question 1

정수를 저장하는 스택을 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 다섯 가지이다.

* push X: 정수 X를 스택에 넣는 연산이다.
* pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.
* size: 스택에 들어있는 정수의 개수를 출력한다.
* empty: 스택이 비어있으면 1, 아니면 0을 출력한다.
* top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

## **입력**

첫째 줄에 주어지는 명령의 수 N (1 ≤ N ≤ 10,000)이 주어진다. 둘째 줄부터 N개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수는 1보다 크거나 같고, 100,000보다 작거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

## **출력**

출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

answer 1 - good

import java.util.Scanner;

public class T\_10828 {

public static void main(String[] args) {

int[] statc = new int[10000];

int now=0;

Scanner scan = new Scanner(System.in);

int input=scan.nextInt();

scan.nextLine();

String menu;

for(int i=0;i<input;i++){

menu=scan.next();

if(menu.equals("push")){

statc[now]=scan.nextInt();

scan.nextLine();

now++;

}else if(menu.equals("pop")){

if(now==0){

System.out.println("-1");

}else{

System.out.println(statc[now-1]);

now--;

}

}else if(menu.equals("size")){

System.out.println(now);

}else if(menu.equals("empty")){

if(now==0){

System.out.println("1");

}else{

System.out.println("0");

}

}else if(menu.equals("top")){

if(now==0){

System.out.println("-1");

}else{

System.out.println(statc[now-1]);

}

}

}

}

}

qustion 2

정수를 저장하는 큐를 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 여섯 가지이다.

* push X: 정수 X를 큐에 넣는 연산이다.
* pop: 큐에서 가장 앞에 있는 정수를 빼고, 그 수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.
* size: 큐에 들어있는 정수의 개수를 출력한다.
* empty: 큐가 비어있으면 1, 아니면 0을 출력한다.
* front: 큐의 가장 앞에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.
* back: 큐의 가장 뒤에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.

## **입력**

첫째 줄에 주어지는 명령의 수 N (1 ≤ N ≤ 10,000)이 주어진다. 둘쨰 줄부터 N개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수는 1보다 크거나 같고, 100,000보다 작거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

## **출력**

출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

answer 1 - good

import java.util.Scanner;

public class T\_10845 {

public static void main(String[] args) {

int[] statc = new int[10000];

int now=0,first=0;

Scanner scan = new Scanner(System.in);

int input=scan.nextInt();

scan.nextLine();

String menu;

for(int i=0;i<input;i++){

menu=scan.next();

if(menu.equals("push")){

statc[now]=scan.nextInt();

scan.nextLine();

now++;

}else if(menu.equals("pop")){

if(now==first){

System.out.println("-1");

}else{

System.out.println(statc[first]);

first++;

}

}else if(menu.equals("size")){

System.out.println(now-first);

}else if(menu.equals("empty")){

if(now==first){

System.out.println("1");

}else{

System.out.println("0");

}

}else if(menu.equals("front")){

if(now==first ){

System.out.println("-1");

}else{

System.out.println(statc[first]);

}

}else if(menu.equals("back")){

if(now==first){

System.out.println("-1");

}else{

System.out.println(statc[now-1]);

}

}

}

}

}

question 3

## **문제**

정수를 저장하는 덱(Deque)를 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 여덟 가지이다.

* push\_front X: 정수 X를 덱의 앞에 넣는다.
* push\_back X: 정수 X를 덱의 뒤에 넣는다.
* pop\_front: 덱의 가장 앞에 있는 수를 빼고, 그 수를 출력한다. 만약, 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
* pop\_back: 덱의 가장 뒤에 있는 수를 빼고, 그 수를 출력한다. 만약, 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
* size: 덱에 들어있는 정수의 개수를 출력한다.
* empty: 덱이 비어있으면 1을, 아니면 0을 출력한다.
* front: 덱의 가장 앞에 있는 정수를 출력한다. 만약 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
* back: 덱의 가장 뒤에 있는 정수를 출력한다. 만약 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.

## **입력**

첫째 줄에 주어지는 명령의 수 N (1 ≤ N ≤ 10,000)이 주어진다. 둘쨰 줄부터 N개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수는 1보다 크거나 같고, 100,000보다 작거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

## **출력**

출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

answer 1 - runtime error

import java.util.Scanner;

public class T\_10866 {

public static void main(String[] args) {

int[] statc = new int[10000];

int now=0,first=0,realfirst=0;

Scanner scan = new Scanner(System.in);

int input=scan.nextInt();

scan.nextLine();

String menu;

for(int i=0;i<input;i++){

menu=scan.next();

if(menu.equals("push\_back")){

statc[now]=scan.nextInt();

scan.nextLine();

now++;

}if(menu.equals("push\_front")){

if(realfirst==0){

realfirst=9999;

first--;

statc[realfirst]=scan.nextInt();

scan.nextLine();

}else{

statc[realfirst]=scan.nextInt();

scan.nextLine();

realfirst--;

first--;

}

}else if(menu.equals("pop\_back")){

if(now==first){

System.out.println("-1");

}else{

System.out.println(statc[now-1]);

now--;

}

}else if(menu.equals("pop\_front")){

if(now==first){

System.out.println("-1");

}else{

System.out.println(statc[realfirst]);

first++;

if(realfirst==9999){

realfirst=0;

}else{

realfirst++;

}

}

}else if(menu.equals("size")){

System.out.println(now-first);

}else if(menu.equals("empty")){

if(now==first){

System.out.println("1");

}else{

System.out.println("0");

}

}else if(menu.equals("front")){

if(now==first ){

System.out.println("-1");

}else{

System.out.println(statc[realfirst]);

}

}else if(menu.equals("back")){

if(now==first){

System.out.println("-1");

}else{

System.out.println(statc[now-1]);

}

}

}

}

}

answer 2 - good

import java.util.Scanner;

public class T\_10866 {

public static void main(String[] args) {

int[] statc = new int[10000];

int realnow=0,now=0,first=0,realfirst=0;

Scanner scan = new Scanner(System.in);

int input=scan.nextInt();

scan.nextLine();

String menu;

for(int i=0;i<input;i++){

menu=scan.next();

if(menu.equals("push\_back")){

if(realnow==9999){

statc[realnow]=scan.nextInt();

scan.nextLine();

realnow=0;

now++;

}else{

statc[realnow]=scan.nextInt();

scan.nextLine();

now++;

realnow++;

}

}if(menu.equals("push\_front")){

if(realfirst==0){

realfirst=9999;

first--;

statc[realfirst]=scan.nextInt();

scan.nextLine();

}else{

realfirst--;

first--;

statc[realfirst]=scan.nextInt();

scan.nextLine();

}

}else if(menu.equals("pop\_back")){

if(now==first){

System.out.println("-1");

}else if(realnow==0){

realnow=9999;

System.out.println(statc[realnow]);

now--;

}else{

System.out.println(statc[realnow-1]);

now--;

realnow--;

}

}else if(menu.equals("pop\_front")){

if(now==first){

System.out.println("-1");

}else{

System.out.println(statc[realfirst]);

first++;

if(realfirst==9999){

realfirst=0;

}else{

realfirst++;

}

}

}else if(menu.equals("size")){

System.out.println(now-first);

}else if(menu.equals("empty")){

if(now==first){

System.out.println("1");

}else{

System.out.println("0");

}

}else if(menu.equals("front")){

if(now==first ){

System.out.println("-1");

}else{

System.out.println(statc[realfirst]);

}

}else if(menu.equals("back")){

if(now==first){

System.out.println("-1");

}else{

if(realnow==0){

System.out.println(statc[9999]);

}else{

System.out.println(statc[realnow-1]);

}

}

}

}

}

}