**1.Design and implement a stack[array] and demonstrate its working with necessary input, display the appropriate output.**

**ALGORITHM:**

STEP 1:Include all the header files which are used in the program and define a constant ‘size’ with specific value.

Step 2: Declare all the functions used in stack implementation.

Step 3: Create a one-dimensional array with fixed value.

Step 4: Define a integer variable ‘top’ and initialize with -1.

Step 5: In main method, display, menu list to operations used and make a suitable function call to perform operations selected by the user on the stack.

**TO PUSH**

Step 1: Check whether stack is full (Top==size-1).

Step 2: If it is full, then display “Stack is full” insertion is not possible. And terminate the function.

Step 3: If it is not full, then increment top value by one (top++) and set stack[top] to value (stack[top]=value).

**TO POP**

Step 1: Check whether the stack is empty (top==-1).

Step 2: If it is empty, them display “Stack is Empty” and terminate the function.

**TO DISPLAY**

Step 1: Check whether the stack is empty (top==-1).

Step 2: If empty display “Stack is Empty” and terminate.

Step 3: If not empty, then define a variable ‘i’ and initialize with top. Display stack[i] value and decrement the value by one (i--).

Step 4: Repeat above stop until ‘i’ value becomes ‘0’.

**PROGRAM**

#include<stdio.h> //standard I/O header file

#include<stdlib.h> //standard library header file

#define size 4//declaring the size to 5

int top=-1,inp\_arr[size],i;//declaring variables

void push();

void pop();

void display();

int main()//main fuction

{

int choice;

while(1)//checks the condition and enters the while loop

{

printf("Operation performed by stack\n");//printing statement

printf("1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n");//printing statement

printf("Enter the choice\n");//printing statement

scanf("%d",&choice);//store of value in variable

switch(choice)//switch statement initialization

{

case 1:push();//if option 1 go o push

break;

case 2:pop();//if option is 2 go to pop

break;

case 3:display();//if option is 3 go to display

break;

case 4:exit(0);//if option is 4 exit the loop

default:printf("Invalid choice\n");//printing staement

}

}

}

void push()//push function

{

int x;

if(top==size-1)//condition check

printf("Overflow\n");

else

{

printf("Enter the element to be inserted:\n");

scanf("%d",&x);//the element to be inserted in stack

top=top+1;//top value increased

inp\_arr[top]=x;

}

}

void pop() //pop function

{

if(top==-1)//condition check

printf("Underflow\n");

else

{

printf("Popped the element %d\n",inp\_arr[top]);

top=top-1;//decrement of top

}

}

void display()//display function

{

if(top>=0)//condition check

{

printf("Elements in the stack:\n");

for(i=top;i>=0;i--)// loop for printing the elements in stack

{

printf("%d\n",inp\_arr[i]);// I value decides the place of top

}

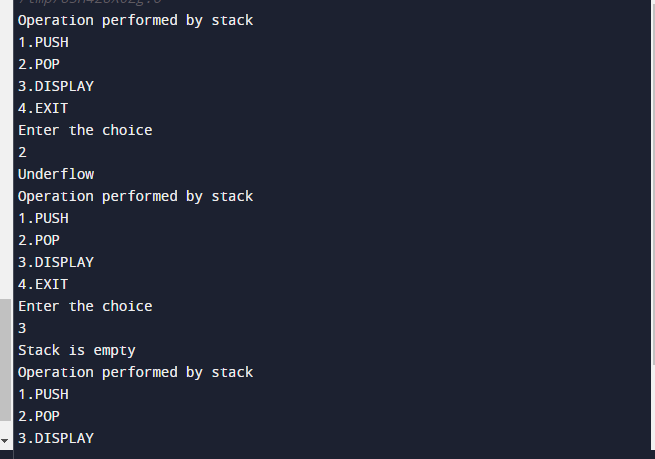
}

else

printf("Stack is empty\n");//empty stack, no elements present

}

**Output**





**GITHUB LINK**