13/11/2020

**Experiment No: 13**

**STACK USING LINKED LIST**

**AIM:**

Write a program to implement Stack using Linked List.

**DATA STRUCTURES USED:**

Linked List

**ALGORITHM:**

Algorithm POP (ITEM)

1. new = GetNode(Node)
2. If (new = NULL) then
3. Print ”memory underflow”
4. Exit
5. Else
6. new->LINK=TOP->LINK
7. new-> DATA=ITEM
8. TOP->LINK=new
9. Endif
10. Stop

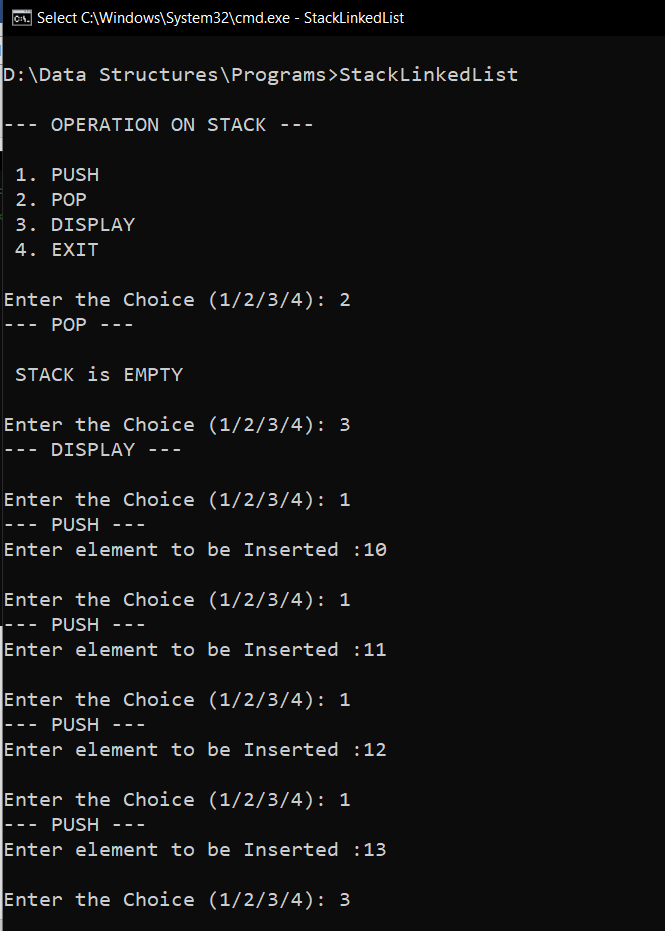
Algorithm POP()

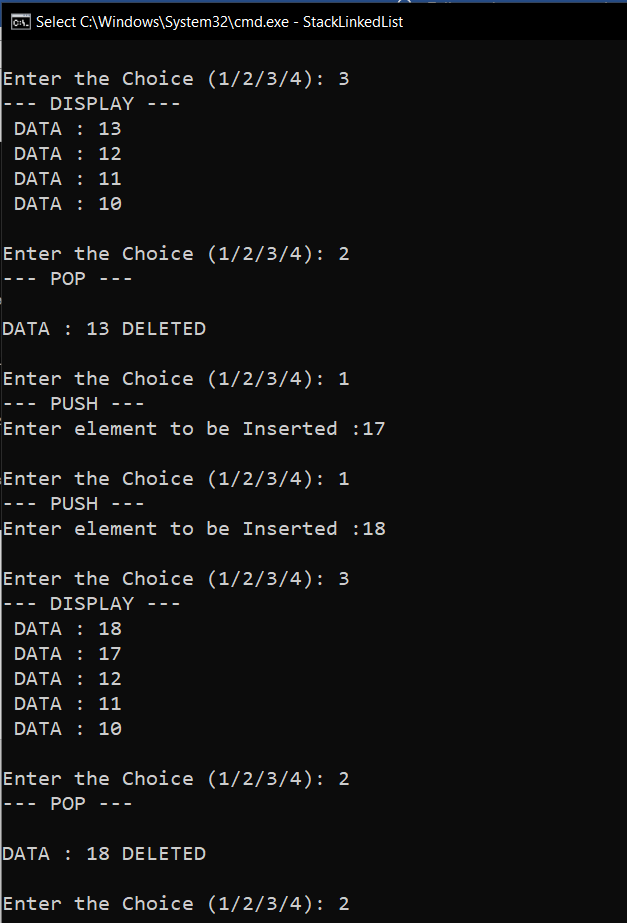
1. ptr=TOP->LINK
2. if(ptr=NULL)then
3. print “The stack is empty”
4. Exit
5. Else
6. TOP->LINK=ptr->LINK
7. ReturnNode(ptr)
8. stop

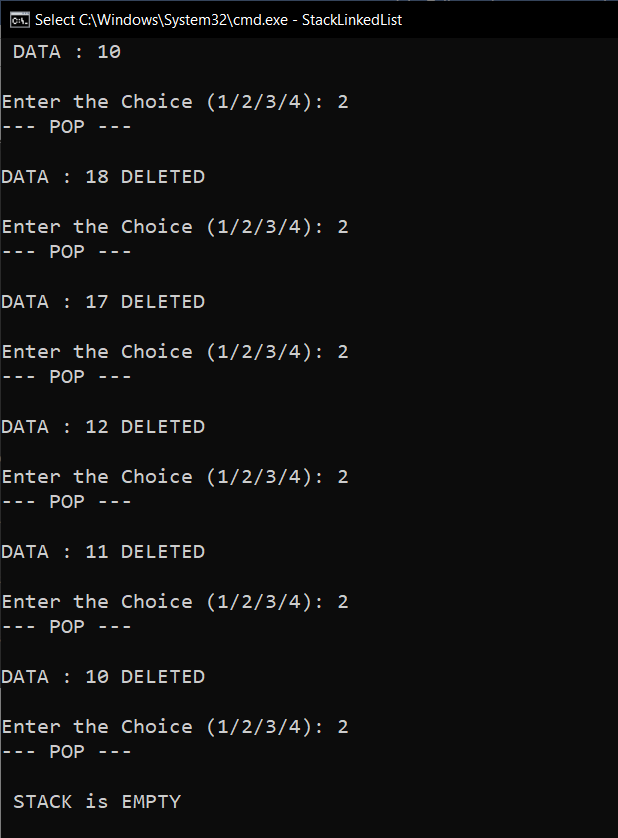
**PROGRAM:**

#include<stdio.h>  
#include<stdlib.h>  
struct node{  
 int data;  
 struct node \*link;  
};  
  
void push(struct node\* top,int x){  
 struct node\* new = (struct node\*)malloc(sizeof(struct node));  
 new->data=x;  
 new->link=NULL;  
 if(new==NULL){  
 printf("\nMEMORY Underflow\n");  
 }else{  
 if(top->link==NULL){  
 top->link=new;  
 }else{  
 new->link=top->link;  
 top->link=new;  
 }  
 }  
}  
  
void pop(struct node\* top){  
 struct node\* ptr=top->link;  
 if(ptr==NULL){  
 printf("\n STACK is EMPTY\n");  
 }else{  
 top->link=ptr->link;  
 printf("\nDATA : %d DELETED\n",ptr->data);  
 free(ptr);  
 }  
}  
  
void display(struct node\* top){  
 struct node\* ptr=top;  
 while(ptr->link!=NULL){  
 ptr=ptr->link;  
 printf(" DATA : %d\n",ptr->data);  
 }  
}  
  
void main(){  
 int n,x,y,key;  
 char ans='y';  
 struct node\* top = (struct node\*)malloc(sizeof(struct node));  
 top->link=NULL;  
 printf("\n--- OPERATION ON STACK --- \n\n");  
 printf(" 1. PUSH \n");  
 printf(" 2. POP \n");  
 printf(" 3. DISPLAY \n");  
 printf(" 4. EXIT \n");  
 while(ans=='y'){  
 printf("\nEnter the Choice (1/2/3/4): ");  
 scanf("%d",&n);  
 switch(n){  
 case 1:printf("--- PUSH ---\n");  
 printf("Enter element to be Inserted :");  
 scanf("%d", &x);  
 push(top,x);  
 break;  
 case 2:printf("--- POP ---\n");  
 pop(top);  
 break;  
 case 3:printf("--- DISPLAY ---\n");  
 display(top);  
 break;  
 case 4:ans='n';  
 break;  
 default:printf("Enter a Valid Input\n");  
 }  
 }  
}

**OUTPUT:**

****





**RESULT:**

The given operations are performed on a Stack implemented using linked list.

Time complexity of PUSH() operation is O(1).

Time complexity of POP() operation is O(1).