**P1)** Design and implement a stack (Array implementation) and demonstrate its working with necessary inputs. Display the appropriate messages in case of exceptions.

**ALGORITHM**

1. 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).

1. pop( )

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!! Deletion is not possible!!!" and terminate the function.

Step 3 - If it is NOT EMPTY, then delete stack[top] and decrement top value by one (top--).

1. Display( )

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!!" and terminate the function.

Step 3 - If it is NOT EMPTY, then define a variable ' I ' and initialize with top. Display stack[i] value and decrement i value by one (i--).

Step 3 - Repeat above step until i value becomes '0'.

**// stack using array (program)**

#include<stdio.h>

#include<stdlib.h>

#define size 10

void push(int);

void pop();

void display();

int stack[size],top=-1;

void main()

{

while(1)

{

int value,choice;

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

printf("\n1.push\n 2.pop \n 3.display\n 4.exit\n");

printf("\n enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("enter the valuer to be inserted");

scanf("%d",&value);

push(value);

break;

case 2:pop();

break;

case 3:display();

break;

case 4:exit(0);

default : printf("wrong selection");

}

}

}

void push(int value)

{

if (top==size-1)

printf("stack is full");

else

{

top++;

stack[top]=value;

printf("insertion sucessfull");

}

}

void pop()

{

if (top==-1)

printf("stack is empty");

else

{

printf("deleted %d",stack[top]);

top--;

}

}

void display()

{

if (top==-1)

printf("stack id empty");

else

{

int i;

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

printf("%d",stack[i]);

}

}

**Output**







