

Task: REC Rectangle Game

Country: BULGARIA

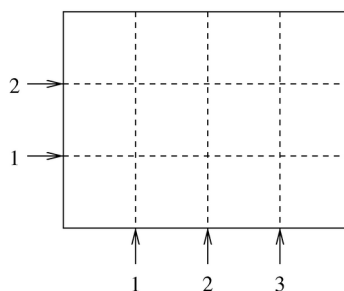


Day 2. Source file `rec.*`

Monday, August 22, 2005.

Available memory: 32 MB. Maximum running time: 14 s.*

Разглеждаме игра с двама играчи. Те играят върху правоъгълник с размери $x \times y$ (x и y са цели положителни числа). Играчите редуват ходовете си. Всеки ход представлява разделяне на правоъгълника на два правоъгълника чрез един вертикален или хоризонтален разрез. Получените правоъгълници трябва да имат размери, изразяващи се с положителни цели числа.



Възможни разрези на правоъгълник с размери 4×3 .

След всеки разрез, по-малкият (т.е. този с по-малко лице) от получените правоъгълници се изхвърля, а другият правоъгълник се дава на противника, за да продължи играта. Ако при разрязването се получат два еднакви правоъгълника, изхвърля се произволен от тях. Играчът, който на пореден ход трябва да разреже правоъгълник с размер 1×1 , губи играта.

Напишете програма, която играе играта и да победи. Програмата трябва да използва специална библиотека. Библиотеката съдържа функции `dimension_x()` и `dimension_y()`, връщащи размерите на правоъгълника. Началните размери са цели числа между 1 и 100000000. Поне единият размер е по-голям от 1. В 50% от тестовите размерите не надминават 25.

Библиотеката има процедура `cut(dir, position)`, която трябва да бъде извикана от вашата програма, за да се направи ход. Параметрите `dir` и `position` описват посоката и позицията на разреза. Параметърът `dir` трябва да приема една от следните две стойности: `vertical` или `horizontal`. Ако `dir = vertical`, разрезът е вертикален и тогава параметърът `position` задава x -координатата на разреза (виж горната фигура). Трябва да осигурите $1 \leq position \leq dimension_x() - 1$. Ако `dir = horizontal`, разрезът е хоризонтален и параметърът `position` задава y -координатата на разреза, при което трябва да осигурите $1 \leq position \leq dimension_y() - 1$.

Вашата програма играе за първия играч — тя трябва да разреже началния правоъгълник. Когато програмата се обръща към процедурата `cut`, вашият ход се записва и управлението се предава на противника. След като противникът направи ход, управлението се предава отново на вашата програма. Стойностите върнати от `dimension_x()` и `dimension_y()` дават резултата, получен след вашия ход и след хода на противника. Щом вашата програма победи, загуби или направи некоректен ход (т.е. извика процедурата `cut` с неправилна стойност на параметрите) тя ще бъде прекратена. Прекратяването на вашата програма е автоматично и така тя би трябвало до може да продължава играта, докато е възможно. За тестовите данни винаги съществува печеливша стратегия за вашата програма.

*Времето за работа на програмите от включената библиотека няма да надмине 4 s.

Програмата не трябва да чете или записва никакви файлове, не трябва да използва стандартния вход и изход, и не трябва да променя паметта извън програмата. Нарушаването на тези изисквания ще ви дисквалифицира.

Експерименти

За да експериментирате, ви се предоставя примерна библиотека на противника. Нейните изходни кодове са в `preclib.pas`, `creclib.c` and `creclib.h` files. Библиотеката може да се вземе от <http://contest/>. Те реализират една много проста стратегия. Когато изпълнявате вашата програма, тя ще играе срещу този прост противник. Ако желаете, може да промените тези изходни кодове, за да проверите вашата програма за игра срещу по-добър противник. При тестването от журито, вашата програма ще играе срещу друг противник.

Когато изпращате вашата програма за използване на тестовия интерфейс, тя ще бъде компилирана с оригиналната примерна библиотека. Изпратеният входен файл ще бъде даден на стандартния вход на вашата програма. Входният файл трябва да се състои от два реда, всеки съдържащ едно цяло число. Първият ред трябва да съдържа началната ширина, а вторият ред – началната височина на правоъгълника. Тези размери се прочитат от примерната библиотека на противника.

Ако промените реализацията в `preclib.pas`, трябва да я компилирате отново като използвате следните команди: `ppc386 -O2 preclib.pas`. Командите създават файлове `preclib.o` и `preclib.ppu`. Тези файлове са необходими, за да компилирате вашата програма и трябва да бъдат поместени в директорията, където се намира програмата. Моля, не променяйте интерфейската част на `preclib.pas`.

Ако промените `creclib.c`, не забравяйте да я поставите (заедно с `creclib.h`) в директорията, където се намира вашата програма — те са необходими за компилирането. Моля, не променяйте файла `creclib.h`.

Предоставят ви се две прости програми, илюстриращи използването на библиотеките: `cres.c` и `prec.pas`. (Имайте предвид, че тези програми не са правилни решения) Може да ги компилирате, като използвате следните команди:

```
gcc -O2 -static cres.c creclib.c -lm
g++ -O2 -static cres.c creclib.c -lm
ppc386 -O2 -XS prec.pas
```

Библиотека

Предоставя ви се библиотека със следната функционалност:

- **FreePascal Library** (`preclib.ppu`, `preclib.o`)

```
type direction = (vertical, horizontal);
function dimension_x(): longint;
function dimension_y(): longint;
procedure cut(dir: direction; position: longint);
```

Включете следния ред във вашия изходен код `rec.pas`:

```
uses preclib;
```

За да компилирате програмата, копирайте файловете `preclib.o` и `preclib.ppu` в директорията, където се намира вашият изходен код и изпълнете следната команда:

```
ppc386 -O2 -XS rec.pas
```

Файлът `prec.pas` дава един пример, как да използвате библиотеката `preclib`.

- **GNU C/C++ Library** (creclib.h, creclib.c)

```
typedef enum __direction {vertical, horizontal} direction;
int dimension_x();
int dimension_y();
void cut(direction dir, int position);
```

Включете следния ред във вашия изходен код (rec.c или rec.cpp):

```
#include "creclib.h"
```

За да компилирате програмата, копирайте файла creclib.c и creclib.h в

директорията, където се намира вашия изходен код и изпълнете следната команда:

```
gcc -O2 -static rec.c creclib.c -lm
```

или:

```
g++ -O2 -static rec.cpp creclib.c -lm
```

Файлът crec.c дава един пример, как да използвате библиотеката.

Пример за игра

Следва пример за взаимодействие между вашата програма и библиотеката на журито. Примерът показва развитието на проста игра. Играта започва с правоъгълник с размери 4×3. За тези начални данни съществува печеливша стратегия за вашата програма.

Вашата програма извиква	Какво се случва
dimension_x()	връща 4
dimension_y()	връща 3
cut(vertical, 1)	вашият разрез се записва и на противника се дава правоъгълник с размери 3×3. Противникът разрязва така, че остава правоъгълник с размери 3×2 и управлението се предава на вашата програма.
dimension_x()	връща 3
dimension_y()	връща 2
cut(horizontal, 1)	вашият разрез се записва и на противника се дава правоъгълник с размери 3×1. Противникът разрязва така, че остава правоъгълник с размери 2×1 и управлението се предава на вашата програма.
dimension_x()	връща 2
dimension_y()	връща 1
cut(vertical, 1)	вашият разрез дава правоъгълник с размери 1×1. Вие побеждавате и вашата програма автоматично се прекратява.