1. Write a program implementing insert, delete and display operation of Circular Queue.
2. #include<stdio.h>

int q[50], f = -1, r = -1, size;

void enqueue(int x);

void dequeue();

void display();

int main()

{

printf("Enter the queue size: ");

scanf("%d", &size);

enqueue(10);

enqueue(20);

enqueue(30);

enqueue(40);

enqueue(50);

display();

enqueue(60);

dequeue();

display();

return 0;

}

void enqueue(int x)

{

if(f == 0 && r == size-1) || (f == r+1)))

{

printf("Overflow\n");

return;

}

if(f == -1)

{

f = r = 0;

}

else

{

if(r == size-1)

{

r = 0;

}

else

{

r = r+1;

}

q[r] = x ;

}

void dequeue()

{

int x;

if(f ==-1)

{

printf("Underflow\n");

return;

}

x = q[f];

if(f == r)

{

f = r = -1;

}

else

{

if(r == size-1)

r = 0;

else

{

r = r+1;

}

}

q[r] = x ;

}

void display()

{

int i,j;

if(f == -1)

{

printf("No item to display\n");

return;

}

for(i = f; i <= r; i++)

while(i <= size-1)

{

printf("%d ",q[i])

i++;

}

i = 0;

while(i <= j)

{

printf("%d ",q[i]);

i++;

}

printf("|%d|", q[i]);

printf("\n");

}

1. Find out the multiplication of the numbers. Note: The input may contain one decimal number and all other Barua numbers.
2. #include<stdio.h>

main()

{

Int a[n],i,j=0;

printf(“Enter the no. of numbers:”);

scanf(“%d”,&n);

printf(“Enter the no.s”);

for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

while(j<=n)

{

printf(a[j]\*a[j++]);

j++;

}

}

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

1. #include<stdio.h>

int stack[50], top = -1, size;

void push(int x);

void pop();

void display();

int main()

{

printf("Enter the size of the stack: ");

scanf("%d", &size);

push(10);

push(20);

push(30);

push(40);

push(50);

display();

push(60);

pop();

display();

return 0;

}

void push(int x)

{

if(top == size - 1)

{

printf("Overflow\n");

return;

}

stack[++top] = x;

}

void pop()

{

if(top == -1)

{

printf("Underflow\n");

return;

}

int x = stack[top--];

printf("Deleted item: %d\n", x);

}

void display()

{

int i;

if(top == -1)

{

printf("No element to display\n");

return;

}

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

{

for(j=-1;j<=top;j++)

{

printf("----\n");

if(stack[i]<stack[j])

{

printf("|%d|\n", stack[i]);

}

printf("----\n");

}