**Q1. Stack operations on char datatype using global variables and arrays**

**#include<stdio.h>**

**#include<string.h>**

**#define size 100**

**char ar[size];**

**int top=-1;**

**void push( char ch)**

**{**

**if(top==size-1)**

**printf("OVERFLOW");**

**else**

**ar[++top]=ch;**

**}**

**void pop()**

**{**

**if(top==-1)**

**printf("UNDERFLOW");**

**else**

**printf("%c removed successfully",ar[top]);**

**top--;**

**}**

**void display()**

**{**

**int x;**

**for(x=top;x>=0;x--)**

**printf("%c",ar[x]);**

**}**

**int main()**

**{**

**int val,x;**

**char ch;**

**while(1){**

**printf("\nEnter your choice:");**

**printf("\n1.push");**

**printf("\n2.pop");**

**printf("\n3.traverse");**

**printf("\n4.exit\n");**

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

**if(val==4)**

**break;**

**else**

**{**

**switch(val)**

**{**

**case 1:**

**printf("\nEnter the item:");**

**scanf("%s",&ch);**

**push(ch);**

**break;**

**case 2:pop();**

**break;**

**case 3:traverse();**

**break;**

**}**

**}**

**}**

**return 0;**

**}**

**OUTPUT**



**2. Stack operations on any datatype using local variables (passing parameters)**

**#include<stdio.h>**

**#include<stdlib.h>**

**#define SIZE 5**

**void push(int arr[],int \*temp,int ele); // function prototype for insertion**

**int pop(int arr[],int \*temp);// function prototype for deletion**

**void display(int arr[],int top);// function prototype for display**

**void main()**

**{**

**int ch,item,del;**

**int s[SIZE],top=-1;**

**for(;;) // infinte loop to ask user choices for different operations**

**{**

**printf("\n1.Push 2.Pop 3.Display 4.Exit\n");**

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

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

**switch(ch)**

**{**

**case 1:printf("Enter the item to be inserted\n");**

**scanf("%d",&item);// read element to be inserted to the stack**

**push(s,&top,item);// call push function to insert element**

**break;**

**case 2:**

**del=pop(s,&top); // call pop function to delete top most element of the stack**

**if(del!=-1)**

**printf("Deleted element is %d\n",del); // print deleted element**

**if(top!=-1)**

**top--;**

**else**

**top=-1;**

**break;**

**case 3:display(s,top); // call display function to see the contents of stack**

**break;**

**case 4:exit(0);**

**}**

**}**

**}**

**void push(int arr[],int \*temp,int ele)**

**{**

**if(\*temp==SIZE-1) //check if stack is full**

**{**

**printf("Stack overflow\n");**

**}**

**else**

**{**

**arr[++\*temp] = ele;**

**printf("added %d\n",ele);**

**//return arr[\*temp];**

**}**

**}**

**int pop(int arr[],int \*temp)**

**{**

**if(\*temp==-1) // check if stack is empty**

**{**

**printf("Stack underflow\n");**

**return -1;**

**}**

**else{**

**// int a=\*temp;**

**// arr[\*temp]=0;**

**// (\*temp)--;**

**return arr[\*temp]; //delete the element and decrement top (post decrement)**

**}**

**}**

**void display(int arr[],int top)**

**{**

**int i;**

**if(top==-1) // check if stack is empty**

**{**

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

**return;**

**}**

**printf("Elements in stack are as follows:\n");**

**for(i=0;i<=top;i++) // print all elements of the stack from index 0 to top**

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

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

**}**

**OUTPUT**



**MADE BY -:**

**NAME-RAHUL KUMAR**

**USN-4NM19IS126**