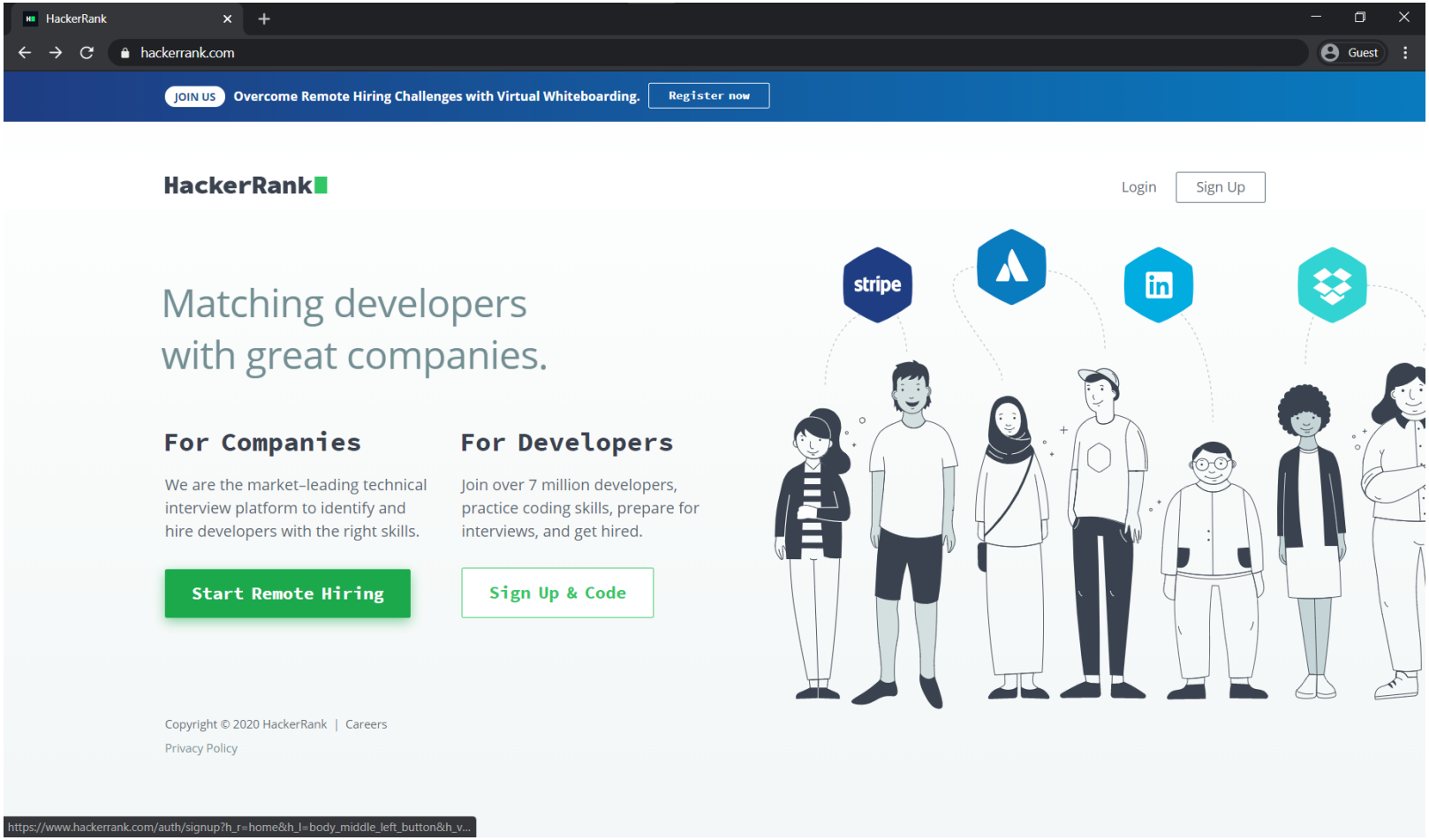
Techimax

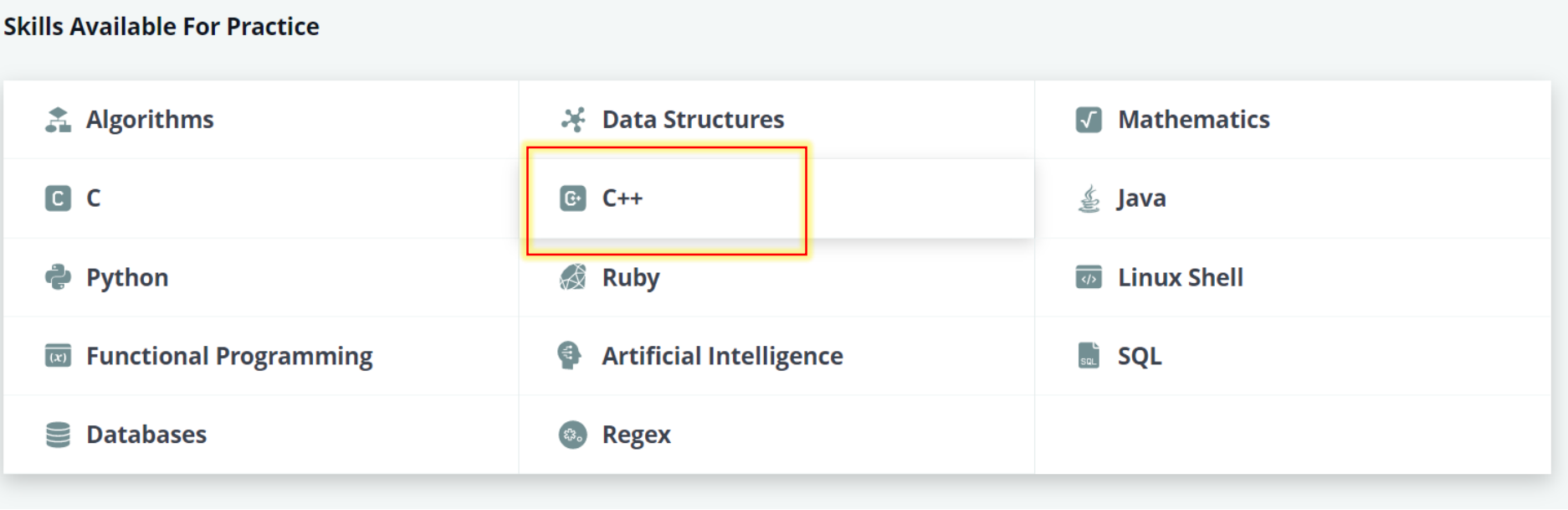
Fundamentals of Programming in C++

Where to Practice ?

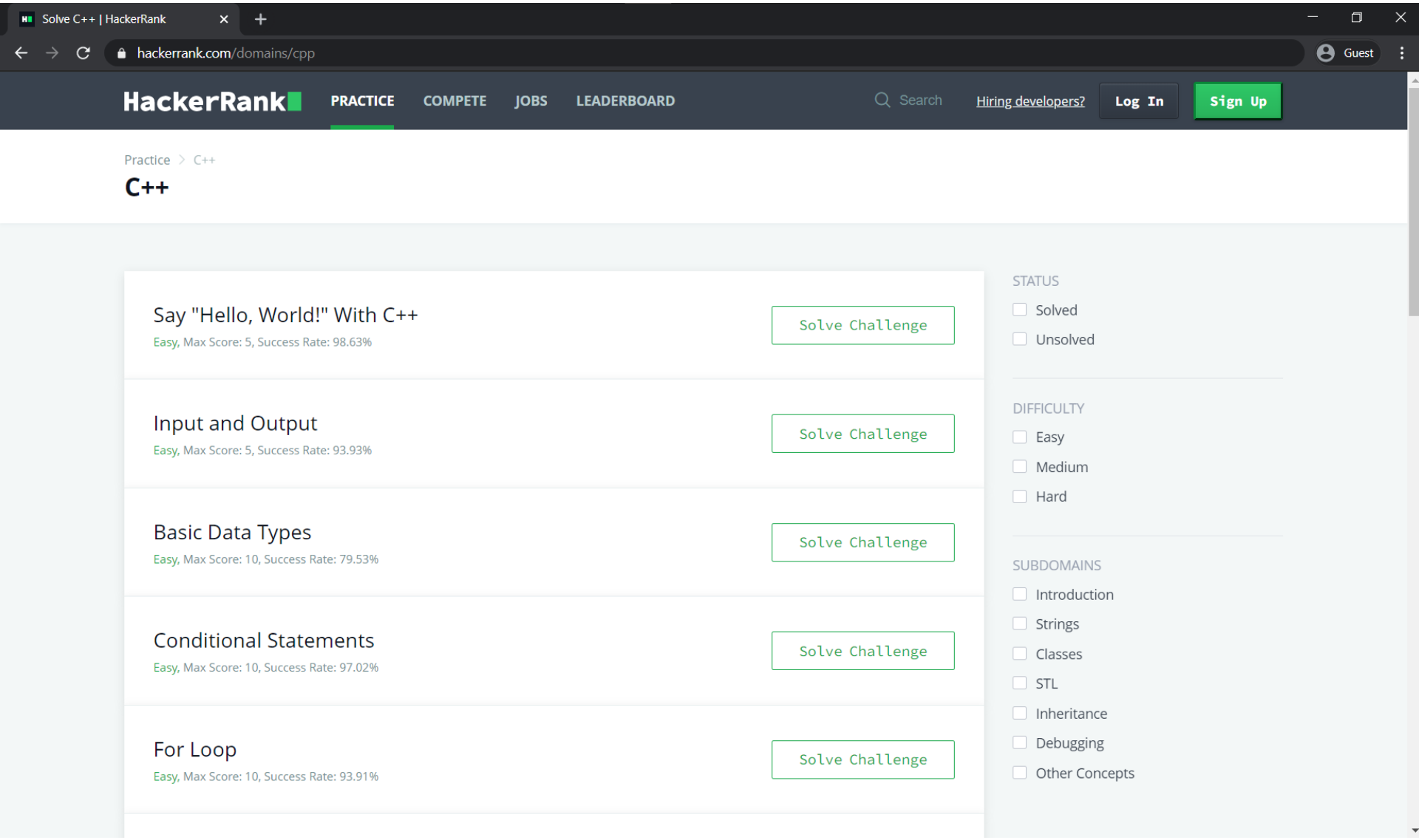
https://www.hackerrank.com/



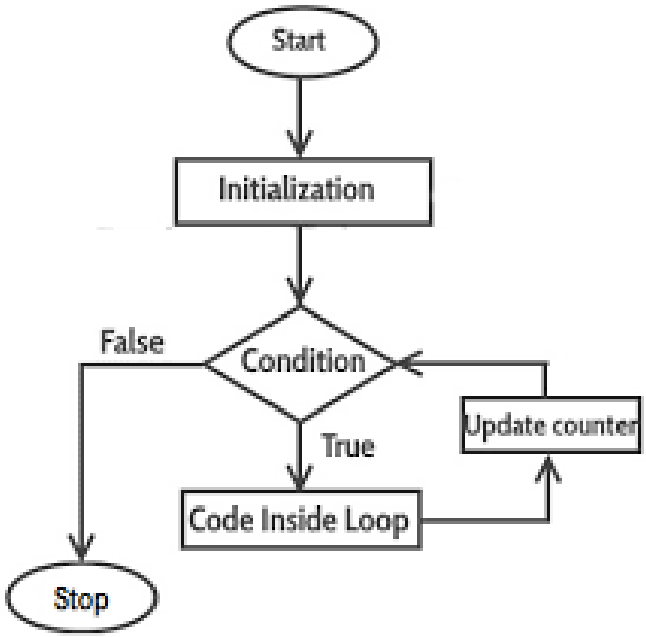
Where to Practice ?



https://www.hackerrank.com/domains/cpp

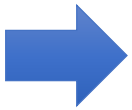


Loops




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

int main()
{
    cout << "I love Programming <3 \n";
    cout << "I love Programming <3 \n";
    cout << "I love Programming <3 \n";
    cout << "I love Programming <3 \n";
    cout << "I love Programming <3 \n";
    cout << "I love Programming <3 \n";
}
```



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

int main()
{
    while (1)
    {
        cout << "I love Programming <3 \n";
    }
}
```

For loop


```
for (initialization expr; test expr; update expr)
{
    // body of the loop
    // statements to execute
}
```

```
for (int i = 0; i < 5; i++)
{
    cout << i << '\n';
}
```

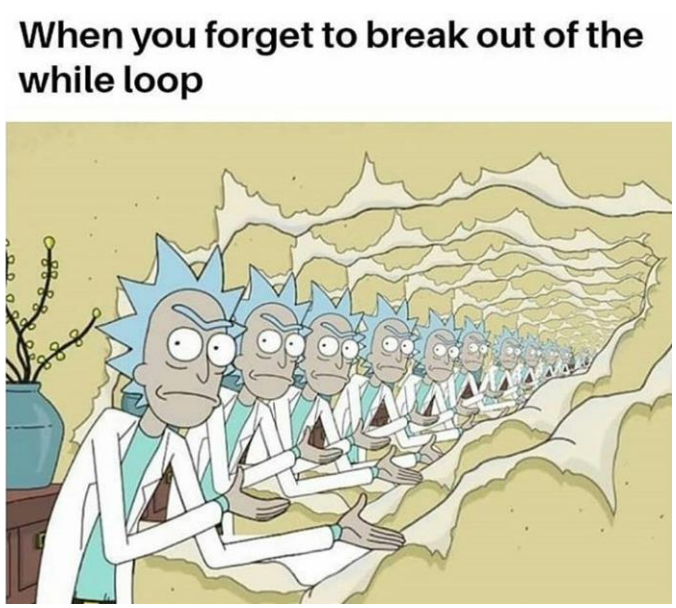
```
0
1
2
3
4
```

While loop

```
initialization expression;
while (test_expression)
{
    // statements

    update_expression;
}
```

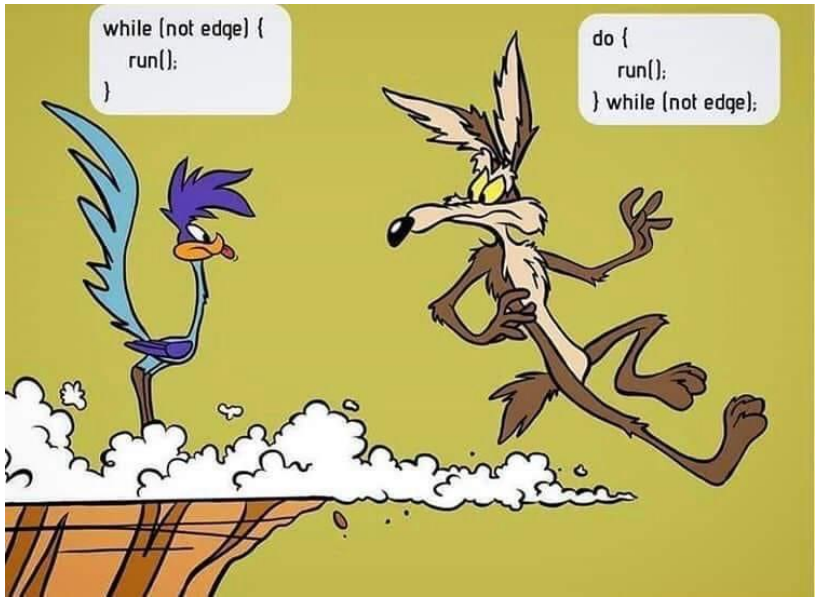
```
int T = 6;
while (T)
{
    cout << T << '\n';
    T--;
}
```



do while loop

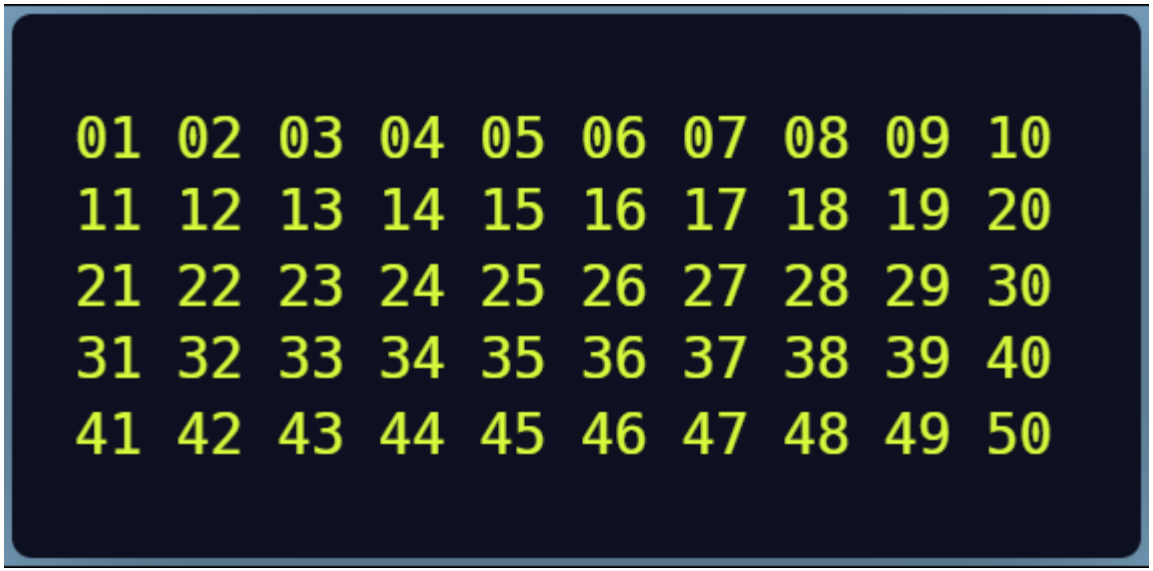
```
initialization expression;
do
{
    // statements

    update_expression;
} while (test_expression);
```




```
int i = 3;
do
{
    cout << "Hey !!\n";
    i++;
} while (i < 3);
```

Lets Print this pattern



```
01 02 03 04 05 06 07 08 09 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
```

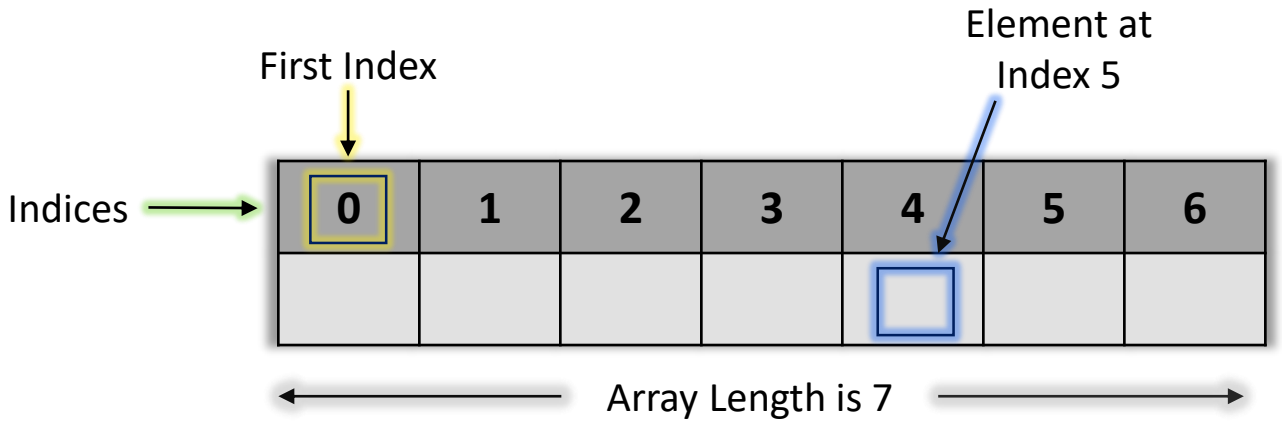
```
int count = 1;
while (count <= 50)
{
    if (count < 10)
    {
        cout << "0" << count << ' ';
    }
    else
    {
        cout << count << ' ';
    }

    if (count % 10 == 0)
    {
        cout << '\n';
    }

    ++count;
}
```

Arrays

A data structure that contains a group of elements of the same type



Initialize Arrays

```
float arrFloat[] = {10.56, 4.6, 3.3, 7.8};
```

```
int arr1[10];
```

```
int n = 10;
int arr2[n];
```

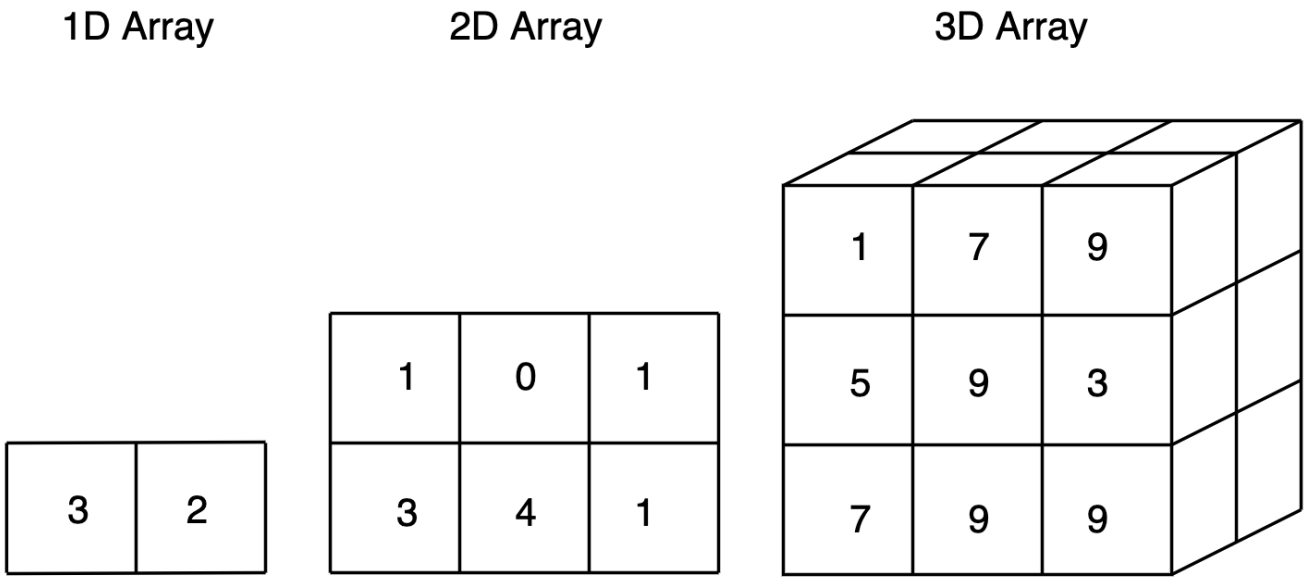
```
int arr[5];
arr[0] = 5;
arr[1] = 3;
arr[2] = 65;
arr[3] = 467;
arr[4] = -98;
```

Accessing Elements of an Array

```
int n;
cin >> n;
int arr[n];
for (int i = 0; i < n; i++)
{
    cin >> arr[i];
}
```

```
for (int i = 0; i < 5; i++)
{
    cout << arr[i] << " ";
}
```

Multidimensional Array

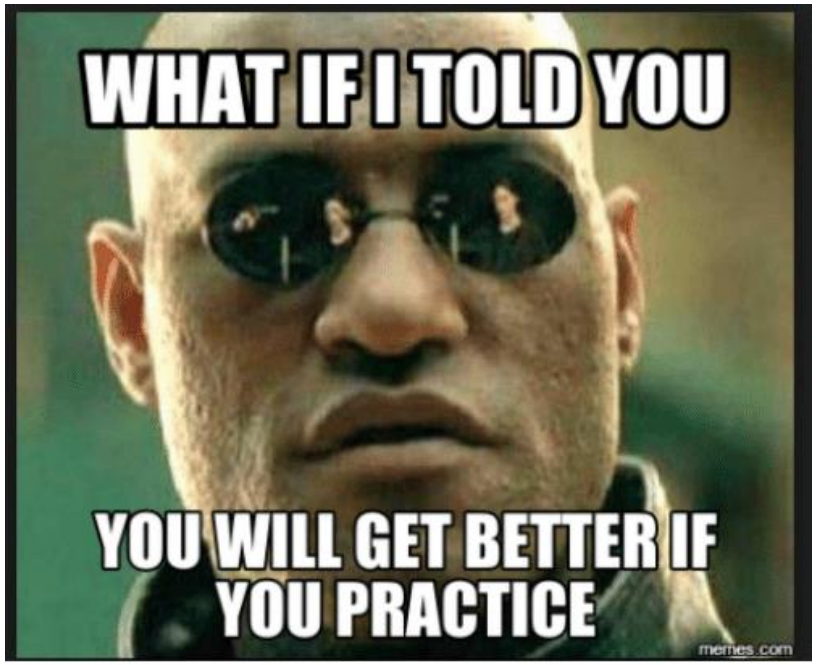



```
int arr_1d[2];
int arr_2d[2][3];
int arr_3d[3][3][2];
```

2D Array

```
int arr[3][2] = {{1, 2}, {3, 4}, {5, 6}};
```

```
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 2; j++)
    {
        cout << arr[i][j] << " ";
    }
    cout << '\n';
}
```



Write a code to print this pattern ->

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Make a Calculator

Take two numbers and
a mathematical operator as input.
print the output of the arithematic operation.

Given the size of an array and its elements,
write a function to get its maximum element.

Matrix Addition

Take the dimensions m*n and
the elements of two 2D arrays
as input from the user and
print the resultant array of the matrix addition.