# Practical-4

**1. Implement a stack using an array having following functionalities:**

**a. isEmpty – to check if the stack if empty or not**

**b. isFull – to check if the stack if full or not**

**c. push – to insert the element into the stack**

**d. pop – to delete an element from the stack**

**e. print_top – to print the top most element of the stack.**

Aim:

To implement the concept of stack.

Theory:

In this practical we implemented the concept of pointers to structure and using it we implemented stack.

Code:

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

struct Stack
{
 int top,size;
 int *array;
};

struct Stack* createStack(int size)
{
 struct Stack *s=(struct Stack *) malloc(sizeof(struct Stack));
 s->size=size;
 s->top=-1;
 s->array=(int *)malloc(sizeof(int)*size);
```

```
  return s;
}

int isEmpty(struct Stack *s)
{
  if(s->top==-1)
  {
    printf("The stack is Empty\n");
    return 1;
  }
  else
  {
    printf("The stack is not empty\n");
    return 0;
  }
}

int isFull(struct Stack *s)
{
  if(s->top==s->size-1)
  {
    printf("The stack is full\n");
    return 1;
  }
  else
  {
    printf("The stack is not full\n");
    return 0;
  }
```

```
}

int push(struct Stack *s)
{
 int item1;
 int g=isFull(s);
 if(g==0)
 {
  printf("Enter the value that is to be pushed=");
  scanf("%d",&item1);
  s->top=s->top+1;
  s->array[s->top]=item1;
  printf("Pushed the element to stack\n");
  return 0;
 }
 return 0;
}

int pop(struct Stack *s)
{
 int p=isEmpty(s);
 if(p==0)
 {
  s->top=s->top-1;
  printf("The element was poped\n");
  return 0;
 }
 return 0;
}
```

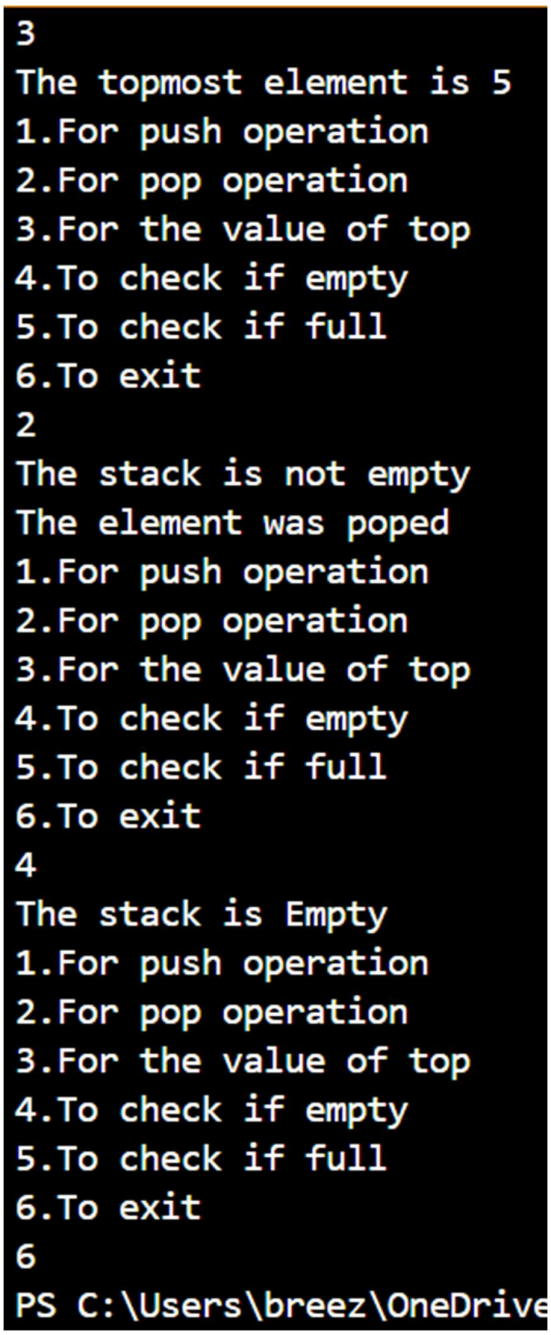
```
void print_top(struct Stack *s)
{
  printf("The topmost element is %d\n",s->array[s->top]);
}

int main()
{
  int size;
  printf("Enter the size value = ");
  scanf("%d",&size);
  struct Stack *s=createStack(size);
  printf("Enter a value for the following operation:\n");
  int x;
  while(1)
  {
    printf("1.For push operation \n2.For pop operation \n3.For the value of top \n4.To check if empty \n5.To check if full \n6.To exit\n");
    scanf("%d",&x);
    switch(x)
    {
      case 1:
        push(s);
        break;
      case 2:
        pop(s);
        break;
      case 3:
        print_top(s);
```

```
      break;
    case 4:
      isEmpty(s);
      break;
    case 5:
      isFull(s);
      break;
    case 6:
      exit(0);
   }
 }
}
```

Output:

```
PS C:\Users\breez\OneDrive - pdpu.ac.in\PDE
ab\Practise-4\" ; if ($?) { gcc Problem1.c
Enter the size value = 2
Enter a value for the following operation:
1.For push operation
2.For pop operation
3.For the value of top
4.To check if empty
5.To check if full
6.To exit
1
The stack is not full
Enter the value that is to be pushed=5
Pushed the element to stack
1.For push operation
2.For pop operation
3.For the value of top
4.To check if empty
5.To check if full
6.To exit
3
The topmost element is 5
1.For push operation
2.For pop operation
3.For the value of top
4.To check if empty
```




Link to all the code:

https://github.com/PanavPatel06/DSA-Lab/tree/main/Practise-4