

BÀI TẬP CHƯƠNG 5 – CẤU TRÚC DỮ LIỆU KIỂU LIST

5.1. Viết chương trình có sử dụng hàm thực hiện các yêu cầu sau:

- Hàm **Input()**: nhập một số nguyên **n** và **n** số nguyên lưu vào List **L**, và một số nguyên **x**;
- Hàm **FirstAndLast(L)** trả về và in lên màn hình List mới chỉ gồm **phần tử đầu tiên và cuối cùng của L**;
- Hàm **Search(L,x)**: xác định **x** có nằm trong **L** hay không. Trả về **True** nếu tìm thấy, còn lại trả về **False**.

```
n=4
3
5
3
7
x=5
[3, 7]
True
```

5.2. Viết chương trình có sử dụng hàm thực hiện các yêu cầu sau:

- Hàm **Input()**: nhập một số nguyên **n** và **n** số nguyên lưu trữ vào một List **L**;
- Hàm **Search(L)**: Tìm và trả về số nhỏ nhất và lớn nhất trong List **L**;
- Hàm **Output(max, min)**: In lên màn hình số lớn nhất **max** và bé nhất **min**;

Lưu ý: không được sử dụng hàm chuẩn **max** và **min** trong python.

```
n=5
10
15
3
2
7
15 2
```

5.3. Nhập từ bàn phím một số nguyên **n** (**n>0**), và **n** số nguyên lưu trữ vào một List.

In lên màn hình:

- **Số lượng** các số nguyên **DƯƠNG**
- **Trung bình cộng** của các số nguyên **chẵn** được lưu trữ trong List trên.

```
n=5
6
-2
-1
2
7
SND=3
TBC=2
```

```
n=4
3
-5
1
-7
SND=2
TBC=0
```

5.4. Nhập từ bàn phím một số nguyên **n** (**n>0**) và **n** số nguyên lưu vào List **A**:

- Hãy **đảo ngược giá trị của các phần tử** trong List **A** và **lưu vào List B**. In giá trị các phần tử trong List **B** sau khi thực hiện đảo;
- Sắp xếp và in lên màn hình List **B** sau khi được sắp xếp tăng dần;

Gợi ý: Sử dụng cấu trúc lặp với $i=(n-1)..0$ để duyệt List **A**. Không được sử dụng phương thức **reverse** và **sort**.

```
n=5
3
1
4
2
5
5 2 4 1 3
1 2 3 4 5
```

```
n=4
2
4
3
5
5 3 4 2
2 3 4 5
```

5.5. Viết chương trình **nhập vào một số nguyên n ($n > 0$), và n số nguyên** lưu trữ vào List A. In lên màn hình: tổng giá trị của các phần tử ở vị trí có thứ tự chẵn trong List A (*biết rằng phần tử thứ 1 có số chỉ mục là 0 sẽ có thứ tự là 1, ...*).

```
n=5
8
10
12
4
7
Tong=14
```

```
n=6
1
23
15
30
12
49
Tong=102
```

Gợi ý: Để xác định phần tử ở vị trí chẵn, sử dụng cấu trúc lặp $i=0..(n-1)$, nếu i lẻ tức là phần tử đó ở vị trí chẵn.

5.6. Viết chương trình nhập vào từ bàn phím **10 số nguyên** và lưu vào một List A. Hãy **hoán đổi giá trị của 2 phần tử nằm cạnh nhau** (theo từng đôi) trong List. Và in lên màn hình List kết quả sau khi xử lý.

```
5
7
2
6
3
7
8
9
1
2
7 5 6 2 7 3 9 8 2 1
```

```
6
3
5
8
7
9
1
2
5
3
3 6 8 5 9 7 2 1 3 5
```

Gợi ý:

- Để hoán đổi giá trị theo cặp phần tử, sử dụng thêm List B để lưu trữ tập kết quả
- Cho vòng lặp $i=0..(n-2)$ $step=2$
 - o $B[i]=A[i+1]$
 - o $B[i+1]=A[i]$

5.7. Viết chương trình nhập vào một số nguyên n và n số nguyên lưu vào List L. Thực hiện loại bỏ những phần tử có giá trị trùng nhau và lưu tập mới vào List M. In lên màn hình các phần tử trong M.

n=5
2
4
2
5
4
2 4 5

5.8. Nhập từ bàn phím hai số nguyên m và n , và giá trị của các phần tử trong hai ma trận A và B có cấp là $m \times n$. Tính và in lên màn hình ma trận C là tổng của A và B.

Biết rằng, tổng hai ma trận được tính như sau:

$$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 1+5 \\ 1+7 & 0+5 & 0+0 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 8 & 5 & 0 \end{bmatrix}$$

m=2
n=3
A:
1
3
1
1
0
0
B:
0
0
5
7
5
0
C:
1 3 6
8 5 0

5.9. Viết chương trình nhập vào hai số nguyên n và m . Nhập n số nguyên lưu trữ vào List L1, nhập m số nguyên lưu trữ vào List L2. Thực hiện xử lý và cho ra một List L3 chỉ chứa những phần tử chung giữa L1 và L2 (không được trùng nhau).

n=3
m=2
L1:
3
2
5
L2:
2
3
L3: 3 2

n=2
m=2
L1:
3
5
L2:
6
1
L3: