24 August 2023 20:49

While Loop
Do while loop

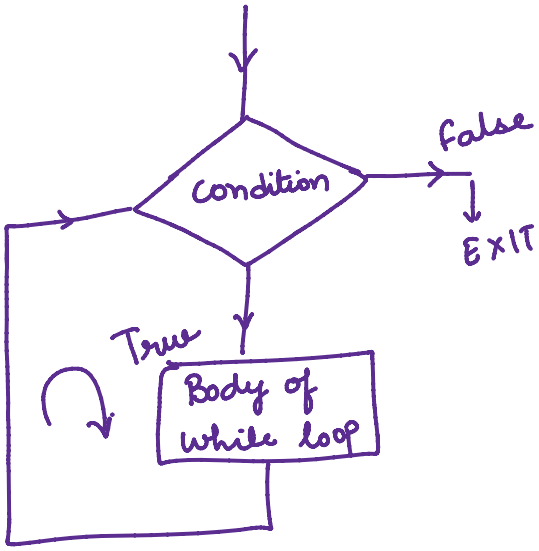
Arrays
Vectors

While :

```
while (condition) {

    ==========

}
```



Print 1 to 5 :

```
int i = 1;
while (i <= 5) {
    cout << i;
```

	2<=5	3<=5	4<=5	5<=5	6<=5
1	2	3	4	5	False

```
while (i <= 5)
{
    cout << i;
    i++;
}
```

1	2	3	4	5	False
i=2	i=3	i=4	i=5	i=6	

For / While — Difference → Syntax

Working → Same

Sum of no.s from 1 to 10 → "For" is preferred

```
sum = 0
sum = sum + 1
sum = sum + 2
sum = sum + 3
sum = sum + 4
sum = sum + 5
  :
sum = sum + 10
```

Sum of digits of a number → "WHILE" is preferred

N = 23

```
sum = 0
sum = sum + 3
sum = sum + 2
```

2 times

N = 564

```
sum = 0
sum = sum + 4
sum = sum + 6
sum = sum + 5
```

3 times

N = 21835

5 times

Sum of digits of a number

Sum of digits of a number

N = 256 ✓ ✓ ✓ Sum = 2 + 5 + 6 = 13

sum = 0

sum = sum + 6 = 6

sum = sum + 5 = 11

sum = sum + 2 = 13

N%10 → Last digit of a number

N = 256^x Sum = 0

{ digit = N%10 = 6
Sum = sum + digit = 0 + 6 = 6
N = N/10 }

256/10 25.6 25

N = 25

{ digit = N%10 = 5
sum = sum + 5 = 11
N = N/10 }

N = 2

digit = N%10 = 2

Sum = sum + digit = 11 + 2 = 13

N = N/10 2/10 = 0

```
while (num > 0) {
    digit = num%10;
    Sum += digit;
    num = num/10;
}
```

23578

23570

8

n%10 → Remainder when divided by 10 ⇒ Last digit of the number

N = 251 Reverse = 152

n = 2, p = 5 n = 25

n*10 + p

20 + 5 = 25

n = 25, p = 3 n = 253

n*10 + p

250 + 3 = 253

```
int rev = 0;
while (N != 0)
{  digit = N % 10;
   rev = rev * 10 + digit;
   N = N / 10;
}
```

N = 2713

rev = 0

N != 0 True: digit = N % 10 = 2713 % 10 = 3

rev = rev * 10 + digit

= 0 * 10 + 3 = 3

N = N / 10 = 271

N != 0 True digit = 1

rev = rev * 10 + digit

= 3 * 10 + 1 = 31

N = N / 10 = 27

N != 0 True: digit = 7

rev = rev * 10 + digit

= 31 * 10 + 7

= 310 + 7 = 317

N = N / 10 = 2

N != 0 True : digit = 2

rev = rev * 10 + digit

N!=0 ...

rev = rev * 10 + digit

= 3170 + 2 = 3172

N = N/10 = 0

N!=0 False

cout << rev. → 3172

N = 305

N!=0 True digit = N%10 = 5

rev = 0*10+5 = 5

N = N/10 = 30

N!=0 True: digit = 0

rev = 5*0+0 = 50

N = N/10 = 3

N!=0 True: digit = 3%10 = 3

rev = 50*10+3 = 503

N = N/10 = 0

N!=0 False

Palindrome

N = 121 ↺ same → Palindrome

rev = 121

```
int temp = n;
while (n!=0) {
```

```
while (n!=0) {
    d = n%10;
    rev = rev*10 + d;
    n = n/10;
}
```

$\Longrightarrow$ n = ? $\boxed{n=0}$

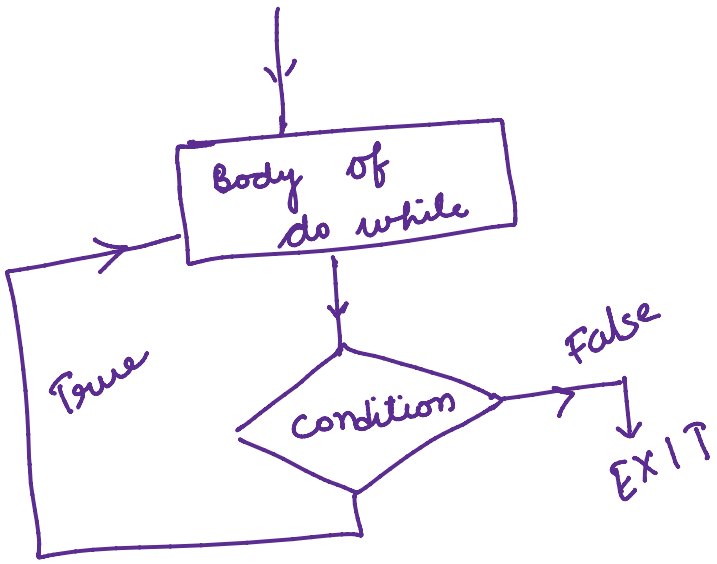
```
if (rev == n) {
    temp
}
```

$\longrightarrow$ Palindrome

Do while loop.

```
do {

} while (condition);
```



Print no. from 1 to 5

```
int i=1;
```

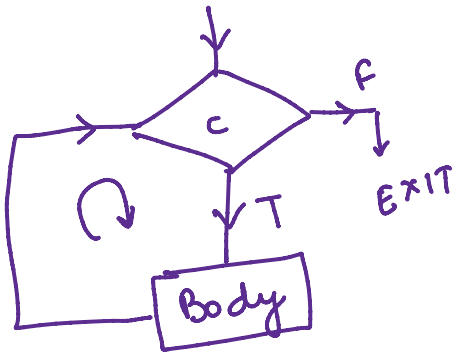
i=1

```
int i=1;
do {
    cout << i;
    i++;
} while (i <=5);
```

i=1 → 1 → i=2 → 2<=5 ↑ → 2 → i=3 → 3<=5 ↑ → 3 → i=4 → 4<=5 ↑ → 4 → i=5 → 5<=5 ↑ → 5 → i=6 → 6<=5 ✗

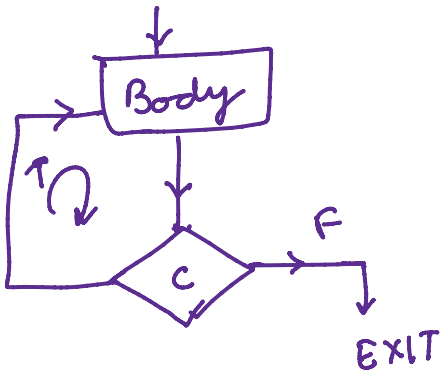
o/p: 12345

While



entry controlled loop

Do-while



Exit controlled loop.

Condition is true:

While = do while

Condition is false:

loop is not executed

loop is executed atleast once

```
int i=1;
while (i>10) {
    cout << i;
    i++;
```

```
int i=1;
do {
    cout << i;
    i++;
```

cout << i;

i++;

}

i=1: i>10 False

No output

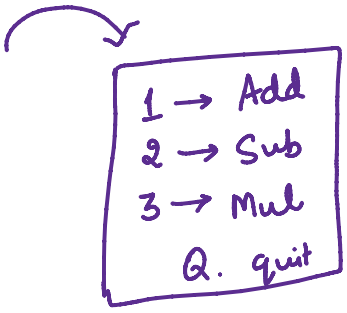
cout <<...

i++;

} while (i>10);

i=1 ↓ Print 1 i=2

2>10 False



Do while loop.

Q.

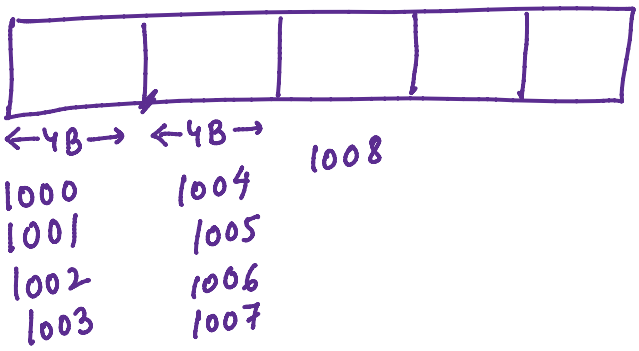
Array

collection of elements of same data type.

"Contiguous"

Array of size 5 : (Int) STATIC

5 * 4 bytes = 20 B.



1003 1007

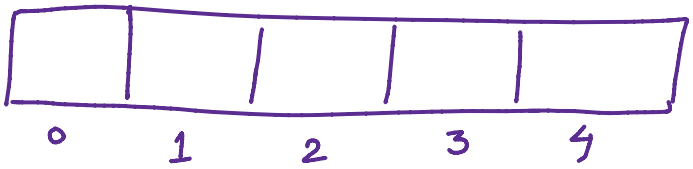
datatype name [size];

int arr[5]; // declaring

int arr[ ] = {5, 2, 3, 9};

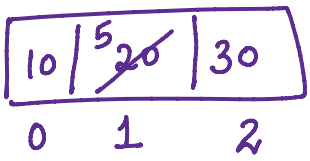
int arr [4] = {1,2, 3,4}

arr [5]



Index starts from 0.

int arr[ ] = {10, 20, 30};



arr [0] → 10
arr [1] → 20
arr [2] → 30

arr [1] = 5;

Print all values of an array:

```
for (int i = 0; i < 3; i++) {
        cout << arr [i];
}
```

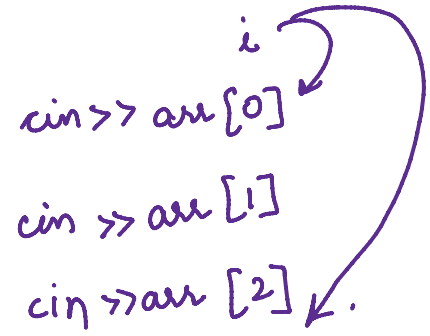
i=0 :	i<3	cout << arr[0]	i++
i=1	i<3	cout << arr [1]	i++

i=1 i<3 } cout << arr[1] i++

i=2 i<3 cout << arr[2] i++

i=3 i<3 False

```
for (int i = 0; i < 3; i++) {
        cin >> arr[i];
}
```

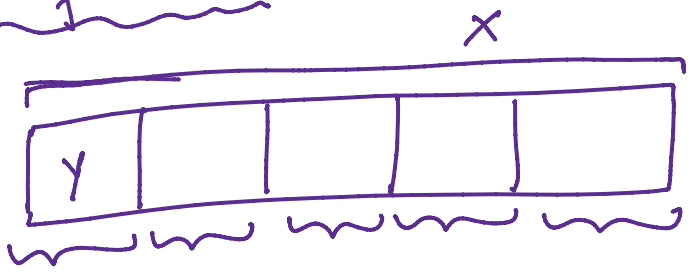


sizeof (arrayname) → size (in bytes)

int arr[5]; → 5 elements of 4B each

Size = 20 B.

Number of elements :



N = X/Y

sizeof (arr) / sizeof (arr[0])

sizeof (arr) / sizeof (arr[0])

OR

size of (arr) / sizeof (datatype)

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

Sum of all the elements in an array

arr = { 5, 2, 3, 1, 8 }

sum = 19

```
sum = 0;
for (int i= 0; i <N; i++) {
    sum += arr[i];
}
```

arr → { 5, 2, 1, 3 }

N = 16/4 = 4

sum = 0

i = 0 i<4 True sum = sum + arr[0] = 0 + 5 = 5 i++

i=1 i<4 sum = sum + arr[1] i++

$i=1$ $i<4$ True

$sum = sum + arr[1]$

$= 5 + 2 = 7$ $i++$

$i=2$ $i<4$ True

$sum = sum + arr[2]$

$= 7 + 1 = 8$ $i++$

$i=3$ $i<4$ True

$sum = sum + arr[3]$

$= 8 + 3 = 11$ $i++$

$i=4$ $i<4$ False

EXIT

Vector : Dynamic Array → Change size

vector < datatype > name;

vector <int> vec;

size → "name.size()"

```
#include<iostream> //Header File
#include<bits/stdc++.h>
using namespace std;
int main(){ //Start of the main function
   vector<int> vec={2,6,4};
   cout<<"Size = "<<vec.size()<<endl;
   for(int i=0;i<vec.size();i++){
      cout<<vec[i]<<" ";
   }
   cout<<endl;
   vec.push_back(9); //Adds an element at the end
   cout<<"Size = "<<vec.size()<<endl;
   for(int i=0;i<vec.size();i++){
      cout<<vec[i]<<" ";
   }
   cout<<endl;
   vec.insert(vec.begin()+1,10); //Insert 10 at index 1
   cout<<"Size = "<<vec.size()<<endl;
   for(int i=0;i<vec.size();i++){
      cout<<vec[i]<<" ";
   }
   cout<<endl;
   vec.pop_back(); //Removes the last element
   cout<<"Size = "<<vec.size()<<endl;
   for(int i=0;i<vec.size();i++){
      cout<<vec[i]<<" ";
   }
   cout<<endl;
   vec.erase(vec.begin()+2);
```

```
    cout<<"Size = "<<vec.size()<<endl;
    for(int i=0;i<vec.size();i++){
        cout<<vec[i]<<" ";
    }
    cout<<endl;
}
```

Size — vec.size()

Insertion:

① push_back(ele)

vec.push_back(5);

↳ Insert 5 at the end.

② insert(index, ele)

vec.insert(vec.begin()+1, 4);

↳ Insert 4 at index 1.

Deletion:

① pop_back()

vec.pop_back();

↳ Remove last element

② erase(index)

vec.erase(vec.begin() + 2)

↳ Remove element at index 2.