Test STDIO

Câu 1:

1. Stack:

Ngăn xếp: Stack

- Abstract Data Type - ADT: Một ngăn xếp là cấu trúc dữ liệu trừu tượng, được sử dụng cho gần như tất cả các ngôn ngữ lập trình.

- Ngăn xếp dạng Array (mảng) hoặc Linked list (liên kết), Pointer (con trỏ), Struct (cấu trúc) và bị chặn 1 đầu, dữ liệu chỉ có thể vào từ 1 đầu và ra ở đó.

- Stack hoạt động theo dạng LIFO hay Last in first out: dữ liệu sẽ được PUSH (đưa vào) vào cuối cùng (trên cùng) của ngăn xếp và sẽ được POP (lấy ra/truy cập) đầu tiên, kích cỡ của ngăn xếp có thể cố định hoặc co dãn để nhận thêm dữ liệu PUSH.

Các hoạt động:

- isFull(): Kiểm tra xem đầy hay chưa.

- isEmpty(): Kiểm tra xem rỗng hay không.

- Peek(): Lấy dữ liệu cuối cùng (trên cùng) mà không xóa đi.

2. Queue:

Hàng đợi: Queue

- Là một ADT, Queue hoạt động theo kiểu FIFO hay First in first out: dữ liệu được enqueue (thêm vào) ở đầu hàng và được dequeue (đưa ra) đầu tiên trái ngược với Stack. Chúng ta có thể xóa dữ liệu ở vị trí được trỏ bởi con trỏ front và đưa dữ liệu vào với con trỏ rear.

- Nếu Stack là 1 hũ bánh, chỉ có thể đưa vào sau cùng và lấy ra đầu tiên thì queue như 1 đường hầm xe nào vào trước sẽ ra trước và cứ như vậy.

- Queue cũng có thể triển khai tương tự stack, được triển khai với sử dụng Array, Pointer, Struct.

Các hoạt động:

Tương tự như của Stack chúng ta có:

- isFull(): kiểm tra xem hàng đợi có đang đầy.

- isEmpty(): kiểm tra hàng đợi có đang rỗng.

- Peek(): Lấy đầu phần tử mà không xóa đi.

Câu 2:

1. Tham Trị:

Kiểu dữ liệu tham trị: Kiểu nguyên thủy

- byte

- short

- int

- long

- float

- double

- boolean

- char

Giá trị trước và sau khi truyền vào không thay thay đổi.

Truyền vào giá trị

2. Tham Chiếu: Reference Type

Kiểu dữ liệu tham chiếu

- Object

- Array

Giá trị trước và sau khi truyền vào có thể thay đổi.

Truyền vào địa chỉ ô nhớ.

Câu 3:

Khai báo 2 mảng học sinh:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Khai báo mảng 3.

Chiều dài mảng 3 = tổng chiều dài 2 mảng:

Ảnh có chứa văn bản, màu cam, đóng

Mô tả được tạo tự động

Tạo số count = 0 và tăng dần sau mỗi vòng lặp, đẩy giá trị mảng 1 và 2 vào mảng 3 với chỉ số bằng count:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Tiến hành sắp xếp bằng mothod sort có sẵn trong gói java.util và in ra mảng đã được sắp xếp:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Sort(): chia mảng ra thành nhiều mảng nhỏ và tiến hành sắp xếp các mảng nhỏ trên các luồng khác nhau và sau đó gom lại, mảng cuối cùng sẽ là mảng đã được sắp xếp.

Câu 4:

Main

Tạo thông báo nhập n từ bàn phím và đặt điều kiện 1 <= n <= 10^6.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Dùng Map để tạo key và value là giá trị nhập vào, ở đây K là Interger V là Long. Và sử dụng hashmap để ghi nhớ và in kết quả ra màn hình:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Đệ quy kết hợp quy hoạch động:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Nếu dữ liệu nhập vào khác NULL thì sẽ dùng get() để lấy dữ liệu nhớ từ vị trí = input (số n nhập vào từ bàn phím).

Nếu n bằng 1 hoặc 2 sẽ trả về 1 (Vì 1 <= n <= 10^6) nên sẽ bắt đầu từ vị trí số fibonancci thứ 1 và thứ 2 = 1.

Nếu không phải cả 2 trường hợp trên, sẽ được đưa xuống để thực hiện fib = (fib-1) + (fib-2) chỗ này sẽ dùng long để lưu lại giá trị và return nhằm thực hiện gọi lại và in kết quả ra màn hình ở main.