Sửa Lab02

* Exercise 2
* Đệ quy

public static int ex2\_recur(int n) {

if (n == 1)

return 3;

return Math.pow(2, n) + Math.pow(n , 2) + ex2\_recur(n - 1);

}

* Stack

public static int ex2(int n) {

MyStack<Integer> stk = new MyStack<Integer>();

while (n > 1) {

stk.push(Math.pow(2, n) + Math.pow(n, 2));

n--;

}

stk.push(3);

int result = 0;

while (!stk.isEmpty()) {

result += stk.pop();

}

return result;

}

Stack<Character>

String str

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

char ch = str.charAt(i);

// hoặc char ch = str.subString(...);

// không nên dùng toCharArray()

// trong trường hợp phiên bản Java không hỗ trợ autoboxing, unboxing (chuyển primitive type sang wrapper classes và ngược lại) thì sử dụng method valueOf()

// Character temp = Character.valueOf(ch);

}

Cách xác định điều kiện dừng: trường hợp nào nhẩm được thì có thể là base case

Ví dụ: 2 mũ 0 = 1

// Tính 2 mũ n

public static int func(int n) {

if (n == 0) { // base cases

return 1;

}

return 2 \* func(n - 1); // recursive calls

}

Đệ quy duyệt mảng

Ví dụ: 2 0 1 4 6

Idea: danh sách các số là số hoặc số với danh sách các số

Exercise 3:

Cơ chế của đệ quy: cơ chế nhớ

Dùng vòng lặp for để làm thì i sẽ luôn được cập nhật

Dùng đệ quy không giữ đc i => Cần 1 biến(tham số) trong trường hợp này là d để lưu lại giá trị trước

Exercise 5:

Thao tác trực tiếp vào chỗ gọi đệ quy là trả về số nguyên được

Exercise 8:

Truyền next vào chỗ gọi đệ quy