STACK

Terminology: Push ,Pop, display ,Top

content

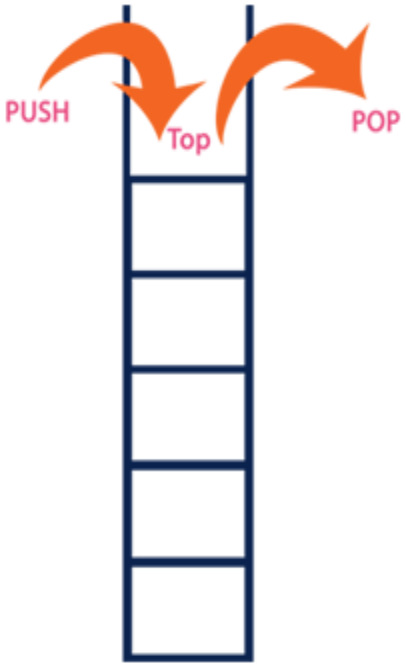
- What is Stack?
- Stack working principle

What is a Stack?

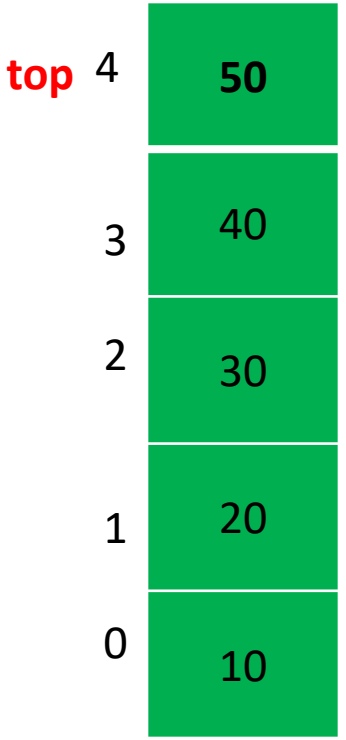
- **Stack** is a **linear data structure** in which the **insertion** and **deletion** operations are performed **at only one end**.
- In a stack, **adding** and **removing** of elements are performed **at a single position** which is known as **"top".**
- That means, a **new element** is **added at top** of the **stack** and an element is **removed from** the **top of the stack**.

Principle

LIFO(**L**ast **I**n **F**irst **O**ut)



Stack Examples



CD-BOX

IDLY PLATES

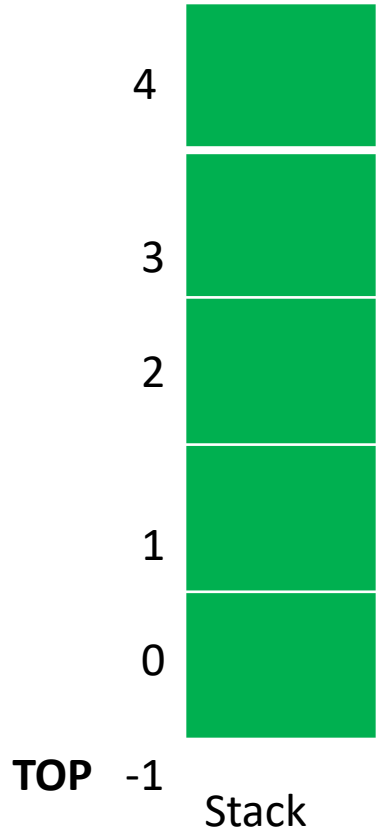
Operations on a Stack

- **Push (To insert an element on to the stack)**
- **Pop (To delete an element from the stack)**
- **Display (To display elements of the stack)**

Push (To insert an element on to the stack)

Initial Top=-1

SIZE=5



Push (To insert an element on to the stack) Algorithm

- **Step 1 -** Check whether **stack** is **FULL.**

  (top == SIZE-1)

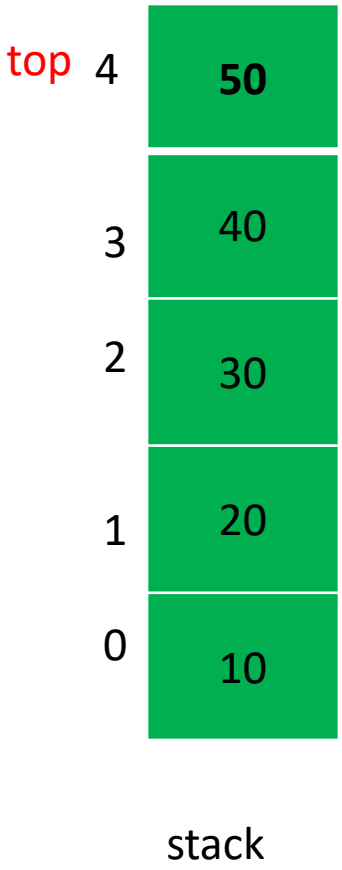
- **Step 2 -** If it is **FULL**, then display

  "Stack is FULL!!! Insertion is not possible!!!" and terminate the function.

Push()

Push(60)

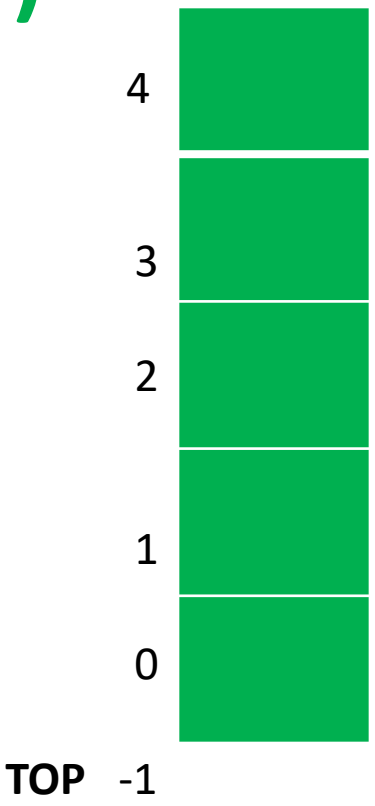
TOP=4 ,i.e. Stack is FULL



Push()

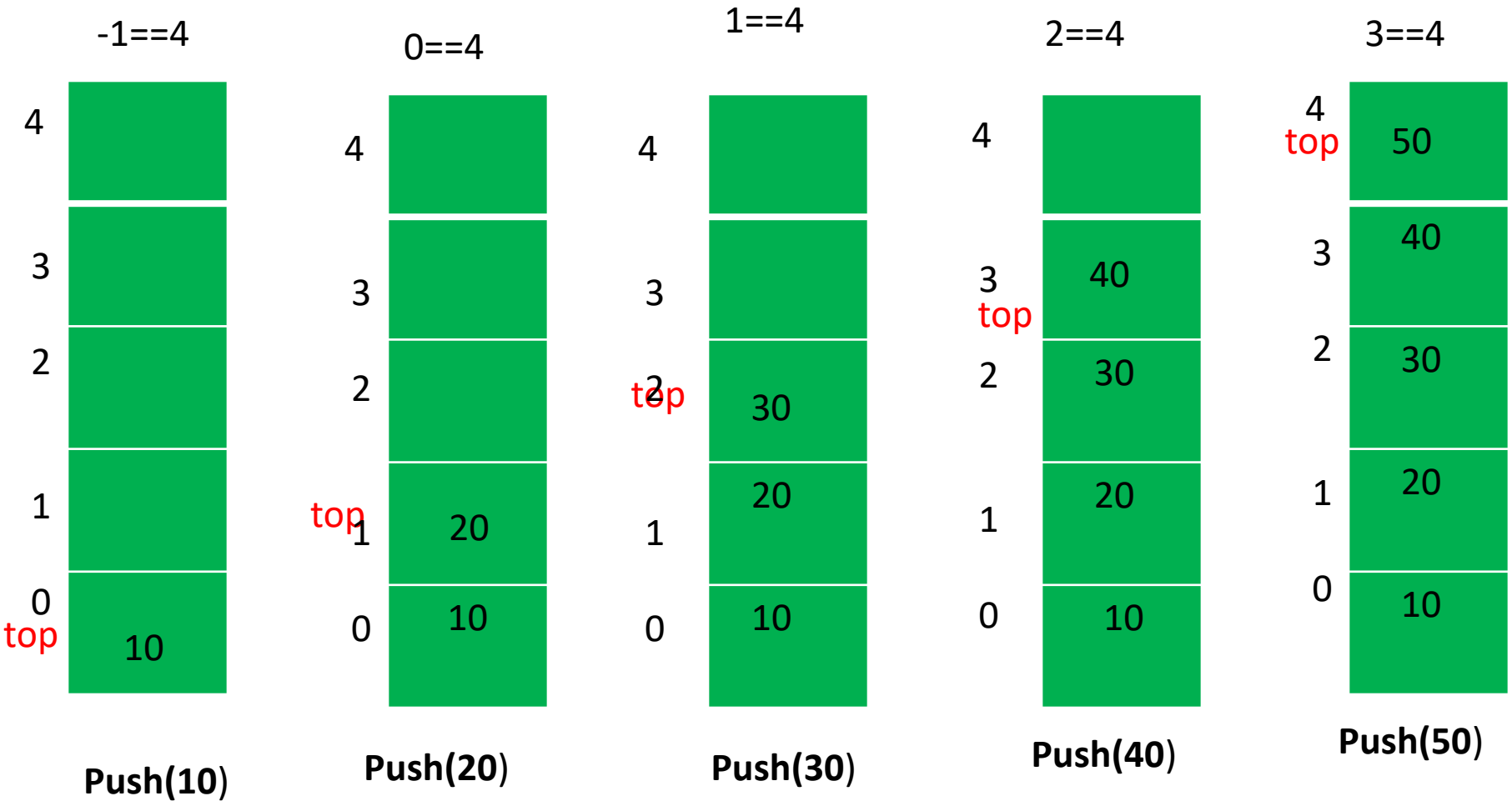
- **Step 3 -** If it is **NOT FULL**, then increment **top** value by one **(top++)** and set stack[top] to value **(stack[top] = value).**

❖TOP=-1 ,i.e. Stack is empty



Push (To insert an element on to the stack)

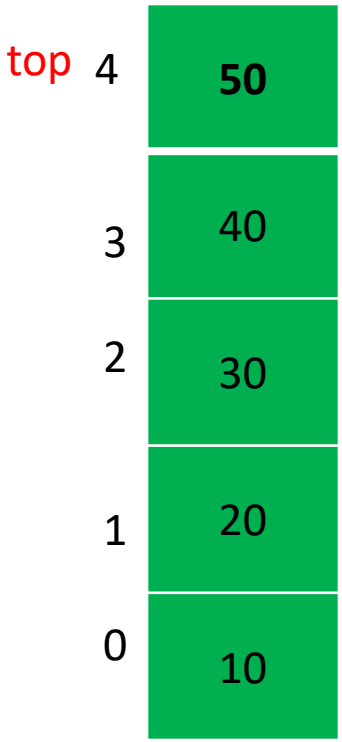
If (top == SIZE-1)



Push()

Push(60) **If (top == SIZE-1)**

TOP=4 ,i.e. Stack is FULL Insertion not possible



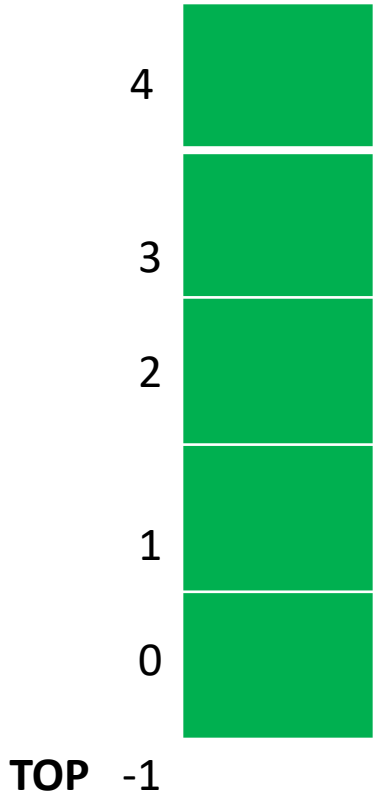
Pop (To delete an element from the stack) Algorithm

- **Step 1 -** Check whether **stack** is **EMPTY.**

  (top == -1)
- **Step 2 -** If it is **EMPTY**, then display **"Stack is EMPTY!!! Deletion is not possible!!!"** and terminate the function.

Pop()

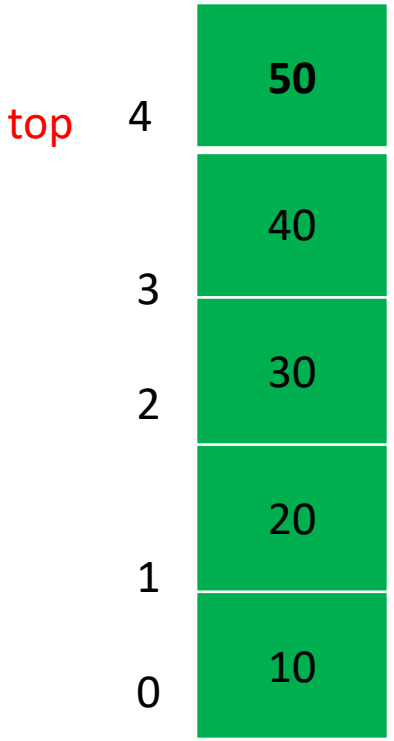
TOP=-1 ,i.e. Stack is empty



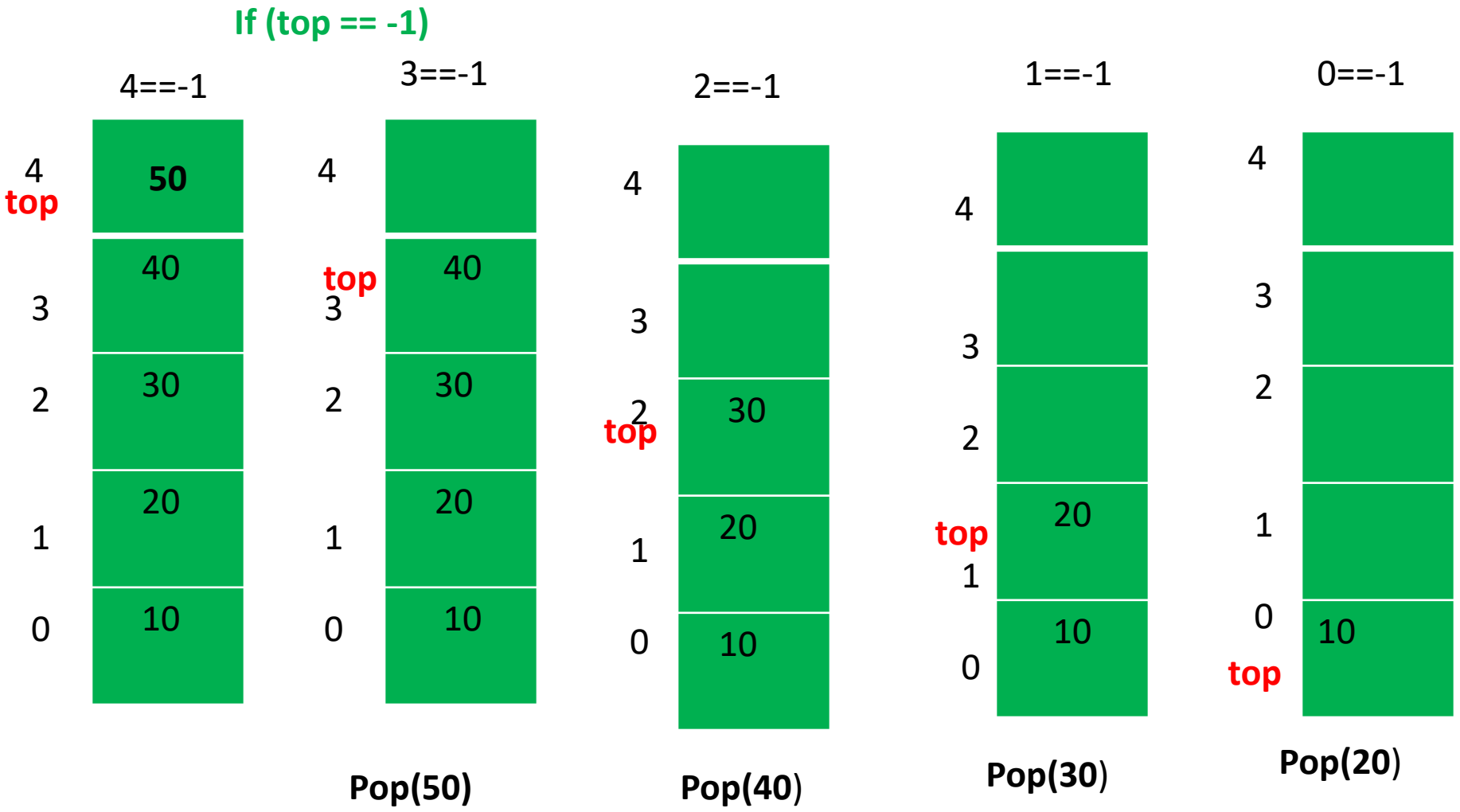
Pop()

TOP=4 ,i.e. Stack is not empty

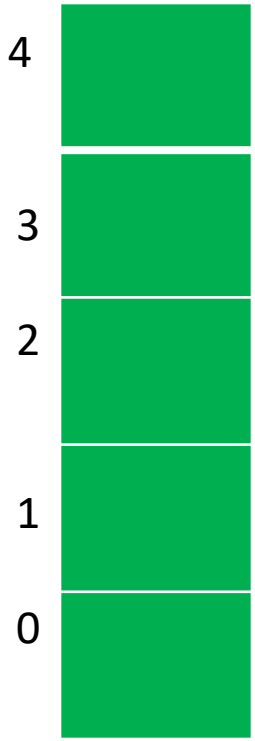
Step 3 - If it is **NOT EMPTY**, then
delete **stack[top]** and
decrement **top** value by one **(top--).**



Pop (To delete an element from the stack)

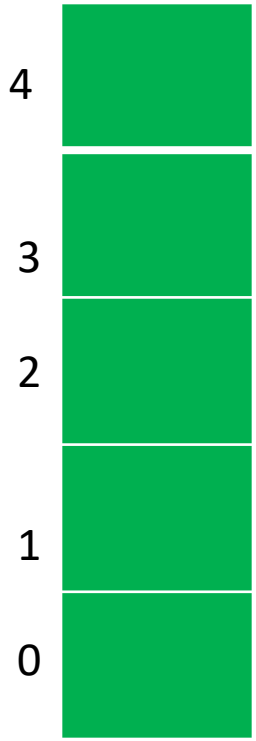


Pop (To delete an element from the stack)



Top=-1

Pop(10)



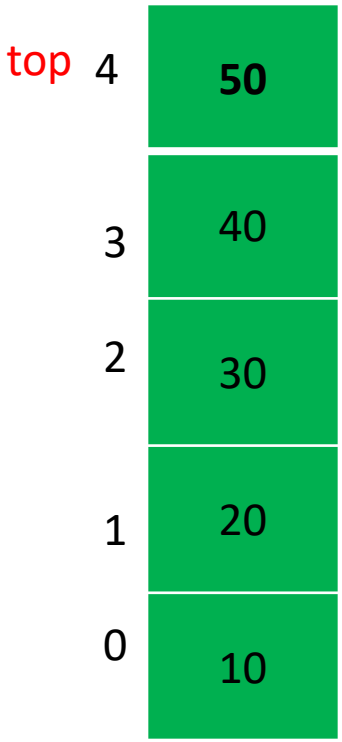
TOP -1

Stack is empty

Display (To display elements of the stack)

- **Step 1 -** Check whether **stack** is **EMPTY. (top == -1)**
- **Step 2 -** If it is **EMPTY**, then display **"Stack is EMPTY!!!"** and terminate the function.
- **Step 3 -** If it is **NOT EMPTY**, then define a variable **'i'** and initialize with **top**. Display **stack[i]** value and decrement **i** value by one **(i--).**
- **Step 4 -** Repeat above step until **i** value becomes **'0'**.

Display()

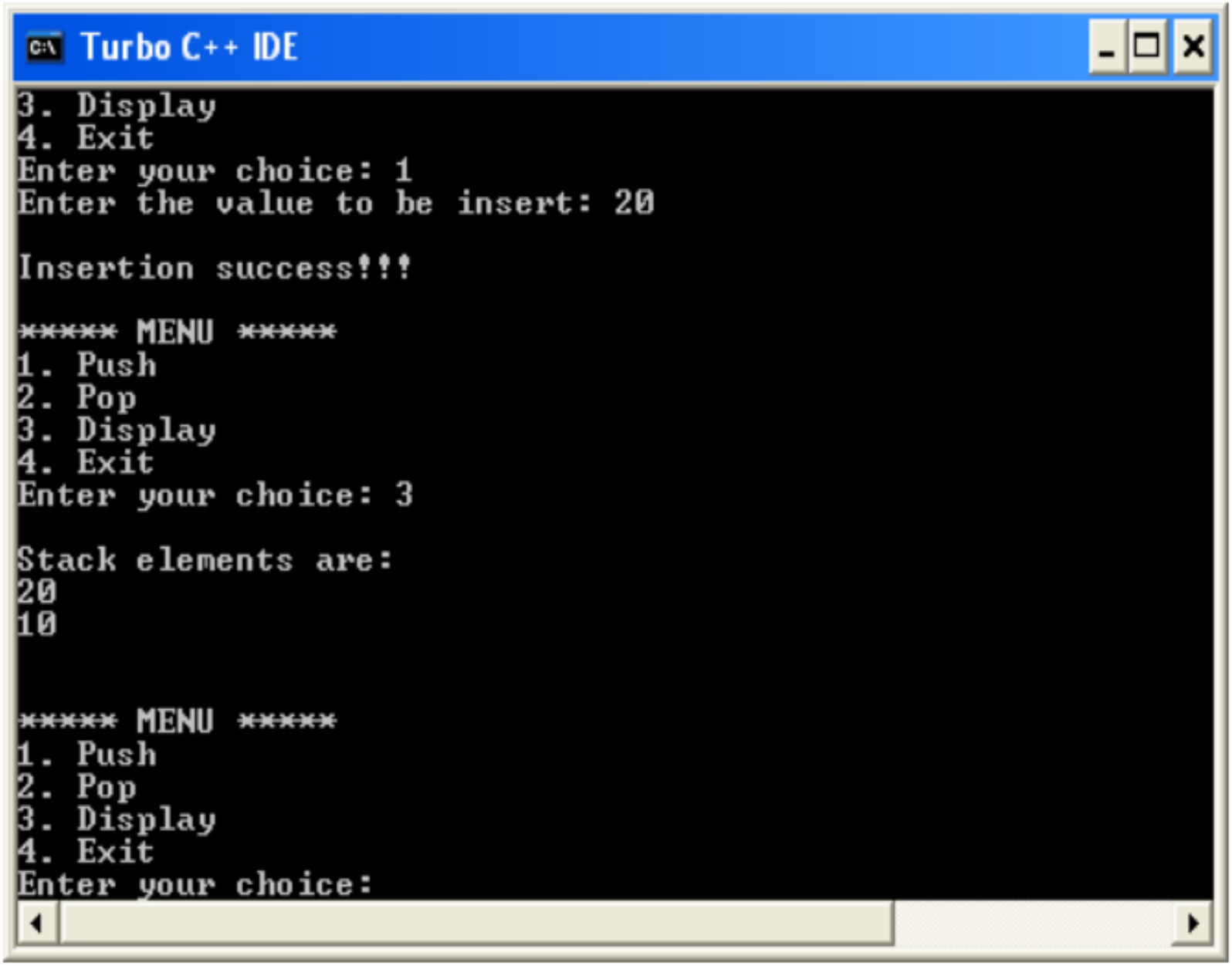


mplementation of Stack using Array

```
#include<stdio.h>
#include<conio.h>

#define SIZE 10

void push(int);
void pop();
void display();

int stack[SIZE], top = -1;

void main()
{
   int value, choice;
   clrscr();
   while(1){
      printf("\n\n**** MENU ****\n");
      printf("1. Push\n2. Pop\n3. Display\n4. Exit");
      printf("\nEnter your choice: ");
      scanf("%d",&choice);
      switch(choice){
         case 1: printf("Enter the value to be insert: ");
                 scanf("%d",&value);
                 push(value);
                 break;
         case 2: pop();
                 break;
         case 3: display();
                 break;
         case 4: exit(0);
         default: printf("\nWrong selection!!! Try again!!!");
      }
   }
```

```
void push(int value){
   if(top == SIZE-1)
      printf("\nStack is Full!!! Insertion is not possible!!!");
   else{
      top++;
      stack[top] = value;
      printf("\nInsertion success!!!");
   }
}
void pop(){
   if(top == -1)
      printf("\nStack is Empty!!! Deletion is not possible!!!");
   else{
      printf("\nDeleted : %d", stack[top]);
      top--;
   }
}
void display(){
   if(top == -1)
      printf("\nStack is Empty!!!");
   else{
      int i;
      printf("\nStack elements are:\n");
      for(i=top; i>=0; i--)
         printf("%d\n",stack[i]);
   }
}
```

Output


Turbo C++ IDE
3. Display
4. Exit
Enter your choice: 1
Enter the value to be insert: 20
Insertion success!!!
***** MENU *****
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Stack elements are:
20
10
***** MENU *****
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: