

BÀI THỰC HÀNH VỀ DANH SÁCH LIÊN KẾT

Bài tập 1

Sử dụng cấu trúc danh sách liên kết thực hiện các yêu cầu sau:

(Với kiểu các phần tử có kiểu phần tử là số nguyên)

- Khai báo cấu trúc dữ liệu của bài toán
- Nhập danh sách
- In ra danh sách vừa nhập
- Nhập giá trị X (có kiểu trùng kiểu phần tử). Hãy cho biết X có trong ds hay không? Nếu có x xuất hiện bao nhiêu lần và xuất hiện ở những vị trí nào?
- Sắp xếp danh sách tăng dần.

Bài tập 2

Sử dụng cấu trúc danh sách liên kết thực hiện các yêu cầu sau: (Với kiểu các phần tử có kiểu phần tử là số nguyên)

- Khai báo cấu trúc dữ liệu của bài toán
- Nhập lần lượt 2 danh sách.
- In ra từng danh sách vừa nhập.
- Sắp xếp từng danh sách tăng dần.
- Trộn danh sách đã được sắp xếp tăng dần thành một danh sách cũng tăng dần.

Bài tập 3

Sử dụng cấu trúc danh sách liên kết thực hiện các yêu cầu sau:

Thông tin của hình Ellipse gồm: x , y , $R1$, $R2$, chu vi và diện tích (trong đó x là hoành độ, y là tung độ của tâm và $R1$, $R2$ là hai bán kính). Hãy khai báo cấu trúc để quản lý danh sách các hình Ellipse như mô tả. Viết chương trình thực hiện các yêu cầu sau (tất cả các yêu cầu phải viết hàm)

- Khai báo cấu trúc dữ liệu của bài toán
- Nhập thông tin cho danh sách hình Ellipse (không nhập chu vi và diện tích).
- Tính chu vi và diện tích của các hình Ellipse vừa nhập.
- Xuất thông tin các hình Ellipse vừa nhập (x , y , $R1$, $R2$, chu vi và diện tích).
- Hãy cho biết hình các Ellipse vừa nhập có tâm nằm ở góc toạ độ hay không?

(chu vi $cv = 2 * \pi * \sqrt{(R_1 * R_1 + R_2 * R_2)}/2$ và diện tích $dt = \pi * R1 * R2$ với $\pi = 3.14$)

Bài tập 4

Sử dụng cấu trúc danh sách liên kết thực hiện các yêu cầu sau:

Thông tin của chữ nhật gồm: chiều dài, chiều rộng, chu vi và diện tích. Hãy khai báo cấu trúc để quản lý các hình chữ nhật như mô tả. Viết chương trình thực hiện các yêu cầu sau (tất cả các yêu cầu phải viết hàm)

- Khai báo cấu trúc dữ liệu của bài toán
- Nhập thông tin cho các hình chữ nhật trong danh sách (không nhập chu vi và diện tích).
- Tính chu vi và diện tích các hình chữ nhật trong danh sách.
- Xuất thông tin các hình chữ nhật (chiều dài, chiều rộng, chu vi và diện tích)
- Liệt kê những hình chữ nhật có diện tích bằng chu vi.
- Sắp xếp danh sách tăng dần theo diện tích.

Biết rằng, chu vi $cv = (d+r)*2$ và diện tích $dt = d*r$ với d, r là độ dài 2 cạnh.

Bài tập 5

Sử dụng cấu trúc danh sách liên kết để quản lý thông tin con người gồm: số chứng minh nhân dân, họ tên, tuổi. Thực hiện các yêu cầu sau (các yêu cầu phải viết ở dạng hàm):

- Khai báo cấu trúc dữ liệu của bài toán
- Nhập danh sách con người
- In thông tin danh sách con người đã nhập.
- Cho biết có bao nhiêu người tuổi từ 18 trở lên.
- Sắp xếp danh sách giảm dần theo tuổi.

Bài tập 6

Sử dụng cấu trúc danh sách liên kết, cài đặt các hàm thực hiện các yêu cầu sau: (Với kiểu các phần tử có kiểu phần tử là kiểu sinh viên có các trường mssv, họ tên, điểm thường xuyên, điểm thi, điểm môn học, xếp loại)

- Khai báo cấu trúc dữ liệu của bài toán
- Nhập danh sách (không nhập điểm môn học).
- Tính điểm môn học cho các sinh viên biết rằng, điểm môn học của mỗi sinh viên sẽ được tính như sau: $\text{điểm môn học} = (\text{điểm thường xuyên} + \text{điểm thi})/2$
- Thực hiện xếp loại cho các sinh viên, biết rằng:
 - + Nếu điểm môn học ≥ 8.5 thì xếp loại “Giỏi”
 - + Nếu $7.0 \leq \text{điểm môn học} < 8.5$ thì xếp loại “Khá”
 - + Nếu $5.5 \leq \text{điểm môn học} < 7.0$ thì xếp loại “Trung bình khá”
 - + Nếu $4.0 \leq \text{điểm môn học} < 5.5$ thì xếp loại “Trung bình”
 - + Các trường hợp còn lại xếp loại “Yếu”
- In ra danh sách sinh viên.
- Cho biết số lượng sinh viên có điểm từ 5.0 trở lên.
- Sắp xếp danh sách tăng dần theo điểm môn học.
- Sắp xếp danh sách tăng dần theo mssv.

Hướng dẫn khai báo

```
typedef ... ElementType; //kiểu của phần tử trong danh sách
```

```
typedef struct Node
```

```
{
```

```
    ElementType Element;    //Chứa nội dung của phần tử
```

```
    Node *Next;            //con trỏ chỉ đến phần tử kế tiếp
```

```
};
```

```
typedef Node *PtrToNode;
```

```
typedef PtrToNode Position;    //kiểu vị trí
```

```
typedef PtrToNode List;        //Danh sách
```