ICPC
MNU COMMUNINTY

Data Types & IF Conditions

PREPERED BY: ABDULLAH HAITHAM

Agenda

- WHAT ARE DATA TYPES
- BASIC DATA TYPES
- TYPE CONVERSIONS
- INPUTS & OUTPUTS
- OPERATORS
- IF CONDITIONS
- PRACTICE

DATA TYPES

DATA TYPES 1

What is Data Types ?

A categories of data , that inform the computer:

- **what kind of value to store**
- **how much memory needed**
- **what operations we can do**

DECLARATION

Syntax:

1.
```
DataType Variable_Name = value;
```

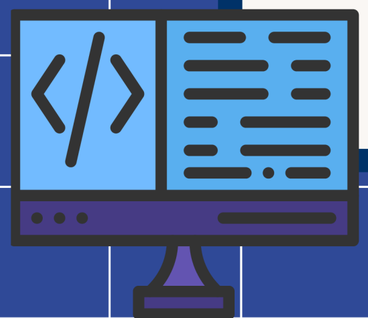
2.
```
DataType Variable_Name;
varaibel_Name = value
```

Example:

```
int example =5;
```

or

```
int example;
example =5;
```



TYPES

short : stores only integers (Size → 2 Bytes)

int : stores Integer values(Size→ 4Bytes)

double : stores both decimals & Integers(Size→ 8Bytes)

float : stores both decimals & Integers(Size→ 4Bytes)

long long : stores Integers values (Size→ 8Bytes)

char : store character (Size→ 1Byte)

string : store a word(group of characters)(Size->Depends)

bool : True or False (Size→ 1Byte)

TYPES

short : stores only integers (Size → 2 Bytes)

int : stores Integer values(Size→ 4Bytes)

double : stores both decimals & Integers(Size→ 8Bytes)

float : stores both decimals & Integers(Size→ 4Bytes)

long long : stores Integers values (Size→ 8Bytes)

char : store character (Size→ 1Byte)

string : store a word(group of characters)(Size->Depends)

bool : True or False (Size→ 1Byte)

SCIENTIFIC NOTATION

Integer Types

- int → Range: -2.1×10^{9} → $+2.1 \times 10^{9}$
- long long → Range: -9.2×10^{18} → $+9.2 \times 10^{18}$
- char → Range: -128 → +127 (small integer / character)

Floating Point Types

- float → Range: ~1.2×10^{-38} → 3.4×10^{38} (~7 digits precision)
- double → Range: ~2.3×10^{-308} → 1.7×10^{308} (~15 digits precision)

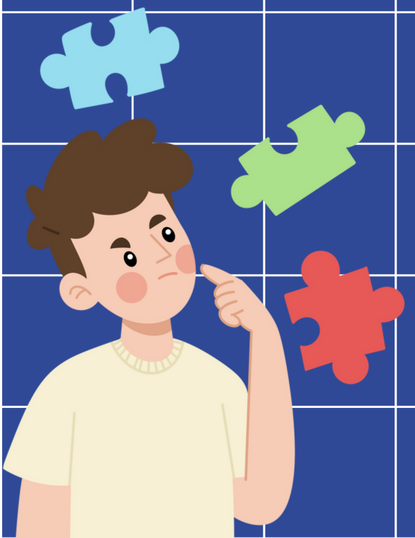
Other Types

- string → sequence of characters (no fixed range, depends on memory)
- bool → values: 0 (false) or 1 (true)

```cpp
#include<bits/stdc++.h>
using namespace std;
int main (){

    int a =5;
    long long b = 10000000000;
    float c = 4.9;
    double d = 8.20;
    char charachter = '?';
    string name = "Welcome to ICPC MNU !!" ;
    bool stat = false;


}
```



TYPE CONVERSION

A process that is used to convert from one Data Type to Another.

Syntax:

```
Data_Type(Varaible);
```

Note : Doesn’t work with conversions to strings !

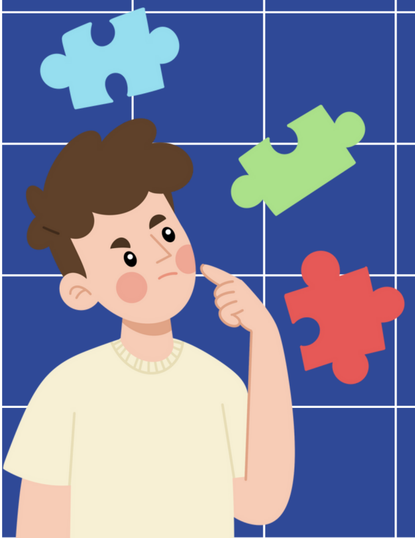
Example:

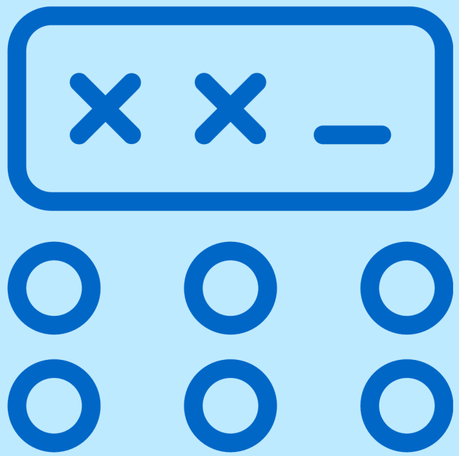
```
float a = 3.7;
int b = int(a);
-----------------
float a = 3.7;
cout<<int(a)<<endl;
```

```cpp
#include<string>
using namespace std;
int main (){

    double value = 3.20;
    //convert a decimcal value to integer
    cout <<int(value)<<endl;

    int value_2 = 550;
    //Convert A numeric integer value to string
    cout <<to_string(value_2)<<endl;
    //Note : Should use #include<string>
}
```





INPUTS/OUTPUTS

INPUT/OUTPUT

```
#include<iostream>
```

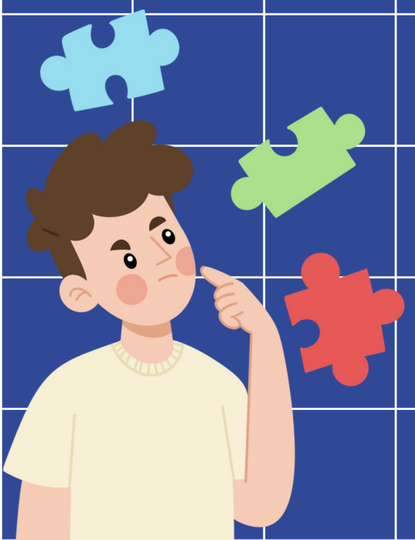
Input

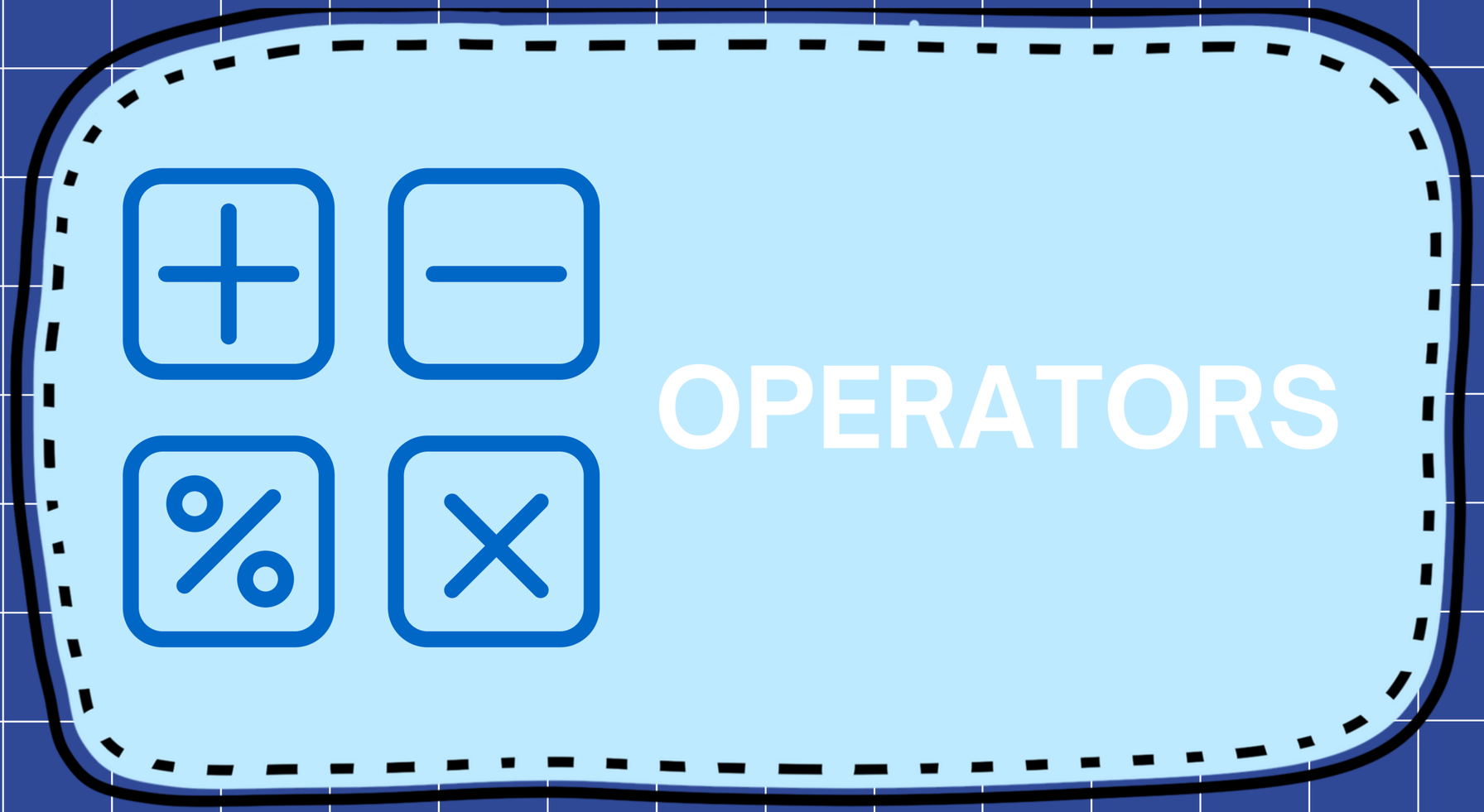
```
cin>>a;
cin>>a>>b
```

Output

```
cout<<a;
cout<<a<<endl;
cout<< a<< “\n”;
cout<<fixed<<setprecision(int)<<a;
Note: Should #include<iomanip>
```

```cpp
#include<iostream>
#include<iomanip>
using namespace std;
int main (){
    float a;
    cin>>a; // 2.4
    cout <<a<<endl; // 2.4
    cout <<a<<"\n"; //2.4
    cout<<fixed<<setprecision(0)<<a<<endl;
    //2
   // Note : To use setprecision(x),
   #include<iomanip>
}
```




OPERATORS

Types of Operators:

1. Unary Operators (++ ,--)
2. Arithmetic Operators (+, - ,* ,/ ,%)
3. Relational Operators (> ,< , >= , <= , == , !=)
4. logical Operators (&& , || , !)
5. Assignment Operators (= , += , -= ,*= , /= ,%=)
6. Ternary Operator (?:)

PROBLEMS

PROBLEMS

1.
Write A program that initialize two variables a,b and print the sum , sub , mul of the two varaibles

2.
Write A program that initialize variable a and print the double value of a.

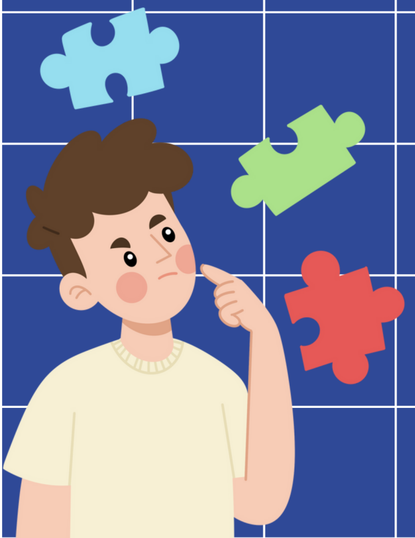
3.

Given a integer value of 1234 print each digit in a separate line.

SOLUTIONS

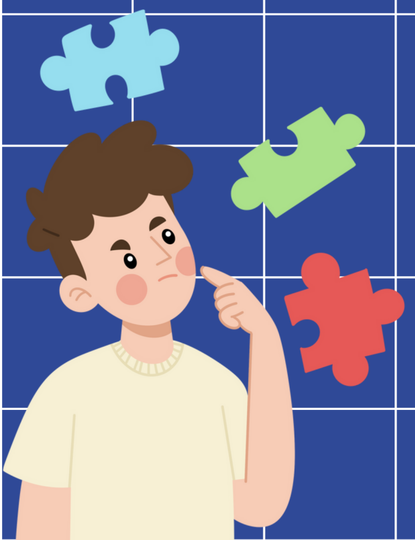
1.

```cpp
#include<iostream>
using namespace std;
int main (){
    int a,b;
    cin>>a>>b;
    int sum,sub,mul;
    sum = a+b;
    sub = a-b;
    mul = a*b;
    cout<<sum<<endl;
    cout<<sub<<endl;
    cout<<mul<<endl;
}
```



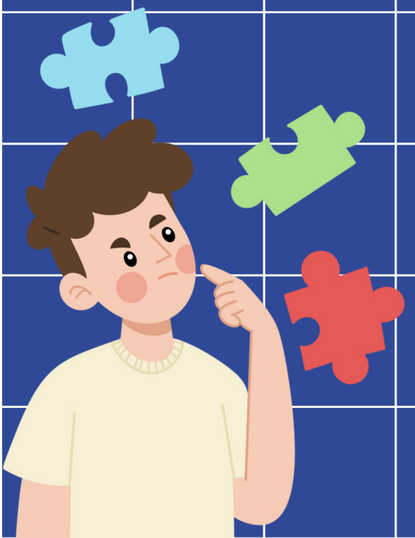
2.

```cpp
#include<iostream>
using namespace std;
int main (){
    int a,b;
    cin>>a;
    a*=2;
    cout<<a<<endl;
}
```



3.

```cpp
#include<iostream>
using namespace std;
int main (){
    int value = 1234;
    int v1 = value % 10;
    int v2 = (value / 10) % 10;
    int v3 = (value / 100) % 10;
    int v4 = (value / 1000) % 10;
    cout<<v1<<endl;
    cout<<v2<<endl;
    cout<<v3<<endl;
    cout<<v4<<endl;
}
```

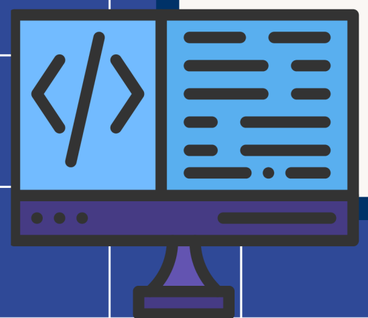


IF CONDITIONS

IF STATEMENT

What is an if statement?

- It’s a decision-making statement.
- It lets the program choose whether to run a block of code or skip it.
- Works like a real-life "if condition happens, then do something."



DECLARATION

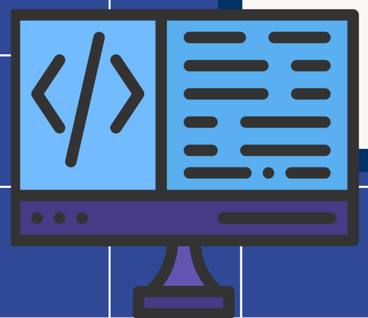
Syntax:

1.

```
if (condition) {
 // code to run if condition is
true
}
```

2. Using Ternary Operator

```
 cout << (condition)? : “stat1”:”stat2" <<endl;
```



```cpp
#include<iostream>
using namespace std;
int main (){
    int x;
    cin>>x;
    cout<<(x>0 ? "YES":"NO")<<endl;
}
```

IF

- Checks one condition
- Runs code only if true

```
if (x > 0)
cout << "+";
```

ELSE

- Works as the “otherwise”
- Runs when the if condition is false

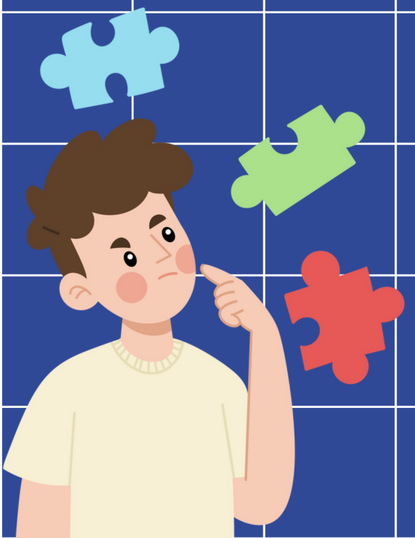
```
else
cout << "Not +";
```

ELSE IF

- Adds more conditions to check
- Used between if and else

```
else if (x == 0)
cout << "Zero";
```

```cpp
int main() {
    int x;
    cin >> x;
    // Simple if
    if (x > 0) {
        cout << "The number is positive" << endl;
    }
    // if - else
    if (x % 2 == 0) {
        cout << "The number is even" << endl;
    } else {
        cout << "The number is odd" << endl;
    }
    // if - else if - else
    if (x > 0) {
        cout << "The number is greater than zero" << endl;
    } else if (x < 0) {
        cout << "The number is less than zero" << endl;
    } else {
        cout << "The number is zero" << endl;
    }
}
```



PROBLEM

I. Welcome for you with Conditions

time limit per test: 1 second

memory limit per test: 64 megabytes

Given two numbers A and B. Print "**Yes**" if A is **greater than or equal to** B. Otherwise print "**No**".

Input

Only one line containing two numbers A and B ($0 \leq A, B \leq 100$).

Output

Print "Yes" or "No" according to the statement.

Examples

input	Copy
10 9	

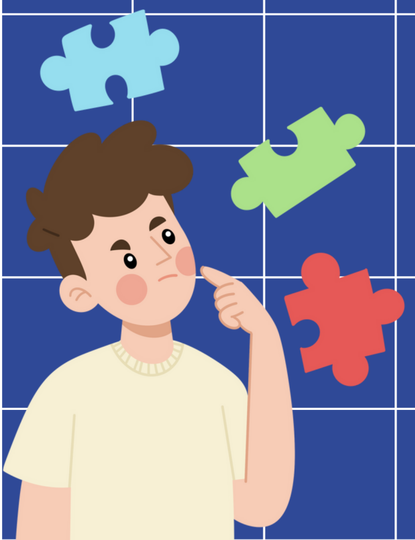
output	Copy
Yes	

input	Copy
5 5	

output	Copy
Yes	

input	Copy
5 7	

output	Copy
No	



SOLUTION

Competitive.cpp > ...

```cpp
#include<iostream>
using namespace std;
int main (){
    int a, b;
cin >> a >> b;
if (a >= b) {
    cout << "Yes";
}
else
cout << "No";
}
```

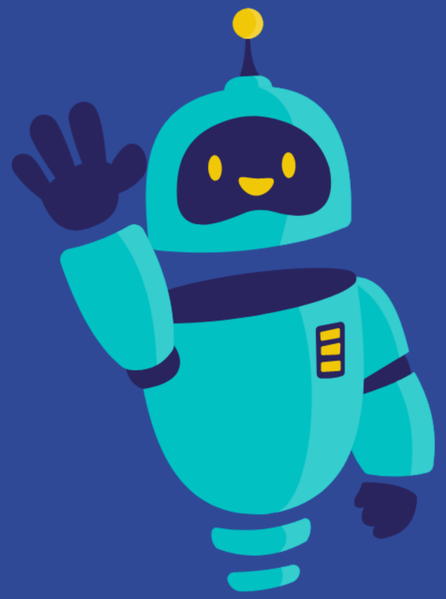
PSEUDO CODE

```
START

 DECLARE integer a, b
    INPUT a, b
     IF a >= b THEN
         PRINT "Yes"
     ELSE
        PRINT "No"
     ENDIF
END
```

CODEFORCE TIME !

THANK YOU!



FOLLOW US !