WEEK-8

Date:03-09-2025

List of programs:

1. Write a C program to implement Stack operations using arrays.
2. Write a C program to implement Stack operations using linked list.

1. Aim: To write a C program to implement Stack operations using arrays.

Program:

```
#include <stdio.h>
#include <stdlib.h>
#define MAXSIZE 5
void push(int x);
void pop();
void display();
int stack[MAXSIZE];
int top=-1;
void main( )
{
  int ch,num;
  while(1)
  {
    printf("\n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n\nEnter your choice:");
    scanf("%d",&ch);
     switch(ch)
     {
       case 1: printf("\n\nENTER THE STACK ELEMENT :  ");

            scanf("%d",&num);
           push(num);
```

```
                break;
         case 2:  pop();
                  break;
         case 3:  display();
                  break;
        case 4:  exit(0);
                 break;
        default:printf("Invalid Choice :");
        }
    }
}
void push(int item)
{
  if(top==MAXSIZE-1)
  {
                printf("\nSTACK FULL");
                return;
        }
        else
        {
                    top++;
                    stack[top]=item;
                    printf("ELEMENT INSERTED\n");
        }
 }
void pop()
{
   int x;
   if(top==1)
   {
      printf("STACK EMPTY");
       return;
      }
```

```
        else
        {
          x=stack[top];
          printf("DELETED ELEMENT is %d\n",x);
          top--;
        }
}
void display()
{
        int i;
        if(top==-1)
        {
                printf("\nSTACK EMPTY");
                return;
        }
        else
        {
                printf("\nSTACK ELEMENTS ARE...\n");
                for(i=top;i>=0;i--)
                {
                        printf("\n%d ",stack[i]);
                        if(i==top)
                        printf("---->TOP");
                }
        }
    }
```

Output:

```
1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:1


ENTER THE STACK ELEMENT :  10
ELEMENT INSERTED

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:1


ENTER THE STACK ELEMENT :  20
ELEMENT INSERTED

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:1


ENTER THE STACK ELEMENT :  30
ELEMENT INSERTED

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:3
```

```
STACK ELEMENTS ARE...

30 ---->TOP
20
10
1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:2
DELETED ELEMENT is 30

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:3

STACK ELEMENTS ARE...

20 ---->TOP
10
```

```
1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice:4


=== Code Execution Successful ===
```

2. **Aim:** To Write a C program to implement Stack operations using linked list.

Program:

```
#include<stdio.h>
 #include<stdlib.h>
struct node
{
int data;
struct node *next;
};
struct node *top=NULL;
void push(int);
void pop();
void display();
void main()
{
 int ch,num;
 printf("\n:: Stack using Linked List ::\n");
 while(1)
 {
   printf(" \n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n\nEnter Your Choice:");
   scanf("%d",&ch);
   switch(ch)
   {
     case 1: printf("\n\nENTER THE STACK ELEMENT : ");
             scanf("%d",&num);
             push(num);
             break;
```

```
        case 2:  pop();
                 break;
        case 3:  display();
                 break;
        case 4:  exit(0);
                 break;
    default: printf("Invalid Choice : ");
         }
      }
    }
    void display()
    {
       struct node *temp = top;
        if(temp==NULL)
           printf("\nSTACK IS EMPTY\n");
         else
          {
            while(temp!=NULL)
              {
                printf("%d-->",temp->data);
                temp=temp->next;
              }
                printf("NULL");
        }
    }
    void push(int num)
    {
      struct node *newNode = (struct node*)malloc(sizeof(struct node));
```

```
 newNode->data = num;
 newNode->next = NULL;
 if(top == NULL)
        top=newNode;
else
 {
   newNode->next = top;
   top = newNode;
  }
  printf("\nELEMENT IS INSERTED\n");
}
void pop()
{
  if(top == NULL)
      printf("\nSTACK IS EMPTY\n");
  else
  {
    temp = top;
    printf("\nDELETED ELEMENT IS %d\n",temp->data);
    top = temp->next;
    free(temp);
  }
}
```

Output:

```
:: Stack using Linked List ::

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:1
ENTER THE STACK ELEMENT : 100

ELEMENT IS INSERTED

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:1
ENTER THE STACK ELEMENT : 200

ELEMENT IS INSERTED

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:1
ENTER THE STACK ELEMENT : 300

ELEMENT IS INSERTED

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:3
300-->200-->100-->NULL
```

```
1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:2

DELETED ELEMENT IS 300

1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:3
200-->100-->NULL
1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter Your Choice:4


=== Code Execution Successful ===
```

Inferences:

- A stack can be implemented using an array where elements are added/removed from one end (top).
- Advantages: Simple, fixed-size memory allocation.
- Disadvantages: Fixed size can lead to overflow; resizing can be costly.
- Stacks using Linked Lists Implementation: Nodes are dynamically allocated; top points to head node.