



Taller de Programación

Parte I

Consideraciones para el uso del Juez en Línea.

Jhonny Felípez Andrade
jrfelizamigo@yahoo.es



Contenido

- La página <http://www.programming-challenges.com>
- La página <http://acm.uva.es/problemset>
- Compiladores del C y JAVA.
 - Como compilar.
 - Como enviar al juez en línea.
- Entradas y salidas estándar.
- Sugerencias sobre programación.



La página

<http://www.programming-challenges.com>



La página

<http://acm.uva.es/problemset>



Compiladores del C y JAVA.

- Como compilar.
- Como enviar al juez en línea.



Entradas y salidas estándar.



Entrada/Salida Estándar en C

```
#include <stdio.h>
main() {
    long p,q,r;
    while (scanf("%ld %ld",&p,&q) != EOF)
    {
        if (q>p) r=q-p;
        else r=p-q;
        printf("%ld\n",r);
    }
}
```

Entrada/Salida Estándar en C++

```
#include <iostream.h>
void main() {
    long a,b,c;
    while (cin>>a>>b) {
        if (b>a) c=b-a;
        else c=a-b;
        cout << c << endl;
    }
}
```




Entrada/Salida Estándar en JAVA

```
import java.io.*;  
import java.util.*;
```

```
class Main
```

```
{
```

```
    static String readLn (int maxLg) // funcion util para leer del stdin
```

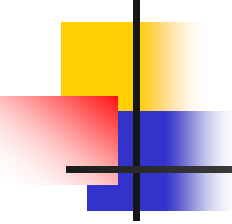
```
{
```

```
    byte lin[] = new byte [maxLg];
```

```
    int lg = 0, car = -1;
```

```
    String line = "";
```

Entrada/Salida Estándar en JAVA. Cont.



```
try
{
    while (lg < maxLg)
    {
        car = System.in.read();
        // las proximas lineas se han agregado para permitir saltar
        // caracter 13 del fin de linea en sistemas Windows.
        if (car==13)
        {
            car = System.in.read();
            car = -1;
        }
        if ((car < 0) || (car == '\n')) break;
        lin [lg++] += car;
    }
}
```

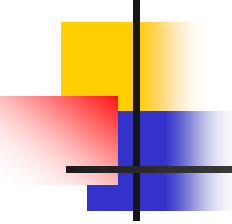


Entrada/Salida Estándar en JAVA. Cont.

```
catch (IOException e)
{
    return (null);
}
```

```
if ((car < 0) && (lg == 0)) return (null); // eof
return (new String (lin, 0, lg));
}
```

Entrada/Salida Estándar en JAVA. Cont.

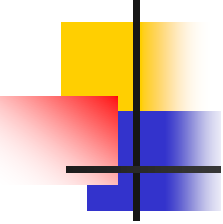


```
void Begin ()
{
    String s;
    StringTokenizer st;
    long p, q, r;
    while ((s = Main.readLn(255)) != null)
    {
        st = new StringTokenizer(s);
        p = Long.parseLong(st.nextToken());
        q = Long.parseLong(st.nextToken());
        if (q>p) r = q - p;
        else r = p - q;
        System.out.print(r + "\n");
    }
}
```



Entrada/Salida Estándar en JAVA. Cont.

```
public static void main(String[] args) // punto
    de entrada del SO
{
    Main myWork = new Main();    // crea una
    instancia dinamica
    myWork.Begin();               // punto de
    entrada
}
}
```



```
import java.io.*;  
import java.util.*;  
class Main {  
void Begin() {  
    Scanner in = new Scanner(System.in);  
    long p, q, r;  
    while (in.hasNext()) {  
        p = in.nextLong();  
        q = in.nextLong();  
        if (q > p)  
            r = q - p;  
        else  
            r = p - q;  
        System.out.print(r + "\n");  
    }  
}
```

```
public static void main(String[] args) // punto de entrada del  
SO
```

```
{
```

```
    Main2 myWork = new Main2(); // crea una instancia  
dinamica
```

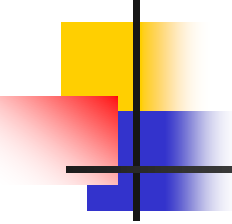
```
    myWork.Begin(); // punto de entrada
```

```
}
```

```
}
```



Sugerencias sobre programación.



Sugerencias sobre Programación

- Escriba los comentarios. Comience los programas y funciones escribiendo unas pocas líneas explicando que es lo que se supone que debe hacer.



Sugerencias sobre Programación. Cont.

- Documente todas las variables.

```
typedef struct {  
    int q[QUEUESIZE+1];    /* cuerpo de la cola */  
    int first;             /* lugar del elemento inicial */  
    int last;              /* lugar del elemento final */  
    int count;             /* cantidad de elementos en la cola */  
} queue;
```



Sugerencias sobre Programación. Cont.

- Use constantes simbólicas. Declárelas al principio.

```
#define NAMELENGTH      30      /* longitud máxima del nombre */  
#define ARRAYLENGTH    100     /* longitud máxima del vector */  
#define PI              3.141593654 /* valor pi */
```



Sugerencias sobre Programación. Cont.

Utilice tipo enumerados. Veamos este ejemplo donde se representan los palos (tréboles, diamantes, corazones) de un mazo de cartas.

```
switch(cursuit) {  
    case 'C':  
        newcard.suit = C;  
        break;  
    case 'D':  
        newcard.suit = D;  
        break;  
    case 'H':  
        newcard.suit = H;  
        break;  
    case 'S':  
        newcard.suit = S;  
        break;  
    . . .  
}
```

Aquí el uso de la variables enumeradas no aporta nada, mas que nuevas oportunidades de cometer errores.



Sugerencias sobre Programación. Cont.

- Use subrutinas para evitar la repetición de código.

```
. . .
while (c != '0') {
    scanf ("%c", &c);
    if (c == 'A') {
        if (row-1 >= 0) {
            temp = b[row-1][col];
            b[row-1][col]=' ';
            b[row][col] = temp;
            row = row - 1;
        }
    }
    else if (c == 'B') {
        if (row+1 <= BOARDSIZE-1) {
            temp = b[row-1][col];
            b[row-1][col]=' ';
            b[row][col] = temp;
            row = row - 1;
        }
    }
}
. . .
```



Bibliografía

- Fundamentos de Programación, Jorge Teran Pomier, 2006.
- Concursos Internacionales de Informática y Programación, Miguel Revilla, Universidad de Valladolid, 2006.



Taller de Programación

Gracias