**ST. XAVIER’S COLLEGE**

**MAITIGHAR, KATHMANDU**

**LAB ASSIGNMENT #13**

**OF**

**Data Structure & Algorithm**



**Submitted To:**  **Submitted By:**

Mr. Bal Krishna Subedi Name: Sajan Sanjel

Lecturer Class: BSc CSIT (2nd Semester)

Department of Computer Science Roll No.: 014BSCIT039

Write a program to implement:

1. Ascending priority queue
2. Descending priority queue
3. ASCENDING PRIORITY QUEUE

**Source Code**

#include<stdio.h>

#include<conio.h>

#include<process.h>

#define SIZE 20

struct queue

{

int item[SIZE];

int rear;

int front;

};

typedef struct queue pq;

void insert(pq\*);

void delet(pq\*);

void display(pq\*);

void main()

{

int ch;

pq \*q;

q->rear=-1;

q->front=0;

clrscr();

printf("Menu for The priority queue program:\n\n");

printf("1:INSERT\n2:DELETE\n3:DISPLAY\n4:EXIT\n\n");

do

{

printf("Please enter youer choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

insert(q);

break;

case 2:

delet(q);

break;

case 3:

display(q);

break;

case 4:

exit(1);

break;

default:

printf("Sorry your choice is incorrect\n");

break;

}

}while(ch<5);

getch();

}

void insert(pq \*q)

{

int d;

if(q->rear==SIZE-1)

printf("Sorry the queue is full\n");

else

{

printf ("Please enter data to be inserted\n");

scanf("%d",&d);

q->rear++;

q->item[q->rear]=d;

}

}

void delet(pq \*q)

{

int i, temp=0, x;

x=q->item[q->front];

if(q->rear<q->front)

{

printf("Sorry the queue is empty\n");

}

else

{

for(i=q->front+1; i<q->rear; i++)

{

if(x>q->item[i])

{

temp=i;

x=q->item[i];

}

}

for(i=temp;i< q->rear-1;i++)

{

q->item[i]=q->item[i+1];

}

q->rear--;

}

}

void display(pq \*q)

{

int i;

if(q->rear < q->front)

printf("Sorry the queue is empty\n");

else

{

printf("Items of the priority queue are:\n");

for(i=(q->front);i<=q->rear;i++)

{

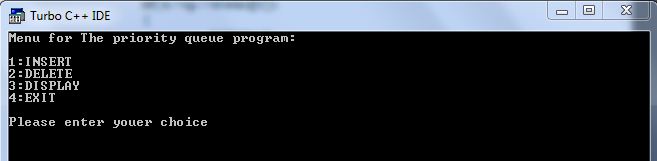
printf("%d\t",q->item[i]);

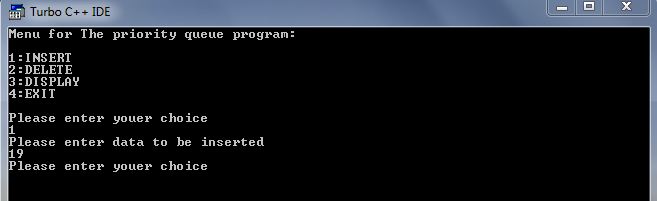
}

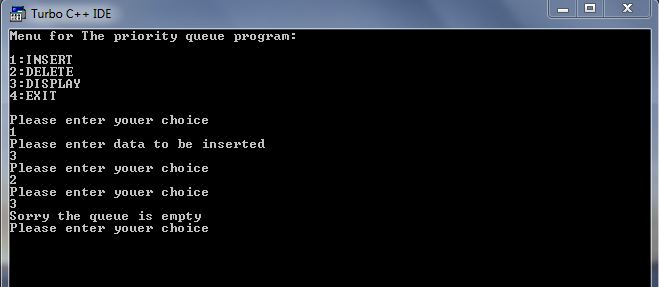
}

}

**Output Screen**







1. DESCENDING PRIORITY QUEUE

**Source Code**

#include<stdio.h>

#include<conio.h>

#include<process.h>

#define SIZE 20

struct queue

{

int item[SIZE];

int rear;

int front;

};

typedef struct queue pq;

void insert(pq\*);

void delet(pq\*);

void display(pq\*);

void main()

{

int ch;

pq \*q;

q->rear=-1;

q->front=0;

clrscr();

printf("Menu for The priority queue program:\n\n");

printf("1:INSERT\n2:DELETE\n3:DISPLAY\n4:EXIT\n\n");

do

{

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

scanf("%d",&ch);

switch(ch)

{

case 1:

insert(q);

break;

case 2:

delet(q);

break;

case 3:

display(q);

break;

case 4:

exit(1);

break;

default:

printf("Sorry your choice is incorrect\n");

break;

}

}while(ch<5);

getch();

}

void insert(pq \*q)

{

int d;

if(q->rear==SIZE-1)

printf("Sorry the queue is full\n");

else

{

printf ("Please enter data to be inserted\n");

scanf("%d",&d);

q->rear++;

q->item[q->rear]=d;

}

}

void delet(pq \*q)

{

int i, temp=0, x;

x=q->item[q->front];

if(q->rear<q->front)

{

printf("Sorry the queue is empty\n");

}

else

{

for(i=q->front+1; i<q->rear; i++)

{

if(x<q->item[i])

{

temp=i;

x=q->item[i];

}

}for(i=temp;i< q->rear-1;i++)

{

q->item[i]=q->item[i+1];

}

q->rear--;

}

}

void display(pq \*q)

{

int i;

if(q->rear < q->front)

printf("Sorry the queue is empty\n");

else

{

printf("Items of the priority queue are:\n");

for(i=(q->front);i<=q->rear;i++)

{

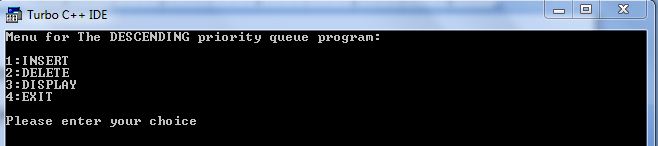
printf("%d\t",q->item[i]);

}

}

}

**Output Screen**

****

