

# Programación Competitiva



# Programación Competitiva

- ✓ Deporte mental que consiste en resolver problemas de índole algorítmico, lógico y matemático, a través de programas computacionales eficientes
- ✓ Busca crear un grupo élite de profesionales de la computación, en los que la industria y la academia centren su atención



# GPC-UPC

- ✓ Grupo de estudios dedicado a la Programación Competitiva
- ✓ Preparamos a los estudiantes en algoritmos y estructuras de datos avanzados
- ✓ Representamos a la universidad en los principales concursos de programación



# ¿Cómo son los problemas?

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

## Problem Statement

There are two buttons, one of size  $A$  and one of size  $B$ .

When you press a button of size  $X$ , you get  $X$  coins and the size of that button decreases by 1.

You will press a button twice. Here, you can press the same button twice, or press both buttons once.

At most how many coins can you get?

## Constraints

- All values in input are integers.
- $3 \leq A, B \leq 20$

## Input

Input is given from Standard Input in the following format:

```
A B
```

## Output

Print the maximum number of coins you can get.

### Sample Input 1 Copy

```
5 3
```

### Sample Output 1 Copy

```
9
```

You can get  $5 + 4 = 9$  coins by pressing the button of size 5 twice, and this is the maximum result.

# ¿Qué conocimientos debo tener?

- ✓ Un lenguaje de programación a nivel básico
- ✓ Inglés básico (deseable)



# ¿Qué aprenderé en el GPC?

## Algoritmos y estructuras de datos

- ✓ Búsqueda binaria
- ✓ Teoría de grafos
- ✓ Teoría de números
- ✓ Programación dinámica y mucho más



# Concursos



- ✓ Compiten equipos de 3 estudiantes universitarios.
  - ✓ Consta de 3 fases de forma presencial, cada una de 5 horas:
- **Fase Regional:** equipos de toda Latinoamérica
  - **Latin America Championship:** 40 mejores equipos latam
  - **World Final:** 120 mejores equipos del mundo



2° puesto nacional 2018



# IEEE Xtreme



- ✓ Compiten equipos de 3 estudiantes universitarios.
- ✓ Consta de una sola fase online a nivel mundial.
- ✓ El concurso dura 24 horas.



IEEE Xtreme 12.0 Global Rankings

Global Rank	Team Name	School	Country	School Region	Country Rank	Region Rank	School Rank
1	W3ComAccessors	University of Illinois - Urbana	USA	RA	1	1	1
2	FatCat	China Univ Of Electronic Science And Tech UESTC	China	AO	1	1	1
3	TiredOfWinning	École Polytechnique Fédérale de Lausanne (EPFL)	Switzerland	RA	1	1	1
4	AuroraP5UT	Princess Sumaya University for Technology	Jordan	RA	1	2	1
5	M4y14y	Georgia Institute of Technology	USA	AO	2	1	1
6	DoubleCycleCover	University of Turku	Finland	RA	1	3	1
7	Apprentices	Univ Federal do Rio Grande do Norte	Brazil	AO	1	1	1
8	ZhenKang	China Univ Of Electronic Science And Tech UESTC	China	AO	2	2	2
9	Bombardier	Sakarya University	Turkey	RA	1	4	1
10	YoungComptelisee	Rice Univ	USA	AO	3	1	1
11	QuantumRegeneration	Escola Politécnica Univ De São Paulo	Brazil	AO	2	2	1
12	TukangOutap	Institut Teknologi Bandung	Indonesia	AO	1	3	1
13	Profler	Chulalongkorn Univ	Thailand	AO	1	4	1
14	TeamSky	Jordan University of Science & Technology	Jordan	RA	2	5	1
15	Antone	American University Beirut	Lebanon	RA	1	6	1
16	UACNem	University Of British Columbia	Canada	AO	1	1	1
17	AcarapComFaria	Univ Federal do Rio de Janeiro	Brazil	AO	3	3	1
18	IdolFinger	Institut Teknologi Bandung	Indonesia	AO	2	5	2
19	WindFireSearch	National Technical University of Athens	Greece	RA	1	7	1
20	LafayetteInnovat	Universidad Peruana de Ciencias Aplicadas	Peru	AO	1	4	1
21	Mathos	Osaka University Of Appl Engng & Tech	Croatia	AO	1	8	1
22	Roundabout	Bangladesh Univ Of Engineering & Tech	Bangladesh	AO	1	9	1

20° puesto mundial (2018)

# TCS Codevita

- ✓ Compiten estudiantes universitarios de forma individual
- ✓ Organizado por Tata Consultancy Services
- ✓ El concurso consta de 3 fases

- **Ronda 1:** estudiantes de todo el mundo (virtual)
- **Ronda 2:** clasificados de la ronda 1 (virtual)
- **Gran Final:** 25 mejores del mundo (presencial)

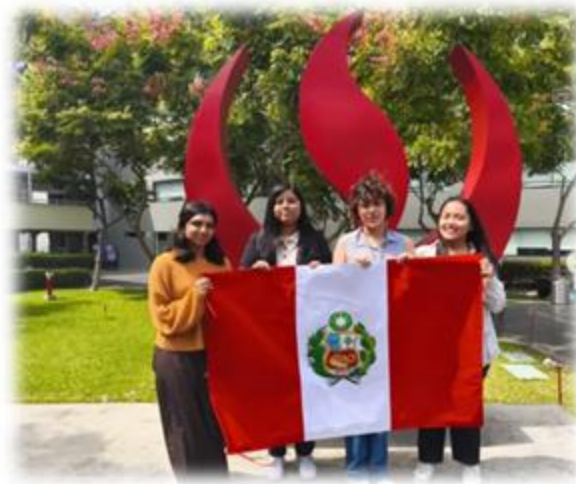


6° puesto mundial (2025)

# Maratón Femenina de Programación



- ✓ Competencia femenina universitaria de forma individual.
- ✓ Organizada por la Sociedad Brasileña de Computación.
- ✓ El concurso consta de 2 fases:
  - **1° Fase:** estudiantes de toda latam (virtual)
  - **Final:** mejores 120 estudiantes de latam (presencial)



Finalistas 2024 y 2025

# Coder Bloom



- ✓ Competencia femenina mensual
- ✓ Modalidad individual
- ✓ El concurso es presencial y dura 3 horas



Ganadoras mensualmente

# Beneficios



# Desarrollo de habilidades

- ✓ Mejorarás tus habilidades para la resolución de problemas
- ✓ Trabajarás en equipo y bajo presión
- ✓ Pondrás en práctica tu inglés



# Beneficios académicos

- ✓ Se te harán más sencillos los cursos de la carrera.
- ✓ Puedes ganar créditos extras (finalizando junior)
- ✓ Puedes llegar a ser tutor (senior)



## Créditos Extraacadémicos

Vida Universitaria te ofrece diferentes alternativas para obtener los cuatro créditos extraacadémicos que son requisito para graduarte en cualquiera de nuestras carreras. ¡Encuentra la que mejor se adapte a ti!

# Training Camp UPC

Capacitación intensiva durante 1 semana  
por parte de un coach mundialista  
internacional





# Internships Internacionales

Incrementarás tus posibilidades de hacer pasantías/trabajar en las mejores empresas a nivel mundial.



# Ganarás muchos premios



A dive into the world under

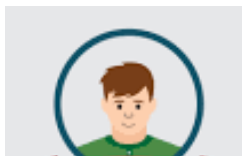


VGP MARINE KINGDOM  
#TCSCodeVita

# Camino en el GPC

**Admisión**  
28 de Junio  
(3 pm – 5 pm)

1 ciclo de entrenamiento



**Beginner**

1 año de entrenamiento



**Junior**

+1 año de entrenamiento



**Senior**



# C++ para competencias



# Antes de empezar...



- Crearse un usuario en Virtual Judge con su código UPC
- Unirse al grupo: [vjudge.net/group/gpctaller25](https://vjudge.net/group/gpctaller25)

# Ambiente de trabajo



Usaremos CSAcademy

<https://csacademy.com/workspace/>

# Programa básico

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Hello world";
7      return 0;
8  }
```

directivas para el preprocesador

namespace

función principal

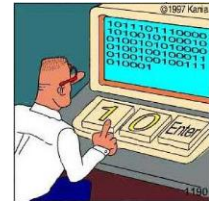
instrucciones en C++

# Compilador g++



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Hello world";
7      return 0;
8  }
```

usado en las  
competencias



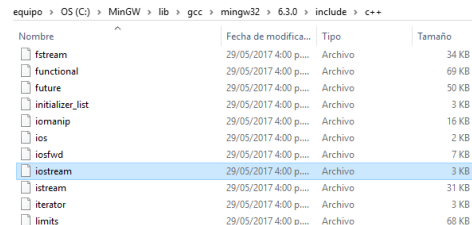


# Preprocesador

- Forma parte del compilador.
- Procesa directivas (comandos) antes de la compilación real de un programa.
- Las directivas inician con el símbolo # y terminan con el final de línea.
- Utiliza una sintaxis diferente a C++.

# Directiva #include

- Indica al preprocesador incluir al programa actual el contenido de otro archivo.
- Los archivos que se incluyen en otros programas se denominan **headers** y son necesarios para usar ciertas funcionalidades.
- Por ejemplo **#include <iostream>** le dice al preprocesador que copie el contenido del archivo **iostream**, el cual nos permite manejar el ingreso y salida de datos.



equipo > OS (C:) > MinGW > lib > gcc > mingw32 > 6.3.0 > include > c++

Nombre	Fecha de modifica...	Tipo	Tamaño
fstream	29/05/2017 4:00 p...	Archivo	34 KB
functional	29/05/2017 4:00 p...	Archivo	69 KB
future	29/05/2017 4:00 p...	Archivo	50 KB
initializer_list	29/05/2017 4:00 p...	Archivo	3 KB
iomanip	29/05/2017 4:00 p...	Archivo	16 KB
ios	29/05/2017 4:00 p...	Archivo	2 KB
iosfwd	29/05/2017 4:00 p...	Archivo	7 KB
<b>iostream</b>	29/05/2017 4:00 p...	Archivo	<b>3 KB</b>
istream	29/05/2017 4:00 p...	Archivo	31 KB
iterator	29/05/2017 4:00 p...	Archivo	3 KB
limits	29/05/2017 4:00 p...	Archivo	68 KB

# ⟨bits/stdc++.h⟩

- Header especial que incluye a todos los headers del estándar C++.
- Solo está presente en el compilador G++.

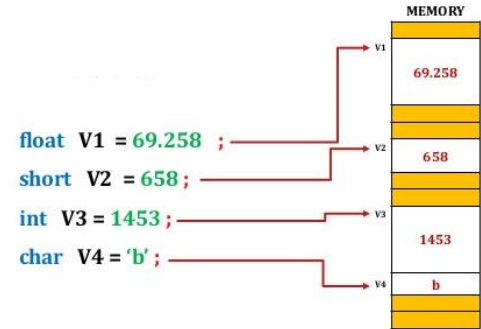
# Directiva #define

- Define las denominadas macros, las cuales nos permiten manejar sustitución de términos en el código.
- Las macros tienen el siguiente formato: `#define macro reemplazo`
- El preprocesador busca todas las ocurrencias de `macro` en el programa y las sustituye por `reemplazo`, antes de iniciar la compilación real.

```
#define MAXN 1000  
#define ll long long
```

# Variables

```
char letra;  
string nombre;  
bool flag;  
int edad;  
long long nro_conexiones;  
float costo;  
double area;
```



# Lectura y escritura

- La entrada estándar viene desde teclado.
- La salida estándar se muestra en consola.

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

int main() {
    int a, b, c;
    cin >> a >> b >> c; //lectura
    c = a + b;
    cout << c; //escritura
    return 0;
}
```

# Escritura de números reales

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

int main() {
    double a, b;
    cin >> a >> b;
    cout << fixed;
    cout << setprecision(2) << a << "\n";
    cout << setprecision(2) << b << "\n";
    return 0;
}
```

# Condicionales / if-else

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

int main() {
    int mi_edad, tu_edad;
    cin >> mi_edad >> tu_edad;
    if (mi_edad < tu_edad) {
        cout << "soy menor";
    } else {
        cout << "soy mayor o tenemos la misma edad";
    }
    return 0;
}
```



# Ejercicios

- [EOlymp 8891 – Exactly one condition out of two](#)
- [Codeforces 50 A – Domino piling](#)



# Repetitivas / while

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

int main() {
    int i = 0;
    while (i <= 10) {
        cout << i << endl;
        i = i + 1;
    }
    return 0;
}
```

# Repetitivas / for

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

int main() {
    for (int i = 0; i <= 10; ++i) {
        cout << i << endl;
    }
    return 0;
}
```

# Ejercicios

- [EOlymp 1603 – The sum of digits](#)
- [HackerRank – Staircase](#)
- [UVA 10812 – Beat the spread!](#)





“Opportunities don’t happen, you create them.”

- Chris Grosser