Experiment No-04: Vector, Stack, and Queue in C++.

Objectives

- Introduce with vector in C++.
- Introduce with stack and its operations in C++.
- Introduce with queue and its operations in C++.

Example 1: Vector in C++. [Vector]

```
/**
vector: Member Functions

1) push_back(element) -----> push_back() is used for inserting an
    element at the end of the vector

2) pop_back()       -----> pop_back() is used to remove the last
    element from the vector. It reduces the size of the vector by one.

8) clear()          ----- > This method clears the whole vector,
    removes all the elements from the vector but do not delete the
    vector.
9) size()           ------> returns the size of the vector

**/

#include<bits/stdc++.h>
using namespace std;

int main()
{
    vector<int>vec1; //int type vector declaration
    vector<string>vec2; // string type vector declaration

    // Push_back operatin on vec1

    for(int i=0;i<5;i++){
        vec1.push_back(i);
    }
    vec1.push_back(100);
    vec1.push_back(10);
    vec1.push_back(23);
    vec1.push_back(9);

// Print the elements of the vector
for(int i=0;i<vec1.size();i++){
    cout<<vec1[i]<<"\t";
}
}
```

Example 2: Stack in C++. [Stack]

```
#include<bits/stdc++.h>
using namespace std;

// Stack Container in C++

int main() {

stack<int>mystack; // variable declaration

mystack.push(42); // push operation
mystack.push(11);
mystack.push(5);
mystack.push(71);
mystack.push(43);


while(!mystack.empty()){
   cout<<mystack.top()<<" ";
   mystack.pop(); // pop operation
}

}
```

Example 3: Queue in C++. [Queue]

```
#include<bits/stdc++.h>
using namespace std;

// Queue Container in C++

int main ()
{
    queue <int> q; // creates an empty queue of integer q

    q.push(2); // pushes 2 in the queue , now front = back = 2
    q.push(3); // pushes 3 in the queue , now front = 2 , and back = 3
    q.push(8);
    q.push(45);
    q.push(60);
    q.push(80);

    while(!q.empty()){

        cout<<q.front()<<" ";

        q.pop();
    }

}
```

Practice Exercise

1. Take 5 integer values into a stack. Find the summation of all the stack elements.
2. Take 6 integer values (0 to 5) into a stack and then find the factorial of each stack element. Store the outputs in another stack. Print the output in the following way:
Factorial : 0 = 1
Factorial : 1 = 1
Factorial : 2 = 4
—-