ICPC Assiut Community

Newcomers Training

Functions

ICPC Assiut
community

Functions

- **What is the Functions ?**

- It's a way to make your code more simple by **divide the big problem to smaller problems.**
- it's a group of statement that is given a name, and which can be called from some point of the program.

- **The most common syntax to define a function is :**

```
Type Name (Parameter1, Parameter2, ...) {
     Statement
}
```

- **Type :** is the type of the value returned by the function.

- **Name :** is the identifier by which the function can be called.

- **Parameter** : Each parameter consists of a type followed by an identifier, each parameter looks very much like a regular variable declaration, the purpose of parameters is to allow passing arguments to the function from the location where it is called from.

- **Statement** : is the function's body. It is a block of statements surrounded by braces { } that specify what the function actually does.

Functions

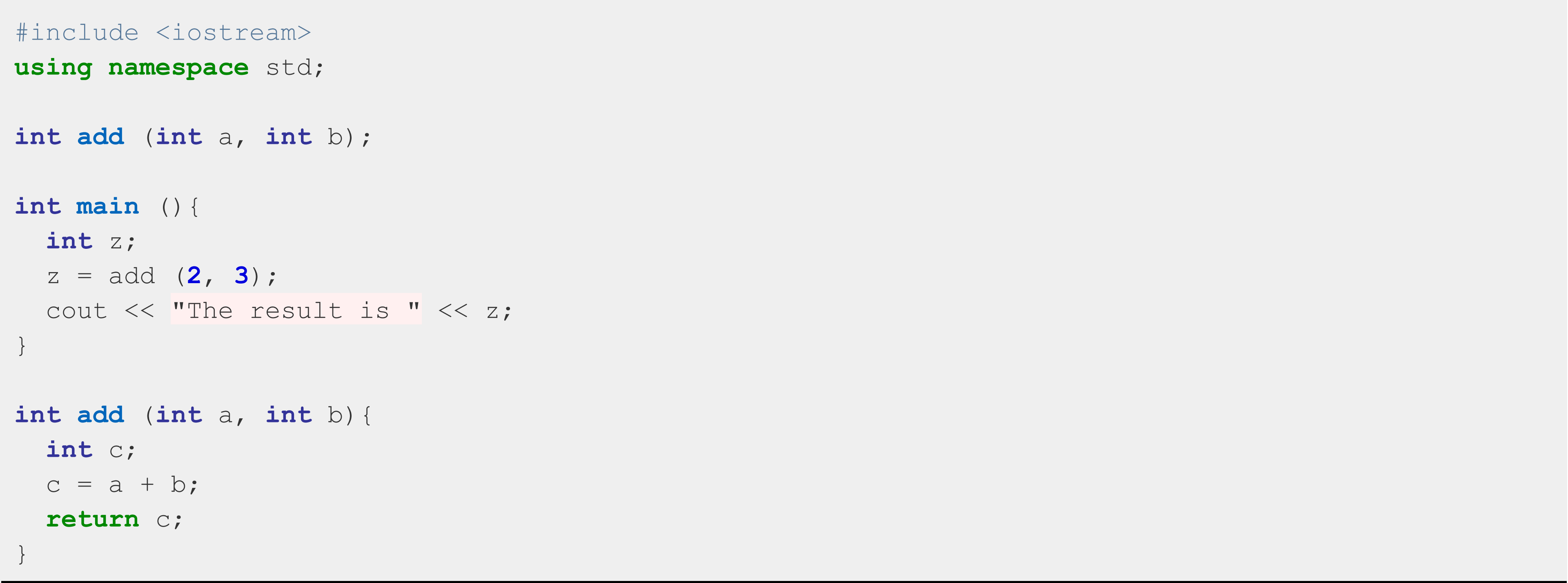
- An Example for the Functions

```cpp
#include <iostream>
using namespace std;

int add (int a, int b){
  int c;
  c = a + b;
  return c;
}

int main (){
  int z;
  z = add (2, 3);
  cout << "The result is " << z;
}
```

```
Output :
The result is 5
```

Functions

- Same example but with another way to write the function



```cpp
#include <iostream>
using namespace std;

int add (int a, int b);

int main (){
  int z;
  z = add (2, 3);
  cout << "The result is " << z;
}

int add (int a, int b){
  int c;
  c = a + b;
  return c;
}
```

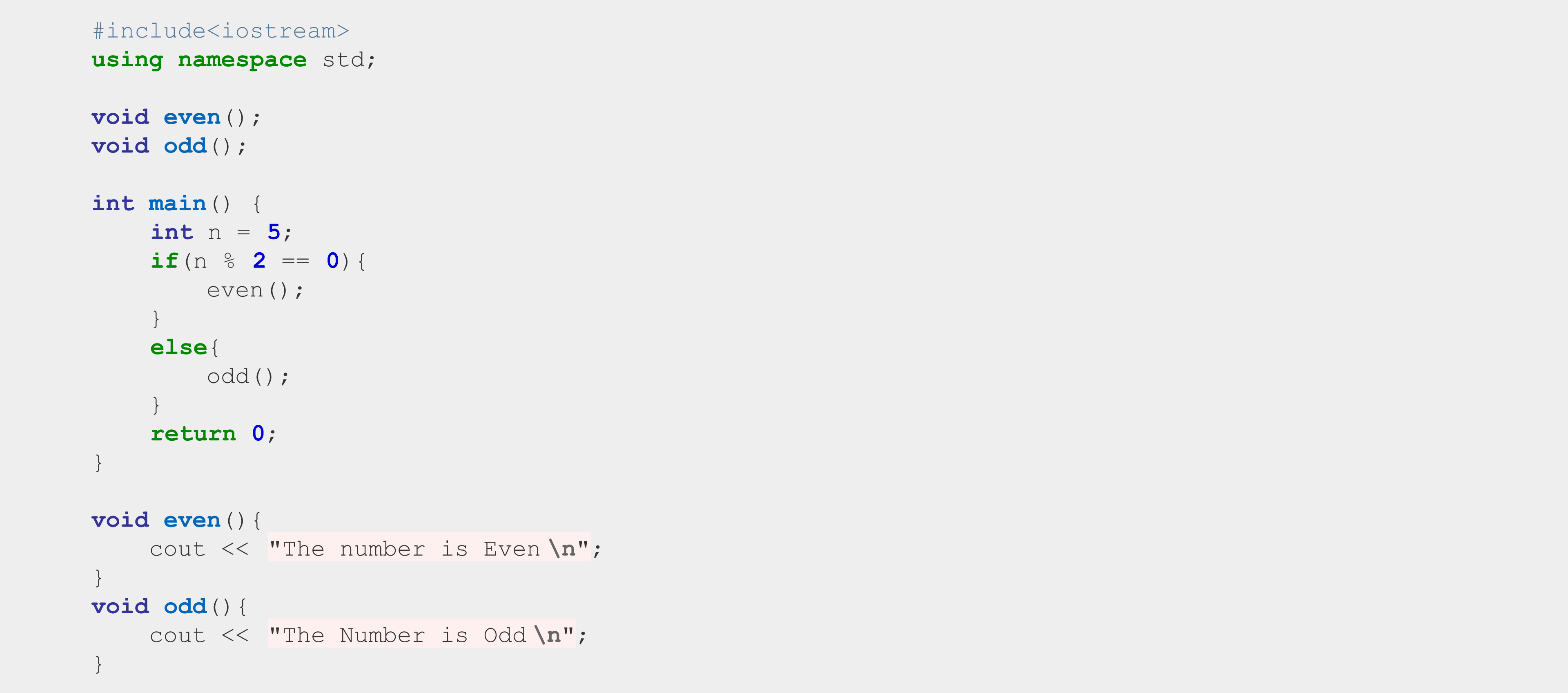
Output :

```
The result is 5
```

Void Functions

Void Functions : it's a function that do operations without returning a value.

Ex :



```
#include<iostream>
using namespace std;

void even();
void odd();

int main() {
    int n = 5;
    if(n % 2 == 0){
        even();
    }
    else{
        odd();
    }
    return 0;
}

void even(){
    cout << "The number is Even \n";
}
void odd(){
    cout << "The Number is Odd \n";
}
```

Functions

- Write a functions that take 2 integers and return the maximum number of them.

```
#include<iostream>
using namespace std;

int Max_number(int a, int b);

int main() {
    int a, b;
    cin >> a >> b;
    cout << Max_number(a, b) << endl;
    return 0;
}

int Max_number(int a, int b){
    if(a >= b){
        return a;
    }
    else{
        return b;
    }
}
```

Problems

- Write a function that take an integer number N and return the summation of numbers between 1...N .

  (1 <= N <= 1000)

- Write a function that take an integer number N and return the factorial of N .

  (1 <= N <= 100)

- Write a function that take an integer number N and return **true** if the number is Prime otherwise return **false**.

- Write a program that take an integer N (size of array) and N numbers in the Array, and a function that take the array and it's size and return the summation of its elements.

Problems

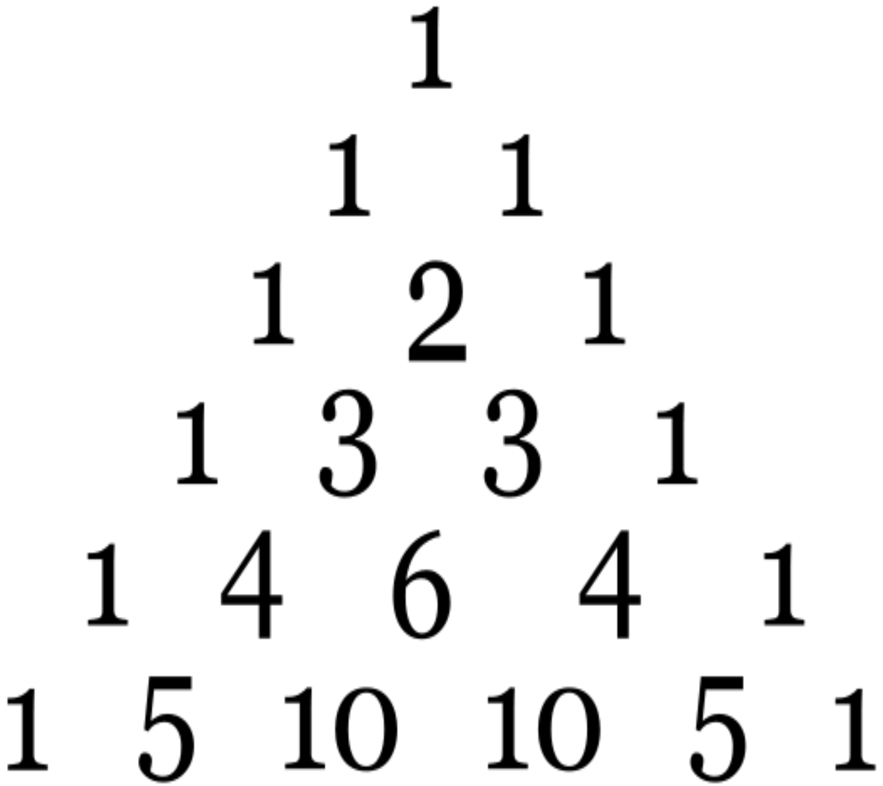
- Write a program that take ***N*** elements in the array and **<u>void function</u>** that print **“All the number is Even”** if all the numbers in the array is even, or print **“All the number is Odd”** if all the numbers in the array is odd, otherwise print **“All the numbers is strangers”**.

  <u>(you can write a void function more than once)</u>

- Write a void function that take a string X and print “Hello, X“ .

- Write a function that take a string X contain a small letters and spaces, and return the string without any spaces.

Pascal's Triangle



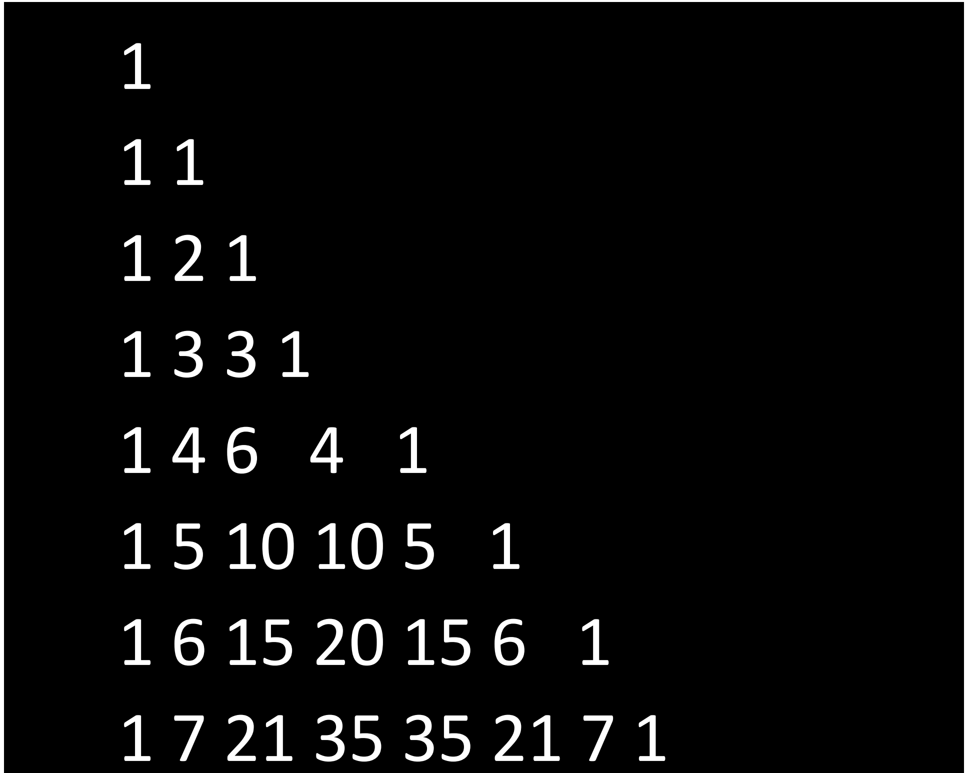
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

Every cell equal to the sum of the two numbers above it

Problems

- Write a void function that take an integer ***N*** and print ***N*** rows from Pascal's Triangle .



```
1
1 1
1 2 1
1 3 3 1
1 4 6   4   1
1 5 10 10 5   1
1 6 15 20 15 6   1
1 7 21 35 35 21 7 1
```

For more information about **Functions** visit this Link

Now it's time to practise and solve the problems of Functions

Functions Sheet

Good luck <3