

## Task: REC Rectangle Game

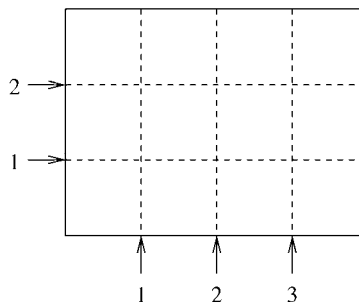
Country: ESP



Día 2. Archivo fuente `rec.*`

Memoria disponible: 32 MB. Tiempo máximo de ejecución: 14 s\*.

Consideremos un juego de dos jugadores. Se les da a los jugadores un rectángulo  $x \times y$  (donde  $x$  e  $y$  son enteros positivos). Los jugadores juegan alternadamente. Un movimiento consiste en dividir un rectángulo en dos rectángulos con un solo corte horizontal o vertical. Los rectángulos resultantes deben tener dimensiones enteras positivas.



Posibles cortes de un rectángulo  $4 \times 3$ .

Después de cada corte, el rectángulo más pequeño (que es el de menor área) se descarta y el otro es pasado al otro jugador. Si el rectángulo se corta en dos mitades iguales, entonces una de las mitades es descartada. El jugador que reciba un rectángulo  $1 \times 1$ , y por lo tanto quede incapacitado para hacer una jugada, pierde el juego.

Tu tarea es escribir un programa que juegue y gane en el juego rectangular. El programa debe usar una librería especial para jugar. La librería le proporciona las funciones `dimension_x()` y `dimension_y()` que devuelven las dimensiones del rectángulo. Las dimensiones iniciales del rectángulo son enteros entre 1 y 100 000 000. Al menos una dimensión es mayor que 1. Aún más, en el 50% de los casos de prueba las dimensiones no excederán 25.

También hay un procedimiento `cut(dir, position)`, el cual debe ser llamado por tu programa para hacer jugadas. Los parámetros `dir` y `position` describen la dirección y la posición del corte, respectivamente. El parámetro `dir` debe tener uno de los dos valores: `vertical` u `horizontal`. Si `dir = vertical`, entonces el corte es vertical, y el parámetro `position` especifica la coordenada  $x$  del corte (vea la figura anterior) y debes asegurarte de que  $1 \leq position \leq dimension_x() - 1$ . Si `dir = horizontal`, entonces el corte es horizontal, y el parámetro `position` indica la coordenada  $y$  del corte y debes asegurarte de que  $1 \leq position \leq dimension_y() - 1$ .

Cuando tu programa empieza, actuará como jugador de un juego. Tu programa juega primero —debe cortar el rectángulo original—. Cuando tu programa llama al procedimiento `cut`, tu movimiento queda registrado y el control pasa al programa de tu oponente. Después de que tu oponente juegue, el control vuelve a tu programa. Los valores devueltos por `dimension_x()` y `dimension_y()` reflejarán el resultado de tu jugada y de la de tu oponente. Tu programa acabará cuando tu programa gane, pierda o haga una jugada ilegal (por ejemplo, llamar a `cut` con parámetros inválidos). La finalización de tu programa es un proceso automático, por lo tanto, tu programa debe seguir haciendo jugadas tanto

\* Puedes asumir que durante la calificación la sobrecarga de la librería no excederá 4s.

como sea posible. Puedes asumir que para los casos de prueba siempre existe una estrategia ganadora para tu programa.

Tu programa no debe leer o escribir en ningún archivo, no debe usar entrada/salida estándar y no debe tratar de modificar ningún lugar de memoria fuera de él. Incumplir cualquiera de estas reglas implica la descalificación.

## Experimentación

Puedes experimentar con la librería, te proporcionan librerías ejemplo de oponentes: sus fuentes son los archivos `preclib.pas`, `creclib.c` y `creclib.h`. Las librerías pueden ser bajadas de <http://contest/>. Ellas implementan una estrategia muy simple. Cuando ejecutes tu programa, estará jugando contra estos jugadores simples. Puedes modificarlas, y prueba tu programa contra un mejor oponente. Sin embargo, durante la evaluación tu programa estará jugando contra un oponente diferente.

Cuando envíes tu programa usando la interfase TEST, será compilado con el ejemplo no modificado de la librería oponente. El archivo de entrada enviado será dado a la entrada estándar de tu programa. El archivo de entrada debe consistir de dos líneas, cada una conteniendo un entero. La primera línea debe contener el ancho inicial y la segunda línea debe contener la altura inicial del rectángulo. Estas dimensiones son leídas por la librería ejemplo del oponente.

Si modificas la parte de implementación de la librería `preclib.pas`, por favor recompíla usando el comando: `ppc386 -O2 preclib.pas`. Este comando produce los archivos `preclib.o` y `preclib.ppu`. Estos archivos son necesarios para compilar su programa y deberían ser colocados en el directorio, donde esta ubicado su programa. Por favor no modifiques la parte de interface de la librería `preclib.pas`.

Si modificas la librería `creclib.c`, por favor recuerda colocarla (junto con `creclib.h`) en el directorio donde esta ubicado tu programa —se necesitan para compilarlo—. Por favor no modifiques el archivo `creclib.h`.

También te dan dos programas simples ilustrando el uso de las librerías anteriores: `crec.c` y `prec.pas`. (Por favor, recuerda, estos programas no son soluciones correctas.) Puedes compilarlas usando los siguientes comandos:

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

## Librería

Te dan una librería que proporciona la siguiente funcionalidad:

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

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

Incluya la siguiente sentencia en su archivo `rec.pas`:

```
uses preclib;
```

Para compilar su programa, copie los archivos `preclib.o` y `preclib.ppu` al directorio donde esta su programa y ejecute el siguiente comando:

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

El archivo `prec.pas` da un ejemplo de como usar la librería `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);
```

Incluye la siguiente sentencia en tu archivo fuente(rec.c or rec.cpp):

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

Para compilar tu programa, copie los archivos preclib.c y preclib.h al directorio donde esta tu programa y ejecute el siguiente comando:

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

ó:

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

El archivo crec.c proporciona un ejemplo de como usar la librería en C.

### Interacción ejemplo

A continuación hay un ejemplo de interacción entre tu programa y la librería evaluadora. Muestra como puede proceder un juego de ejemplo. El juego comienza con un tablero 4×3. Hay una estrategia ganadora para esta posición.

Llamadas de tu programa	¿Qué pasa?
dimension_x()	devuelve 4
dimension_y()	devuelve 3
cut(vertical, 1)	tu corte es registrado y un rectángulo 3×3 es pasado a tu oponente, quien lo corta en un tablero 3×2, después de esto, el control es pasado de nuevo a tu programa
dimension_x()	devuelve 3
dimension_y()	devuelve 2
cut(horizontal, 1)	tu corte es registrado y un rectángulo 3×1 es pasado a tu oponente, quien lo corta en un tablero 2×1, después de esto, el control es pasado de nuevo a tu programa
dimension_x()	devuelve 2
dimension_y()	devuelve 1
cut(vertical, 1)	tu corte da un rectángulo 1×1, por lo tanto, ganas, tu programa acaba automáticamente