**ST. XAVIER’S COLLEGE**

**MAITIGHAR, KATHMANDU**

**LAB ASSIGNMENT #13**

**OF**

**Data Structure & Algorithm**



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

Mr. Bal Krishna Subedi Name: Surya Khanal

Lecturer Class: BSc.CSIT (2nd Semester)

Department of Computer Science Roll No.: 014BSCIT046

Q.N. 1

1. STATEMENT: WAP TO IMPLEMENT ASCENDING PRIORITY QUEUE.
2. 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\t2.Delete\t3.Display\t4.Exit\n");

do

{

printf("Enter Choice:\t");

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("Enter from 1-4\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++)

{

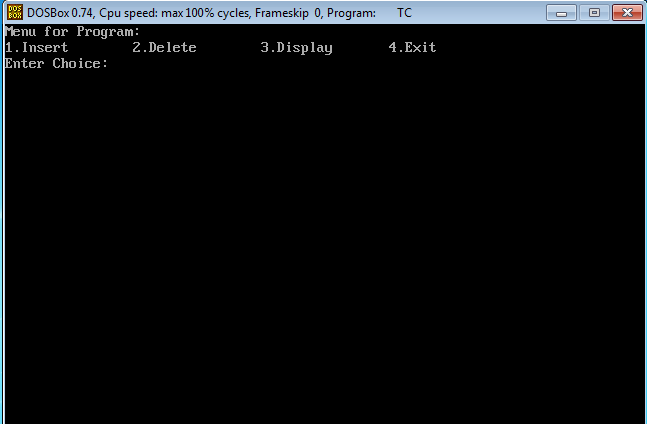
printf("%d\n",q->item[i]);

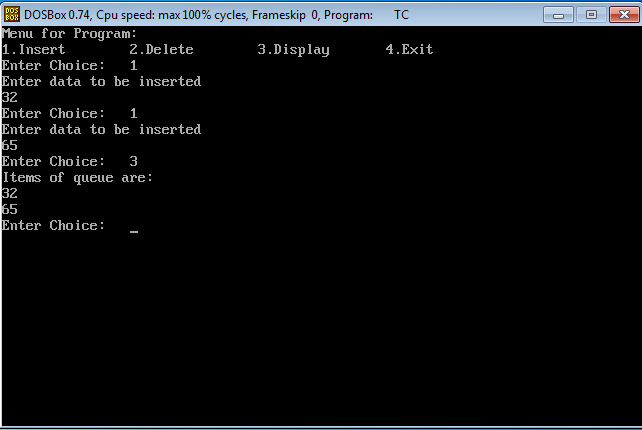
}

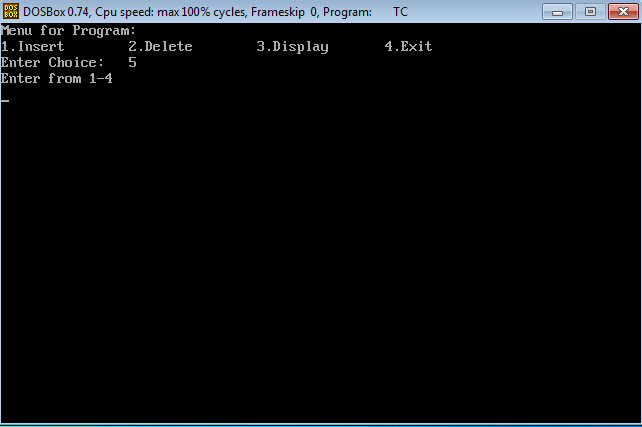
}

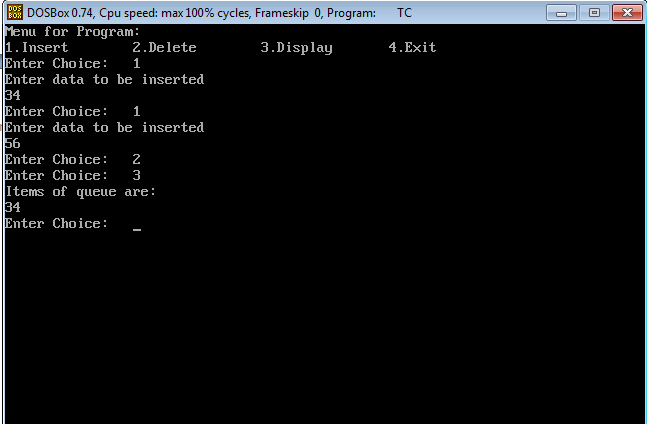
}

1. OUTPUT SCREEN:









1. CONCLUSION:

HENCE, A PROGRAM TO IMPLEMENT ASCENDING PRIORITY QUEUE WAS DETERMINED.

Q.N. 2

1. STATEMENT: WAP TO IMPLEMENT DESCENDING PRIORITY QUEUE.
2. 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\t2.Delete\t3.Display\t4.Exit\n");

do

{

printf("Enter Choice:\t");

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("Enter from 1-4\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++)

{

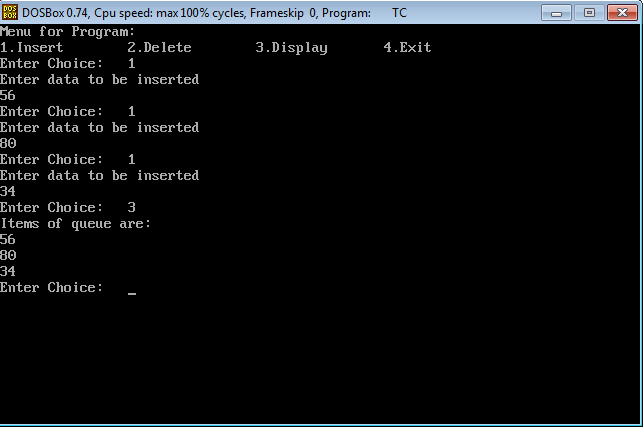
printf("%d\n",q->item[i]);

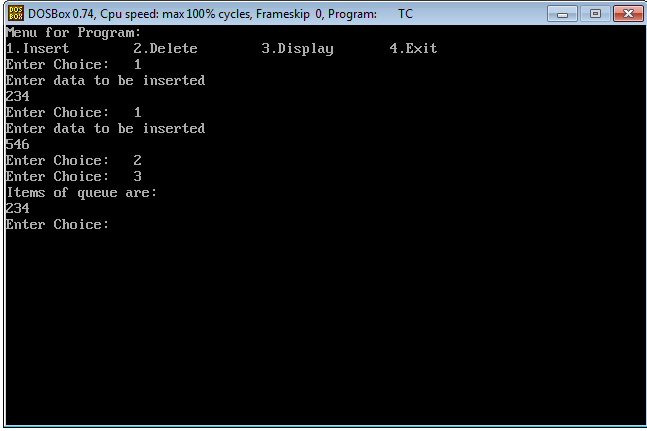
}

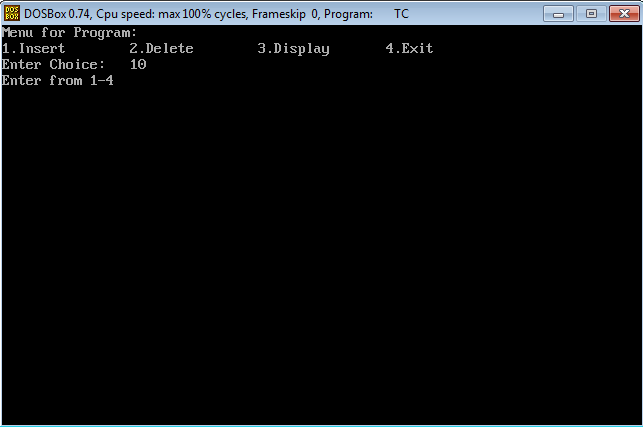
}

}

1. OUTPUT SCREEN:







1. CONCLUSION:

HENCE, A PROGRAM TO IMPLEMENT DESCENDING PRIORITY QUEUE WAS DETERMINED.