NAME: THARUN RAJ I
ROLL NO: 230701362

EX NO: 04
PROGRAM NAME: IMPLEMENTATION OF STACK USING ARRAY AND LINKED LIST IMPLEMENTATION

CODE 1: USING ARRAY STACK

```
#include <stdio.h>
#include<string.h>
# define SIZE 5
int top=-1,s[SIZE];
int isempty(){
   if(top==-1)
      return(1);
   else
      return(0);
}
int isfull(){
   if(top==SIZE-1)
      return(1);
   else
      return(0);
}
void push(int elt){
   if(isfull()){
```

```
        printf("stack overflow");
    }
    else{
    top++;
    s[top]=elt;}
}
void pop()
{
    if(isempty())
        printf("stact underflow");
    else{
        printf("%d\n",s[top]);
        top--;
    }
}
void disptop(){
    if(isempty())
        printf("stack empty no elts\n");
    else{
        pop();
    }
}
void disp(){
    if(isempty())
```

```
        printf("stack empty no elts\n");
    else{
        int f=0;
        while(f<=top){
            printf("%d ",s[f]);
            f++;
        }
    }
    printf("\n");
}
int main(){
    push(3);
    push(6);
    push(9);
    push(11);
    push(0);
    pop();
    disp();
    disptop();

}
```

OUTPUT 1:

0

3 6 9 11

11

Process returned 0 (0x0)   execution time : 1.156 s

Press any key

CODE 2:LINKED LIST STACK

```
#include<stdio.h>
#include<stdlib.h>
struct node{
   int data;
   struct node* link;
}*ptr,*first,*last;
int isEmpty()
{if(first==NULL)
   return(1);
  else
   return(0);

}
void push(int elt)
   {
  struct node* new=(struct node*)malloc(sizeof(struct node));
  new->data=elt;
  new->link=NULL;
```

```
  if(isEmpty()){
     first=new;
     last=new;
  }
  else{
    last->link=new;
    last=new;
  }
}
void pop(){
   struct node*prev;
   ptr=first;
   if(isEmpty()){
      printf("\nNo elements to pop\n");
   }
   else{
   while(ptr->link!=NULL){
      prev=ptr;
      ptr=ptr->link;
   }
   prev->link=NULL;
   printf("%d ",ptr->data);
   last=prev;
   if(ptr==first){
```

```
        first=NULL;
        last==NULL;
    }
    free(ptr);}
}
void disptop(){
    if(isEmpty())
        printf("List is empty\n");
    else
    printf("\n%d is the top elt\n",last->data);

}
void disp(){

    ptr=first;
    while(ptr!=NULL)
    {
        printf("%d ",ptr->data);
        ptr=ptr->link;
    }
    printf("\n");
}
int main()
{
```

```
    push(2);
    push(3);
    push(4);
    push(5);
    disp();
    pop();
    disptop();
}
```

OUTPUT 2:

```
2 3 4 5
5
4 is the top elt

Process returned 0 (0x0)   execution time : 0.687 s
Press any key to continue.
```