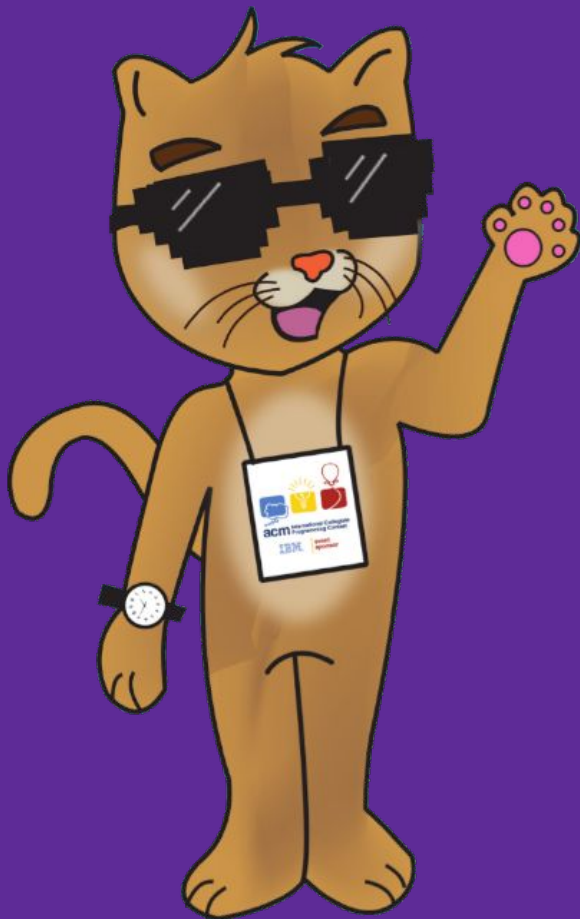


PU++

Club de programación competitiva



Intro

Luis Erick Montes Garcia

Intro

Luis Erick Montes Garcia

Mexico & Centroamerica ICPC Finals 2017-2019.

Intro

Luis Erick Montes Garcia

Mexico & Centroamerica ICPC Finals 2017-2019.
Microsoft Intern Software engineer.

Intro

Pedro Juan Salvador Sanchez Perez

Intro

Pedro Juan Salvador Sanchez Perez

Mexico & Centroamerica ICPC Finals 2019.

Intro

Pedro Juan Salvador Sanchez Perez

Mexico & Centroamerica ICPC Finals 2019.
Lider de PU++.

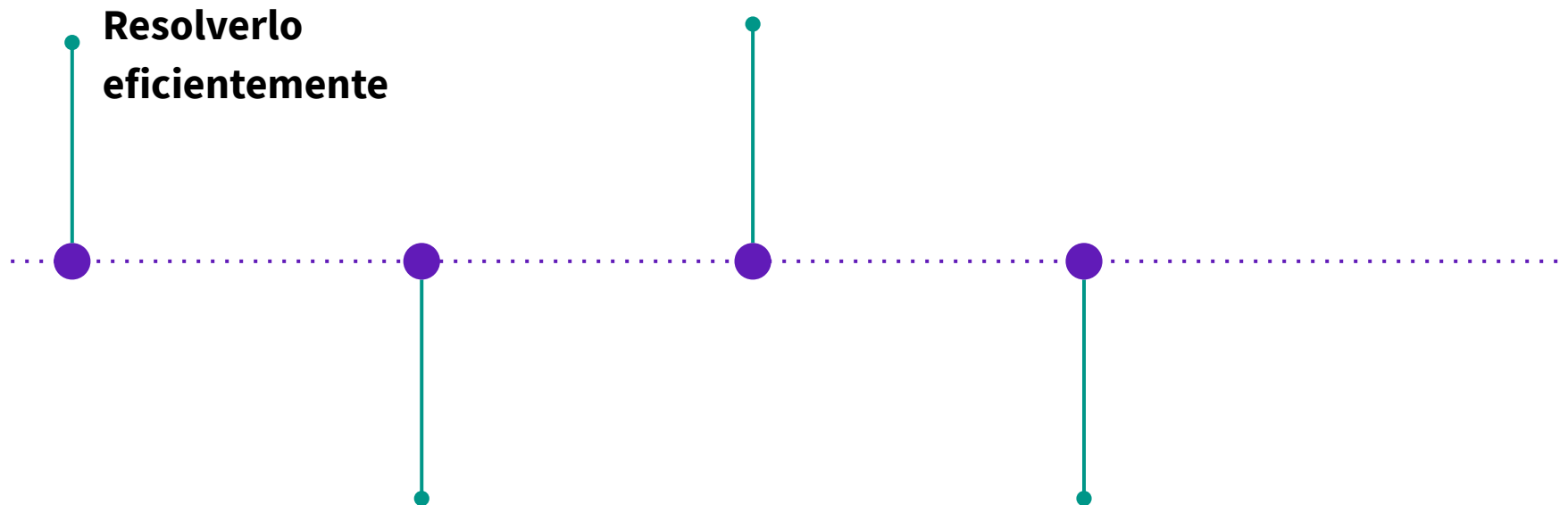
La meta

La meta

Dado cualquier problema

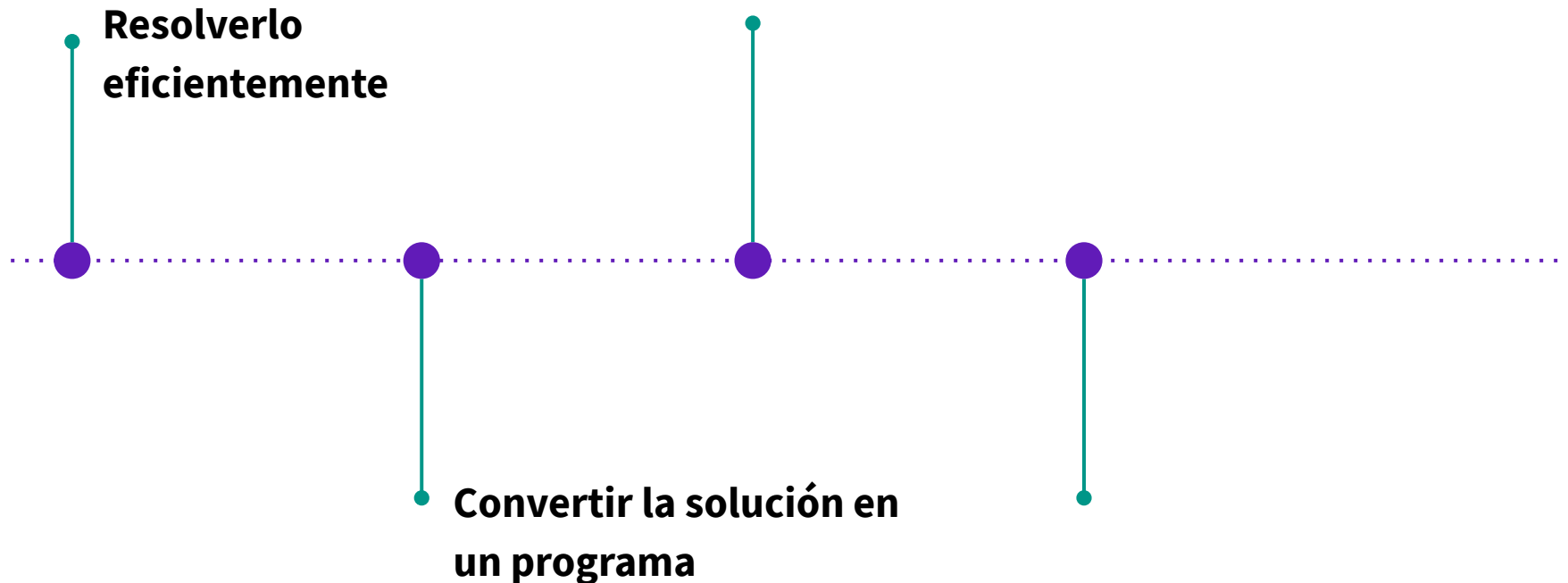
La meta

Dado cualquier problema



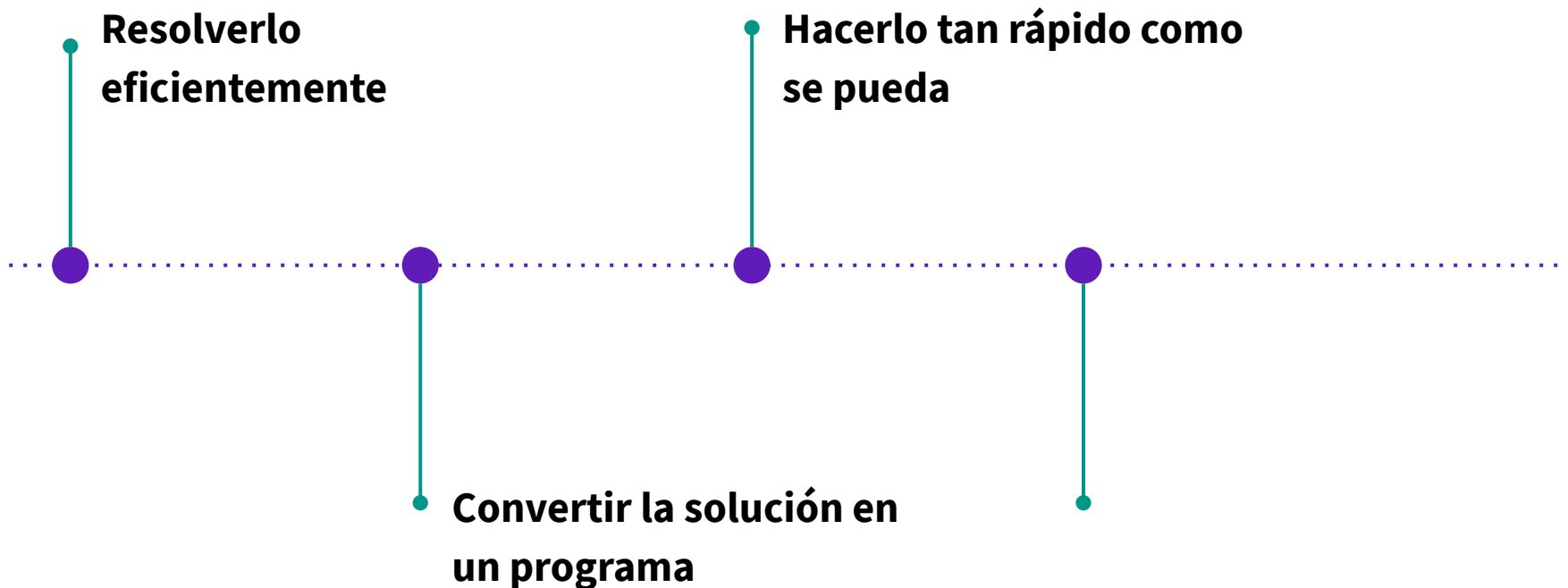
La meta

Dado cualquier problema



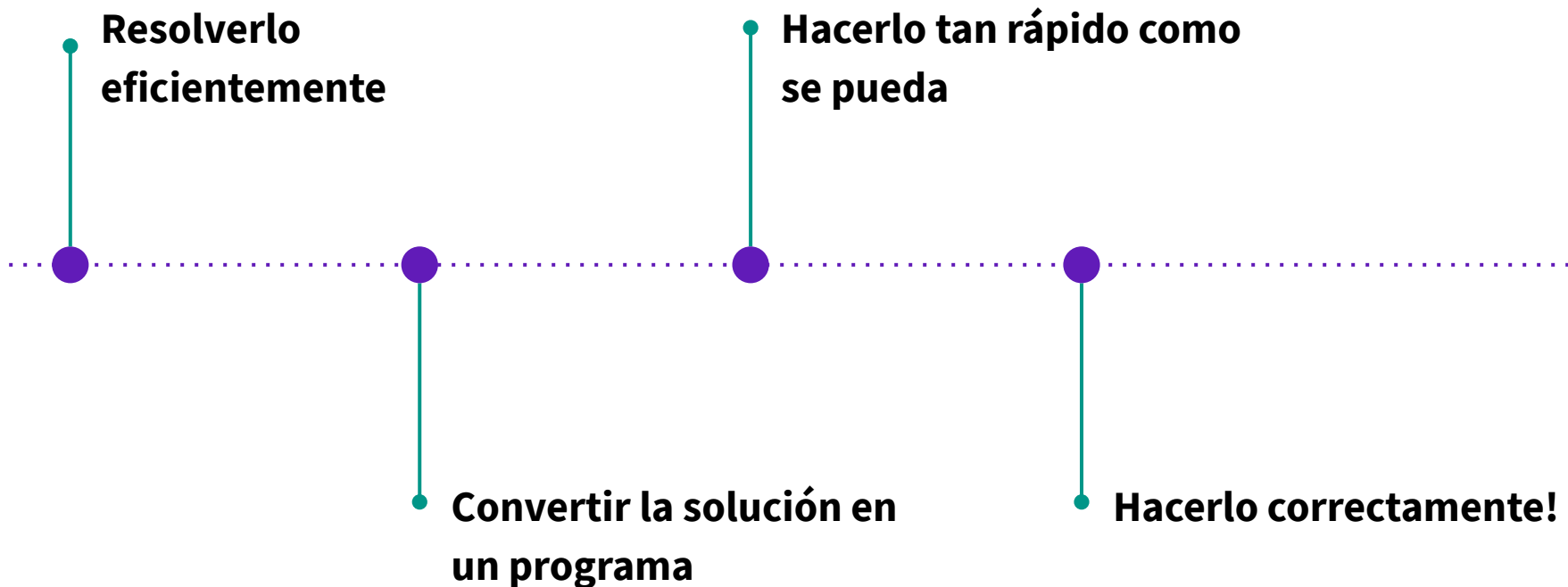
La meta

Dado cualquier problema



La meta

Dado cualquier problema



¿Cómo?

¿Cómo?

- Estudiando problemas comunes

¿Cómo?

- Estudiando problemas comunes
- Estudiando Algoritmos, Estructuras de Datos y Matemáticas

¿Cómo?

- Estudiando problemas comunes
- Estudiando Algoritmos, Estructuras de Datos y Matemáticas
- Practicando

¿Cómo?

- Estudiando problemas comunes
- Estudiando Algoritmos, Estructuras de Datos y Matemáticas
- Practicando
- PRACTICANDO

¿Cómo?

- Estudiando problemas comunes
- Estudiando Algoritmos, Estructuras de Datos y Matemáticas
- Practicando
- PRACTICANDO
- PRACTICANDO!!!!!!

Problemas

Problemas

Un problema consiste en:

Problemas

Un problema consiste en:

- Descripción del problema.

Problemas

Un problema consiste en:

- Descripción del problema.

Durante una crisis energética en Nueva Zelanda el invierno pasado (causada por una escasez de lluvia y por lo tanto bajos niveles en las represas hidroeléctricas), se desarrolló un plan de contingencia para dejar sin energía áreas del país de manera sistemática y totalmente justa. El país se dividió en N regiones (Auckland era la región número 1, y Wellington la número 13). Un número m sería elegido "al azar", y la energía se desactivaría primero en la región 1 (claramente el punto inicial más justo) y luego en cada región m -ésima, volviendo a 1 después de N , e ignorando regiones ya apagadas. Por ejemplo, si $N = 17$ y $m = 5$, la energía se desconectaría a las regiones en el orden: 1, 6, 11, 16, 5, 12, 2, 9, 17, 10, 4, 15, 14, 3, 8, 13, 7.

Por lo tanto, claramente, lo más justo es apagar la región de Wellington por último (ya que es donde está el problema). Por lo tanto, dado un N , el número 'aleatorio' m debe ser elegido cuidadosamente para que, claramente, lo más justo es apagar la región de Wellington por último (ya que es donde está el problema).

El problema de Josephus es conocido por la leyenda de Flavio Josephus, un historiador judío que vivía en el siglo I. Según el relato de Josephus sobre el asedio de Yodfat, él y sus 40 compañeros soldados quedaron atrapados en una cueva, cuya salida fue bloqueada por los romanos. Ellos eligieron el suicidio antes que la captura, y decidieron que formarían un círculo y empezarían a matarse saltando de tres en tres. Josephus dice que, por fortuna o quizás por la mano de Dios, él fue el último y se entregó a los romanos.



Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.

Entrada

Hay **NC** ($1 \leq \text{NC} \leq 30$) casos de pruebas. En cada caso de prueba de entrada tendremos un par de números enteros positivos **n** ($1 \leq n \leq 10000$) y **k** ($1 \leq k \leq 1000$). El número **n** representa la cantidad de gente en el círculo, numerados desde 1 hasta **n**. El número **k** representa el tamaño del paso entre hombres dentro del círculo.

Siga el ejemplo con 5 hombres y paso 2: En este ejemplo el elemento restante es 3.

Entrada

La entrada consistirá de una serie de líneas, cada una contiene el número de regiones **N** ($13 \leq N \leq 100$). La entrada finaliza con una línea con el número 0.

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.
- Descripción de la salida.

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.
- Descripción de la salida.

Output

For each input test case, your program must print a single line containing a single character, 'S' if Mary has a winning strategy or 'N' otherwise.

Salida

La salida consiste en una línea por cada entrada. Cada línea consiste en un número m de acuerdo al esquema anterior.

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.
- Descripción de la salida.
- Ejemplos de entradas y salidas.

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.
- Descripción de la salida.
- Ejemplos de entradas y salidas.

Input Sample		Output Sample	
5		S	
.....		N	
5		S	
..X..		N	
6			
X.X.X.			
12			
.....			
0			

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.
- Descripción de la salida.
- Ejemplos de entradas y salidas.
- Tiempo límite en segundos.

Problemas

Un problema consiste en:

- Descripción del problema.
- Descripción de la entrada.
- Descripción de la salida.
- Ejemplos de entradas y salidas.
- Tiempo límite en segundos.
- Memoria límite en Bytes.

Ejemplo

Ejemplo

El problema de Josephus es conocido por la leyenda de Flavio Josephus, un historiador judío que vivía en el siglo I. Según el relato de Josephus sobre el asedio de Yodfat, él y sus 40 compañeros soldados quedaron atrapados en una cueva, cuya salida fue bloqueada por los romanos. Ellos eligieron el suicidio antes que la captura, y decidieron que formarían un círculo y empezarían a matarse saltando de tres en tres. Josephus dice que, por fortuna o quizás por la mano de Dios, él fue el último y se entregó a los romanos.



Entrada

Hay **NC** ($1 \leq \text{NC} \leq 30$) casos de pruebas. En cada caso de prueba de entrada tendremos un par de números enteros positivos **n** ($1 \leq n \leq 10000$) y **k** ($1 \leq k \leq 1000$). El número **n** representa la cantidad de gente en el círculo, numerados desde 1 hasta **n**. El número **k** representa el tamaño del paso entre hombres dentro del círculo.

Siga el ejemplo con 5 hombres y paso 2: En este ejemplo el elemento restante es 3.



C++



C++



- Es más rápido por lo general

C++



- Es más rápido por lo general
- Tiene Bibliotecas muy útiles

C++



- Es más rápido por lo general
- Tiene Bibliotecas muy útiles
- Es más manipulable

C++



- Es más rápido por lo general
- Tiene Bibliotecas muy útiles
- Es más manipulable
- La mayoría de la bibliografía para C.P. esta en C++

Ejemplo: Hello World

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

Hello.cpp

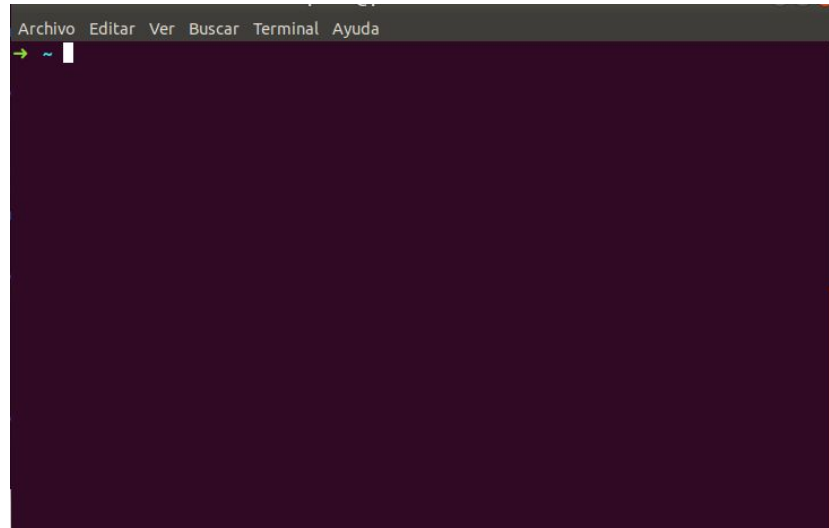
Ejemplo: Hello World

Cómo ejecutar un programa en C++ (Terminal)

Ejemplo: Hello World

Cómo ejecutar un programa en C++ (Terminal)

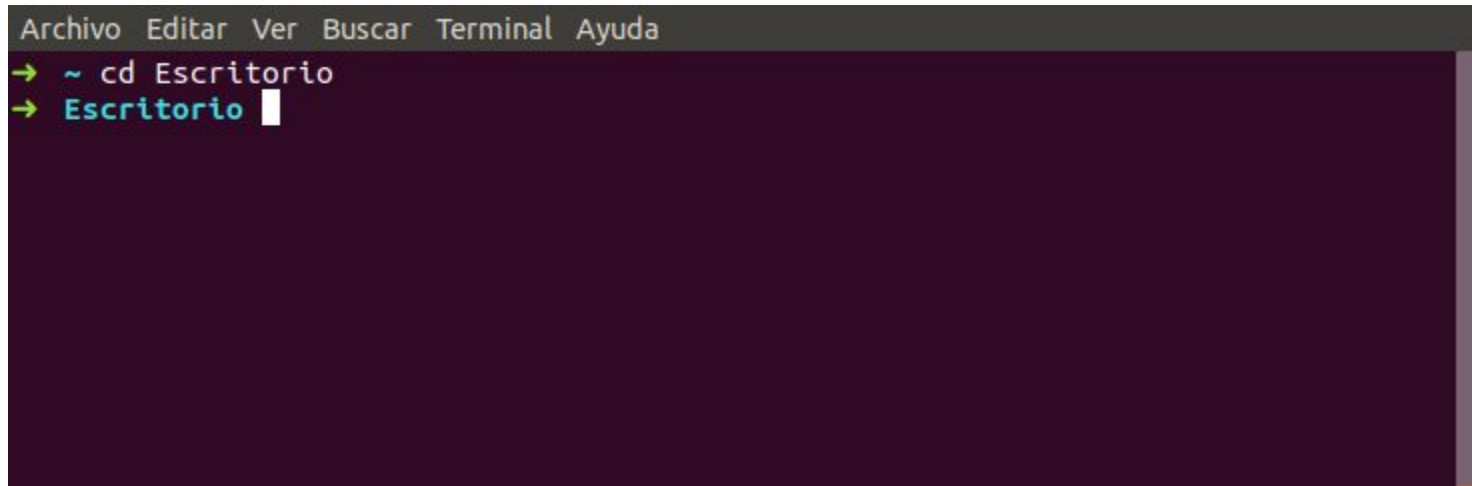
- Abrir una terminal.



Ejemplo: Hello World

Cómo ejecutar un programa en C++ (Terminal)

- Abrir una terminal.
- Localizar la terminal en el directorio donde esta el archivo.

A screenshot of a terminal window with a dark purple background. At the top, there is a menu bar with the following items: 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. Below the menu bar, the terminal shows two lines of text. The first line is a green prompt character followed by '~ cd Escritorio'. The second line is a green prompt character followed by 'Escritorio' and a white cursor. The rest of the terminal area is empty.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
→ ~ cd Escritorio
→ Escritorio
```

Ejemplo: Hello World

Cómo ejecutar un programa en C++ (Terminal)

- Abrir una terminal.
- Localizar la terminal en el directorio donde esta el archivo.
- Escribir la siguiente línea:
 - `g++ [Archivo] -o [Nombre]`

Archivo Editar Ver Buscar Terminal Ayuda

→ ~ cd Escritorio

→ **Escritorio** g++ Hello.cpp -o Hello

Ejemplo: Hello World

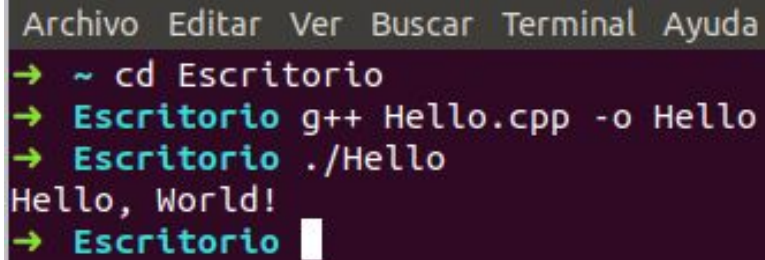
Cómo ejecutar un programa en C++ (Terminal)

- Abrir una terminal.
- Localizar la terminal en el directorio donde esta el archivo.
- Escribir la siguiente línea:
 - g++ [Archivo] -o [Nombre]
- Escribir la siguiente línea:
 - ./[Nombre]

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
→ ~ cd Escritorio
→ Escritorio g++ Hello.cpp -o Hello
→ Escritorio ./Hello
```

Ejemplo: Hello World

EXITO!!



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
→ ~ cd Escritorio
→ Escritorio g++ Hello.cpp -o Hello
→ Escritorio ./Hello
Hello, World!
→ Escritorio
```

A terminal window with a dark purple background and a grey menu bar at the top containing the options: Archivo, Editar, Ver, Buscar, Terminal, and Ayuda. The terminal shows a sequence of commands and their output. The first command is a green arrow followed by '~ cd Escritorio'. The second command is a green arrow followed by 'Escritorio g++ Hello.cpp -o Hello'. The third command is a green arrow followed by 'Escritorio ./Hello'. The output of the third command is 'Hello, World!'. The final line shows a green arrow followed by 'Escritorio' and a white cursor.

Ejemplo: Hello World

EXITO!!



Tu primer problema.

Tu primer problema.



Muchos Más...

Tu primer problema.



CODEFORCES

Sponsored by Telegram

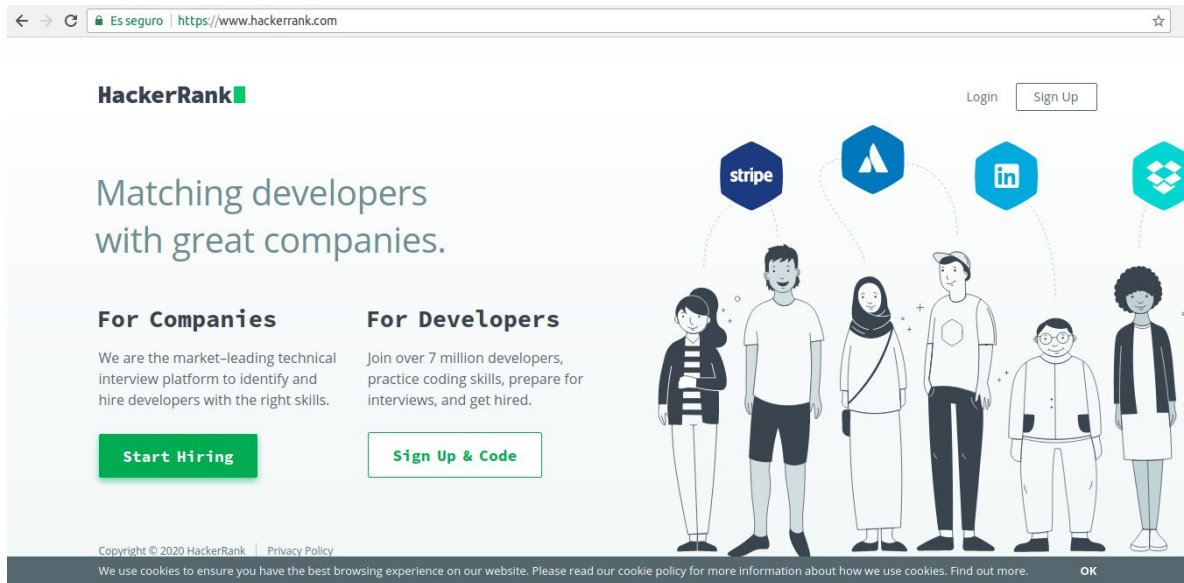
β je



.....

Tu primer problema.

Entra a Hackerrank.com



Tu primer problema.

Entra a [Hackerrank.com](https://leetcode.com)

Crear una cuenta

Tu primer problema.

Entra a Hackerrank.com

Practice

Dashboard

Bookmarked Challenges

Your Skills

INTERVIEW PREPARATION

NEW

Interview Preparation Kit

4% (3/69 challenges solved)

Continue Practice

PROBLEM SOLVING

Problem Solving

19% (302.94 points to gold badge)

Continue Practice



Tu primer problema.

Entra a Hackerrank.com

Algorithms | [Data Structures](#)

Solve Me First

Easy, Max Score: 1, Success Rate: 98.49%



Solved 

Tu primer problema.

Descripción

Problem

[Submissions](#)[Leaderboard](#)[Discussions](#)[Editorial](#)

Complete the function solveMeFirst to compute the sum of two integers.

Function prototype:

```
int solveMeFirst(int a, int b);
```

where,

- a is the first integer input.
- b is the second integer input

Return values

- sum of the above two integers

Sample Input

```
a = 2  
b = 3
```

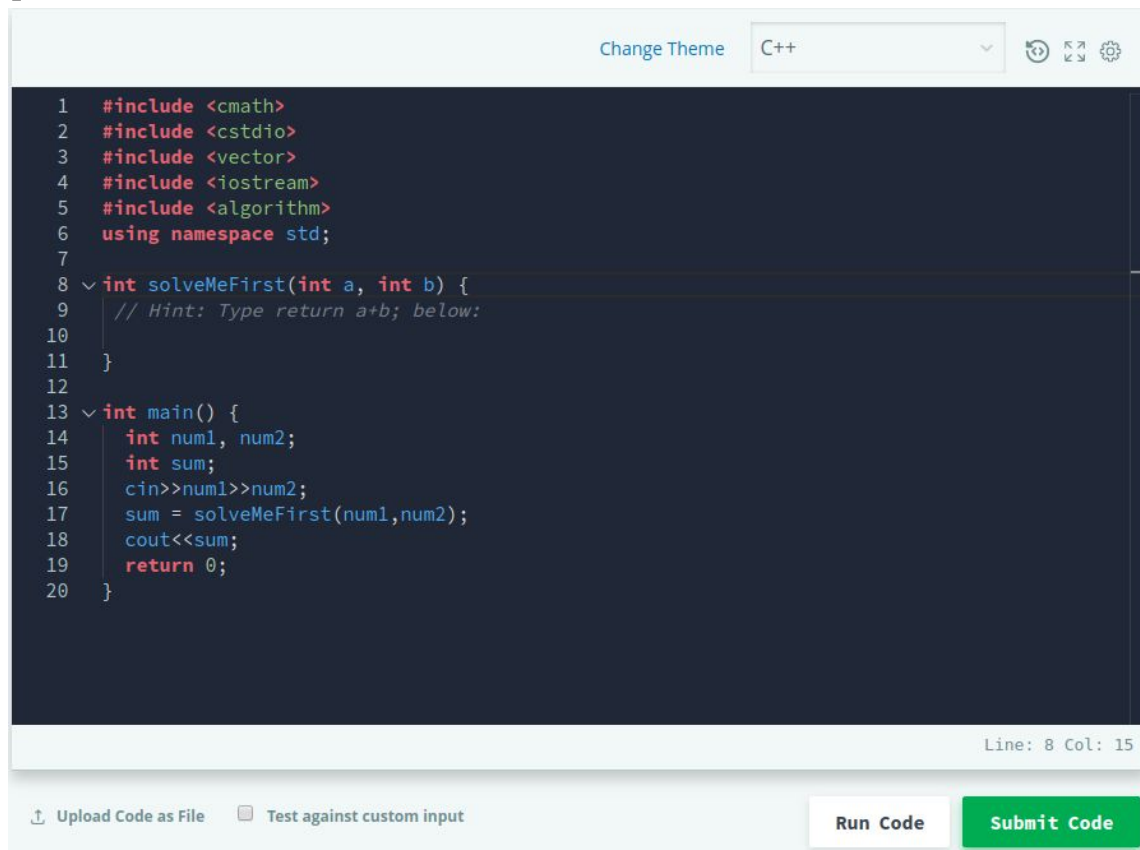
Sample Output

```
5
```

Explanation

The sum of the two integers a and b is computed as: $2 + 3 = 5$.

Tu primer problema.

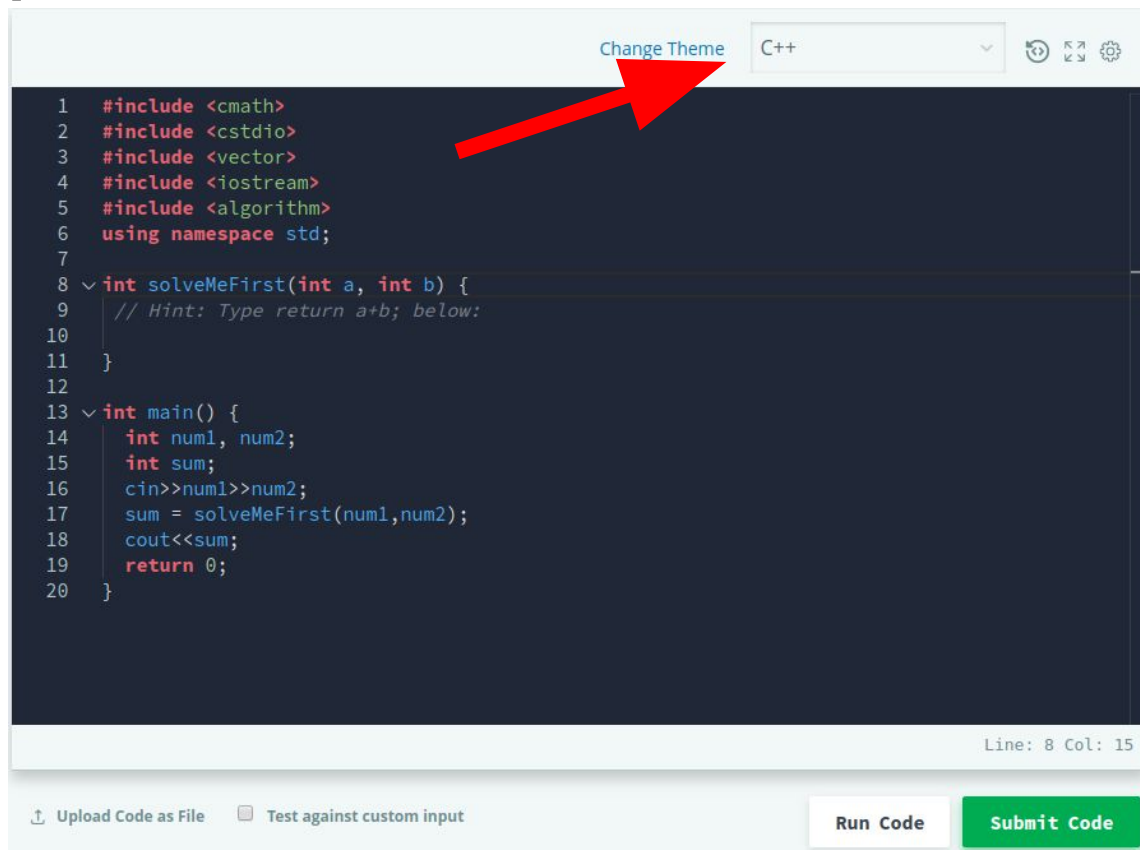


```
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
4  #include <iostream>
5  #include <algorithm>
6  using namespace std;
7
8  int solveMeFirst(int a, int b) {
9      // Hint: Type return a+b; below:
10
11 }
12
13 int main() {
14     int num1, num2;
15     int sum;
16     cin>>num1>>num2;
17     sum = solveMeFirst(num1,num2);
18     cout<<sum;
19     return 0;
20 }
```

Line: 8 Col: 15

[Upload Code as File](#) ☐ [Test against custom input](#) [Run Code](#) [Submit Code](#)

Tu primer problema.




The image shows a screenshot of a C++ code editor interface. A red arrow points to the 'Change Theme' button in the top right corner. The code is written in C++ and is as follows:

```
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
4  #include <iostream>
5  #include <algorithm>
6  using namespace std;
7
8  int solveMeFirst(int a, int b) {
9      // Hint: Type return a+b; below:
10
11  }
12
13  int main() {
14      int num1, num2;
15      int sum;
16      cin>>num1>>num2;
17      sum = solveMeFirst(num1,num2);
18      cout<<sum;
19      return 0;
20  }
```

At the bottom of the editor, there is a status bar showing 'Line: 8 Col: 15'. Below the editor, there are two buttons: 'Run Code' and 'Submit Code'. There are also links for 'Upload Code as File' and 'Test against custom input'.

Tu primer problema.



```
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
4  #include <iostream>
5  #include <algorithm>
6  using namespace std;
7
8  int solveMeFirst(int a, int b) {
9      // Hint: Type return a+b; below:
10
11 }
12
13 int main() {
14     int num1, num2;
15     int sum;
16     cin>>num1>>num2;
17     sum = solveMeFirst(num1,num2);
18     cout<<sum;
19     return 0;
20 }
```

Line: 8 Col: 15

[Upload Code as File](#) ☐ [Test against custom input](#) [Run Code](#) [Submit Code](#)

Tu primer problema.

```
1
8  ✓ int solveMeFirst(int a, int b) {
9    // Hint: Type return a+b; below:
10   |
11   }
12
```

Tu primer problema.

Función

```

8  ✓ int solveMeFirst(int a, int b) {
9      // Hint: Type return a+b; below:
10
11 }
12

```

Tu primer problema.

Función



```
1  
2  
3  
4  
5  
6  
7  
8  
9 int solveMeFirst(int a, int b) {  
10     // Hint: Type return a+b; below:  
11  
12 }
```

Tu primer problema.

Función



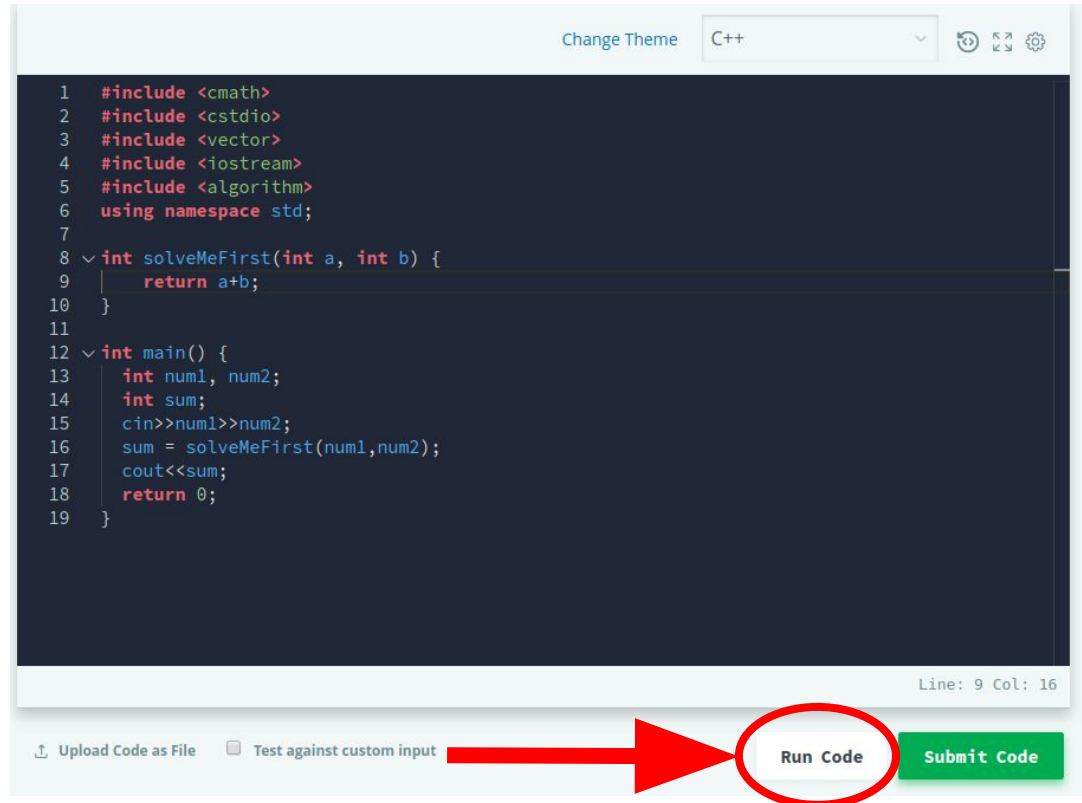
```
1  
2  
3  
4  
5  
6  
7  
8  
9 int solveMeFirst(int a, int b) {  
10     // Hint: Type return a+b; below:  
11  
12 }
```

return

Tu primer problema.

```
8  ✓ int solveMeFirst(int a, int b) {  
9    |     return a+b;  
10   |  
11   | }
```

Tu primer problema.



The image shows a code editor interface for C++. At the top, there is a 'Change Theme' button and a language dropdown menu set to 'C++'. The editor contains the following C++ code:

```
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
4  #include <iostream>
5  #include <algorithm>
6  using namespace std;
7
8  int solveMeFirst(int a, int b) {
9      return a+b;
10 }
11
12 int main() {
13     int num1, num2;
14     int sum;
15     cin>>num1>>num2;
16     sum = solveMeFirst(num1,num2);
17     cout<<sum;
18     return 0;
19 }
```

At the bottom of the editor, there is a status bar showing 'Line: 9 Col: 16'. Below the editor, there are two buttons: 'Run Code' and 'Submit Code'. A red arrow points to the 'Run Code' button, which is circled in red.

Tu primer problema.

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ Sample Test case 1

Input (stdin)

[Download](#)

1	2
2	3

Your Output (stdout)

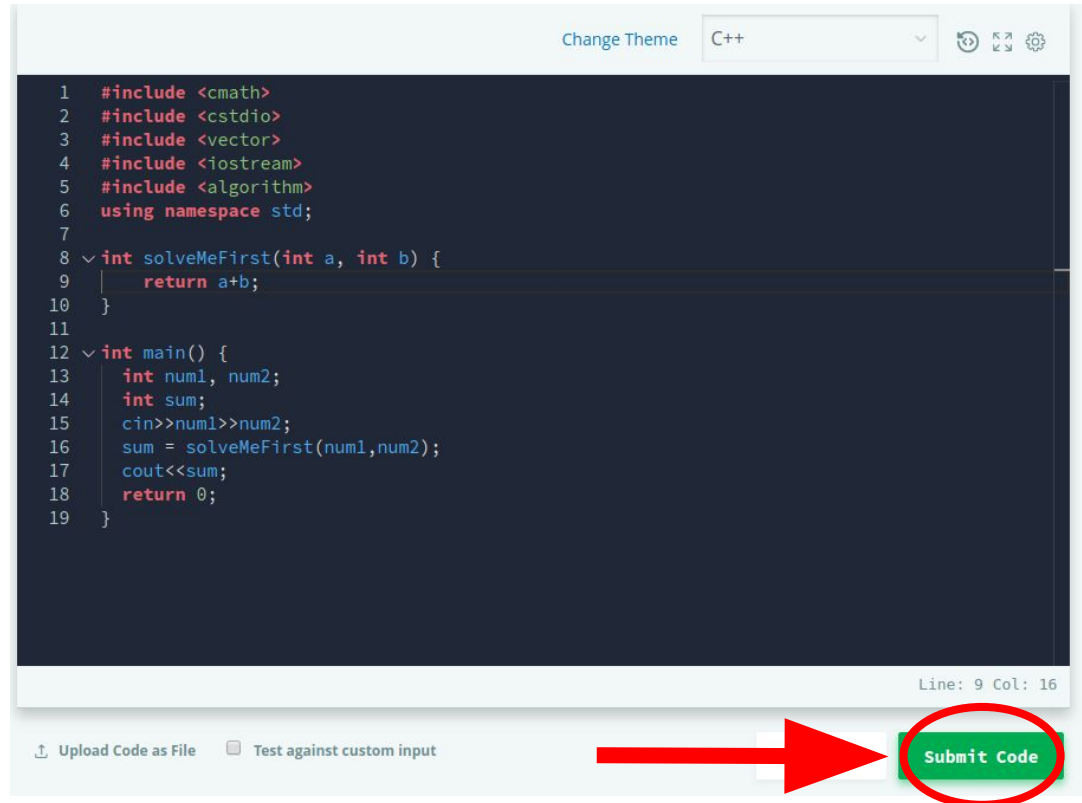
1	5
---	---

Expected Output

[Download](#)

1	5
---	---

Tu primer problema.



The image shows a C++ code editor interface. At the top, there is a 'Change Theme' button and a language dropdown menu set to 'C++'. The code is written in a dark-themed editor with line numbers 1 through 19. The code defines a function `solveMeFirst` and a `main` function. The `main` function reads two integers, calls `solveMeFirst`, and prints the result. At the bottom, there are two options: 'Upload Code as File' and 'Test against custom input'. A large red arrow points to a green 'Submit Code' button, which is also circled in red.

```
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
4  #include <iostream>
5  #include <algorithm>
6  using namespace std;
7
8  int solveMeFirst(int a, int b) {
9      return a+b;
10 }
11
12 int main() {
13     int num1, num2;
14     int sum;
15     cin>>num1>>num2;
16     sum = solveMeFirst(num1,num2);
17     cout<<sum;
18     return 0;
19 }
```

Line: 9 Col: 16

☐ Upload Code as File ☐ Test against custom input

Submit Code

Tu primer problema.

Congratulations

You solved this challenge. Would you like to challenge your friends?



[Next Challenge](#)

Tu primer problema.

Congratulations

You solved this challenge. Would you like to challenge your friends?



Next Challenge



Gracias.

