

APIX

## Concurso de programación Xtrelan 2011

Juez Online Mooshak



### Colaboran

Grupo de Ingeniería de Medios



Grupo de Ingeniería de Software Quercus

Software Engineering Group  
QUERCUS



## Contenido

1. Bases del concurso.....	2
2. Premios.....	3
3. Registro.....	3
2. Utilización de Mooshak.....	4
3. Algunos consejos para resolver los problemas correctamente .....	6
4. Ejemplo de problema resuelto .....	7
5. Errores frecuentes .....	9

La prueba de programación de la **Xtrelan2011** se realizará utilizando un juez automático llamado Mooshak. Esta es una aplicación que permite comprobar si un programa resuelve correctamente el problema planteado. Os recomendamos que preparéis la prueba “entrenando” con esta aplicación. De esa forma, el día del concurso, todo resultará más cómodo. A continuación, encontraréis unas sencillas indicaciones para utilizar Mooshak.

### 1. Bases del concurso.

Estas bases estarán sujetas a modificación por la organización del evento hasta el día de comienzo del mismo. Por favor, presta atención a ellas en la web oficial de la party.

1. El objetivo del concurso es divertirse programando. ¿Qué otro podría ser?
2. La competitividad es la base de los concursos. Demuestra que eres mejor programador.
3. Las herramientas (en este caso los lenguajes) son menos importantes que los resultados. Escoge tu lenguaje preferido: **ANSI C** (gcc 4.4.1), **Java6** (Sun JDK 1.6.0\_18) o **C++** (g++ 4.4.1). Compites en igualdad de condiciones.
4. El concurso consiste en resolver problemas pequeños, aunque no necesariamente simples, programando una lógica entre una entrada de texto y una salida de texto.
5. El concurso se rige por las reglas del ACM ICPC (*International Collegiate Programming Contest*), que se describen a continuación:
  - En primer lugar, los participantes se ordenarán por el mayor número de problemas resueltos correctamente, es decir, aceptados por el juez on-line.
  - En caso de empate en el número de problemas resueltos, se sumarán para cada problema resuelto el tiempo tardado desde el inicio del concurso hasta el instante en que dicho problema es aceptado por el juez on-line. Los participantes con el mismo número de problemas resueltos se ordenarán atendiendo a dicha suma de tiempos, de menor a mayor.
  - En la anterior suma, cada envío no aceptado realizado a un problema sumará una penalización de 20 minutos, siempre que el problema sea después aceptado (es decir, los envíos incorrectos no penalizan si el problema no es resuelto finalmente).
  - La clasificación estará disponible en el propio juez on-line durante el

desarrollo de la prueba y se ocultará automáticamente una hora antes de la finalización de la misma. No obstante, esta clasificación podría sufrir cambios si los jueces detectan alguna irregularidad (como copia entre equipos, fallos en el juicio automático, etc.).

6. El slogan del torneo es *Haz difícil lo fácil*. No lo olvides.
7. Resolver un problema implica hacerlo de la mejor manera. La eficiencia es requisito indispensable de los buenos programas.
8. Los tiempos de los problemas cuentan desde el inicio del concurso. Sólo tú decides el orden en que los intentas resolver.
9. El concurso estará abierto para los inscritos desde las 14:00 horas del día 29 de abril hasta las 14:00 horas del día 1 de mayo. Durante este periodo de tiempo los enunciados de los problemas **estarán disponibles en el juez on-line de Xtrelan 2011. Atento, los minutos corren desde el momento de apertura del concurso.**
10. La filosofía consiste en que lo que se descubre es mejor que lo que se gana. ¿Qué te parece?
11. Para participar en el concurso será necesario inscribirse previamente en la intranet de Xtrelan. No se admitirán inscripciones a medio concurso.
12. Los enunciados estarán en castellano o inglés, lo cual no es ningún problema para un programador experto.
13. Cada vez que recibas un resultado "Accepted" por parte del juez online, verás que en tu página de estadísticas aparece como resuelto con el segundo exacto en que se verificó. Ese es el instante que cuenta a efectos de concurso.

## 2. Premios.

**Primer premio:** iPad 2 wifi 64GB

**Segundo premio:** iPad 2 wifi 16GB

## 3. Registro.

Para poder participar en el concurso de programación de Xtrelan'2011, deberás realizar tu inscripción en la intranet de Xtrelan'2011. Allí te facilitarán tu identificación de acceso a Mooshak para el concurso.

Previo a la participación, y para que te familiarices con el uso de Mooshak, se ha creado un escenario de preparación que podrás encontrar accediendo a <http://oiex.unex.es/~mooshak>. En esta ocasión, deberás registrarte on-line haciendo "click" en la opción para registrarse.



Figura 1. Mensaje de bienvenida



A continuación deberéis rellenar un sencillo formulario:

Figura 2. Formulario para el registro en la aplicación

Utilizando este formulario, debéis introducir el nombre por el que “Mooshak” os va a conocer. Debéis tener en cuenta que Mooshak sólo aceptará letras, dígitos y el guión bajo “\_”, pero no acepta espacios en blanco, vocales con tilde, ñ's, o cualquier otro carácter “especial”. Por lo tanto, tu nombre puede ser “PepePerez” o “Pepe\_Perez” pero no “José M<sup>a</sup> Pérez”.

Inmediatamente, recibirás en tu correo electrónico un mensaje en el que se te indicará la contraseña que necesitas para utilizar la aplicación.

## 2. Utilización de Mooshak.

Una vez que tengas tu contraseña para entrar en la aplicación, deberás volver a la pantalla de entrada a “Mooshak” (Figura 1) y pulsar el botón “entrar”. A continuación deberás rellenar una nuevo formulario con tu nombre de usuario y tu contraseña.



Figura 3. Formulario de entrada a la aplicación

Si eliges la opción de invitado puedes entrar en la aplicación sin haberte registrado previamente y así podrás ver los resultados obtenidos por los diferentes participantes pero no podrás ver los enunciados de los problemas ni participar en el concurso.

Si has entrado como un usuario registrado, la pantalla que vas a encontrar es la siguiente:

#	Tiempo de Concurso	País	Equipo	Problema	Lenguaje	Resultado	Estado
8	1:51:58		Bachillerato Seux	C	Java	0 Runtime Error	final
7	0:39:55		Otros ana elena 4416	D	C++	0 Accepted	final
6	0:39:33		Otros ana elena 4416	D	C++	0 Invalid Function	final
5	0:39:10		Otros ana elena 4416	A	C++	0 Accepted	final
4	0:07:48		Otros jj 6191	C	C++	0 Accepted	final
3	0:07:27		Otros jj 6191	D	C++	0 Accepted	final
2	0:07:08		Otros jj 6191	B	C++	0 Accepted	final
1	0:06:48		Otros jj 6191	A	C++	0 Accepted	final

Figura 4. Pantalla de “trabajo” de Mooshak

En primer lugar deberás decidir cuál es el problema que quieres resolver, para ello podrás utilizar la pestaña señalada en la figura con la flecha roja, en la parte inferior de la pantalla podrás ver el enunciado de dicho problema.

Deberás resolver el problema con un programa escrito en C, C++ o Java. En el apartado siguiente te daremos algunos consejos para realizar estos programas. Una vez escrito el programa deberás enviar el fichero que hayas creado a Mooshak. La pestaña “Examinar..” que indica la flecha verde te permitirá seleccionar tu fichero con comodidad dentro del sistema de carpetas de tu ordenador. Cuando hayas seleccionado el fichero, pulsa el

botón “Enviar”. Inmediatamente verás el resultado en la parte inferior de la pantalla. Si el programa es correcto aparecerá el mensaje “Accepted” en color verde y si no lo es aparecerá un mensaje en color rojo.

Si quieres tener más información sobre estos mensajes de error, pulsa la tecla “Ayuda” y encontrarás las respuestas a las preguntas más frecuentes.

La opción “Ver” te permite ver el enunciado del problema seleccionado.

La opción “Preguntar” te permite enviar una pregunta al administrador de la aplicación.

La opción “Salir”, evidentemente, te permite salir de la aplicación.

### 3. Algunos consejos para resolver los problemas correctamente

Para que Mooshak sepa qué lenguaje has utilizado para resolver el problema, debes darle al fichero que contiene la solución un nombre con la extensión adecuada.

Lenguaje	Extensión	Ejemplo de nombre para el fichero
C	c	solucion.c
C++	cpp	solucion.cpp
Java	java	solucion.java

Para comprobar si un programa resuelve correctamente el problema propuesto, Mooshak lo ejecuta y compara la salida con una serie de casos de pruebas que están previamente definidos. Por este motivo, es muy importante que la entrada y la salida del programa se ajusten a lo indicado en el enunciado.

- No dejes espacios en blanco o líneas en blanco que no estén indicadas en el enunciado.
- Cuida los caracteres utilizados en los mensajes. Por ejemplo, no es lo mismo escribir:
  - “5 es primo”
  - “5 es Primo”
  - “5 es primo”
- No es necesario utilizar ningún archivo de entrada o salida. La entrada se lee desde teclado, es decir, desde la entrada estándar. La salida se escribe en la pantalla (salida estándar). Ejemplos de instrucciones de entrada y salida:

Lenguaje	instrucción de entrada	instrucción de salida
C	<code>scanf( “%d” , &amp;A);</code>	<code>printf ( “%d\n” , A*2);</code>
C++	<code>cin &gt;&gt; A;</code>	<code>cout &lt;&lt; A*2 &lt;&lt; endl;</code>
Pascal	<code>read (A);</code> o <code>readln(A);</code>	<code>writeln (A*2);</code>
java	<code>usar System.in.readLine();</code>	<code>System.out.println(A*2);</code>

- No debéis utilizar mensajes que actúen como interfaces de usuario como por

ejemplo “Introduzca el dato: “.

- No utilizéis funciones como **getch()** en C, ya que éstas esperan a que un usuario presione una tecla y el juez automático considerará que el programa está “parado”.
- También es muy importante que utiliceis el menor número posible de librerías.

Cualquier infracción a estas reglas puede representar un error para el evaluador automático. Esto puede ocurrir aunque el programa funcione correctamente con tu compilador habitual. Por lo tanto, ten en cuenta estas recomendaciones al resolver los problemas.

#### 4. Ejemplo de problema resuelto

A continuación presentamos la solución, en diferentes lenguajes de programación, de un problema muy sencillo.

##### **Enunciado del problema**

Escribir un programa que, dado un número entero, escriba el doble de dicho número.

##### **Entrada:**

La entrada consta de varios casos de prueba. La primera línea indica el número de casos. Las líneas restantes tienen un número entero cuyo doble hay que calcular.

##### **Salida:**

Por cada caso de prueba, el programa debe escribir el doble del número indicado.

Ejemplo de entrada:

4  
9  
16  
-5  
3

Ejemplo de salida:

18  
32  
-10  
6

En las soluciones que aparecen a continuación todos los programas tienen la misma estructura. Además, hemos utilizado, en todos los casos, las siguientes variables:

- **num\_datos** se utilizará para almacenar el valor de la primera línea, es decir, el número de casos de prueba
- **dato** almacenará, de forma secuencial, el resto de los números que constituyen la entrada del programa, es decir, los números cuyo doble vamos a calcular
- **i** es una variable *índice* que utilizaremos en el bucle encargado de leer y procesar los datos.

**Solución en C:**

```
#include <stdio.h>
int main() {
    int num_datos, dato, i;
    scanf( "%d" , &num_datos);
    for (i = 0; i < num_datos; i++) {
        scanf( "%d" , &dato);
        printf( "%d\n" , dato * 2);
    };
    return 0;
}
```

**Solución en C++:**

```
#include <iostream>
using namespace std;
int main() {
    int num_datos, dato;
    cin >> num_datos;
    for (int i = 0; i < num_datos; i++) {
        cin >> dato;
        cout << dato * 2 << endl;
    };
    return 0;
}
```

**Solución en Java:**

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

@SuppressWarnings("all")
public class MultiplicarPorDos {
    public static void main(String[] args) throws java.io.IOException {
        BufferedReader in;
        in = new BufferedReader(new InputStreamReader(System.in));
        int num_datos = 0;
        num_datos = Integer.parseInt(in.readLine());
        for( int i = 0; i < num_datos; i++ ) {
            int dato = Integer.parseInt(in.readLine());
            System.out.println(dato*2);
        }
    }
}
```

```
}  
}  
}
```

## 5. Errores frecuentes

Cuando envías a Mooshak un programa que resuelve un determinado problema, tu objetivo será que aparezca el mensaje “Accepted”. Eso significa que el programa ha funcionado según lo esperado. Te recomendamos que antes de enviar el programa hagas alguna prueba para comprobar que su funcionamiento es correcto ya que, en la competición, los envíos incorrectos serán penalizados.

Sin embargo, es posible que recibas un mensaje que te indica la existencia de un problema. Estos son los errores que pueden aparecer:

### Compile Time Error

El sistema no ha podido compilar el programa debido a algún error. Si haces “doble click” sobre el mensaje, podrás ver otros mensajes que explican la situación (indicando habitualmente la línea en la que se ha detectado el problema), y que te ayudarán a encontrar el error. Habríamos tenido un error de este tipo si hubiéramos enviado el siguiente código para el ejemplo del apartado 4. Un despiste al escribir el nombre de una variable hace que el compilador no la reconozca.

```
#include <iostream>  
using namespace std;  
int main() {  
    int num_datos, dato;  
    cin >> num_datossss;  
    for (int i = 0; i < num_datos; i++) {  
        cin >> dato;  
        cout << dato * 2 << endl;  
    };  
    return 0;  
}
```

El mensaje de ayuda que aparecería sería el siguiente:

error: 'num\_datossss' was not declared in this scope

Este tipo de errores se pueden evitar, compilando previamente el programa antes de enviarlo.

### Runtime Error

El programa ha sido compilado con éxito pero la ejecución ha fallado.

El siguiente ejemplo provocaría un error de este tipo. El programa es gramaticalmente correcto y se compila sin problemas. Sin embargo, la ejecución de la línea destacada en rojo (concretamente: `dato/i`) provocaría un error, ya que el primer valor que toma la variable `i` es 0 y no es posible dividir por ese número.

```
int main() {
    int num_datos, dato;
    cin >> num_datos;
    for (int i = 0; i < num_datos; i++) {
        cin >> dato;
        cout << dato / i <<endl;
    };
    return 0;
}
```

### Time Limit Exceeded

El programa ha tardado en ejecutarse un tiempo mayor que el disponible.

### Wrong Answer

El programa se ha compilado y ejecutado correctamente, pero la salida que ha generado no coincide con la esperada por el juez. Por ejemplo, el programa que aparece a continuación funciona perfectamente y, sin embargo, no resuelve el problema planteado ya que no multiplica los datos por 2 sino por 3. En este caso es posible que no se haya entendido bien el enunciado del problema a resolver.

```
int main() {
    int num_datos, dato;
    cin >> num_datos;
    for (int i = 0; i < num_datos; i++) {
        cin >> dato;
        cout << dato * 3 <<endl;
    };
    return 0;
}
```

### Presentation Error

La salida del programa está muy cerca de la salida esperada, pero hay algún pequeño detalle que las diferencia. Estás muy cerca de la solución, probablemente has escrito algún espacio o alguna línea en blanco. En el siguiente ejemplo, el problema está en la línea destacada en rojo que escribe en la pantalla un salto de línea inesperado.

```
int main() {
    int num_datos, dato;
    cin >> num_datos;
    for (int i = 0; i < num_datos; i++) {
        cin >> dato;
        cout << dato * 2 <<endl;
    };
    cout << endl;
```

```
return 0;  
}
```

**Invalid Submission**

El envío no ha sido correcto. Es posible que el fichero enviado no tenga un nombre válido.

**Output Limit Exceeded**

La salida que genera el programa es demasiado grande.

**Program Size Exceeded**

El código del programa es demasiado largo.

**Invalid function**

Se ha violado alguna regla de Mooshak, por ejemplo, utilizar librerías no estándar o terminar el programa con “return 1” o “exit 1”.

**6. Licencia y uso inapropiado.**

Queda **terminantemente prohibido** el uso de cualquier medio, dispositivo, aplicación o modificación alguna que afecte al funcionamiento normal de la plataforma del juez online, con el fin de obtener la más mínima ventaja, ya sea a través de. El uso de cualquiera de estas medidas supondrá la **expulsión inmediata** del participante en cuestión.

Los problemas a resolver serán publicados a través de la plataforma Mooshak justo a la hora de comienzo del concurso

La organización se reserva el derecho a modificar las bases del torneo sin previo aviso en función de la afluencia de participantes o posibles problemas de logística que puedan surgir. Así como a tomar las medidas adecuadas si existe algún tipo de comportamiento inapropiado de los participantes, pudiendo suponer la expulsión de los mismos del concurso.

La lectura, comprensión y aceptación completa y absoluta de todos los puntos de estas bases, es requisito imprescindible para la participación en el torneo.

Todas las marcas son propiedad de sus respectivos autores. Todos los derechos reservados.