Question 1

Write a program implementing insert, delete and display operation of Circular Queue.

Ans).

int q[5];

void insertq ()  
{ int x;

if ((front==0 && rear== size-1) || (front>0 && rear==front-1))

printf(“Overflow”);return;

else

{

printf(“Enter number to insert”);

scanf(“%d”,&x);

if(rear==size-1 && front>0)

{

rear=0; q[rear]=x;

}

else

{

if((front==0 && rear==-1)||(rear!=front-1))

q[++rear]=x;

}

}

}

void qdelete ()

{ int a;

if(rear==-1 && front ==0)

printf(“Underflow”);return;

else

{

if(rear==front)

{

a=q[front];

rear=-1;

front=0;

}

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

{

a=q[front];

front=0;

}

else

a=q[front++];

printf(“Deleted = %d”,a);

}

void display()

{

if(rear==-1 && front ==0)

printf(“Underflow”);return;

if(front>rear)

{

for(int i=0;i<=rear;i++)

printf(“%d\t”,q[i]);

for(int i=front;i<=size-1;i++)

printf(“%d\t”,q[i]);

}

else

{

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

printf(“%d\t”,q[i]);

}

}

Question 2

A Barua number is a number which consists of only zeroes and ones and has only one 1. Barua

number will start with 1. Given numbers, find out the multiplication of the numbers. Note: The

input may contain one decimal number and all other Barua numbers. (Assume that each number

is very large and total number of values give is also very large)

Input 1: 100 10 12 1000

Output 1: 12000000

Input 2: 100 121 1000000000000000

Output 2: 12100000000000000000

Input 3: 10 100 1000

Output 3: 1000000

Ans).

int n,p=1,c=0,num;

printf(“Total numbers to be inputted”);

scanf(“%d”,&n);

for(int i=1;i<=n;i++)

{

scanf(“%d”,&num);

while(num%10==0)

{

c++;

num=num/10;

}

p=p\*num;

}

printf(“%d”,p);

for(int i=1;i<=c;i++)

printf(“0”);

Question 3

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

Ans).

int stack[5],top=-1,mintop=-1,size=5,minstack[5];

void push(int x)

{

if (top==size-1)

{ printf(“Overflow”);

return;

}

else

{

stack[++top]=x;

}

}

void pop()

{

if(top==-1)

{

printf(“Underflow”);

return;

}

else

{

top--;

}

}

void minpush(int x)

{

if (mintop==size-1)

{ printf(“Overflow”);

return;

}

else

{

minstack[++mintop]=x;

}

}

void minpop()

{

if(mintop==-1)

{

printf(“Underflow”);

return;

}

else

{

mintop--;

}

}

void display ()

{

if(top==-1)

{

printf(“Underflow”);

return;

}

else

{

printf(“Stack”);

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

printf(“stack[top]);

}

}

void mindisplay ()

{

if(mintop==-1)

{

printf(“Underflow”);

return;

}

else

{

printf(“MinStack”);

for(int i=mintop;i>=0;i--)

printf(“minstack[mintop]);

}

}

void main ()

{ int ch;

do

{

printf(“Choice”);

scanf(“%d”,&ch);

switch(ch)

{

case 1:

scanf(“%d”,&x);

push(x);

if(mintop==-1)

minpush(x);

else if(x<minstack[mintop])

minpush(x);

else

minpush(minstack[mintop]);

}

break;

case 2:

pop();

minpop();

break;

case 3:

display();

mindisplay();

break;

case 4: printf(“no\n”);

}while(ch!=4);

printf(“Smallest= %d”,minstack[mintop]);

}