Department of
Computer Science and Engineering

Title: Implement Stack using Arrays

Data Structure Lab

CSE 106

Green University of Bangladesh

1 Objective(s)

- To attain knowledge on the Stack data structure and how it works.
- To implement Stack using Array.

2 Problem analysis

Stack is a linear data structure in which the insertion and deletion operations are performed at only one end. In a stack, adding and removing of elements are performed at a single position which is known as "top". That means, a new element is added at top of the stack and an element is removed from the top of the stack. In stack, the insertion and deletion operations are performed based on LIFO (Last In First Out) principle. In a stack, the insertion operation is performed using a function called "push" and deletion operation is performed using a function called "pop".

Operation	Picture	Execution
Stack is empty	3 2 1 0 ← sp = 0 st	
`push('A');`	3 2 1 0 A ← sp = 1 st	`st[0] = 'A'; sp = 0 + 1;`
`push('B');`	3 2 1 B 0 A ← sp = 2 st	`st[1] = 'B'; sp = 1 + 1;`
`push('C');`	3 ← sp = 3 2 C 1 B 0 A st	`st[2] = 'C'; sp = 2 + 1;`

(a) The push operation illustrated

Operation	Picture	Execution
`data = pop();`	3 2 C ← sp = 2 1 B 0 A st	`sp = 3 - 1; return sp[2];`
`data = pop();`	3 2 C 1 B ← sp = 1 0 A st	`sp = 2 - 1; return sp[2];`
`data = pop();`	3 2 C 1 B 0 A ← sp = 0 st	`sp = 1 - 1; return sp[2];`

(b) The pop operation illustrated

Figure 1: Stack operations

3 Algorithm

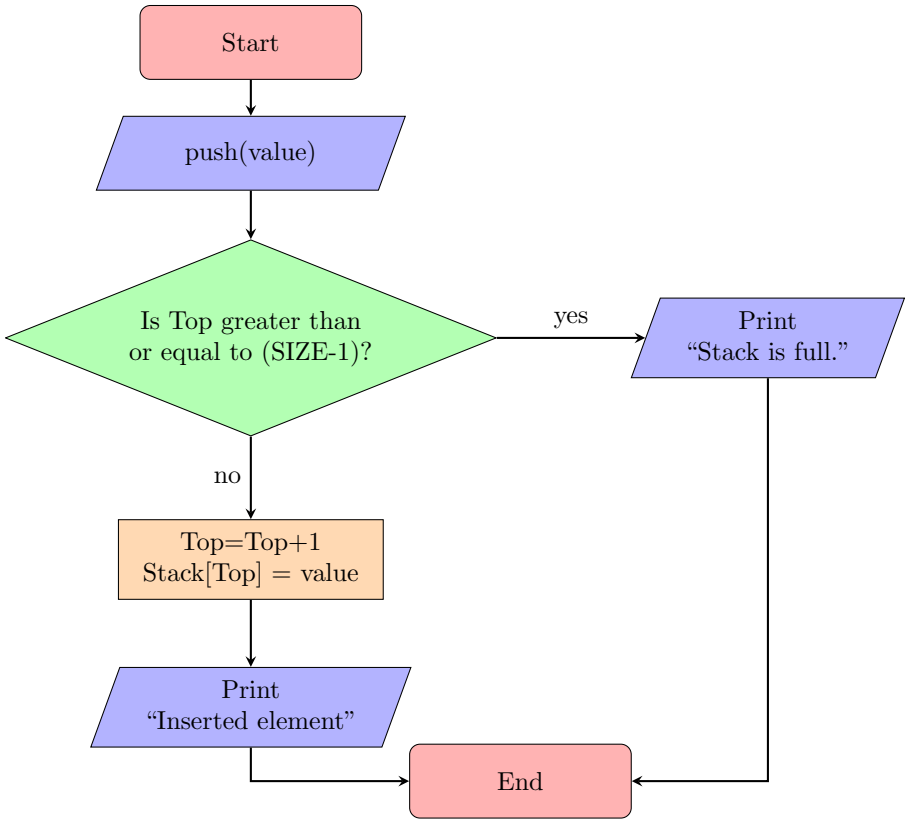
Algorithm 1: Push Operation of Stack

Input: Element

```
/* Algorithm for push operation                                    */
```

```
1 if Top >= SIZE-1 then
2 |   Print "Stack is full"
3 |   return
4 end
5 else
6 |   Top=Top+1
7 |   Stack[Top] = Element
8 |   print "Inserted element"
9 end
```

3.1 Flowchart

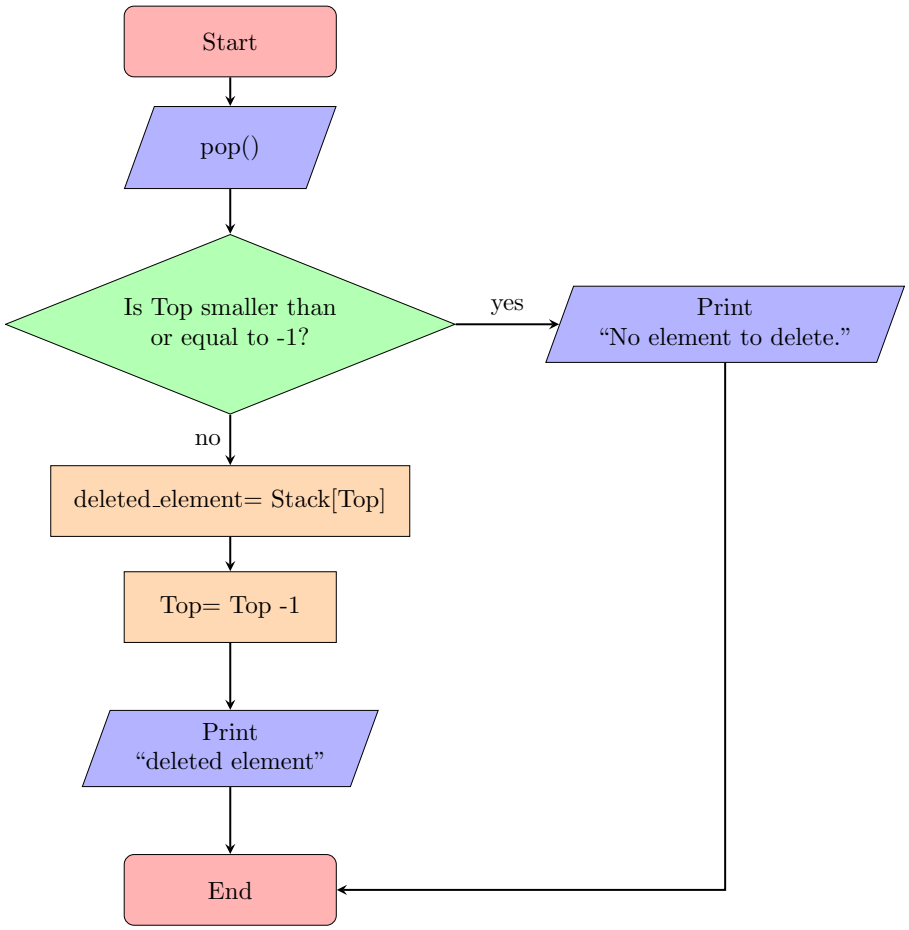


Algorithm 2: Pop Operation of Stack

Output: Element

```
/* Algorithm for pop operation                                    */
1 if Top <= -1 then
2 |   Print "No element to delete"
3 |   return
4 end
5 else
6 |   Set Del_element = Stack[Top]
7 |   Top = Top-1
8 |   return Del_Element
9 end
```

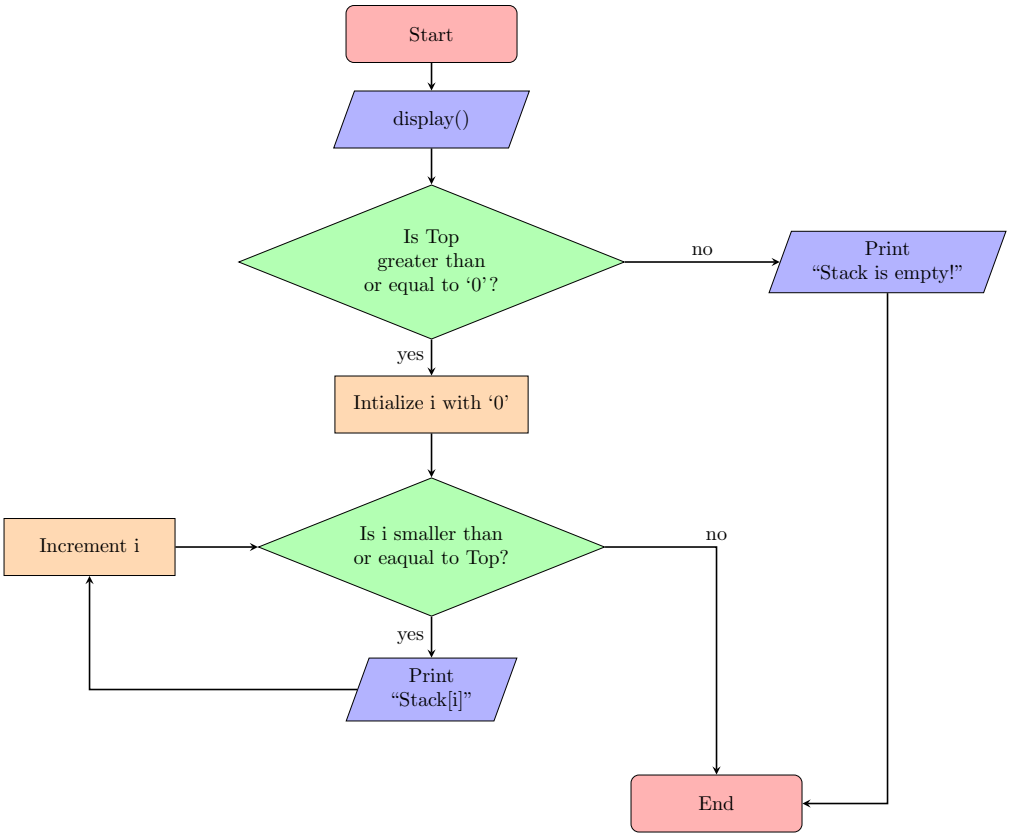
3.2 Flowchart



Algorithm 3: Display Operation of Stack

```
/* Algorithm for display operation                                    */
if Top >= 0 then
    for i = 0 to Top do
        print "stack[i]"
    end
end
else
    print "stack is empty"
    return
end
```

3.3 Flowchart



4 Implementation in C

```c
#include <stdio.h>
#include<stdlib.h>
#define size 5
int stack[size],top=-1;

void push(int val){
    if(top>=size-1)
        printf("Stack is full.\n");
    else{
        top++;
```

```
11          stack[top]=val;
12          printf("Inserted element: %d\n",stack[top]);
13      }
14  }
15
16  void pop(){
17      if(top<=-1)
18      {
19          printf("No element to delete.\n");
20      }
21      else
22      {
23          printf("The popped element = %d\n",stack[top]);
24          top--;
25      }
26  }
27
28  void show(){
29      if(top>=0){
30          printf("Stack Elements are:\n");
31      for(int i=0;i<=top;i++)
32              printf("%d ",stack[i]);
33    }
34    else
35    {
36        printf("Stack is Empty\n");
37      }
38  }
39
40  int main(int argc, char const *argv[]) {
41      int ch,val;
42      do{
43
44      printf("\nMenu");
45      printf("\n1. PUSH");
46      printf("\n2. POP ");
47      printf("\n3. SHOW STACK");
48      printf("\n4. Exit");
49      printf("\nEnter your choice 1 to 4=");
50      scanf("%d",&ch);
51      switch (ch)
52      {
53          case 1:
54                  printf("Enter the value to be pushed=");
55                  scanf("%d",&val);
56                  push(val);
57                  break;
58          case 2:
59                  pop();
60                  break;
61
62          case 3:
63                  show();
64                  break;
65          case 4:
66                  exit(0);
67                  break;
68
```

```
69          default:
70                  printf("Invalid choice!");
71                  break;
72      }
73
74      }while (ch<=3);
75      return 0;
76  }
```

5 Input/Output (Compilation,Debugging & Testing)

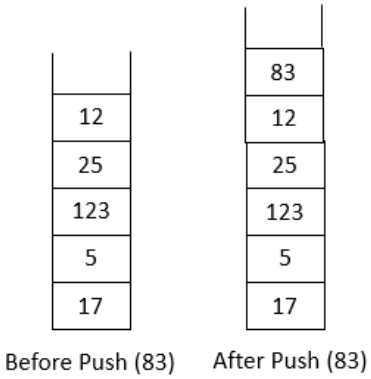
Input	Output
Choice: 3(initially)	Stack is Empty.
Choice: 2(initially)	No element to delete.
Choice: 1 Enter the value to be pushed: 1	Inserted element: 1
Choice: 1 Enter the value to be pushed: 5	Inserted element: 5
Choice: 1 Enter the value to be pushed: 13	Inserted element: 13
Choice: 3	Stack elements are: 1 5 13
Choice: 1 Enter the value to be pushed: 8	Inserted element: 8
Choice: 1 Enter the value to be pushed: 20	Inserted element: 20
Choice: 3	Stack elements are: 1 5 13 8 20
Choice: 1 Enter the value to be pushed: 35	Stack is full
Choice: 3	Stack elements are:1 5 13 8 20
Choice: 2	The popped element: 20
Choice: 3	Stack elements are: 1 5 13 8
Choice: 2	The popped element: 8
Choice: 2	The popped element: 13
Choice: 3	Stack elements are:1 5
Choice: 2	The popped element: 5
Choice: 3	Stack elements are:1
Choice: 2	The popped element: 1
Choice: 3	Stack is empty
Choice: 2	No element to delete.
Choice: 4	Press any key to continue.
Choice: 5	Invalid choice.Press any key to continue.

6 Discussion & Conclusion

Based on the focused objective(s) to understand about the stack operations, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

7 Lab Task (Please implement yourself and show the output to the instructor)

1. Implement the following STACK shown in the following figure and perform the operations.



2. Delete the item 123 from the STACK and show the result. Also Find the TOP value from the STACK.
3. Implement the STACK for both numeric and character items.
4. Write appropriate code for controlling OVERFLOW and UNDERFLOW.

8 Lab Exercise (Submit as a report)

- Consider the following arithmetic infix expression:

$$A + (B * C - (D/E \uparrow F) * G) * H$$

  Write a program to transform it into equivalent postfix expression using stack.

9 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.