Assignments -02

1. Implement Stack using i) fixed memory and ii) dynamic memory representations. In both representations, code the two primitive stack functions. Add a display function to show the content of your stack after every operation.

Solutions

**Code:**

i)

STACK IMPLEMENTATION USING ARRAY (FIXED MEMORY)

*/\*This is the code for stack implementation using array*

*Done By*

*RAHUL SAHA*

*Roll: 001910701009*

*ETCE, UG2 ; Subject : DSA Lab \*/*

*#include<stdio.h>*

*#include<stdlib.h>*

*#include<conio.h>*

*#define MAX 40*

*int Arr[MAX];*

*int top = -1;*

*void display();*

*void push()*

*{*

*int item;*

*printf("Enter the element: ");*

*scanf("%d",&item);*

*if(top == MAX-1)*

*{*

*printf("\n----OVERFLOW DETECTED---\n");*

*return;*

*}*

*Arr[++top] = item;*

*display();*

*}*

*void pop()*

*{*

*if(top == -1)*

*{*

*printf("\n---UNDERFLOW DETECTED---\n");*

*return;*

*}*

*printf("\nElement %d is popped out\n", Arr[top]);*

*top--;*

*display();*

*}*

*void display()*

*{*

*printf("The stack is ");*

*if(top==-1)*

*printf("\n STACK EMPTY");*

*for(int i=top;i>=0;i--)*

*printf("%d ",Arr[i]);*

*}*

*int main()*

*{*

*int choose,num=1;*

*while(num==1)*

*{*

*printf("\n Enter 1 to push element \n Enter 2 to pop from the stack\n Enter 3 to view the latest stack");*

*printf("\n Enter your choice: ");*

*scanf("%d",&choose);*

*if(choose == 1)*

*push();*

*else if(choose == 2)*

*pop();*

*else if(choose == 3)*

*display();*

*else*

*printf("Enter valid choice\n");*

*printf("\nEnter 1 to continue and 0 to exit");*

*printf("\nDo you want to continue?: ");*

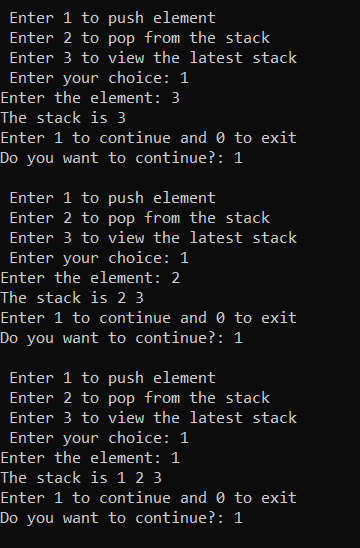
*scanf("%d",&num);*

*}*

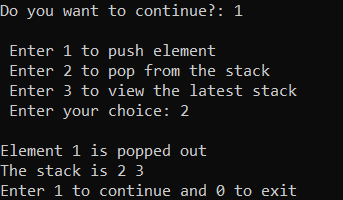
*}*

**Output(s):**

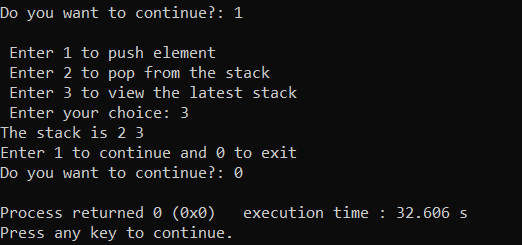
PUSH



POP



DISPLAY



**Code**

ii)

STACK IMPLEMENTATION USING LINKED LIST (DYNAMIC MEMORY ALLOCATION)

*/\*This is the code for stack implementation using linked list*

*Done By*

*RAHUL SAHA*

*Roll: 001910701009*

*ETCE, UG2 ; Subject : DSA Lab \*/*

*#include<stdio.h>*

*#include<stdlib.h>*

*#include<conio.h>*

*struct Node{*

*int data;*

*struct Node\* next;*

*};*

*struct Node\* top;*

*struct Node\* create\_node(val)*

*{*

*struct Node\* temp = (struct Node\*)malloc(sizeof(struct Node));*

*temp->data = val;*

*temp->next = NULL;*

*};*

*void display();*

*void push()*

*{*

*int item;*

*printf("Enter the element: ");*

*scanf("%d",&item);*

*struct Node\* new\_node = create\_node(item);*

*if(top == NULL)*

*{*

*top = new\_node;*

*display();*

*return;*

*}*

*new\_node->next = top;*

*top = new\_node;*

*display();*

*}*

*void pop()*

*{*

*if(top == NULL)*

*{*

*printf("\n----UNDERFLOW DETECTED----\n");*

*return;*

*}*

*struct Node\* del\_node = top;*

*if(del\_node->next == NULL)*

*{*

*free(del\_node);*

*printf("List is empty\n");*

*return;*

*}*

*top = del\_node->next;*

*printf("\n Element %d is popped out", del\_node->data);*

*free(del\_node);*

*display();*

*}*

*void display()*

*{*

*struct Node\* temp = top;*

*printf("\n The list is: ");*

*while(temp!=NULL)*

*{*

*printf("\n %d ",temp->data);*

*temp = temp->next;*

*}*

*}*

*int main()*

*{*

*top =NULL; //initialize*

*int choice,num=1;*

*while(num==1)*

*{*

*printf("\n Enter 1 to push element\n Enter 2 to pop from the stack\n Enter 3 to view the latest stack");*

*printf("\n Enter your choice: ");*

*scanf("%d",&choice);*

*if(choice == 1)*

*push();*

*else if(choice == 2)*

*pop();*

*else if(choice == 3)*

*display();*

*else*

*printf("Enter valid choice\n");*

*printf("\nEnter 1 to continue and 0 to exit");*

*printf("\nDo you want to continue?: ");*

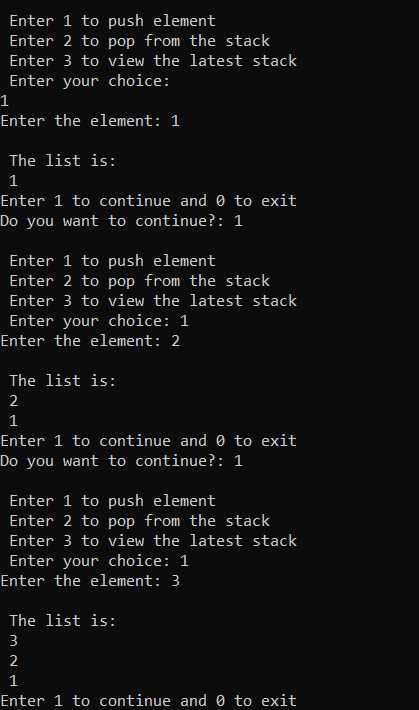
*scanf("%d",&num);*

*}*

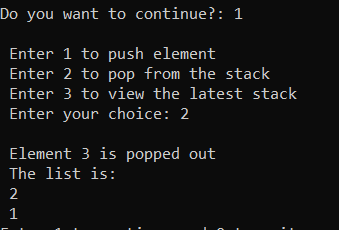
*}*

**Output(s) :**

PUSH



POP



DISPLAY

