IT DS 201 LAB

SUBMITTED BY ADITYA SINGH 2K19/EP/005

Program 6: Write a program to implement two stacks using a single array.

CODE

```
#include<iostream>
#include<stdlib.h>
using namespace std;

class twoStacks {
    int *arr; int size; int top1, top2;
public:
   twoStacks(int n){
       size = n;
       arr = new int[n];
       top1 = -1; top2 = size;
   }
   void push1(int x) {
       if (top1 < top2 - 1) {
           top1++;
           arr[top1] = x;
       }
       else{
           cout << "Stack Overflow";
           exit(1);
       }
   }
   void push2(int x) {
       if (top1 < top2 - 1) {
           top2--;
           arr[top2] = x;
       }
       else{
           cout << "Stack Overflow";
           exit(1);
       }
   }
```

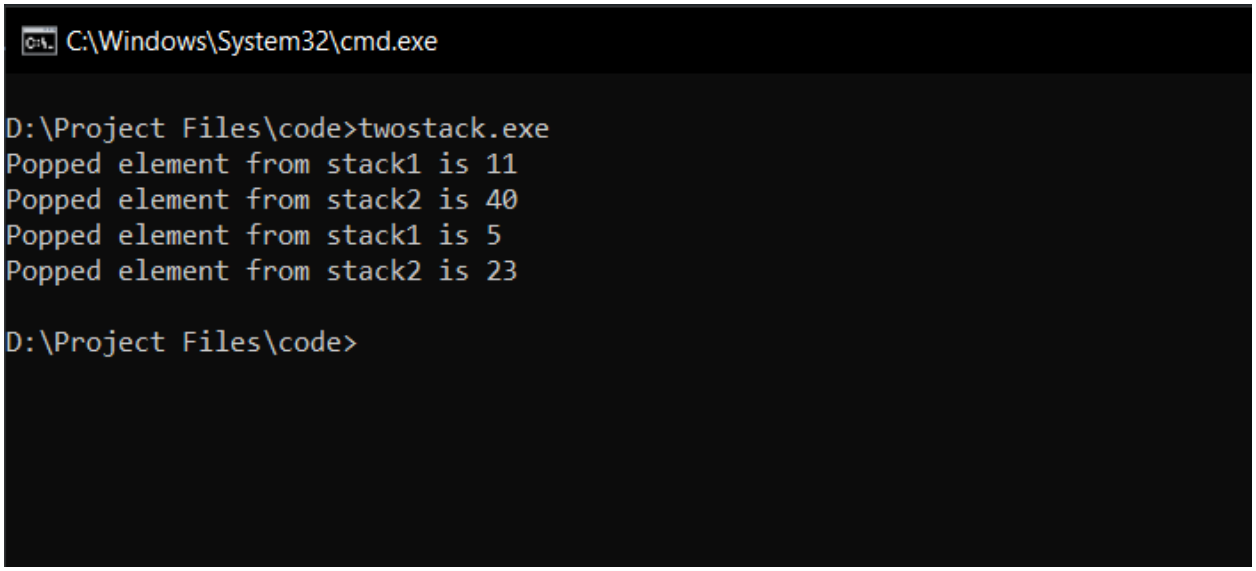
```
    int pop1() {
        if (top1 >= 0 ) {
           int x = arr[top1];
           top1--;
           return x;
        }
        else{
            cout << "Stack UnderFlow";
            exit(1);
        }
    }
    int pop2() {
        if (top2 < size) {
           int x = arr[top2];
           top2++;
           return x;
        }
        else{
            cout << "Stack UnderFlow";
            exit(1);
        }
    }
};
```

ALGORITHM

1. This method efficiently utilizes the available space. It doesn't cause an overflow if there is space available in arr[].
2. The idea is to start two stacks from two extreme corners of arr[]. stack1 starts from the leftmost element, the first element in stack1 is pushed at index 0.
3. The stack2 starts from the rightmost corner, the first element in stack2 is pushed at index (n-1).
4. Both stacks grow (or shrink) in opposite directions.
5. To check for overflow, all we need to check is for space between top elements of both stacks.

INPUT/OUTPUT

```
int main()
{
    twoStacks ts(5);
    ts.push1(5);
    ts.push2(10);
    ts.push2(15);
    ts.push1(11);
    ts.push2(7);
    cout << "Popped element from stack1 is " << ts.pop1() << endl;
    ts.push2(40);
    cout << "Popped element from stack2 is " << ts.pop2() << endl;
    cout << "Popped element from stack1 is " << ts.pop1() << endl;
    ts.push2(23);
    cout << "Popped element from stack2 is " << ts.pop2() << endl;
    return 0;
}
```



END