

باسمه تعالی

شانزدهمین دوره‌ی آموزشی المپیاد کامپیوتر

امتحان عملی اصلی دوم

پنج‌شنبه دوم شهریورماه ۱۳۸۵

وقت: پنج ساعت

نصیری شرق، اوئیس قرن

توصیه‌های ایمنی

- در خط اول هر یک از برنامه‌های تان نام خود را با فرمت روبه‌رو قرار دهید: `//name: YourNameHere`
مثال: `//name: Tom Verhoeff`
فراموش کردن این مهم در هر یک از برنامه‌های شما، موجب از دست دادن تمام نمره‌ی آن سؤال خواهد شد. دقت کنید که اولین فاصله‌ی خالی این سطر، بعد از : می‌باشد.
- توصیه می‌شود قبل از ترک جلسه‌ی امتحان حتماً یک‌بار تست‌های برگه^۱ را به برنامه‌های تان بدهید.
- وقت امتحان چهار و نیم ساعت است و پس از اتمام وقت، برق سالن قطع خواهد شد. حداقل زمان حضور در جلسه نیز یک ساعت می‌باشد.
- در ده دقیقه‌ی ابتدایی به هیچ سؤالی پاسخ داده نمی‌شود. در این زمان حتماً صورت هر دو سؤال را بخوانید. سؤالات شما، صرفاً در باب رفع ابهام، در یک ساعت اول جواب داده خواهند شد.
- نام کاربری شما در این امتحان **Exam2** و رمز عبور آن **esteghamat** می‌باشد. در طول امتحان فقط از همین کاربری استفاده کنید.
- برنامه‌های تان را در شاخه‌ی ~ با نام‌های **three.cpp** و **chanbare.cpp** ذخیره کنید. دقت کنید که هنگام جمع‌آوری فایل‌ها تنها وجود دو فایل `/home/Exam2/three.cpp` و `/home/Exam2/chanbare.cpp` در رایانه‌ی شما بررسی خواهد شد و بعداً هیچ عذری در این زمینه پذیرفتنی نیست.
- محدودیت زمانی برنامه‌های شما **چهار** ثانیه (روی سرور) در نظر گرفته شده است. بدیهی است پس از این مدت برنامه‌ی شما کلاً پا خواهد شد. هیچ‌گونه محدودیت حافظه برای برنامه‌های شما اعمال نخواهد شد؛ با این حال توجه داشته باشید که استفاده از حافظه‌ی زیاد باعث کند شدن برنامه می‌شود.
- از آن‌جا که حجم فایل‌های ورودی مسائل این امتحان ممکن است زیاد باشد، اکیداً توصیه می‌شود برای خواندن ورودی و نوشتن خروجی از توابع `fscanf()` و `fprintf()` استفاده کنید. به عنوان مثال، برنامه‌ی زیر یک عدد صحیح n از فایل `sample.in` خوانده و مقدار $3n$ را در `sample.out` می‌نویسد:

```
#include <cstdio>

FILE *fin = fopen("sample.in", "r"); // opens the file for reading
FILE *fout = fopen("sample.out", "w"); // opens the file for writing

int main() {
    int n;
    fscanf(fin, "%d", &n); // Don't forget the & symbol
    fprintf(fout, "%d\n", 3*n); // \n is the same as endl
    return 0;
}
```

^۱ Sample input

۱ درخت ۱۰۰ نمره

آقای «نون» در آرشیو برنامه‌های زمان المپیادش یک فایل متنی پیدا کرده است!
... در سطر اول این فایل یک عدد طبیعی t نوشته شده است و پس از آن t بلوک از اعداد بدین صورت قرار گرفته‌اند که در سطر اول هر بلوک، یک عدد طبیعی e نوشته شده و بعد از آن دقیقاً e زوج عدد صحیح آمده است.
آقای «نون» حدس می‌زند این فایل یال‌های t تا درخت ریشه‌دار^۲ است!
... آیا حدس آقای «نون» درست است؟!

ورودی: فایل three.in

در سطر اول فایل ورودی، t تعداد بلوک‌های ورودی آمده است. سپس در هر بلوک، ابتدا یک عدد e ، تعداد یال‌های آن بلوک آمده است. پس آن در هر یک از e سطر بعدی، دو عدد x y آمده است که به معنی وجود یک یال جهت‌دار از x به y است.
می‌توانید فرض کنید که در صورتی که یال‌های ورودی تشکیل یک درخت بدهند، رئوس درخت شماره‌های ۱ تا $e + 1$ را خواهند داشت؛ اما لزومی ندارد که ریشه (در صورت وجود) رأس شماره‌ی یک باشد.

خروجی: فایل three.out

در t سطر خروجی، به ازای هر یک از بلوک‌های ورودی، در صورتی که آن بلوک یک «درخت ریشه‌دار» است، شماره‌ی رأس ریشه را بنویسید. در غیر این صورت، عبارت `This is not a three!` را بنویسید.

محدودیت‌ها

- $1 \leq t \leq 10$.
- $1 \leq e \leq 1,000,000$.
- اعداد ورودی در تایپ `int` جا می‌شوند.

مثال

three.in	three.out
2	3
2	This is not a three!
3 1	
3 2	
3	
1 2	
2 3	
1 3	

توصیه

این سؤال را دست کم نگیرید!

^۲ یک درخت ریشه‌دار، درخت جهت‌داری است که در آن فقط و فقط از یک رأس (ریشه) می‌توان به هر یک از رئوس دقیقاً یک مسیر جهت‌دار یافت.

۲ چنبره ۱۰۰ نمره

یک چنبره جدولی است $n \times n$ بطوری که سطر اول و سطر آخرش مجاور هم هستند و ستون اول و آخرش هم مجاور یکدیگرند. بعنوان مثال خانه‌ی $(1, 1)$ که خانه‌ی گوشه‌ی بالا و سمت چپ جدول است مجاور خانه‌های $(1, n)$ (خانه‌ی گوشه‌ی بالا و سمت راست جدول) و $(n, 1)$ (خانه‌ی گوشه‌ی پایین و سمت چپ جدول) است. یک چنبره را با اعدادی پر کرده و اسم آن را A قرار داده‌ایم. شما بایستی یک چنبره‌ی B را طبق قانونی که در زیر گفته شده پر کنید.

یک عدد k به شما داده شده می‌شود. برای هر خانه‌ی (i, j) از جدول مربعی به ضلع $2k + 1$ حول آن در نظر بگیرید بطوری که خانه‌ی مرکزی مربع خانه‌ی (i, j) باشد؛ شما بایستی مجموع اعداد این مربع از چنبره‌ی A را در خانه‌ی (i, j) از چنبره‌ی B قرار دهید.

ورودی: فایل chanbare.in

در سطر اول فایل ورودی بترتیب n و k نوشته شده است. در n سطر بعدی در هر سطر n عدد نوشته شده است بطوری که اعداد سطر $i + 1$ ورودی اعداد سطر i از چنبره‌ی A را نشان می‌دهد.

خروجی: فایل chanbare.out

فایل خروجی n سطر دارد و شما بایستی در هر سطر n عدد بنویسید بطوری که اعداد سطر i ام خروجی اعداد سطر i چنبره‌ی B را نشان می‌دهد.

محدودیت‌ها

$$1 \leq 2k + 1 \leq n \leq 1000.$$

توصیه

می‌دانیم اعداد چنبره‌های A و B در long long جا می‌شوند. ولی ممکن است مجموع اعداد چنبره‌ی A خیلی بیشتر از بزرگترین عددی باشد که در long long جا می‌شود. برای خواندن یک عدد x در long long از فایل ورودی از دستور `fscanf(fin, "%lld", &x)` و برای نوشتن آن در فایل خروجی از دستور `fprintf(fout, "%lld", x)` استفاده کنید.

مثال

chanbare.in	chanbare.out
4 1	13 13 11 14
3 1 1 2	13 13 12 13
1 1 2 1	11 12 11 11
1 2 1 1	14 13 11 13
2 1 1 1	