```
   #include <bits/stdc++.h>

using namespace std;



template <typename N> class Node {

     public:

   N value;
   Node * next;

   Node(N val){

      value=val;
      next=NULL;
   }
};

template <typename Q >class Queue{

   Node <Q>*front;
   Node <Q>* rear;

   public :

   Queue (){
      front = NULL;
      rear = NULL;
   }

   //Enqueue-->push

   void push (Q val){

      Node <Q>* NewNode = new Node <Q> (val);

      if ( front == NULL)
      {
         front = NewNode;
         rear = NewNode;
         return;
```

```
    }
rear->next = NewNode;
    rear = rear->next;
 }

 //Deque---->POP

 Q pop (){

    Q chk;
    if(rear == NULL){
       cout<<"Que is underflow && Que have no element "<<endl;
       return chk;
    }

    Node <Q>* dellNode;

    dellNode = front;
    front = front->next;
    if(front == NULL){
       rear ==NULL;
    }
    chk = dellNode->value;
    delete dellNode;
    return chk;
 }

 //Peek Front () , Back()

 Q Front (){
    Q chk;
    chk = front->value;
    return chk;
 }

 Q Back (){
    Q chk;
    chk = rear->value;
    return chk;
 }

 bool empty (){
    if (front == NULL && rear == NULL)
       return true;
```

```
        else
        return false;
    }
};

int main() {

    Queue <int>q;
    int n,i;
    cin>>n;
    for (i=0; i<n; i++){
        int chk;
        cin>>chk;
        q.push(chk);
    }

    while (!q.empty()){
        cout<<q.pop()<<" "<<endl;
    }
    if (!q.empty ()){
        cout<<q.Front()<<endl;
    }

    if (!q.empty() ){
        cout<<q.Back()<<endl;
    }
    return 0;
}
```