

# BÀI TẬP THỰC HÀNH

## MÔN CTDL VÀ GT

### Bài 1

1. Cài đặt hàm nhập mảng 1D số nguyên, n phần tử.
2. Cài đặt hàm phát sinh n phần tử ngẫu nhiên cho mảng 1D số nguyên.
3. Cài đặt hàm phát sinh n phần tử ngẫu nhiên có giá trị tăng/giảm dần cho mảng 1D số nguyên.
4. Cài đặt hàm xuất các phần tử trong mảng 1D.
5. Cài đặt hàm tìm kiếm tuyến tính (LinearSearch), với quy ước tên hàm:

**public static int LinearSearch1(int a[], int x);**

Input: a, x

Output: vị trí của phần tử x trong mảng a

**public static void LinearSearch2(int a[], int x);**

Input: a, x

Output: vị trí của phần tử x trong mảng a, số phép so sánh

**public static void LinearSearch3(int a[], int x);**

Input: a, x

Output: tất cả vị trí của phần tử x xuất hiện trong mảng a, số phép so sánh

6. Cài đặt hàm tìm kiếm nhị phân (BinarySearch) với quy ước tên hàm:

**public static int BinarySearch1(int a[], int x);**

Input: a, x

Output: vị trí của phần tử x trong mảng a

**public static int BinarySearch2(int a[], int x);**

Input: a, x

Output: vị trí của phần tử x trong mảng a, số phép so sánh

**public static int BinarySearch3(int a[], int x);**

Input: a, x

Output: vị trí của phần tử x trong mảng a, số phép so sánh, số phép gán chỉ số

7. Tính thời gian chạy của 2 algorithms: Linear and Binary Search (tính theo: nano and mili second)

Gợi ý:

**A. Phát sinh số ngẫu nhiên:**

(1) Khai báo thư viện

```
import java.util.Random;
```

(2) Sử dụng dụng hàm tạo số ngẫu nhiên.

```
Random rd = new Random();
```

```
System.out.println(rd.nextInt(1000));
```

(3) Có thể sử dụng câu lệnh for để tạo ra nhiều số ngẫu nhiên.

**B. Tính thời gian thực hiện:** chọn cách tính bằng nano giây hoặc mili giây.

```
long start, start1, end, end1, mgiay, ngay;  
start = System.nanoTime(); //start lấy thời gian theo nanosecond  
start1 = System.currentTimeMillis(); //start1 lấy thời gian theo millisecond  
  
...đoạn code cần tính thời gian ...  
  
end = System.nanoTime(); //end lấy thời gian theo nanosecond  
ngay = (end - start);  
System.out.println("\nGiải thuật thực hiện trong " + ngay + " Nano  
Second");  
  
end1 = System.currentTimeMillis(); //end1 lấy thời gian theo millisecond  
mgiay = (end1 - start1);  
System.out.println("\nGiải thuật thực hiện trong "+ ngay +" Mili Second ");
```

*Chú ý:* Nên tạo menu cho bài 1

**Bài 2:** Viết chương trình khai báo Stack (dùng mảng) chứa số nguyên, với menu thực hiện như sau:

1. Push vào Stack
2. Pop ra khỏi Stack
3. Xem phần tử trên đỉnh Stack
4. Kiểm tra Stack rỗng
5. Kiểm tra Stack đầy
6. Số phần tử trong Stack
7. Duyệt Stack
8. Thoát
9. Nhập các phần tử vào Stack
10. Chuyển số thập phân sang nhị phân lưu vào Stack.

Nhấn một số để chọn:

**Bài 3:** Viết chương trình khai báo Queue (dùng mảng) chứa số nguyên, với menu thực hiện như sau:

1. Khai báo Queue chứa số nguyên
2. Hàm tạo Queue
3. Hàm kiểm tra Queue đầy
4. Hàm kiểm tra Queue rỗng
5. Hàm thêm 1 phần tử vào Queue
6. Lấy 1 phần tử khỏi Queue
7. Hàm lấy phần tử đầu Queue (front)
8. Hàm lấy phần tử cuối Queue (rear)
9. Hàm main gọi thực hiện các hàm trên

Nhấn một số để chọn:

#### **Bài 4: Linked Lists**

Cài đặt các hàm sau trên DSLK đơn số nguyên:

1. Chèn phần tử ở đầu DSLK

2. Chèn phần tử ở đuôi DSLK

3. Chèn phần tử tại vị trí (2,3,4,...)

4. Xóa phần tử tại vị trí bất kỳ

5. Tính kích thước DSLK

6. Xuất các node của DSLK

7. Chèn một nút tại vị trí Pos trong DSLK

8. Xóa một nút tại vị trí Pos trong DSLK

9. Đếm số lượng node trong DSLK

10. Tìm node có giá trị lớn nhất

11. Tìm node có giá trị x

12. In những node có giá trị chẵn

13. Tính giá trị trung bình cộng các node lẻ

14. Tìm node có giá trị âm nhỏ nhất

15. Tìm node có giá trị dương lớn nhất

16. Sắp xếp danh sách tăng dần

**Bài 5:** Sử dụng DSLK đơn để quản lý thông tin SV gồm: Masv, Hoten, Ngaysinh, Gioitinh, DTB,...

1. Xây dựng cấu trúc lớp SINHVIEN
2. Xây dựng cấu trúc lớp Nodes

3. Xây dựng cấu trúc LinkedLists
4. Nhập danh sách sinh viên
5. Xuất danh sách sinh viên
6. Tìm sinh viên theo mã số sinh viên
7. Tìm sinh viên có DTB lớn nhất
8. Tìm sinh viên có DTB nhỏ nhất
9. Sắp xếp danh sách sinh viên giảm dần theo DTB