#include <iostream>

using namespace std;

class Queue

{

private:

int arr[100];

int frontIndex;

int lastIndex;

int capacity;

int sizee;

public:

Queue(int capacity)

{

this->capacity = capacity;

frontIndex = 0;

lastIndex = -1;

sizee = 0;

}

void enqueue(int value)

{

if (isFull())

{

cout << "Queue is full.\nCannot enqueue." << endl;

return;

}

lastIndex = (lastIndex + 1) % capacity;

arr[lastIndex] = value;

sizee++;

}

int dequeue()

{

if (isEmpty())

{

cout << "Queue is empty.\nCannot dequeue." << endl;

return -1;

}

int frontElement = arr[frontIndex];

frontIndex = (frontIndex + 1) % capacity;

sizee--;

return frontElement;

}

bool isFull()

{

return sizee == capacity;

}

bool isEmpty()

{

return sizee == 0;

}

int display()

{

if (isEmpty())

{

cout << "Queue is empty." << endl;

return -1;

}

return arr[frontIndex];

}

int last()

{

if (isEmpty())

{

cout << "Queue is empty." << endl;

return -1;

}

return arr[lastIndex];

}

int Size()

{

return sizee;

}

};

int main()

{

int T;

long long x;

cout << "Enter your capacity \n";

cin >> x;

cout <<"#1enqueue \n#2dequeue \n";

cout << "Enter the number of attempts : ";

cin >> T;

Queue q(x);

while (T--)

{

int choic;

cout << "Enter your choic : ";

cin >> choic;

if (choic == 1)

{

int n ;

cout << "your valio : ";

cin >> n;

q.enqueue(n);

}

else if (choic == 2)

{

q.dequeue();

}

}

cout << "Front element: " << q.display() << endl;

cout << "Last element: " << q.last() << endl;

cout << "Front element after dequeue: " << q.display() << endl;

cout << "Queue size after dequeue: " << q.Size() << endl;

return 0;

}

