، نمره‌ای نخواهید گرفت؛ دقت کنید که بقیه‌ی رئوس می‌توانند هر مختصات حقیقی داشته باشند، هرچند مختصاتی که به شما داده می‌شود، همواره صحیح است.
برای هر مورد آزمون، یک عدد به‌عنوان تعداد پرسش‌های موردنیاز برای یافتن حتمی پاسخ درنظر گرفته‌ایم (با کمی اغماض). اگر این عدد `best` و تعداد پرسش‌های شما `queries` باشد، و پاسخ‌تان صحیح باشد، به‌اندازه‌ی

$$\left(\frac{۲۱ - \texttt{queries}}{۲۱ - \texttt{best}}\right)^{۲}$$

از نمره‌ی آن مورد را دریافت می‌کنید. البته درصورتی که تعداد پرسش‌های‌تان کم‌تر از `best` باشد، تمامی نمره‌ی آن مورد را کسب می‌کنید.

تعامل نمونه

فراخوانی	روی‌داد
`init()`	مقدار $n = ۴$ بازگردانده می‌شود.
`get_query()`	مختصات رأس اصلی (۱, ۱) بازگرانده می‌شود.
`get_vertex(1)`	مختصات رأس شماره‌ی یک (۴, ۳) برمی‌گردد.
`report(INSIDE)`	اعلام می‌کنیم که نقطه درون چندضلعی است.