**Academic Year: 2023-24 Name of Student:**

**Semester: III Student ID:**

**Class / Branch/ Div: SE- IT A/ B/ C Roll No.**

**Subject: Data Structure Lab Date of Submission:**

**Name of Instructor:**

**Experiment No.1**

**Aim:** Implementation of stack and queue using array

**Code:**

**Program on Stack using array:**

#include<stdio.h>

int stack [100], choice, n, top, x, i;

void push(void);

void pop(void);

void display(void);

int main()

{

top=-1;

printf("\n Enter the size of STACK : ");

scanf("%d",&n);

printf("\n STACK OPERATIONS USING ARRAY");

printf("\n 1.PUSH\n 2.POP\n 3.DISPLAY\n 4.EXIT");

do

{

printf("\n Enter the Choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:

{

push();

break;

}

case 2:

{

pop();

break;

}

case 3:

{

display();

break;

}

case 4:

{

printf("\nExiting...");

break;

}

default:

{

printf ("\n Please Enter a Valid Choice(1/2/3/4)");

}

}

}

while(choice!=4);

return 0;

}

void push()

{

if(top>=n-1)

{

printf("\nSTACK is over flow");

}

else

{

printf(" Enter a value to be pushed:");

scanf("%d",&x);

top++;

stack[top]=x;

}

}

void pop()

{

if(top<=-1)

{

printf("\n Stack is under flow");

}

else

{

printf("\nThe popped elements is %d",stack[top]);

top--;

}

}

void display()

{

if(top>=0)

{

printf("\n The elements in STACK \n");

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

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

printf("\n Enter Next Choice");

}

else

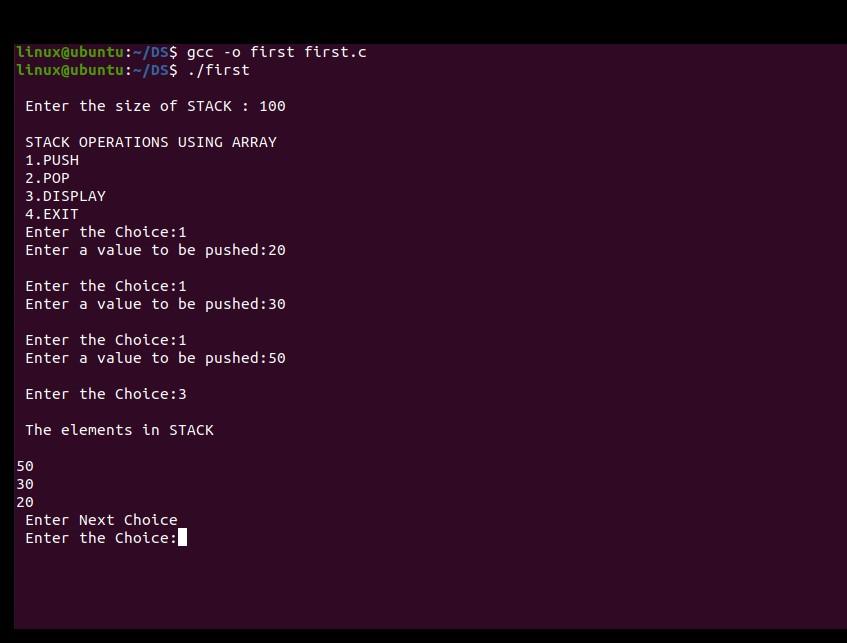
{

printf("\n The STACK is empty");

}

}

**OUTPUT:**



**Program on Queue using array:**

#include <stdio.h>

#define MAX 10 // Changing this value will change length of array

int queue[MAX];

int front = -1, rear = -1;

void insert(void);

int delete\_element(void);

int peek(void);

void display(void);

int main()

{

int option, val;

do

{

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

printf("\n 1. Insert an element");

printf("\n 2. Delete an element");

printf("\n 3. Peek");

printf("\n 4. Display the queue");

printf("\n 5. EXIT");

printf("\n Enter your option : ");

scanf("%d", &option);

switch(option)

{

case 1:

insert();

break;

case 2:

val = delete\_element();

if (val != -1)

printf("\n The number deleted is : %d", val);

break;

case 3:

val = peek();

if (val != -1)

printf("\n The first value in queue is : %d", val);

break;

case 4:

display();

break;

}

}while(option != 5);

return 0;

}

void insert()

{

int num;

printf("\n Enter the number to be inserted in the queue : ");

scanf("%d", &num);

if(rear == MAX-1)

printf("\n OVERFLOW");

else if(front == -1 && rear == -1)

front = rear = 0;

else

rear++;

queue[rear] = num;

}

int delete\_element()

{

int val;

if(front == -1 || front>rear)

{

printf("\n UNDERFLOW");

return -1;

}

else

{

val = queue[front];

front++;

if(front > rear)

front = rear = -1;

return val;

}

}

int peek()

{

if(front==-1 || front>rear)

{printf("\n QUEUE IS EMPTY");

return -1;

}

else

{

return queue[front];

}

}

void display()

{

int i;

printf("\n");

if(front == -1 || front > rear)

printf("\n QUEUE IS EMPTY");

else

{

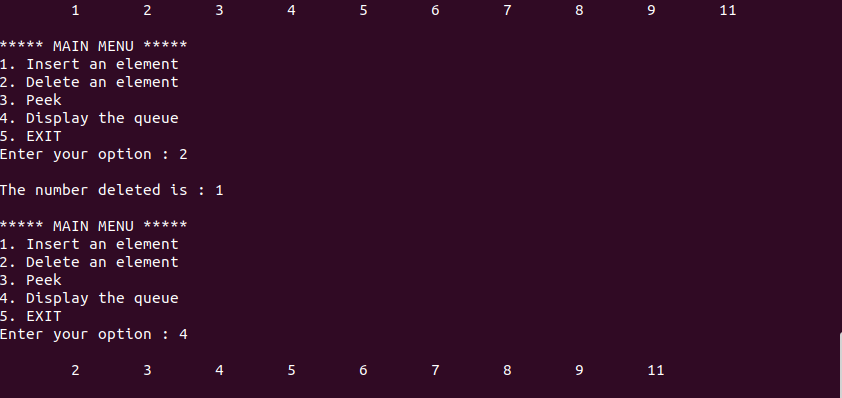
for(i = front;i <= rear;i++)

printf("\t %d", queue[i]);

}

}

**OUTPUT:**



**Conclusion: -**

Thus, we have implemented stack and queue data structure using array. In this we can create a stack and queue using different operations like push (), pop (), enqueue (), dequeue (), display () etc.