## Bài tập ctdlgt-bt05

### Bài 1

Lấy mã nguồn chương trình và dữ liệu về các cài đặt (implementation) khác nhau của hai cấu trúc Stack và Queue tại [link mã nguồn môn học](http://algs4.cs.princeton.edu/code/). Chạy lại chương trình với dữ liệu test. Sau đó đọc mã nguồn. Chú ý: Thư viện [algs4.jar](http://algs4.cs.princeton.edu/code/algs4.jar)

**Chú ý**: Trong thư viện của Java có sẵn Stack và Queue để sử dụng khi làm các bài tập tiếp sau. Link tới ví dụ cho [Stack](http://www.javacoffeebreak.com/faq/faq0037.html) và cho [Queue](https://docs.oracle.com/javase/tutorial/collections/interfaces/queue.html).

### Bài 2 (Solved)

[Java List](https://www.hackerrank.com/challenges/java-list/problem)

Tóm tắt: Có hai loại thao tác trên một list (danh sách): a) thêm giá trị *y* vào vị trí *x* và b) xoá đi phần tử tại vị trí *x*.

Cho danh sách *L* của *N* số nguyên. Cho *Q* thao tác. Hãy thực hiện *Q* thao tác trên danh sách *L*; sau đó in ra danh sách kết quả *L* cuối cùng.

Nên sử dụng Linked List hay ArrayList khi làm bài này.

**import** java.io.\*;

**import** java.util.\*;

**import** java.text.\*;

**import** java.math.\*;

**import** java.util.regex.\*;

**public** **class** Solution {

**public** **static** **void** main(String[] args) {

*/\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*

        Scanner scanner = **new** Scanner(System.in);

**int** N = scanner.nextInt();

        List<Integer> arr = **new** ArrayList<Integer>();

**for** (**int** i = 0; i < N; i++) {

**int** tmp = scanner.nextInt();

            arr.add(tmp);

        }

**int** queries = scanner.nextInt();

**for** (**int** i = 0; i < queries; i++) {

            String s = scanner.next();

**if** (s.equals("Insert")) {

**int** pos = scanner.nextInt();

**int** data = scanner.nextInt();

                arr.add(pos, data);

            }

**else** **if** (s.equals("Delete")) {

                arr.remove(scanner.nextInt());

            }

        }

**for** (**int** i= 0; i < arr.size(); i++) System.out.print(arr.get(i) + " ");

    }

}

### Bài 3 (Solved)

[Java Arraylist](https://www.hackerrank.com/challenges/java-arraylist/problem)

Tóm tắt: Cho *n* hàng, mỗi hàng có 0 hay 1 hay nhiều các số nguyên. Số các số nguyên mỗi hàng không giống nhau. Có *q* câu truy vấn, mỗi câu yêu cầu in ra giá trị tại côt *y* hàng *x*. Nếu không có giá trị tại cột *y* hàng *x* thì in ra “ERROR".

Nên sử dụng arraylist của các arraylist (của các số nguyên) khi làm bài này.

**import** java.io.\*;

**import** java.util.\*;

**import** java.text.\*;

**import** java.math.\*;

**import** java.util.regex.\*;

**public** **class** Solution {

**public** **static** **void** main(String[] args) {

*/\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*

        Scanner scanner = **new** Scanner(System.in);

        ArrayList<ArrayList> arr = **new** ArrayList();

**int** N = scanner.nextInt();

**for** (**int** i = 0; i < N; i++) {

            ArrayList<Integer> subarr = **new** ArrayList<>();

**int** n = scanner.nextInt();

**for** (**int** j = 0; j < n; j++) {

                subarr.add(scanner.nextInt());

            }

            arr.add(subarr);

        }

**int** queries = scanner.nextInt();

**for** (**int** i= 0; i < queries; i++) {

**int** x = scanner.nextInt();

**int** y = scanner.nextInt();

**try** {

                System.out.println(arr.get(x-1).get(y-1));

            } **catch** (IndexOutOfBoundsException e) {

                System.out.println("ERROR!");

            }

        }

        scanner.close();

    }

}

### Bài 4 (Solved)

[Balanced Brackets](https://www.hackerrank.com/challenges/balanced-brackets)

Tóm tắt: Kiểm tra xem các dấu đóng mở ngoặc {[()]} có đúng qui tắc hay không.

Sử dụng stack trong bài này. Nên xem lại ví dụ xử lý và tính toán biểu thức trong slide bài giảng.

    public static String isBalanced(String s) {

        while (true) {

            String originalString = s;

            s = s.replace("()", "");

            s = s.replace("{}", "");

            s = s.replace("[]", "");

            if (s.length() == originalString.length()) break;

        }

        if (s.length() > 0) return "NO";

        return "YES";

    // Write your code here

    }

Cách 2:

        public static String isBalanced(String s) {

        Stack<Character> stack = new Stack<>();

        int i;

        for (i = 0; i < s.length(); i++) {

            if (s.charAt(i) == '(') stack.push('(');

            else if (s.charAt(i) == '[') stack.push('[');

            else if (s.charAt(i) == '{') stack.push('{');

            else if (stack.isEmpty()) return "NO";

            else if (stack.peek().equals('(') && s.charAt(i) == ')') stack.pop();

            else if (stack.peek().equals('[') && s.charAt(i) == ']') stack.pop();

            else if (stack.peek().equals('{') && s.charAt(i) == '}') stack.pop();

            else break;

        }

        if (stack.isEmpty() && i == s.length()) return "YES";

        return "NO";

    // Write your code here

    }

### Bài 5 (Solved)

[Equal Stacks](https://www.hackerrank.com/challenges/equal-stacks)

Tóm tắt: Cho ba chồng đĩa (mỗi đĩa có độ dày khác nhau, biểu diễn bằng 1 số). Tìm độ cao chung lớn nhất của các chồng đĩa con từ dưới lên.

**public** **static** **int** equalStacks(List<Integer> h1, List<Integer> h2, List<Integer> h3) {

*// Write your code here*

**int** sum1 = 0, sum2 = 0, sum3 = 0;

**for** (**int** x: h1) sum1 += x;

**for** (**int** x: h2) sum2 += x;

**for** (**int** x: h3) sum3 += x;

**int** i = 0, j = 0, k = 0;

**while** (!(sum1 == sum2 && sum2 == sum3)) {

**if** (sum1 > sum2 || sum1 > sum3)         sum1 -= h1.get(i++);

**else** **if** (sum2 > sum1 || sum2 > sum3)    sum2 -= h2.get(j++);

**else** **if** (sum3 > sum2 || sum3 > sum1)    sum3 -= h3.get(k++);

        }

**return** sum1;

}

### Bài 6 (Solved)

[Simple Text Editor](https://www.hackerrank.com/challenges/simple-text-editor)

Tóm tắt: Cài 1 chương trình soạn thảo với 4 thao tác: append, delete, print và undo.

Để cài undo, có thể sử dụng stack.

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

    public static void main(String[] args) {

        /\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/

        Scanner scanner = new Scanner(System.in);

        int Q = scanner.nextInt();

        String text = "";

        Stack <String> oldVersion = new Stack<>();

        //Stack <Integer> request = new Stack<>();

        for (int i = 0; i < Q; i++) {

            int request = scanner.nextInt();

            switch (request) {

                //append

                case 1:

                {

                    //save the last version of the text before append

                    oldVersion.push(text);

                    //append the string using operator +=

                    String append = scanner.next();

                    text += append;

                    break;

                }

                //delete

                case 2:

                {

                    //save the last version of the text before delete

                    oldVersion.push(text);

                    //input the number of charater we want to delete

                    int k = scanner.nextInt();

                    //create a new string and save the remaining of the old string after delete

                    if (k == text.length()) text = "";

                    else {

                        int len = text.length();

                        String newText = text.substring(0, len - k);

                        text = newText;

                    }

                    break;

                }

                //print

                case 3:

                {

                    //input the order number character we want to print

                    int k = scanner.nextInt();

                    System.out.println(text.charAt(k - 1));

                    break;

                }

                //undo

                case 4:

                {

                    if (!oldVersion.isEmpty()) {

                        String oldText = oldVersion.pop();

                        text = oldText;

                    }

                    break;

                }

            }

        }

        scanner.close();

    }

}

### Bài 7 (Solved)

[Queue Using Two Stacks](https://www.hackerrank.com/challenges/queue-using-two-stacks)

Tóm tắt: Cài đặt queue với các phương thức enqueue, dequeue và print (in ra phần đầu ở đầu).

**Chú ý**: Có thể *không* cài đặt queue bằng 2 stack như trong đề bài. Thay vào đó tự cài đặt queue riêng theo cách của mình; Hay sử dụng Queue interface trong Java (link ví dụ ở trên).

(Tuy nhiên, người nào làm đúng yêu cầu của HackerRank là cài queue bằng 2 stack và qua (passed) tất các test thì vẫn tốt hơn.)

Cach1: Using Java Queue Interface

**import** java.io.\*;

**import** java.util.\*;

**public** **class** Solution {

**public** **static** **void** main(String[] args) {

*/\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*

        Scanner scanner = **new** Scanner(System.in);

        Queue<Integer> queue = **new** LinkedList<>();

**int** Q = scanner.nextInt();

**for** (**int** i = 0; i < Q; i++) {

**int** request = scanner.nextInt();

**switch** (request) {

**case** 1:

                {

**int** tmp = scanner.nextInt();

                    queue.add(tmp);

**break**;

                }

**case** 2:

                {

                    queue.poll();

**break**;

                }

**case** 3:

                {

                    System.out.println(queue.peek());

**break**;

                }

            }

        }

        scanner.close();

    }

}

Cach2: Use two stacks

**import** java.io.\*;

**import** java.util.\*;

**public** **class** Solution {

**public** **static** **void** main(String[] args) {

*/\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*

        Stack<Integer> stack1 = **new** Stack<>();

        Stack<Integer> stack2 = **new** Stack<>();

        Scanner scanner = **new** Scanner(System.in);

**int** N = scanner.nextInt();

**for** (**int** i = 0; i < N; i++) {

**int** request = scanner.nextInt();

**switch** (request) {

**case** 1:

                {

**int** tmp = scanner.nextInt();

                    stack1.push(tmp);

**if** (stack2.isEmpty()){

**while** (!stack1.isEmpty()) {

                            stack2.push(stack1.pop());

                        }

                    }

**break**;

                }

**case** 2:

                {

**if** (stack2.isEmpty()){

**while** (!stack1.isEmpty()) {

                            stack2.push(stack1.pop());

                        }

                    }

                    stack2.pop();

**break**;

                }

**case** 3:

                {

**if** (stack2.isEmpty()){

**while** (!stack1.isEmpty()) {

                            stack2.push(stack1.pop());

                        }

                    }

                    System.out.println(stack2.peek());

**break**;

                }

            }

        }

    }

}