



# POBI

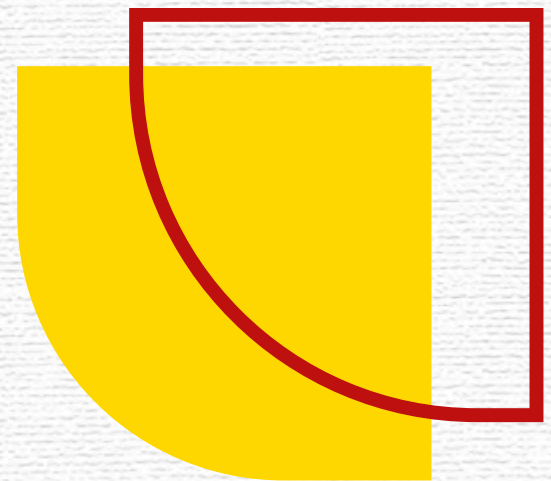
Preparatório para a Olimpíada Brasileira de  
Informática da UFC Quixadá

× × × ×



# PROGRAMAÇÃO COMPETITIVA

- Resolver problemas com lógica e código
- Tempo e memória como restrições
- Avaliação automática das soluções





# COMPETIÇÕES QUE VAMOS FOCAR





# FERRAMENTAS DE ESTUDO



<https://www.mycompiler.io/pt>







beecrowd | 1001

## Extremamente Básico

Adaptado por Neilor Tonin, URI Brasil

Timelimit: 1

Leia 2 valores inteiros e armazene-os nas variáveis **A** e **B**. Efetue a soma de **A** e **B** atribuindo o seu resultado na variável **X**. Imprima **X** conforme exemplo apresentado abaixo. Não apresente mensagem alguma além daquilo que está sendo especificado e não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "*Presentation Error*".

### Entrada

A entrada contém 2 valores inteiros.

### Saída

Imprima a mensagem "X = " (letra X maiúscula) seguido pelo valor da variável **X** e pelo final de linha. Cuide para que tenha um espaço antes e depois do sinal de igualdade, conforme o exemplo abaixo.

Exemplos de Entrada	Exemplos de Saída
10 9	X = 19
-10 4	X = -6
15 -7	X = 8

PROBLEMA

1001

LINGUAGEM

C99

SOURCE CODE

```
1 #include <stdio.h>
2
3 int main() {
4
5     /**
6      * Escreva a sua solução aqui
7      * Code your solution here
8      * Escriba su solución aquí
9      */
10
11     return 0;
12 }
```

CONSTRUA A SUA SOLUÇÃO E ENVIE!

ENVIAR

# Soma Fácil

Faça um programa para ler dois valores inteiros  $A$  e  $B$  e imprima o resultado da expressão  $(A + B)$ .

## Entrada

A entrada consiste de duas linhas. A primeira linha contém o inteiro  $A$  e a segunda linha contém o inteiro  $B$ .

## Saída

A saída consiste de uma linha contendo o resultado da expressão  $(A + B)$ .

## Restrições

- Os inteiros fornecidos não terão valor absoluto maior que 1000.

Exemplos de Entrada	Exemplos de Saída
2 3	5
-2 3	1
10 89	99

Traduzido por **Arthur Freitas**

## Detalhes

[🚀 ESCREVER SOLUÇÃO](#)
[🧩 SUBMISSÕES](#)
[👤 COMUNIDADE](#)
[📖 ANOTAÇÕES](#)
[🔍 ME MOSTRE A SOLUÇÃO](#)

**Tempo Limite:** 1 second(s)

**Limite de Memória:** 256 mb

**Tipo de Acesso:** Gratuito

**XP:** 10

**Melhor Submissão** **Accepted**

**Cronômetro:** ⌚

**Adicionado por :** **Thiago  
Nepomuceno**

**Resolvido por:** 11652 usuários

**No Neps desde:** 08/05/2018



[← PÁGINA DO EXERCÍCIO](#)

## Sua Solução

Linguagem de Programação

C++ (g++ 10.2.1; c++20 partial)



ENVIAR SOLUÇÃO

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     // Lendo a entrada do exercício
7     int A, B;
8     cin >> A >> B;
9
10    cout << A + B << endl;
11
12    return 0;
13 }
14
```

## Testes Customizados

Entrada

Saída



TESTAR SOLUÇÃO





# LINGUAGEM C++

Por que usar C++ em programação competitiva?

- É rápida, essencial pra problemas com limite de tempo
- Linguagem compilada e de alto desempenho
- Grande controle sobre uso de memória (ponteiros)



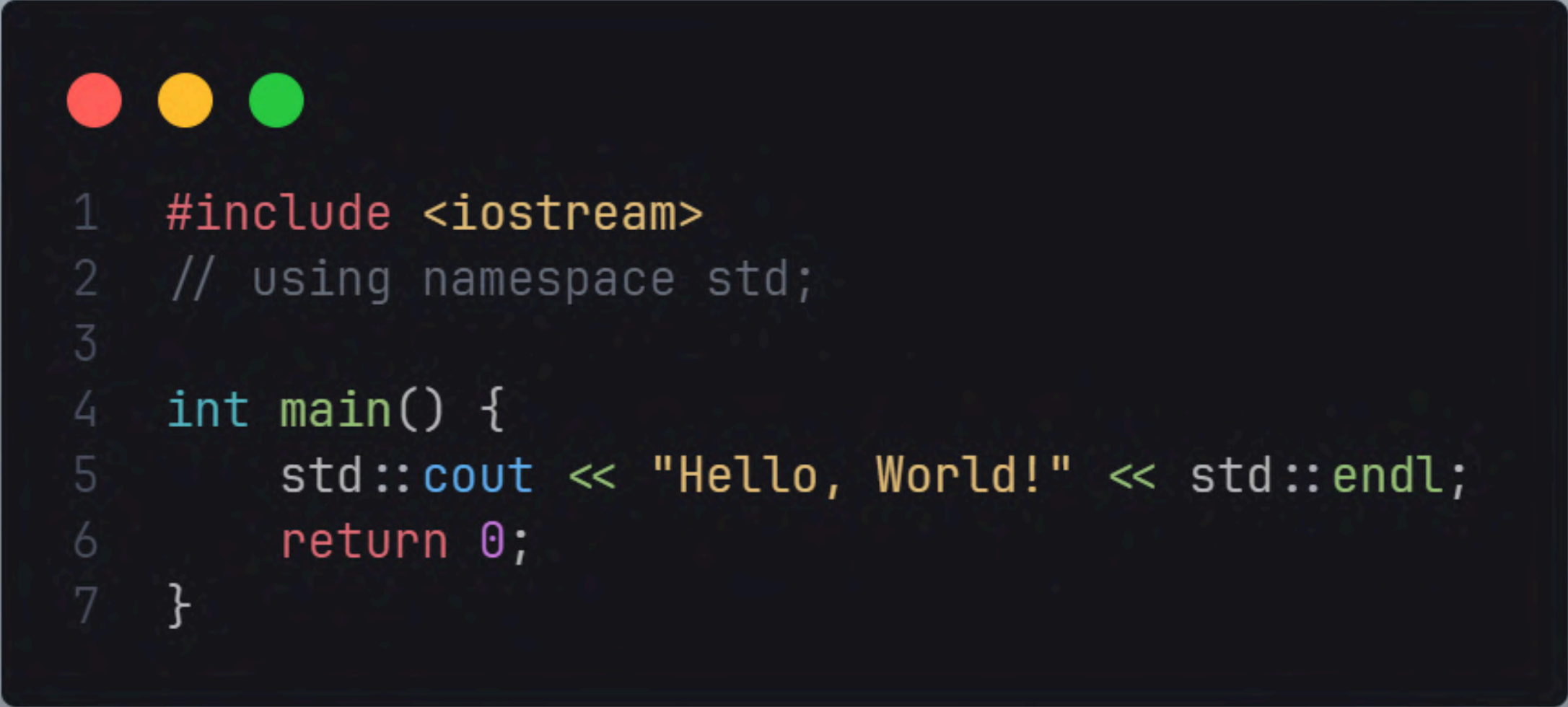
# PRIMEIRO PROGRAMA 'HELLO WORLD'



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



# PRIMEIRO PROGRAMA 'HELLO WORLD'



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



# ENTRADA E SAÍDA



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b, c;
6      cin >> a >> b >> c;
7      cout << a << " " << b << " " << c << endl;
8      cout << a + b + c << endl;
9  }
```



# TIPOS DE DADOS

Tipo	Descrição	Tamanho
bool	Valores Aceitos true ou false	1 byte ou 8 bits
char	Intervalos de -128 a 127	1 byte ou 8 bits
signed char	Intervalos de -128 a 127	1 byte ou 8 bits
unsigned char	Intervalos de 0 a 255	1 byte ou 8 bits
int	Intervalos de -2.147.483.648 a 2.147.483.647	4 bytes ou 32 bits
short int	Intervalos de -32.768 a 32.767	2 bytes ou 16 bits
signed int	Intervalos de -2.147.483.648 a 2.147.483.647	4 bytes ou 32 bits
unsigned int	Intervalos de 0 a 4.294.967.295	4 bytes ou 32 bits
unsigned short int	Intervalos de 0 a 4.294.967.295	2 bytes ou 32 bits
float	Intervalos de 1,2e-38 até 3,4e+38.	4 bytes ou 32 bits
double	Intervalos de 2,2e-308 a 1,8e+308	8 bytes ou 64 bits
long double		
long	Intervalos de -2.147.483.648 a 2.147.483.647	4 bytes ou 32 bits
signed long	Intervalos de -2.147.483.648 a 2.147.483.647	4 bytes ou 32 bits
unsigned long	Intervalos de 0 a 4.294.967.295	4 bytes ou 32 bits
long long		
wchat_t	Intervalos de -32.768 a 32.767	4 bytes ou 32 bits
signed wchar_t	Intervalos de -32.768 a 32.767	4 bytes ou 32 bits
unsigned wchar_t	Intervalo de 0 a 65.535	4 bytes ou 32 bits



# ESTRUTURAS DE DECISÃO



```
1  if (condicao) {  
2      // este bloco é executado  
3      // se a condicao for verdadeira  
4  } else if (condicao2) {  
5      // este bloco é executado  
6      // se a condicao2 for verdadeira  
7  } else {  
8      // este bloco é executado  
9      // se nenhuma das condicoes for verdadeira  
10 }
```



# IF - ELSE IF - ELSE

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a = 8;
6
7      if (a > 10) {
8          cout << "a é maior que 10" << endl;
9      } else if (a < 10) {
10         cout << "a é menor que 10" << endl;
11     } else {
12         cout << "a é igual a 10" << endl;
13     }
14
15     return 0;
16 }
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a = 20;
6
7      if (a > 10) {
8          cout << "a é maior que 10" << endl;
9      } else if (a < 10) {
10         cout << "a é menor que 10" << endl;
11     } else {
12         cout << "a é igual a 10" << endl;
13     }
14
15     return 0;
16 }
```



# OPERADORES LÓGICOS

Operadores Relacionais e Lógicos		
Tipo	Operador	Sintaxe
Menor que	<	a < b
Maior que	>	a > b
Menor ou igual que	<=	a <= b
Maior ou igual que	>=	a >= b
Igual a	==	a == b
Diferente de	!=	a != b
Não (Lógico)	!	!a
E (Lógico)	&&	a && b
Ou (Lógico)		a    b





# PRATIQUE!!!

<https://neps.academy/br/exercise/X>

**$X = [145, 147]$**





# SWITCH CASE

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int num1, num2;
6      char operator;
7      cin >> num1 >> num2 >> operator;
8
9      if (operator == '+') {
10         cout << num1 + num2 << endl;
11     } else if (operator == '-') {
12         cout << num1 - num2 << endl;
13     } else if (operator == '*') {
14         cout << num1 * num2 << endl;
15     } else if (operator == '/') {
16         cout << num1 / num2 << endl;
17     } else {
18         cout << "operador inválido\n";
19     }
20
21     return 0;
22 }
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int num1, num2;
6      char operator;
7      cin >> num1 >> num2 >> operator;
8
9      switch (operator) {
10         case '+':
11             cout << num1 + num2 << endl;
12             break;
13         case '-':
14             cout << num1 - num2 << endl;
15             break;
16         case '*':
17             cout << num1 * num2 << endl;
18             break;
19         case '/':
20             cout << num1 / num2 << endl;
21             break;
22
23         default:
24             cout << "operador inválido\n";
25             break;
26     }
27
28     return 0;
29 }
```





# PRATIQUE!!!

<https://neps.academy/br/exercise/X>


$X = [50, 87, 88, 822]$





# EXERCÍCIOS

Beecrowd > Academic > Acessar Disciplina

**ACESSAR**

INSIRA A CHAVE PARA ACESSAR A DISCIPLINA

**INFORMAÇÃO DA DISCIPLINA**

Por favor insira o ID e a chave fornecida para acessar a disciplina.

ID DISCIPLINA

14424

CHAVE

XvFza4u

ENTRAR

**POBI 2025**  
POR Roberto Cabral Rabêlo Filho

**bcwd.me/d-14424**  
**XvFza4u**

BEECROWD.COM