**ST.XAVIER’S COLLEGE**

**MAITIGHAR, KATHMANDU**

****

**DATA STRUCTURE AND ALGORITHM LAB ASSIGNMENT#13**

**SUBMITTED BY:**

SARITA KARKI

2ND SEM

014BSCIT040

**SUBMITTED TO:**

|  |  |  |
| --- | --- | --- |
|  | **Signature** | **Remarks** |
| Bal Krishna Subedi  Lecturer, Department of Computer Science |  |  |

Write a program to implement:

1. Ascending priority queue
2. Descending priority queue

A) ASCENDING PRIORITY QUEUE

#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 program:\n");

printf("1:insert\n2:delete\n3:display\n4:exit\n");

do

{

printf("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 wrong\n");

break;

}

}while(ch<5);

getch();

}

void insert(pq \*q)

{

int d;

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

printf("Queue is full\n");

else

{

printf ("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("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("Queue is empty\n");

else

{

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

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

{

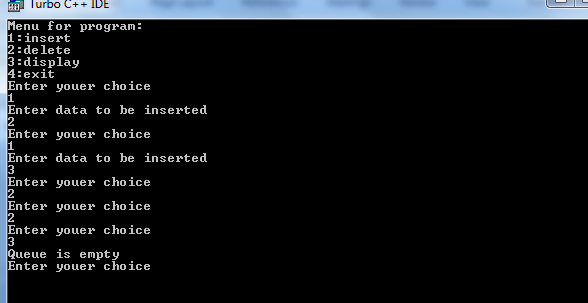
printf("%d\t",q->item[i]);

}

}

}

**OUTPUT:**

****

II) DESCENDING PRIORITY

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 program:\n");

printf("1:insert\n2:delete\n3:display\n4:exit\n");

do

{

printf("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 wrong\n");

break;

}

}while(ch<5);

getch();

}

void insert(pq \*q)

{

int d;

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

printf("Queue is full\n");

else

{

printf ("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("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("Queue is empty\n");

else

{

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

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

{

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

}

}

}

OUTPUT:

