**1.**

#include<stdio.h>

int cirQueue[100] ;

int  front = -1 , rear = -1 , size;

void enqueue(int x){

    if((front == 0 && rear == size-1) || (front == (rear+1)%size)){

        printf("OVERFLOW\n");

        return;

    }

    else if(front == -1){

        front = rear = 0;

        cirQueue[rear] = x;

    }

    else if(rear == size -2 && front != 0){

        rear = 0;

        cirQueue[rear] = x;

    }

    else{

        rear++;

        cirQueue[rear] = x;

    }

}

void dequeue(){

    if(front == -1){

        printf("UNDERFLOW\n");

        return;

    }

    printf("%d is deleted\n",cirQueue[front]);

    if(front == rear){

        front = rear = -1;

    }

    else if(front == size-1)

        front = 0;

    else

        front++;

}

void display(){

    int i;

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

        printf("%d\t",cirQueue[i]);

    printf("\n");

}

int main(){

    printf("Enter size of QUEUE\n");

    scanf("%d",&size);

    int x;

    for (int i = 0; i < size; i++)

    {

        printf("Enter data to QUEUE\n");

        scanf("%d",&x);

        enqueue(x);

    }

    display();

    dequeue();

    display();

    dequeue();

    display();

    return 0;

}

**2.**

#include<stdio.h>

int main(){

    int size;

    printf("Enter input size\n");

    scanf("%d ", &size);

    long long inp[size],p=1,c=0;

    for (int i = 0; i < size; i++)

        scanf("%lld",&inp[i]);

    for (int i = 0; i < size; i++)

    {

        long x = inp[i];

        while (x%10 == 0)

        {

            c++;

            x=x/10;

        }

        p \*= x;

    }

    printf("%lld",p);

    for (int i = 0; i < c; i++)

    {

        printf("0");

    }

    printf("\n");

    return 0;

}

**3.**

#include<stdio.h>

int stack[100], minStack[100], minTop = 0, top =-1, size;

void push(int x){

    if(top == size - 1){

        printf("OVERFLOW\n");

        return;

    }

    stack[++top];

    if(minStack[minTop] > x){

        minStack[++minTop] = x;

    }

    else{

        int i = minTop;

        minStack[++minTop] = minStack[i];

    }

}

void pop(){

    if(top == -1){

        printf("UNDERFLOW\n");

        return;

    }

    top = top - 1;

    minTop = minTop -1;

}

int minEleme(){

    return minStack[minTop];

}

int main(){

    printf("Enter size of STACK\n");

    scanf("%d",&size);

    int x;

    minStack[0] = 999;

    for (int i = 0; i < size; i++)

    {

       printf("Enter data to STACK\n");

       scanf("%d",&x);

       push(x);

    }

    printf("min elemnt %d\n",minEleme());

    pop();

    printf("min elemnt %d\n",minEleme());

    return 0;

}