Assignment - 41 **A Job Ready Bootcamp in C++, DSA and IOT**

Stack

1. Create a stack of int type, push 5 elements in it and print it on the console screen.

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

int main()
{
    stack<int> s;
    // push 5 element in stack
    for (int i = 1; i <= 5; i++)
        s.push(i);

    // pop 5 element in stack and print
    while (!s.empty())
    {
        cout << s.top() << " ";
        s.pop();
    }

    cout << "\nSize of stack is " <<s.size() << endl;
    return 0;
}
```

```
Output:
5 4 3 2 1
Size of stack is 0
```

2. Create a stack of int type, and find the top most element in a stack.

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

int main()
{
    stack<int> s;
    int n;
    cout << "Enter 10 elements: ";
    for (int i = 1; i <= 10; i++)
    {
        cin >> n;
        s.push(n);
    }

    cout << "The top most element in stack is " << s.top() << endl;

    return 0;
}
```

```
Output:
Enter 10 elements: 10 50 60 87 32 214 256 87 65 512
The top most element in stack is 512
```

3. Create a stack, and implement main operations like push(), pop(), peek(), empty() and size().

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

void dispStack(stack<int> S)
{
```

```
    while (!S.empty())
    {
        cout << S.top() << " ";
        S.pop();
    }
    cout << endl;
}
int peek(stack<int> S)
{
    return (S.top());
}
int main()
{
    stack<int> s;
    s.push(10);
    s.push(20);
    s.push(30);
    s.push(40);
    s.push(50);
    cout << "Size of stack: " << s.size() << endl;
    cout << "Peek is " << peek(s) << endl;
    dispStack(s);

    return 0;
}
```

```
Output:
Size of stack: 5
Peek is 50
50 40 30 20 10
```

4. Reverse the Words of a String using Stack.
   Example:
   Input: str = “I Love To Code”
   Output: Code To Love I

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

// function to reverse the words
// of the given string without using strtok().
void reverse(string s)
{
    // create an empty string stack
    stack<string> S;

    // create an empty temporary string
    string temp = "";

    // traversing the entire string
    for (int i = 0; i < s.length(); i++)
    {
        if (s[i] == ' ')
        {

            // push the temporary variable into the stack
            S.push(temp);

            // assigning temporary variable as empty
            temp = "";
        }
        else
        {
```

```
            temp = temp + s[i];
        }
    }

    // for the last word of the string
    S.push(temp);

    while (!S.empty())
    {

        // Get the words in reverse order
        temp = S.top();
        cout << temp << " ";
        S.pop();
    }
    cout << endl;
}

int main()
{
    string s = "I Love To Code";
    reverse(s);
    return 0;
}
```

```
Output:
Code To Love I
```

5. Create stack1 of int type, and create another stack of the same type with name stack2 and copy all the elements of stack1 into stack2 in the same order.

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

int main()
{
    stack<int> stack1;
    stack<int> stack2;
    for (int i = 1; i <= 5; i++)
        stack1.push(i * 10);     // 10 20 30 40 50 =====> 50 40 30 20 10
    for (int i = 1; i <= 5; i++)
        stack2.push(i + 1);      // 2 3 4 5 6 ======> 6 5 4 3 2

    stack1.swap(stack2);

    cout << "Stack1: ";
    while (!stack1.empty())
    {
        cout << stack1.top() << " ";
        stack1.pop();
    }

    cout << "\nStack2: ";
    while (!stack2.empty())
    {
        cout << stack2.top() << " ";
        stack2.pop();
    }

    return 0;
}
```

```
Output:
```

```
Stack1: 6 5 4 3 2
Stack2: 50 40 30 20 10
```

6. Reverse a string using a stack.
   Example:
   Input: str = "Reverse me"
   Output: em esreveR

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

void reverse(string &str)
{
    // create an empty stack
    stack<int> s;

    // Push each character in the string to the stack
    for (int i = 0; str[i]; i++)
        s.push(str[i]);

    for (int i = 0; i < str.length(); i++)
    {
        str[i] = s.top();
        s.pop();
    }
}

int main()
{
    string str = "Reverse me";
    reverse(str);
    cout << str;

    return 0;
}
```

```
================================================================================
Output:
em esreveR
```

7. Create a stack of int type and sort it.

```cpp
#include <iostream>
#include <stack>
#include <algorithm>
#include <vector>
using namespace std;

void sortStack(stack<int> s);

int main()
{
    stack <int> S;
    S.push(5);
    S.push(64);
    S.push(48);
    S.push(78);
    S.push(25);
    S.push(10);

    sortStack(S);
```

```
    return 0;
}

void sortStack(stack<int> s)
{
    vector <int> v;
    while (!s.empty())
    {
        v.push_back(s.top());
        s.pop();
    }
    sort(v.begin(), v.end());
    for (int i = 0; i < v.size(); i++)
        cout << v.at(i) << " ";
}
```

```
Output:
5 10 25 48 64 78
```

8. Create a stack of int type and sort it in descending order.

```
#include <iostream>
#include <stack>
#include <algorithm>
#include <vector>
using namespace std;

void sortStack(stack<int> s);

int main()
{
    stack <int> S;
    S.push(5);
    S.push(64);
    S.push(48);
    S.push(78);
    S.push(25);
    S.push(10);

    sortStack(S);

    return 0;
}

void sortStack(stack<int> s)
{
    vector <int> v;
    while (!s.empty())
    {
        v.push_back(s.top());
        s.pop();
    }
    sort(v.begin(), v.end());
    reverse(v.begin(), v.end());
    for (int i = 0; i < v.size(); i++)
        cout << v.at(i) << " ";
}
```

```
Output:
78 64 48 25 10 5
```

9. Create back button functionality using stack.

```
#include <iostream>
#include <stack>
```

```
using namespace std;

int main()
{
main_menu:
    cout << "1. Open menu 1" << endl;
    cout << "2. Open menu 2" << endl;
    cout << "3. Exit 3" << endl;
    cout << "Enter your choice: " << endl;
    int choice;
    cin >> choice;
    switch (choice)
    {
    case 1:
        cout << "Press 1 to back: ";
        cin >> choice;
        if (choice == 1)
        {
            goto main_menu;
        }
        break;

    case 2:
        cout << "Press 1 to back: ";
        cin >> choice;
        if (choice == 1)
        {
            goto main_menu;
        }
        break;
    case 3:
        exit(0);
    default:
        cout << "Wront input" << endl;
    }
}
```

```
================================================================================
Output:
1. Open menu 1
2. Open menu 2
3. Exit 3
Enter your choice:
1
Press 1 to back: 1
1. Open menu 1
2. Open menu 2
3. Exit 3
Enter your choice:
2
Press 1 to back: 1
1. Open menu 1
2. Open menu 2
3. Exit 3
Enter your choice:
3
```

10. Given an array, print the Next Greater Element (NGE) for every element using stack.
    Example:
    Input: arr[] = [4 , 5 , 2 , 25]
    Output: 4 –> 5
    5 –> 25
    2 –> 25

25 –> -1

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

void printNGE(int arr[], int n)
{
    stack<int> s;

    /* push the first element to stack */
    s.push(arr[0]);

    // iterate for rest of the elements
    for (int i = 1; i < n; i++)
    {

        if (s.empty())
        {
            s.push(arr[i]);
            continue;
        }

        while (s.empty() == false && s.top() < arr[i])
        {
            cout << s.top() << " --> " << arr[i] << endl;
            s.pop();
        }

        /* push next to stack so that we can find
        next greater for it */
        s.push(arr[i]);
    }

    /* After iterating over the loop, the remaining
    elements in stack do not have the next greater
    element, so print -1 for them */
    while (s.empty() == false)
    {
        cout << s.top() << " --> " << -1 << endl;
        s.pop();
    }
}

/* Driver code */
int main()
{
    int arr[] = {4, 5, 2, 25};
    int n = sizeof(arr) / sizeof(arr[0]);
    printNGE(arr, n);
    return 0;
}
```

==========

```
Output:
4 --> 5
2 --> 25
5 --> 25
25 --> -1
```