



PYTHON PROGRAMING

Dr. Dinh Chi Hieu



Kiểm tra giữa kỳ

Lớp 05 / 06

- Quy tắc đặt tên file:

- File mã nguồn: `HovaTen_MaSinhVien_Bai{1, 2, ...}.py/ipynb`
- File kết quả: `HovaTen_MaSinhVien_Bai{1, 2, ...}_ketqua.png /jpg`
- Ví dụ:
`BuiCongSon_B22DCCN686_Bai1.py`
`BuiCongSon_B22DCCN686_Bai1_ketqua_1.jpg`
`BuiCongSon_B22DCCN686_Bai1_ketqua_2.jpg`

- Quy tắc nộp bài:

- Các file mã nguồn + file ảnh chụp kết quả được nén (zip / rar) và gửi theo file đính kèm email với tiêu đề: `HovaTen_MaSinhVien_KiemTraGiuaKy`.
- Gửi tới: hieudc@ptit.edu.vn & hieusdc@gmail.com
- Thời gian: trước 23h59 ngày 12/10/2024

Bài 1

Tìm hình vuông lớn nhất

- Cho ma trận nhị phân $M \times N$ gồm các phần tử 0 hoặc 1.
 - Tìm hình vuông lớn nhất chỉ gồm toàn phần tử 1. Tính diện tích.
 - Tìm hình chữ nhật lớn nhất gồm toàn phần tử 1. Tính diện tích.

Bài 1

Tìm hình vuông lớn nhất

1	0	1	0	0	1	0	1
1	0	1	1	1	1	0	0
1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1
0	1	1	0	1	1	1	1

```
INPUT = [[1, 0, 1, 0, 0, 1, 0, 1],  
         [1, 0, 1, 1, 1, 1, 0, 1],  
         [1, 1, 1, 1, 1, 1, 1, 1],  
         [1, 0, 1, 1, 1, 1, 1, 1],  
         [0, 1, 1, 0, 1, 1, 1, 1]]
```

```
OUTPUT_1 = 9  
          [[3, 2], [5, 4]]  
          [[5, 3], [7, 5]]  
          [[6, 3], [8, 5]]
```

```
OUTPUT_2 = 12  
          [[3, 2], [6, 4]]  
          [[5, 3], [8, 5]]
```

Bài 2

Chuỗi không giảm

- Cho mảng A với N phần tử thực
- Tìm K phần tử phải loại bỏ để A trở thành mảng gồm các phần tử không giảm?
- Có bao nhiêu Phương án? Kết quả của từng Phương án?
- K nhỏ nhất bằng bao nhiêu? Kết quả?

INPUT = [3, 4, 2, 4.2, 3.23, 5]

OUTPUT = K = 2

[3, 4, 4.2, 5]

Bài 3

Tìm hoán vị

- Cho các phần tử là nguyên âm: 'e', 'u', 'i', 'o', 'a'
 - Phần tử tiếp theo 'a', nếu có, phải là 'e'.
 - Phần tử tiếp theo 'e', nếu có, phải là 'a' hoặc 'i'
 - Phần tử tiếp theo 'i', nếu có, không được là 'i'
 - Phần tử tiếp theo 'o', nếu có, phải là 'u' hoặc 'i'
 - Phần tử tiếp theo 'u', nếu có, phải là 'a'
- Có bao nhiêu chuỗi chiều dài **N** được tạo nên từ luật trên?

Bài 4

Dictionary

- Nhập một đoạn văn bản
 - Phân tách thành các từ riêng biệt
 - Xây dựng biến kiểu từ điển:

```
Word_freq = {  
    word : # of occurrences  
}
```

- Sắp xếp Word_freq theo thứ tự giảm dần của số lần xuất hiện của từ
- Vẽ đồ thị thể hiện số lần xuất hiện (histogram)

