
PHẦN THỰC HÀNH MẢNG 1 CHIỀU

Đề nghị sinh viên phải viết thủ tục hàm cho các chương trình sau:

Bài tập 01: Viết chương trình nhập xuất mảng 1 chiều (List).

Hàm nhập mảng 1 chiều

```
def input_data(n):  
    arr=[]  
    for i in range(n):  
        x = int(input(f"Input item[{i}]: "))  
        arr.append()  
    return arr
```

Code ràng buộc điều kiện dữ liệu đầu vào. Sinh viên có thể tham khảo

```
def input_list(n):  
    arr = []  
    i = 0  
    while True:  
        x = input(f"Input item[{i}]: ")  
        try:  
            arr.append(int(x))  
            i += 1  
            if i >= n:  
                break  
        except:  
            print("Chưa nhập đúng số nguyên. Nhập lại!")  
    return arr
```

a. Tính tổng của mảng 1 chiều

```
def tinh_tong(arr):  
    Tạo biến tong = 0  
    Duyệt từng vị trí trong mảng  
    Cộng lần lượt các giá trị lại  
    return tong
```

b. Tính tích mảng 1 chiều

```
def tinh_tich(arr):  
    Tạo biến tích = 1  
    Duyệt từng vị trí trong mảng  
        Nhân lần lượt các giá trị lại  
    return tích
```

c. Tìm giá trị lớn nhất

```
def tim_max(arr):  
    Tạo biến max = phần tử đầu tiên của list  
    Duyệt từng vị trí trong mảng  
        So sánh các giá trị với phần tử max  
    return max
```

d. Tìm giá trị nhỏ nhất

```
def tim_min(arr):  
    Tạo biến min = phần tử đầu tiên của list  
    Duyệt từng vị trí trong mảng  
        So sánh các giá trị với phần tử min  
    return min
```

e. Tìm các số lẻ trong mảng

```
def so_le(arr):  
    Tạo 1 list rỗng result = []  
    Duyệt từng vị trí trong mảng chính  
    Kiểm tra điều kiện nếu số nào chia hết cho 2 khác 0. Thì thêm phần tử vào  
    return result
```

f. Tìm các số chẵn trong mảng

def so_chan(arr):

Tạo 1 list rỗng result = []

Duyệt từng vị trí trong mảng chính

Kiểm tra điều kiện nếu số nào chia hết cho 2 bằng 0. Thì thêm phần tử vào

return result

g. Tính tổng các số nguyên tố trong mảng

def kiem_tra_snt(arr):

Số nguyên tố là số chia hết cho 1 và chính nó.

Viết hàm kiểm tra số nguyên tố trước.

Sau đó viết hàm tính tổng các số nguyên tố để cộng lại các giá trị là snt lại

def tinh_tong_snt(arr):

Khai báo biến tong = 0

return tong

Bài tập 02: Bài tập làm thêm

Mô tả:

Bạn được cung cấp một danh sách N bài hát đã từng được nghe của một người dùng ứng dụng Zingmp3.

Bạn cần xây dựng từ danh sách trên một chuỗi dài nhất các bài hát liên tiếp trong đó không có bài hát nào được lặp lại

Input: A – Mảng chứa id bài hát

Output: Độ dài cần tìm.

Example: Input: A = {1,2,1,3,2,7,4,2,5,5}

Output: 6 (vì chuỗi hình thành được là: {1,2,3,7,4,5})

PHẦN THỰC HÀNH MẢNG 2 CHIỀU

Nhập mảng 2 chiều trong Python:

```
def input_list_2d(m, n): # m - số hàng, n - số cột
    result = []
    j = 0
    while True:
        result_row = []
        i = 0
        while True:
            v = input(f"Nhập phần tử {j},{i}: ")
            try:
                result_row.append(int(v))
                i += 1
                if i >= n:
                    break
            except:
                print("Chưa nhập đúng số nguyên. Nhập lại!")

        result.append(result_row)
        j += 1
        if j >= m:
            break

    return result
```

Bài 00: Viết chương trình sinh list mới bằng cách lấy ngẫu nhiên 5 phần tử từ list gốc.

```
import random
def them_ngau_nhien_ele(arr):
    Tạo một list rỗng result = []
    Duyệt vòng lặp 5 lần
    Mỗi lần lặp thêm 1 phần tử lấy random từ list chính
    random.choice(arr)
    return result
```

Bài 03: Viết chương trình tạo ra list mới bằng cách ghép một chuỗi s vào các phần tử list cũ.

Bài 04: Viết chương trình chia một list thành 2 phần với độ dài phần đầu được nhập vào từ bàn phím.

Bài 05: Viết chương trình in ra phần tử có số lần xuất hiện nhiều nhất trong một list.

Nếu có nhiều phần tử có cùng số lần xuất hiện nhiều nhất thì in ra 1 trong số chúng.

tạo 1 hàm đếm số lần của 1 giá trị bất kỳ có trong mảng
duyệt lại mảng và lấy giá trị max từ hàm đếm số lần ở trên
hàm main để gọi hàm và trả kết quả

Bài 06: Viết chương trình đếm các chuỗi trong một list thỏa mãn:

+ Độ dài từ 2 trở lên

+ Ký tự đầu tiên và cuối cùng của chuỗi đó giống nhau

Tạo 1 biến count = 0
Duyệt các phần tử của List
Ví dụ input a = ['hung','tan','ngon','aa']
Output count = 1
Kiểm tra điều kiện độ dài lớn hơn 2 và ký tự đầu && ký tự cuối giống nhau

Bài 07: Viết chương trình kiểm tra 2 list có phần tử chung hay không.

Tạo 2 list
Tạo 1 biến đếm
Duyệt từng phần trong list 1
Kiểm tra nếu phần tử list 1 có trong list 2 thì tăng biến đếm lên
Check biến đếm.

Bài 08: Cho list các số nguyên dương A.

Xây dựng chương trình đếm số lượng tập gồm 2 phần tử $A[i]$ và $A[j]$ thỏa mãn $A[i] > A[j]$ và $i < j$.

Nhập một mảng list các phần tử số nguyên
Tạo 1 biến count = 0
Duyệt vòng lặp j trong khoảng (1,len(arr))
 Duyệt vòng lặp i đến j
 Kiểm tra nếu $arr[i] > a[j]$ thì tăng biến count lên
Xuất count

Bài 09: Viết chương trình tính tích của 2 ma trận vuông cấp 3 (Gợi ý: dùng list chứa list).

Nhập 2 ma trận:

input

mx_1 = [[1, 2, 3],
 [4, 5, 6],
 [1, 0, 0]]

mx_2 = [[2, 1, 0],
 [9, 0, 5],
 [0, 3, 7]]

Output: [[20, 10, 31], [53, 22, 67], [2, 1, 0]]

result = []

Duyệt theo dòng

 Duyệt theo cột

 Duyệt lần lượt 3 phần tử

 Nhân các giá trị lại

Trò chơi đoán số.

```
import random

def binary_guess():
    if left <= right:
        return (left + right) // 2
    else:
        return None

def hint(n, msg):
    global left, right

    if msg == "less":
        left = n + 1
    else:
        right = n - 1

max = 1000
print("Trò chơi đoán số!")
print(f"Bạn đoán một con số nguyên trong phạm vi 1-{max}.")

left, right = 1, max
secret = random.randint(1, max)
count = 0
while True:
    count += 1
    input("Press Enter to guess.")
    n = binary_guess()
    print(f"Đoán lần {count} số {n}")

    if n < secret:
        print("Số bạn đoán nhỏ quá!")
        hint(n, "less")
    elif n > secret:
        print("Số bạn đoán lớn quá!")
        hint(n, "greater")
    else:
        print(f"Bạn đoán đúng số {secret} sau {count} lần.")
        break
```