Algorithm

Step 1: IF REAR = MAX - 1

Write OVERFLOW

Go to step

[END OF IF]

Step 2: IF FRONT = -1 and REAR = -1

SET FRONT = REAR = 0

ELSE

SET REAR = REAR + 1

[END OF IF]

Step 3: Set QUEUE[REAR] = NUM

Step 4: EXIT

// implementation of queues in array //

#include<stdio.h>

#include<stdlib.h>

#define max size 30

Int front=-1,rear=-1;

Int queue[maxsize];

Void insert()

{

Int item;

Printf(“enter the elements”);

Scanf(“%d”,&item);

If(rear==maxsize-1)

{

Printf(“overflow”);

Return;

}

If(front==-1&&rear==-1)

{

Front=0;

Rear=0;

}

Else

Rear=rear+1;

Queue[rear]=item;

Printf(“value inserted”);

}

Void delete()

{

Int item;

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

{

Printf(“underflow”);

Return;

}

Else

{

Item=queue[front];

If(front==rear)

{

Front=-1;

Rear=-1;

}

Else

Front=front+1;

}

}

Void display()

{

Int I;

If(rear==-1)

Printf(“empty queue”);

Else

{

Printf(“printing the values”);

For(i=fronti<=rear,i++)

Printf(“%d”,queue[i]);

}

}

Int main()

{

Int choice;

While(1)

{

Printf(“1.insert an element\n,2.delete an element\n,3.display the queue\n,4.exit\n”);

Printf(“enter the choice”);

Scanf(“%d”&choice);

Switch(choice)

{

Case 1:

Insert();

Break;

Case 2 :

Delete();

Break;

Case 3:

Display();

Break;

Case 4:

Exit(0);

Default;

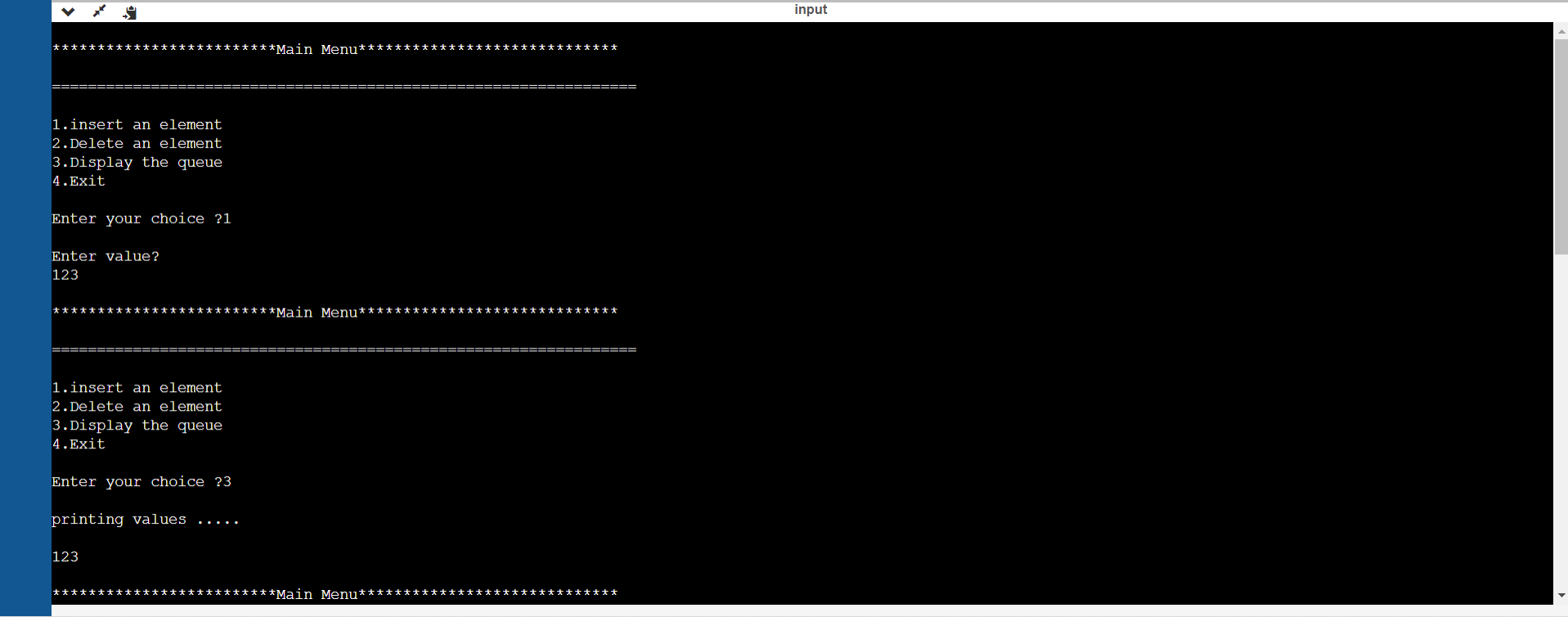
Printf(“enter vaild choice ??\n”);

}

}

}

OUTPUT :



GITHUB LINK :