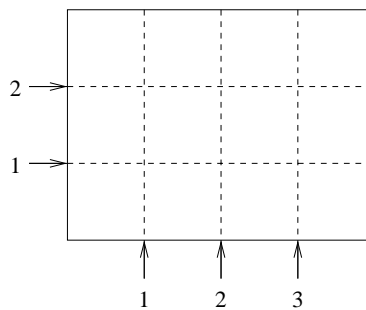




## 第2日 (Monday, 22-08-2005) Rectangle Game

メモリ制限 : 32MB 制限実行時間 : 14 秒

次のような2人ゲームを考える．大きさが  $x \times y$  の長方形が与えられ ( $x$  と  $y$  は正の整数) , 2人のプレイヤーは交互に自分の手を打つ．1回の手で行なうことは, 1つの長方形を縦または横に切って2つの長方形に分割することである．分割してできる長方形の大きさは正でなければならない．



大きさ  $4 \times 3$  の長方形の分割の仕方として可能なもの．

1回分割を行なうと, 小さいほうの長方形は捨てられ, もう一方の長方形が次のプレイヤーに渡される (もし, 同じ大きさの長方形に分割されたときは, どちらか1つだけが捨てられる) . 大きさ  $1 \times 1$  の長方形を受け取ったプレイヤーは (長方形をそれ以上分割できないので) 負けである．

あなたがやらなければならないことは, この長方形ゲームを行なうプログラムを書くことである．プログラムを書くにあたっては, このゲーム用に用意されている特別なライブラリを使わなければならない．このライブラリには以下の関数が用意されている :

- ・ `dimension_x()` と `dimension_y()` は長方形の大きさを返す．ゲームの最初に与えられる長方形の大きさは 1 から 100 000 000 の範囲内にある．少なくとも1つの辺は長さが1より大きい．さらに, テストデータの 50% において, 長方形の大きさは 25 を超えない．

- ・ 別の手続き `cut(dir, position)` もある．あなたのプログラムは `cut(dir, position)` を呼ぶことによってゲームを進めなければならない．パラメータの `dir` と `position` は, 分割の方向と位置をそれぞれ表わす．パラメータ `dir` は2つの値 `vertical` または `horizontal` のどれかでなければならない．もし `dir = vertical` のときには分割は垂直方向であり, パラメータ `position` は分割の  $x$  座標を表わす (上の図を参照のこと) .  $1 \leq position \leq dimension_x() - 1$  でなければならない．もし, `dir = horizontal` のときには分割は水平方向であり, パラメータ `position` は分割の  $y$  座標を表わす．  $1 \leq position \leq dimension_y() - 1$  でなければならない．

あなたのプログラムはスタートすると, このゲームにおける一人のプレイヤーとして動作する．あなたのプログラムがゲームの先手である — すなわち, 最初に与えられた長方形を分割する．あなたのプログラムが手続き `cut` を呼ぶと, あなたの手は記録され, 制御は相手のプレイヤーに

移る．相手のプレーヤーが手を打つと，制御はあなたのプログラムに戻る．関数 `dimension_x()` と `dimension_y()` によって与えられる値はあなたの手とそれに対する相手の手が打たれた結果残った長方形の大きさである．あなたのプログラムが勝つか，負けるか，あるいは誤った動作をする（すなわち，正しくないパラメータで手続き `cut` を呼ぶ）と，すぐに実行は強制終了させられる．あなたのプログラムは自動的に終了するので，それまでは可能な限り手を打ち続けなければならない．あなたのプログラムが必ず勝てる（先手必勝法がある）ようなテストデータしか採点に使われないと仮定してよい．

あなたのプログラムはファイル入出力を行なってはいけなし，標準入出力を使ってもいけない．また，自分のプログラム領域以外のメモリーは一切書き換えてはならない．これらのどれに違反しても，失格となる．

## 1 Experimentation

ライブラリーを試しに試してみるができるように，あなたには対戦相手のプレーヤーのライブラリが一例として与えられる．そのソースファイルは `preclib.pas`，`creclib.c` と `creclib.h` である．これらは非常に単純な戦略を用いている．あなたが自分のプログラムを走らせると，この単純な戦略に従って手を打つプレーヤーを相手に戦うことになる．もっと強い対戦相手が欲しかったら，これらのファイルを修正して使っても良い．ただし，採点のときにあなたのプログラムが相手にするのはこれとは別のプログラムである．

テスト用のインターフェースを使ってあなたのプログラムを提出すると，それはまだ修正されていない対戦相手プログラムのライブラリーと一緒にコンパイルされる．提出した入力ファイルはあなたのプログラムの標準入力に与えられる．この入力ファイルは 2 行でなければならず，各行には 1 つずつ整数が書いてあること．最初の行はゲーム開始時の長方形の横幅，2 行目はその縦幅であること．対戦相手の例示プログラムはこれらの大きさを読み込む．

もし，あなたがライブラリ `preclib.pas` の実装部を書き換えたなら，次のコマンドを使って再コンパイルすること：`ppc386 -O2 preclib.pas`．

このコマンドはファイル `preclib.o` と `preclib.ppu` を生成する．これらのファイルはあなたのプログラムをコンパイルするために必要であり，あなたのプログラムが置いてあるディレクトリに置いておかなければならない．ライブラリ `preclib.pas` のインターフェース部分を書き換えないようにされたい．

もし，ライブラリ `creclib.c` を修正したときは，それを（`creclib.h` と一緒に）あなたのプログラムが置いてあるディレクトリに置くことを忘れないように．それらはコンパイルに必要である．ファイル `creclib.h` を書き換えなないこと．

上述のライブラリの使い方を例示するために 2 つの簡単なプログラム `crec.c` と `prec.pas` を与える（これらは正解のプログラムではないことに注意されたい．）これらは次のようなコマンドによりコンパイルすることができる：

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

## 2 Library

次の機能を持つライブラリが与えられる：

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

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

次のステートメントをあなたのソースファイル `rec.pas` に含めること：

```
uses preclib;
```

あなたのプログラムをコンパイルするには、ファイル `preclib.o` と `preclib.ppu` をあなたのソースファイルが置いてあるディレクトリにコピーして、次のコマンドを投入すること：

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

ファイル `rec.pas` には、ライブラリ `preclib` の使い方が例示されている。

- **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);
```

次のステートメントをあなたのソースファイル (`rec.c` または `rec.cpp`) に含めること：

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

あなたのプログラムを完成するためには、ファイル `creclib.c` と `creclib.h` をあなたのソースファイルが置いてあるディレクトリにコピーし、次のコマンドを投入してコンパイルすること：

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

または

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

ファイル `rec.c` には C のライブラリの使い方が例示されている。

### 3 Sample interaction

あなたのプログラムと採点プログラムとのやり取りの例を以下に示す．これによって，ゲームがどのように進められるかがわかるであろう．この例では，ゲームは大きさ  $4 \times 3$  の長方形で行なわれる．この例には先手必勝法が存在する．

あなたのプログラムからの呼び出し	何が起こるか
<code>dimension_x()</code>	4 が返ってくる
<code>dimension_y()</code>	3 が返ってくる
<code>cut(vertical, 1)</code>	あなたの行なった分割は記録され， $3 \times 3$ 長方形が対戦相手に渡され，相手はそれを $3 \times 2$ の長方形に分割した．このあと，制御はあなたのプログラムに戻る
<code>dimension_x()</code>	3 が返ってくる
<code>dimension_y()</code>	2 が返ってくる
<code>cut(horizontal, 1)</code>	あなたの行なった分割は記録され， $3 \times 1$ の長方形が対戦相手に渡され，相手はそれを $2 \times 1$ の長方形に分割した．このあと，制御はあなたのプログラムに戻る
<code>dimension_x()</code>	2 が返ってくる
<code>dimension_y()</code>	1 が返ってくる
<code>cut(vertical, 1)</code>	あなたの行なった分割によって $1 \times 1$ の長方形ができるので，あなたの勝ち．あなたのプログラムは自動的に終了する

(訳：守屋悦朗)