

## Nội dung thi

- 2 đề khác nhau
- Mỗi đề 5 câu, làm trong 60 phút.
- Code. (thực hành)
- Đề bài và nộp thông qua app client (giống lúc kiểm tra).
- IDE: Visual studio code.
- Làm và nộp bằng file .py

## Giới hạn:

- Bài 1: Biến, String, Logic cơ bản: rẽ nhánh, vòng lặp
- Bài 2: Thao tác với danh sách (List)
- Bài 3: Thao tác với từ điển (Dictionary)
- Bài 4: Class, Object, List of Object ( sorted() và lambda function)
  - Đọc
    - Danh sách các sinh viên
    - Danh sách các công nhân
    - Danh sách các sản phẩm.
  - Sắp xếp danh sách theo một tiêu chí nào đó, rồi in ra kết quả.
    - Sắp xếp theo tổng điểm
    - Sắp xếp công nhân theo thâm niên (số năm làm việc)
    - Sắp xếp theo các sản phẩm được mua nhiều nhất
- Bài 5: Đọc file Json hoặc CSV và tính toán rồi in ra kết quả.
  - JSON: nested list, nested dictionary
  - CSV: comma separated value

## Ví dụ bài 1:

Yêu cầu tính giá vé vào công viên của một người khi biết số tuổi của người đó. Điều kiện là:

<4 tuổi: miễn phí

4-18 tuổi: 10 nghìn

18-60 tuổi: 20 nghìn

>60 tuổi: 15 nghìn

Ví dụ:

Input:

- Dòng đầu ghi số bộ test

- Mỗi dòng tiếp theo, ghi một bộ test là số tuổi của người đó

Output:

in ra số tiền vé người đó phải thanh toán, nếu miễn phí thì in ra FREE

Input	Ouput
4	10
7	20
25	15
66	FREE
3	

Bài mẫu:

```
count=int(input())
```

```
for _ in range(count):
```

```
    age=int(input())
```

```
    if age<4:
```

```
        print('FREE')
```

```
    elif age<18:
```

```
        print(10)
```

```

elif age<60:
    print(20)
else:
    print(15)

```

Bài 2:

## XỬ LÝ CHUỖI 1

Viết chương trình xử lý chuỗi theo các bước như sau:

- Tách các từ trong chuỗi ban đầu
- Loại bỏ các ký tự không thuộc chữ hoa hoặc chữ thường
- Đếm và ghi ra độ dài của mỗi từ đó theo đúng thứ tự của các từ trong chuỗi. (nếu độ dài của từ đó bằng 0 sẽ không in ra gì)

Input

- Dòng đầu là số bộ test
- Mỗi dòng tiếp theo, mỗi dòng là một chuỗi chưa được xử lý

Output

- Ghi ra kết quả xử lý chuỗi theo các bước kể trên.

Ví dụ

Input	Output
4	4 3 2 3
Con1g Hoa9 Xa# Hoi	3 4
Hoc vie?n 101	
123456789	1
123 a	

- Trong đó eng là một từ tiếng Anh (không có dấu cách), viet là nghĩa tiếng Việt (có thể gồm dấu và dấu cách, nhưng không có xuống dòng).
- Nếu từ tiếng Anh đã tồn tại, cập nhật nghĩa mới (ghi đè).

## 2. Tra cứu từ:

- Cú pháp: find eng
- Trả về nghĩa tiếng Việt của từ eng.
- Nếu từ không có trong từ điển, in ra "Not found".

## Đầu vào

- Dòng đầu tiên: số nguyên  $Q$  ( $1 \leq Q \leq 2 \times 10^5$ ) — số lượng thao tác.
- Tiếp theo  $Q$  dòng, mỗi dòng mô tả một thao tác theo định dạng ở trên.

## Đầu ra

- Với mỗi thao tác find, in ra một dòng là kết quả tương ứng.

## Ví dụ:

Input	Output
7	xin chào
add hello xin chào	Not found
add cat con mèo	chào bạn
find hello	con mèo
find dog	
add hello chào bạn	
find hello	
find cat	

```
count=int(input())
```

```
dict1={}
```

Ghi chú:

Ký tự viết thường: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

Ký tự viết hoa: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

## Code mẫu bài 2

```
n = int(input())
for _ in range(n):
    s = input().split() #list string
    result = []

    for i in s:
        clean_i = ''.join(c for c in i if c.isalpha())# filter
no character

        if len(clean_i)>0:
            result.append(str(len(clean_i)))

    print(" ".join(result))#3 4 2
```

## Bài 3: Dictionary

### Từ điển Song ngữ

Bạn được giao nhiệm vụ xây dựng một **từ điển song ngữ** Anh – Việt.

Hệ thống cần xử lý hai loại thao tác:

#### 1. **Thêm từ mới** vào từ điển:

- Cú pháp: add eng viet

```

for _ in range(count):
    s=input().split()
    if s[0]=='add':
        key=s[1]
        list_value= [s[i] for i in range(2, len(s))]
        value=" ".join(list_value)
        dict1[key]=value
    elif s[0]=='find':
        print(dict1.get(s[1], 'Not found'))

```

Bài 4. Ví dụ:

### Quản lý danh sách sinh viên

Có một danh sách sinh viên, mỗi sinh viên có thông tin:

- **Mã sinh viên** (chuỗi, không trùng)
- **Họ tên** (chuỗi, có thể chứa khoảng trắng)
- **Điểm 3 môn**: Toán, Lý, Hóa (số thực)
- **Khóa** (năm nhập học, số nguyên)
- **Lớp** (chuỗi, ví dụ "CNTT1", "KT2"...)

Yêu cầu:

1. Với mỗi sinh viên, tính **tổng điểm** (Toán + Lý + Hóa).
2. Sắp xếp danh sách sinh viên **giảm dần** theo tổng điểm.
  - Nếu tổng điểm bằng nhau, sắp xếp **tăng dần** theo mã sinh viên.
3. In ra danh sách sinh viên đã sắp xếp theo định dạng:

ID Name Total Class Course

- **ID**: mã sinh viên
- **Name**: họ tên

- **Total:** tổng điểm (hiển thị với **2 chữ số thập phân**)
- **Class:** tên lớp
- **Course:** khóa

### Input

- Dòng đầu tiên: số nguyên  $T$  — số test case ( $1 \leq T \leq 10$ ).
- Mỗi test case:
  - Dòng đầu tiên: số nguyên  $N$  — số sinh viên ( $1 \leq N \leq 100$ ).
  - Tiếp theo  $N$  dòng, mỗi dòng chứa:

ID Name Toan Ly Hoa Course Class
----------------------------------

- ID: chuỗi không dấu cách, độ dài  $\leq 10$
- Name: chuỗi, có thể chứa khoảng trắng (đọc đến trước điểm Toán)
- Toan, Ly, Hoa: số thực,  $0 \leq \text{điểm} \leq 10$
- Course: số nguyên (năm)
- Class: chuỗi không dấu cách, độ dài  $\leq 10$

### Output

Với mỗi test case, in:

Case #x: <ID> <Name> <Total> <Class> <Course> ...
---

Trong đó  $x$  là số thứ tự test case (bắt đầu từ 1).

### Ví dụ:

Input	Output
1	Case #1:
3	SV001 Nguyen Van A 24.50 CNTT1 2022
SV001 Nguyen Van A 8.5 7.0 9.0 2022 CNTT1	SV002 Tran Thi B 24.50 CNTT1 2021

SV002 Tran Thi B 9.0 8.0 7.5 2021 CNTT1	SV003 Le Van C 24.50 CNTT2 2022
SV003 Le Van C 8.5 7.0 9.0 2022 CNTT2	

Bài mẫu:

```
class Student:
    def __init__(self, id, name, toan, ly, hoa, course, lop):
        self.id = id
        self.name = name
        self.toan = float(toan)
        self.ly = float(ly)
        self.hoa = float(hoa)
        self.course = int(course)
        self.lop = lop
        self.total = self.toan + self.ly + self.hoa
    def __str__(self):
        return f"{self.id} {self.name} {self.total:.2f} {self.lop} {self.course}"

def main():
    T = int(input().strip())
    for t in range(1, T + 1):
        n = int(input().strip())
        students = []
        for _ in range(n):
            parts = input().strip().split()
            id = parts[0]
```



```

        toan = parts[-5]
        ly = parts[-4]
        hoa = parts[-3]
        course = parts[-2]
        lop = parts[-1]
        name = " ".join(parts[1:-5])

        students.append(Student(id, name, toan, ly, hoa,
course, lop))

    students.sort(key=lambda x: (-x.total, x.id))
    print(f"Case #{t}:")
    for s in students:
        print(s)

if __name__ == "__main__":
    main()

```

## Bài 5.

Tips là dữ liệu về tiền bo (tips) cho nhà hàng bao gồm danh sách các từ điển cho mỗi bàn bao gồm các trường:

```

{
  "tips": [
    {
      "total_bill": "16.99",
      "tip": "1.01",
      "sex": "Female",
      "smoker": "No",
      "day": "Sun",
      "time": "Dinner",
      "size": "2"
    },
    {
      "total_bill": "10.34",
      "tip": "1.66",
      "sex": "Male",
      "smoker": "No",

```

total\_bill: tổng số chi

tip: tiền bo

sex: giới tính (Male hoặc Female)

smoker: hút thuốc hoặc không hút (Yes hoặc No)

day: ngày trong tuần

time: buổi trong ngày

size: số người có trên bàn ăn.

Yêu cầu:

Đọc từ file json và đưa ra các chỉ số thống kê (sum, avg, max, min) về tổng tiền hóa đơn (total\_bill) theo yêu cầu đề bài

Input:

- Dòng đầu đưa vào số bộ test
- Lần lượt đưa vào giới tính (sex) và trạng thái hút thuốc (smoker)

Output:

- Đưa ra 1 dòng gồm các chỉ số thống kê. Các số thập phân lấy 4 chữ số sau dấu phẩy.

Input	Output
1	1919.7500 19.7912 48.3300
Male No	7.5100

```
import json

with open('tips.json', encoding='utf-8') as f:
    du_lieu = json.load(f)['tips']

n = int(input())
for _ in range(n):
    dong = input().strip().split()
    gioi_tinh = dong[0]
    hut_thuoc = dong[1]
    #hoa_don_phu_hop = [
    #    float(item['total_bill'])
    #    for item in du_lieu
    #    if item['sex'] == gioi_tinh and item['smoker'] ==
hut_thuoc
    #]
    item_phu_hop=[]
    for item in du_lieu:
        if item['sex'] == gioi_tinh and item['smoker'] ==
hut_thuoc:
            item_phu_hop.append(item)

    if item_phu_hop:
```

```
tong = sum(item_phu_hop)
tb = tong / len(item_phu_hop)
lon_nhat = max(item_phu_hop)
nho_nhat = min(item_phu_hop)
print(f"{tong:.4f} {tb:.4f} {lon_nhat:.4f}
{nho_nhat:.4f}")
else:
    print("0.0000 0.0000 0.0000 0.0000 0.0000")
```