

# Reto Latinoamérica 2020

## Programación Competitiva

### Guía Introductoria

Nivel: Secundaria, Preparatoria.

#### ¿Qué es programación competitiva?

La **programación competitiva** es un **deporte mental**, el cual se basa en resolver problemas mediante el uso de la **computadora**!

Los problemas de programación competitiva están diseñados para ser resueltos mediante el uso de **lógica, matemáticas y programación**. Al igual que cualquier combinación de estos temas para medir las habilidades de sus deportistas, pero cuidado que hay muchos caminos para la respuesta de un problema, por lo que existe un límite de tiempo para restringir que respuestas.

#### ¿Cómo es un problema de programación competitiva?

Un problema de programación competitiva te va a presentar una tarea que tienes que resolver, sin embargo, el cómo te la presentan pueden variar de problema a problema, en ocasiones siendo muy vagos de manera intencional para retar la capacidad de comprensión de sus deportistas.

Un ejemplo de dos redacciones sería:

- Muestra el resultado de  $23 * 34$ .
- José trabaja en una empresa de venta de tazas, en su empleo el cual le gusta mucho recibe un pago de 23 dólares al día los cuales siempre guarda en su alcancía. Su hijo Tomás vio que han pasado 34 días desde que su alcancía estuvo vacía y quisiera saber cuánto dinero estará dentro de la alcancía si su padre siempre guardó su paga del día y nunca saco nada. Ayuda a Tomás diciéndole cuánto dinero hay dentro de la alcancía.

Aunque los dos textos preguntan lo mismo, ambos tienen distintas maneras de hacerlo. Lo cual puede ser parte crucial al resolver un problema, ya que entre varios datos uno puede perder la noción de cuáles son los verdaderamente importantes, o no notar que no se mencionó un detalle importante intencionalmente.

Aunque el problema puede ser demasiado ambiguo o haber omitido accidentalmente un detalle muy importante, para lo cual en medio de un concurso se puede hacer una **clarificación** en la

cual los organizadores del concurso procederán a leer y a responder acorde (Ojo, si consideran que responder otorgaría una parte de la solución u opinan que el enunciado con una mejor lectura contestaría la misma pregunta podrían optar por no contestar, o dar una respuesta tipo “Vuelve a leer”).

## ¿Cómo se resuelven los problemas con programación?

Dado el ejemplo anterior puede que usar una computadora sea algo innecesario para simplemente contestar que  $23 * 34$  es 782. Al igual que muchos ejemplos más pueden parecer lo suficientemente sencillo que con solo un lápiz y una hoja de papel bastaría para resolverlos. ¡Pero la adición de la computadora nos permite poner retos más dinámicos!

José podría pasar a ganar  $A$  dólares al día, y su hijo Tomás podría pasar a consultar la alcancía después de que pasen  $B$  días. En caso de que  $A$  sea 23 y  $B$  sea 34 tendríamos el ejemplo anterior, con un resultado de 782.

Sin embargo, con un  $A$  igual a 17 y un  $B$  igual a 3, claramente nuestro resultado ya no sería 782.

Por lo que nuestra solución debería reajustarse a los nuevos valores. Aquí es donde entra el papel de los **lenguajes de programación**, los cuales son un conjunto de instrucciones que sigue una computadora para el procesamiento de datos, los cuales sin problemas pueden adaptarse a dar diferentes respuestas dependiendo de qué valores le ingresamos, los cuales llamamos **valores de entrada**, o **entrada del problema**. Una vez que tenemos nuestro programa hecho este debería producir la respuesta correcta y mostrar en pantalla la respuesta, la cual solemos llamar **valor de salida**, o **salida del problema**.

Al final la respuesta de nuestro problema no será la respuesta de solamente un caso, sino el sistema que soluciona todos los casos, en otras palabras, nuestro **código** que realizamos bajo un **lenguaje de programación**.

## Sobre lenguajes de programación:

Los lenguajes de programación tanto como el lenguaje de las personas nos indican que palabras podemos usar en qué contexto, como en español usamos un “Hola” para saludar, en inglés usamos “Hello”, y al igual que nuestro lenguaje están sujetos a contexto y a la lógica. Tanto como si no llamamos a alguien por su nombre no volteara, o decir “Adiós” como forma de saludo causaría mucha confusión. Unos ejemplos sería si quisiéramos decir “Imprime el número 5” en el lenguaje de C++ pondremos “`cout << 5;`”, y en C pondremos “`printf(“5”);`”, el conjunto de todo lo que escribimos en un lenguaje de programación es nuestro **código** el cual es importante que esté todo escrito en el mismo lenguaje no esperaras que la gente te vea como un gran comunicador si a mitad de tu frase cambias de idioma.

Al final nuestro código estará todo escrito en un lenguaje de programación y tendrá que cumplir con dar la respuesta correcta para cada entrada posible.

El siguiente es un ejemplo de un código escrito en C++ que resolvería el problema de ejemplo:

```

1  #include<iostream>
2  using namespace std;
3  int main() {
4      int A, B;
5      cin >> A >> B;
6      cout << A * B << endl;
7      return 0;
8  }
```

Existe una gran cantidad de lenguajes de programación y cada uno con sus peculiaridades. Sin embargo, en programación competitiva (a nivel preparatoria) los **únicos permitidos son C, C++ y Java**. (Las versiones de estos varían de concurso a concurso).

### ¿Ya tengo mi código, ahora qué hago?

Después de tener tu código listo ¡eso será tu respuesta del problema! Así que solo procede mandarla a un **juez en línea** para que la evalúe y te de ¡tú resultado! Pero cuidado, el juez en línea va a recibir tu código, pero tienes que especificarle bien en que lenguaje de programación está escrito para que él lo interprete.

### ¿Qué es un juez en línea?

Un juez en línea es un evaluador automático, que va a ejecutar tu programa y poner todos sus **casos de prueba** insertados de manera ordenada, los **casos de prueba** son el conjunto de todas las **entradas del problema** que se van a evaluar, ya que aunque tu código debería responder bien para cada entrada posible no significa que se va a intentar toda entrada posible.

Los casos de prueba tienden a rondar alrededor de 20, aunque hay muchos de relleno habrá unos hechos específicamente para tratar de ver si tu código es vulnerable en alguna parte. Para cada caso de prueba el juez en línea sabe cuál es la respuesta correcta, la cual va a comparar con la tuya y ver si cumplen ser iguales, aunque en algunos casos puede variar, la variación será especificada en el problema. Una vez que haya hecho todas las evaluaciones el juez en línea te va a dar una **respuesta** de la evaluación de tu código.

### ¿Qué respuestas me puede dar un juez en línea?

Un juez en línea va a probar correr tu código con sus **casos de prueba**, pero es importante saber que eso no garantiza muchas cosas, por ejemplo, puede que el código *¡ni siquiera se pueda correr!* En esos casos el juez mencionara que tu programa tuvo un error y no pudo terminar de probar los casos. O en caso de que el programa esté bien va a mostrar el deseado “*Respuesta correcta*”. Aunque cada juez en línea varía en cómo muestra sus mensajes, nos enfocaremos en el juez en línea más usado en Secundaria y Preparatoria que es OmegaUp.

OmegaUp mostrará los resultados en la parte de “*Estatus*” en su interfaz, y los posibles estados son:

1. **Respuesta Correcta:** Significa que tu programa paso correctamente ¡todos los casos de prueba!
2. **Error de Ejecución:** Tu programa en algún punto hizo algo que no debía hacer y genero un error, hay muchas formas en las que puede pasar tanto como dividir entre 0, raíz cuadrada de un número negativo o acceder a memoria a la cual no deberías acceder.
3. **Tiempo Límite Excedido:** En un punto tu programa tardó más del límite especificado por el problema, hasta podría tardar ¡tiempo infinito! Por eso el compilador después de pasar cierto tiempo detendrá el programa y no te evaluará la respuesta, no necesariamente significa que la respuesta que ibas a dar sea incorrecta.
4. **Parcialmente correcta:** Tuviste al menos un caso de prueba que diste la respuesta correcta, pero también al menos uno en el que no.
5. **Respuesta incorrecta:** No hubo ningún caso en que dieras una respuesta correcta.

Lo bueno de OmegaUp es que da calificaciones parciales, si de 10 casos hay uno que te dio un TLE (Tiempo Límite Excedido) pero los otros 9 estuvieron correctos te va a dar puntos acordes a tu desempeño, si los casos estuvieran repartidos equitativamente tendrías 90 puntos de 100, aunque aun así el mensaje sería “*Tiempo Límite Excedido*”, el cual es más descriptivo que un simple “*Parcialmente correcta*”.

Los casos de prueba pueden estar ponderados no equitativamente, y también pueden estar agrupados. Al estar agrupados hay varios grupos los cuales solo te darán los puntos del grupo si contestaste bien todos sus casos. Eso evitará que imprimir solo “Si” o “No” en problemas que son sus únicas dos respuestas te de la mitad de los puntos. Al igual que si hay casos que al autor le parezca especialmente importantes les puede dar más puntos y no tener una distribución equitativa.

*Ojo: Los jueces en línea son estrictos con las salidas, si te piden imprimir un 4, el imprimir “La respuesta es 4” lo van a considerar incorrecto, al igual que por más bonito que se vea, poner “Por favor ingrese la entrada: también se considerará incorrecto dado que no es una persona que checa las salidas, sino un programa.*

## ¿Cómo es un concurso de programación competitiva?

Estos se llevan a cabo en plataformas como OmegaUp u otras parecidas, un concurso tiene una hora de inicio y fin, usualmente estos concursos suelen durar 5 horas, puede que hasta más, depende de la cantidad de problemas o su dificultad.

Los concursos suelen constar de 6 problemas y durar 5 horas, varias personas ingresan al concurso al mismo tiempo para resolver los problemas. Como se mencionó anteriormente estos problemas dan cierta cantidad de puntos, y el que tenga **más puntos** al finalizar ¡será el ganador!

Sin embargo, si dos concursantes tienen el mismo puntaje el ganador será el que consiguió ese puntaje primero. Por lo cual la programación competitiva también es un deporte de velocidad.

## ¿Cómo subo mi respuesta a un problema?

En OmegaUp podemos subir nuestras respuestas en la parte de nuevo envío que viene en la parte de abajo de cada problema.

Envíos

Enviado	GUID	Estatus	Porcentaje	Lenguaje	Memoria	Tiempo	Detalles
Nuevo envío							

Al presionarla nos va a mostrar un espacio en el cual podemos poner nuestro código y especificar el lenguaje de programación en el que está escrito.

Después podemos pegar el escrito de nuestro código o seleccionar un archivo de la extensión correcta que tenga guardado nuestro código (Ejemplo: .cpp para c++, .c para lenguaje C) y luego dar "Submit" o "Enviar".

Lenguaje C++11 (g++ 7.4)

Tu código se guardará en el archivo: Main

**Pega el código de tu programa aquí**

```

1 #include<iostream>
2 using namespace std;
3 int main() {
4     int A, B;
5     cin >> A >> B;
6     cout << A * B << endl;
7     return 0;
8 }
9
10


```

**O sube un archivo**

Choose File No file chosen

Submit

Y después solamente nos queda esperar por ¡nuestra respuesta correcta!

Enviado	GUID	Estatus	Porcentaje	Lenguaje	Memoria	Tiempo	Detalles
2019-11-08 01:38:30	<a href="#">2f41b77b</a>	Respuesta correcta	100.00%	cpp11	3.04 MB	0.02 s	

Nuevo envío