Competencias de programación

UNR

Fernando Fiori – fiorifj@gmail.com
Sebastián Zimmermann – zimmermannsebas@gmail.com

(Clase adaptada de la de Pablo Zimmermann adaptada de la de Joaquín Rodrigues y Fidel I. Schaposnik)

30 de Mayo de 2019

Contenidos

- Introducción a las competencias de programación
- Problema de ejemplo
  - El enunciado
  - El código
- Estrategias para la competencia
- Algunas cuestiones organizativas...

Introducción

ACM International Collegiate Programming Contest

Es una competencia en equipos de 3 estudiantes universitarios. Prima el trabajo en equipo, el análisis de problemas y el desarrollo rápido de algoritmos.

Introducción

ACM International Collegiate Programming Contest

¿En qué consiste?

- 3 participantes por equipo;
- 1 computadora;
- $\sim$ 12 problemas;
- 5 horas de competencia.
- 4 Lenguajes para usar: C, C++, Java, Python.

Introducción

ACM International Collegiate Programming Contest

¿En qué consiste?

- 3 participantes por equipo;
- 1 computadora;
- $\sim$ 12 problemas;
- 5 horas de competencia.
- 4 Lenguajes para usar: C, C++, Java, Python.

NO se puede

- conectarse a internet;
- usar material electrónico alguno;
- intentar hacer trampa de cualquier tipo.

SÍ se puede

- consultar material impreso;
- utilizar librerías estándar de C++ ó Java.

Introducción (cont.)

¿Quién gana?

- El equipo que resolvió más problemas;
- Si hay empate, el equipo que lo hizo más rápido y con menos envíos erróneos.

Ejemplo con 2 equipos y 3 problemas (A, B y C):

- Equipo 1:
  - Problema A a los $15'$ $\Longrightarrow$ Yes
  - Problema B a los $32'$ $\Longrightarrow$ No
  - Problema B a los $40'$ $\Longrightarrow$ Yes
- Equipo 2:
  - Problema A a los $12'$ $\Longrightarrow$ Yes
  - Problema B a los $25'$ $\Longrightarrow$ No
  - Problema C a los $45'$ $\Longrightarrow$ Yes
  - Problema B a los $59'$ $\Longrightarrow$ No

Introducción (cont.)

¿Quién gana?

- El equipo que resolvió más problemas;
- Si hay empate, el equipo que lo hizo más rápido y con menos envíos erróneos.

Ejemplo con 2 equipos y 3 problemas (A, B y C):

- Equipo 1:
  - Problema A a los $15'$ $\Longrightarrow$ Yes
  - Problema B a los $32'$ $\Longrightarrow$ No
  - Problema B a los $40'$ $\Longrightarrow$ Yes
- Equipo 2:
  - Problema A a los $12'$ $\Longrightarrow$ Yes
  - Problema B a los $25'$ $\Longrightarrow$ No
  - Problema C a los $45'$ $\Longrightarrow$ Yes
  - Problema B a los $59'$ $\Longrightarrow$ No

El equipo 1 hizo $A + B = 2$ problemas en $15' + 40' + 20' = 75'$.
El equipo 2 hizo $A + C = 2$ problemas en $12' + 45' = 57'$.

Introducción (cont.)

¿Cómo es un problema?

- Una historia en la que se describe el problema a resolver.
- La **entrada** explica cómo nos describen la situación y da límites a los tamaños de las variables del problema.
- La **salida** explica cómo debemos presentar nuestra respuesta.
- Algunos **ejemplos** con instancias particulares del problema para entender mejor la situación.

Introducción (cont.)

¿Cómo es un problema?

- Una historia en la que se describe el problema a resolver.
- La **entrada** explica cómo nos describen la situación y da límites a los tamaños de las variables del problema.
- La **salida** explica cómo debemos presentar nuestra respuesta.
- Algunos **ejemplos** con instancias particulares del problema para entender mejor la situación.

¿Qué significa resolverlo?

- Diseñar un algoritmo capaz de encontrar la respuesta buscada.
- Lograr que el algoritmo sea suficientemente **rápido** para el tamaño de los problemas que se nos van a presentar.
- Implementar el algoritmo (¡sin errores!) en **un solo archivo de código fuente**.

Introducción (cont.)

Notas al margen:

- La solución más simple de un problema puede no ser la que necesitamos (tener en cuenta "velocidad" del algoritmo, SPOILER EDyA: formalmente se le llama "complejidad").
- Los límites de las variables que describen al problema nos indican cuán eficiente debe ser nuestra solución. (Apuntar a menos de $\sim 2 \times 10^8$ operaciones).
- La entrada siempre se ajusta a las especificaciones.
- Si nuestro algoritmo falla con algún caso particular, podemos suponer que el juez lo sabe y no lo va a dejar pasar.
- La entrada se lee del standard input (`cin`/`scanf`).
- La salida se imprime al standard output (`cout`/`printf`).

Introducción (cont.)

¿Cómo nos evalúan?

Introducción (cont.)

¿Cómo nos evalúan? Enviamos nuestro código fuente a los jueces:

- Compilan nuestro programa:
  - Si no compila, "Compilation Error".
- Lo ejecutan con los casos de prueba **secretos** como entrada:
  - si tarda "demasiado" ($>$ 2 seg aprox), "Time Limit Exceeded";
  - si la salida no es **exactamente** la correcta, "Wrong Answer" (aunque algunos jueces piolas te avisan con un "Presentation Error" si sólo te comiste algún espacio o salto de línea);
  - si la salida es idéntica a la que ellos tienen, "Accepted".
- Podemos reintentar tantas veces como queramos hasta obtener Accepted.
- Sólo los problemas correctamente resueltos dan puntos (y tiempo de penalidad). Corolario: batir fruta no es más perjudicial que no hacer nada respecto a un problema (en términos de scoreboard).

¿Por qué está bueno participar?

¿Por qué está bueno participar?

- Motivación personal.
- Trabajo en equipo.
- Aprendizaje en conjunto a una comunidad.
- Desestructuración del aprendizaje.
- Compartir internacionalmente, generar contactos que tengan intereses parecidos.
- Tomar las cosas aprendidas como herramientas: combinarlas y adaptarlas a la resolución de un problema concreto.
- Existen varias formas de hacer las cosas: autocrítica y mejoramiento constantes.
- Desarrolla mucho la creatividad y la intuición lógico-matemática.
- La posibilidad de viajar a distintos lugares del mundo.

URI Online Judge | 1001

Extremadamente Básico

Adaptado por Neilor Tonin, URI Brazil

Timelimit: 1

Leer 2 valores enteros y almacenarlos en variables, llamadas **A** y **B** , y sumarlas, asignando el resultado a la variable **X**. Mostrar **X** como se muestra abajo. No muestre ningún mensaje además de lo que se especifica y no olvide imprimir un salto de línea luego del resultado, de otra forma Ud. recibirá un "*Presentation Error*".

Entrada

El archivo de entrada contiene 2 valores enteros.

Salida

Muestre la variable **X** de acuerdo al siguiente ejemplo, con un espacio en blanco antes y después del signo igual. 'X' está en mayúsculas y Ud. tiene que imprimir un espacio antes y después del signo '='.

Ejemplos de Entrada	Ejemplos de Salida
10 9	X = 19
-10 4	X = -6
15 -7	X = 8

Problema Simple (código)

```cpp
#include<bits/stdc++.h>
using namespace std;

int main ()
{
   int a,b;
   cin >> a >> b;
   cout << "X = " << a+b << endl;
   return 0;
}
```

El código para el problema

Problema Simple (código)

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main ()
5  {
6    int a,b;
7    cin >> a >> b;
8    cout << "X = " << "19" << endl;
9    return 0;
10 }
```

Que pasa si enviamos esto??

Estrategias para la competencia

- Debemos leer todos los problemas e identificar lo que se nos pide en cada uno.
- Los problemas pueden ser de distinto tipo: grafos, programación dinámica, geometría, strings (o cadenas), estructuras de datos, teoría de números, etc.
- Debemos identificar los problemas más fáciles para resolverlos primero (pueden ser muchísimo más sencillos que los más difíciles).
- Cuanto más sencillo sea el código de la solución, tardaremos menos y tendremos menos posibilidad de cometer errores.

Cuestiones organizativas

Recursos útiles:

- urionlinejudge.com.br (base brasilera de problemas)
- coj.uci.cu (base cubana de problemas)
- codeforces.com (competencias individuales una vez x semana)
- tg://join?invite=AM7mwwCnEZu4kCTnVMCO8g (grupo de Telegram de ACM-ICPC en la UNR)
- Libro de Halim "Competitive Programming 3: The New Lower Bound of Programming Contests".
- Clases teóricas training camps pasados: 2016, 2015, 2014.

Fechas

- 8 al 19 de Julio $\rightarrow$ Training Camp (FAMAF - Córdoba)
- Septiembre $\rightarrow$ Torneo Argentino de Programación (TAP)
- Noviembre $\rightarrow$ Regional Latinoamericana (Clasificados del TAP)

Preguntas?

Contest

Link Contest:

https://www.urionlinejudge.com.br/judge/es/tournaments/rank/2190