**ÔN TẬP CUỐI HỌC KỲ 1 – 2024 – LẬP TRÌNH PYTHON**

**MỤC LỤC**

[BÀI 1. SỐ LỘC PHÁT 2](#_Toc186495704)

[BÀI 2. TÍNH TỔNG PHÂN THỨC 2](#_Toc186495705)

[BÀI 3. TÍCH LẺ CHIA TỔNG CHẴN 2](#_Toc186495706)

[BÀI 4. SỐ XEN KẼ CHẴN 3](#_Toc186495707)

[BÀI 5. MẬT MÃ CAESAR 3](#_Toc186495708)

[BÀI 6. CHUỖI 0 DÀI NHẤT 4](#_Toc186495709)

[BÀI 7. TẠO HAI CHUỖI 4](#_Toc186495710)

[BÀI 8. KÝ TỰ XUẤT HIỆN NHIỀU LẦN NHẤT 5](#_Toc186495711)

[BÀI 9. KHOẢNG CÁCH EUCLIDCEAN 5](#_Toc186495712)

[BÀI 11. KHOẢNG CÁCH MINKOWSKI 6](#_Toc186495713)

[BÀI 12. ĐỘ TƯƠNG TỰ JACCARD 7](#_Toc186495714)

[BÀI 13. CHẤM CÔNG 7](#_Toc186495715)

[BÀI 14. ĐIỂM TRUNG BÌNH 8](#_Toc186495716)

[BÀI 15. LỚP TAM GIÁC VÀ CHU VI 9](#_Toc186495717)

[BÀI 16. VÒNG BẢNG WORLD CUP 9](#_Toc186495718)

[BÀI 17. CHUYẾN BAY 10](#_Toc186495719)

[BÀI 18. HOÁ ĐƠN THANH TOÁN 11](#_Toc186495720)

[BÀI 19. LOÀI HOA 12](#_Toc186495721)

[BÀI 20. LOÀI CÁNH CỤT 14](#_Toc186495722)

[BÀI 21. MẬT KHẨU HỢP LỆ 15](#_Toc186495723)

[BÀI 22. THỐNG KÊ CÁC VỊ TRÍ XUẤT HIỆN 15](#_Toc186495724)

[BÀI 23. SỐ HẠNH PHÚC 16](#_Toc186495725)

[BÀI 24. LÃI SUẤT NGÂN HÀNG 16](#_Toc186495726)

[BÀI 25. PHÉP TOÁN SỐ HỌC CƠ BẢN 17](#_Toc186495727)

[BÀI 26. CẶP SỐ MAY MẮN 18](#_Toc186495728)

# **BÀI 1. SỐ LỘC PHÁT**

Một số kết thúc bởi ba chữ số 668 được gọi là số lộc lộc phát.

Cho một số nguyên dương không quá 500 chữ số, kiểm tra số đó có phải số lộc lộc phát hay không.  
**Input**  
Dòng đầu ghi số bộ test.

Mỗi bộ test ghi số nguyên dương cần kiểm tra (không quá 500 chữ số)  
**Output**  
Ghi ra kết quả kiểm tra tương ứng (YES hoặc NO)

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 123456 1111668 | NO  YES |

# **BÀI 2. TÍNH TỔNG PHÂN THỨC**

Nhập số nguyên dương N (1 < N < 10000).

Viết chương trình tính tổng:

* nếu N lẻ
* nếu N chẵn

Kết quả được in ra với 5 chữ số phần thập phân.

**Input**

Dòng đầu ghi số bộ test, không quá 10.

Mỗi test ghi một số N

**Output**

Với mỗi bộ test, ghi ra kết quả trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 0.50000 0.66667 |

# **BÀI 3. TÍCH LẺ CHIA TỔNG CHẴN**

Viết chương trình tính giá trị của một dãy số bằng cách lấy tích các số tại vị trí lẻ (bỏ qua giá trị 0) chia cho tổng giá trị các số tại vị trí chẵn. Nếu tổng các số tại vị trí chẵn bằng 0 thì in ra INVALID

**Input**

- Dòng đầu ghi số bộ test

- Các dòng tiếp theo, mỗi dòng ghi một số có số chữ số không quá 100

**Ouput**

In ra giá trị tính được theo yêu cầu, làm tròn đến 6 số sau dấu phẩy.

Nếu tổng bằng 0 in ra INVALID

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 1012 30304 | 0.500000 INVALID |

**Giải thích test**

Test 1: Tích lẻ =1\*1=1, tổng chẵn: 0 + 2 = 2, kết quả là 0.5

Test 2: Tích lẻ =3\*3\*4=36, tổng chẵn: 0 + 0 = 0, kết quả là INVALID

# **BÀI 4. SỐ XEN KẼ CHẴN**

Một số xen kẽ chẵn là một số được định nghĩa như sau:

1. Số chữ số của số đó là chẵn.
2. Số đầu tiên của số đó khác số thứ ba,
3. Các chữ số ở vị trí chẵn có giá trị bằng nhau

Viết chương trình kiểm tra một số có phải số chẵn hay không.

**Input**: Dòng đầu ghi số bộ test. Mỗi dòng tiếp theo ghi một số

**Output**: Ghi kết quả kiểm tra

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 123123 12324262 | NO YES |

# **BÀI 5. MẬT MÃ CAESAR**

Trong mật mã học, mật mã Caesar, còn được gọi là mật mã Caesar, mật mã dịch chuyển, mã Caesar hoặc dịch chuyển Caesar, là một trong những kỹ thuật mã hóa đơn giản và được biết đến rộng rãi nhất. Nó là một loại mật mã thay thế, trong đó mỗi chữ cái trong bản rõ được thay thế bằng một chữ cái với một số vị trí cố định trong bảng chữ cái. Ví dụ, với sự dịch chuyển sang trái là 3, D sẽ được thay thế bằng A, E sẽ trở thành B, v.v. Phương pháp này được đặt theo tên của Julius Caesar, người đã sử dụng nó trong thư từ riêng tư của mình.

Viết chương trình Python để tạo mã hóa Caesar trong đó các ký tự chỉ nằm trong tập chữ hoa hoặc chữ thường tiếng Anh (26 chữ cái). Danh sách dịch chuyển là danh sách vòng.

**Input:**

Dòng đầu ghi số bộ test

Mỗi test ghi một xâu ký tự và một số thể hiện số bước dịch chuyển. Nếu là số dương thì dịch sang phải, số âm là dịch sang trái.

**Output:**

Với mỗi test ghi kết quả biến đổi

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  abc 2  BC 3 | cde  EF |

# **BÀI 6. CHUỖI 0 DÀI NHẤT**

Viết chương trình Python để tìm độ dài lớn nhất của các số 0 liên tiếp trong một chuỗi nhị phân cho trước.

**Input:**

Dòng đầu ghi số bộ test

Mỗi test ghi một xâu nhị phân trên một dòng.

**Output:**

Với mỗi test ghi kết quả tìm được. Nếu không tìm được thì ghi 0

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 3  111000010000110  111000111  1111111 | 4  3  0 |

# **BÀI 7. TẠO HAI CHUỖI**

Viết chương trình Python để tạo hai chuỗi từ một chuỗi cho trước. Tạo chuỗi đầu tiên bằng cách sử dụng những ký tự chỉ xuất hiện một lần và tạo chuỗi thứ hai bao gồm các ký tự xuất hiện nhiều lần trong chuỗi đã cho.

**Input:**

Dòng đầu ghi số bộ test. Mỗi test gồm một dòng là một xâu ký tự liền, không có khoảng trống.

**Output:**

Với mỗi bộ test ghi ra 2 dòng kết quả tìm được, dòng đầu là chuỗi tạo bởi nối các ký tự không lặp, dòng thứ hai tạo bởi nối các ký tự lặp. Nếu không có ký tự nào trong chuỗi output (có lặp hoặc không lặp) thì ghi là NONE

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  aabbcceffgh  abcabc | egh  abcf  NONE  abc |

# **BÀI 8. KÝ TỰ XUẤT HIỆN NHIỀU LẦN NHẤT**

Viết chương trình Python để tìm ký tự xuất hiện tối đa trong một chuỗi đã cho. Nếu số lần xuất hiện bằng nhau thì in ra ký tự xuất hiện trước.

**Input**

Dòng đầu ghi số bộ test

Các dòng tiếp theo, mỗi dòng ghi một xâu ký tự không có khoảng trống ở giữa

**Ouput**

Ghi ký tự tìm được

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  abca  xccxbxcc | a  c |

# **BÀI 9. KHOẢNG CÁCH EUCLIDCEAN**

Khoảng cách giữa 2 vector trong toán học phổ biến là khoảng cách euclidean với công thức

trong đó x và y là hai vector n chiều.

Viết chương trình tính khoảng cách euclidean của hai vector x và y.

**Input**

- Dòng đầu ghi số bộ test

- Hai dòng tiếp theo ghi 2 vector, mỗi vector là một bộ số trong đó các số cách nhau bởi một hoặc nhiều dấu cách

**Ouput:**

Khoảng cách giữa 2 vector, làm tròn đến 5 số sau dấu phẩy. Nếu 2 vector không cùng số chiều thì in ra INVALID

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 1 4 0 0  3 5 7 8  4 6 | 4.12311 INVALID |

**BÀI 10. KHOẢNG CÁCH MANHATTAN**

Khoảng cách manhattan giữa 2 vector trong toán học được định nghĩa với công thức:

trong đó x và y là hai vector n chiều.

Viết chương trình tính khoảng cách manhattan của hai vector x và y.

**Input**

Dòng đầu ghi số bộ test

Hai dòng tiếp theo ghi 2 vector, mỗi vector là một bộ số trong đó các số cách nhau bởi một hoặc nhiều dấu cách

**Ouput:**

Khoảng cách giữa 2 vector, làm tròn đến 5 số sau dấu phẩy. Nếu 2 vector không cùng số chiều thì in ra INVALID

Ví dụ

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 1 4 0 0  3 5 7 8  4 6 | 5.00000  INVALID |

# **BÀI 11. KHOẢNG CÁCH MINKOWSKI**

Khoảng cách Minkowski giữa 2 vector trong toán học được định nghĩa với công thức:

trong đó x và y là hai vector n chiều, p là hệ số mũ và hệ số căn (1/p).

Viết chương trình tính khoảng cách Minkowski của hai vector x và y.

**Input**

- Dòng đầu ghi số bộ test

- Hai dòng tiếp theo ghi 2 vector, mỗi vector là một bộ số trong đó các số cách nhau bởi một hoặc nhiều dấu cách

- Dòng tiếp theo ghi hệ số mũ p là một số nguyên

**Ouput**

Khoảng cách giữa 2 vector, làm tròn đến 5 số sau dấu phẩy.

Nếu 2 vector không cùng số chiều thì in ra INVALID

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  1 4  0 0  2  2 4 6  2 0  3 | 4.12311  INVALID |

# **BÀI 12. ĐỘ TƯƠNG TỰ JACCARD**

Độ tương tự Jaccard giữa 2 vector trong toán học được định nghĩa với công thức:

trong đó A và B là hai vector n chiều. Tử số là số các phần tử chung của A và B, mẫu số là số các phần tử xuất hiện ở A hoặc xuất hiện ở B

Viết chương trình tính Độ tương tự Jaccard của hai vector A và B.

**Input**

- Dòng đầu ghi số bộ test

- Hai dòng tiếp theo ghi 2 vector, mỗi vector là một bộ số trong đó các số cách nhau bởi một hoặc nhiều dấu cách

**Ouput:**

Độ tương tự của 2 vector, làm tròn đến 5 số sau dấu phẩy.

Nếu 2 vector không cùng số chiều thì in ra INVALID

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  1 4 3 2  0 0 1 2  2 4 6  2 0 | 0.40000  INVALID |

# **BÀI 13. CHẤM CÔNG**

Công ty chấm công các nhân viên để tính thời gian làm việc, từ đó tính xem nhân viên có làm việc đủ thời gian không. Thời gian làm việc chuẩn là 8 giờ. Thời gian làm việc của công nhân tính bằng thời gian ra về trừ thời gian đến và trừ một giờ (60 phút) nghỉ trưa.

**Input**

Dòng đầu của dữ liệu vào ghi số lượng nhân viên chấm công (không quá 20).

Thông tin về một công nhân chấm công được ghi lại trên 4 dòng lần lượt là:

* Mã nhân viên (xâu ký tự độ dài không quá 10, không có khoảng trống)
* Tên nhân viên (xâu ký tự độ dài không quá 100, có thể có khoảng trống).
* Giờ vào (định dạng hh:mm, **có thể chưa đủ 2 chữ số ở cả giờ và phút**)
* Giờ ra (định dạng hh:mm, **có thể chưa đủ 2 chữ số ở cả giờ và phút**)

**Ouput**

Ghi ra danh sách nhân viên đã được sắp xếp theo thời gian làm việc giảm dần cùng với thông tin thiếu (THIEU) hoặc đủ thời gian tiêu chuẩn (DU).

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 01T Nguyen Van An 08:00 17:30 06T Tran Hoa Binh 09:05 17:00 | 01T Nguyen Van An 8 gio 30 phut DU 06T Tran Hoa Binh 6 gio 55 phut THIEU |

# **BÀI 14. ĐIỂM TRUNG BÌNH**

Viết chương trình khai báo lớp Thí Sinh gồm các thông tin: Họ tên, Ngày sinh, Điểm môn 1, Điểm môn 2, Điểm môn 3 và Điểm trung bình. Trong đó điểm trung bình được tính bằng môn thấp nhất nhân đôi cộng điểm hai môn còn lại và chia tổng cho 4.

Đọc thông tin thí sinh từ bàn phím và in ra màn hình danh sách thí sinh được sắp xếp theo điểm trung bình giảm dần

**Input**

Dòng đầu ghi số bộ test

Test gồm 5 dòng lần lượt, mỗi dòng ghi 1 thông tin: Họ tên, Ngày sinh, Điểm môn 1, Điểm môn 2, Điểm môn 3. Họ tên không quá 50 chữ cái, Ngày sinh viết đúng chuẩn dd/mm/yyyy. Các giá trị điểm là số thực (float).

**Output**

Ghi ra danh sách thí sinh bao gồm: Họ tên, Ngày sinh và Điểm Trung bình. Mỗi thông tin cách nhau một khoảng trống. Điểm được ghi ra với 1 số sau dấu phẩy.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 Tran Van Nam 06/07/2002 4.5 7.9 5.4 Nguyen Hoang Ha 11/10/2001 4.5  10.0  5.5 | Nguyen Hoang Ha 11/10/2001 6.1 Tran Van Nam 06/07/2002 5.6 |

# **BÀI 15. LỚP TAM GIÁC VÀ CHU VI**

Một tam giác gồm 3 điểm (points) và mỗi điểm có tọa độ (x,y) trên trục tọa độ.

Xây dựng các lớp và chương trình phù hợp để tính chu vi tam giác.

**Input**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test ghi trên 1 dòng 6 số thực có giá trị tuyệt đối không quá 1000 lần lượt là tọa độ của 3 điểm.

**Output**

* Nếu 3 điểm không thể tạo thành tam giác thì in ra INVALID
* Nếu 3 điểm tạo thành 1 tam giác thì in ra chu vi của tam giác đó, làm tròn đến 6 chữ số phần thập phân.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  0 0 0 5 0 199  1 1 1 1 1 1  0 0 0 5 5 0 | INVALID  INVALID 17.071068 |

# **BÀI 16. VÒNG BẢNG WORLD CUP**

Vòng bảng WorldCup đã thi đấu xong các trận đấu trong vòng bảng. Kết quả của các đội được sắp xếp theo số điểm, nếu đội nào cao điểm hơn sẽ xếp trên.

Nếu hai đội bằng điểm nhau thì sẽ xếp theo hiệu số thắng-thua, đội nào cao hơn thì xếp trên.

Trong trường hợp hai đội bằng nhau cả về điểm số và hiệu số thắng thua thì sẽ xếp theo số bàn thắng ghi được, đội nào cao hơn sẽ xếp trên.

Viết chương trình in ra danh sách các đội đã được xếp hạng trên nguyên tắc trên.

**Input**

Dòng đầu tiên đưa vào danh sách đội bóng N.

Những dòng kế tiếp đưa vào N đội bóng. Mỗi đội bóng gồm 2 dòng dữ liệu, dòng đầu ghi tên đội bóng, dòng thứ hai ghi lần lượt điểm số, hiệu số thắng-thua, số bàn thắng ghi được. Mỗi giá trị cách nhau bởi dấu cách.

**Output**

Đưa ra bảng xếp hạng danh sách quốc gia đã sắp xếp

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Uruguay  4 0 2  Korean  4 0 4 | Korean 4 0 4 Uruguay 4 0 2 |

# **BÀI 17. CHUYẾN BAY**

Dữ liệu về chuyến bay được lưu trữ trong tệp "flights.json" có cấu trúc từ điển lồng nhau bao gồm  
- Ngoài cùng là từ điển có khóa là flights

- Giá trị của khóa danh sách các từ điển, mỗi phần tử bao gồm các khóa và giá trị text như hình:

{

    "flights": [

    {

        "year": "1949",

        "month": "January",

        "passengers": "112"

    },

    {

        "year": "1949",

        "month": "February",

        "passengers": "118"

    },

    {

        "year": "1949",

        "month": "March",

        "passengers": "132"

    }

    ]

}

***Chú ý:*** *Đây chỉ là ví dụ để hiểu định dạng, không phải là ví dụ của số liệu trong test mẫu phía dưới*

Trong đó giá trị của khóa year là năm thống kê, giá trị của khóa month là tháng thống kê, giá trị của khóa passagers là số hành khách chở được tương ứng.

**Yêu cầu:** Đọc dữ liệu từ file JSON và in ra các thông tin thống kê theo năm.

**Input**

Dòng đầu là số bộ Test

Các dòng tiếp theo, mỗi dòng ghi số năm và chỉ số cần thống kê (min, max, sum, avg)

**Output**

In ra giá trị tìm được. Riêng giá trị trung bình (avg) thì in ra làm tròn đến 5 chữ số sau dấu phẩy. Nếu không tìm được giá trị phù hợp in ra "Invalid"

**Test ví dụ:**

* 1. **File flights.json**

****

* 1. **Nhập xuất**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 1959 sum 1950 avg | 5140 139.71429 |

# **BÀI 18. HOÁ ĐƠN THANH TOÁN**

Tips là dữ liệu về tiền thanh toán và tiền bo (tips) cho nhà hàng bao gồm danh sách các từ điển trong tệp Json lồng nhau, mỗi từ điển đại diện  cho mỗi bàn được thanh toán và bo bao gồm các trường như hình dưới:

**Thông tin các thuộc tính**

total\_bill: tổng số chi

tip: tiền bo

sex: giới tính

smoker: hút thuốc hoặc không hút

day: ngày trong tuần

time: buổi trong ngày

size: số người có trên bàn ăn.

**(File JSON có cấu trúc như hình dưới đây)**

{

    "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",

        "day": "Sun",

        "time": "Dinner",

        "size": "3"

      }

    ]

}

***Chú ý:*** *Đây chỉ là ví dụ để hiểu định dạng, không phải là ví dụ của số liệu trong test mẫu phía dưới*

**Yêu cầu:**

Đọc từ file "tips.json" và đưa ra giá trị hóa đơn trung bình (total\_bill) theo ngày và số người trên bàn ăn

**Input:**

* Dòng đầu tiên đưa vào số bộ test
* Dòng thứ 2 ghi ngày và số người trên bàn ăn cách nhau 1 khảng trắng

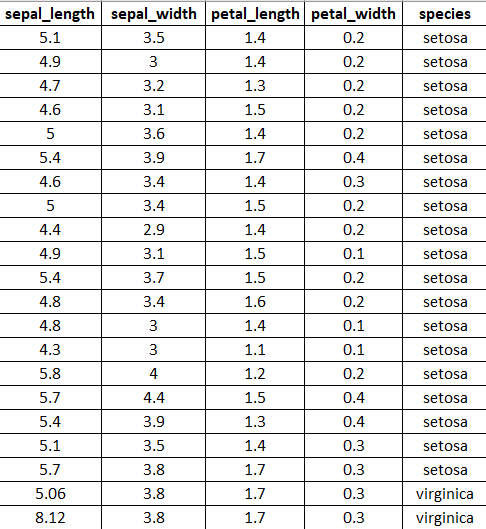
**Output:**

* Đưa ra giá trị trung bình của tổng hóa đơn (total\_bill). Các giá trị thập phân cách nhau bởi 1 khoảng trắng

|  |  |
| --- | --- |
| Input | Output |
| 2  Sun 2  Fri 10 | 17.5600  Invalid |

# **BÀI 19. LOÀI HOA**

Dữ liệu về hoa được lưu trữ trong tệp “iris.csv”  có cấu trúc như hình:



**Trong đó:**

Giá trị sepal\_length là chiều dài của đài hoa

Giá trị sepal\_width là chiều rộng của đài hoa

Giá trị petal\_length là chiều dài của cánh hoa

Giá trị petal\_length là chiều rộng của cánh hoa

Gía trị Species là tên loài.

**Yêu cầu:**

Đọc dữ liệu từ file "iris.csv" và in ra thống kê (min, max, sum, avg) theo chiều dài/rộng x của loài y cho trước. Nếu loài y hoặc tên chiều dài x không có trong tệp dữ liệu thì in ra “Invalid”

**Input**

Dòng đầu ghi số bộ test

Các dòng tiếp theo, mỗi dòng ghi tên loài(species), tên chiều dài và chỉ số cần thống kê (min, max, sum, avg) các nhau bởi khoảng trắng.

**Ouput**

In ra giá trị tính được. Nếu giá trị là float thì làm tròn đến 2 chữ số sau dấu phẩy

**Ví dụ:** (Sample file cũng chính là file trên hình)

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  virginica sepal\_length avg  sunflower petal\_length max | 6.59  Invalid |

**Giải thích test:**

Test 1: Trung bình chiều dài cánh hoa (sepal\_length) của loài virginica là (5.06 + 8.12)/2 = 6.59

Test 2: Loài sunflower không có trong tệp dữ liệu nên ghi Invalid

# **BÀI 20. LOÀI CÁNH CỤT**

File penguins.csv lưu thông tin về các loài chim cánh cụt đang ở trên các đảo bao gồm các thông tin như sau:

A table with numbers and a number

Description automatically generated

Ý nghĩa của các trường dữ liệu được thể hiện trong bảng như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** |
| 1 | species | loài | chuỗi |
| 2 | island | đảo đang ở | chuỗi |
| 3 | bill\_length\_mm | độ dài cánh | số |
| 4 | bill\_depth\_mm | độ sâu | số |
| 5 | flipper\_length\_mm | flipper lengh (mm) | số |
| 6 | body\_mass\_g | trọng lượng cơ thể | số |
| 7 | sex | giới tính | chuỗi |

(Lưu ý: các ô có thể không có dữ liệu do việc thu thập)

Viết chương trình Python đọc file và in ra trung bình độ dài cánh (bill\_length\_mm) và độ sâu cánh (bill\_depth\_mm) theo loài và đảo đang ở.

**Input**

Dòng đầu là số n bộ test

Các dòng tiếp theo, mỗi dòng ghi một bộ test bao gồm tên loài và tên đảo

**Ouput**

Mỗi kết quả ghi trên một dòng. Ghi độ dài cánh và độ sâu cánh trung bình, làm tròn đến 4 số sau dấu phẩy. Nếu không tìm thấy ghi Invalid

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Adelie Torgersen  Adelie PhuQuoc | 39.1125 18.1234  Invalid |

# **BÀI 21. MẬT KHẨU HỢP LỆ**

Một website yêu cầu người dùng nhập vào username và password để đăng ký. Viết một chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.

Các tiêu chí để kiểm tra mật khẩu:

1, Chứa ít nhất một ký tự từ [a-z]

2, Chứa ít nhất một ký tự viết hoa: [A-Z]

3, Chứa ít nhất một chữ số: [0-9]

4, Chứa ít nhất một trong các ký tự đặc biệt sau: [$#@!]

5, Độ dài ngắn nhất là 6

6, Độ dài lớn nhất là 12.

Chương trình nhận đầu vào là một chuỗi các mật khẩu được phân tách bằng dấu phẩy và sẽ kiểm tra chúng theo các tiêu chí trên. Mật khẩu phù hợp với các tiêu chí sẽ được in ra màn hình, mỗi mật khẩu được phân tách bằng dấu phẩy.

**Input:** Chuỗi các mật khẩu được phân tách bằng dấu phẩy

**Output:**

* Các mật khẩu phù hợp với các tiêu chí, mỗi mật khẩu phân tách nhau bằng dấu phẩy.
* Nếu không có mật khẩu nào hợp lệ, in ra "INVALID PASSWORD"

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ABd1234@1,a F1#,2w3E\*,2We3345 | ABd1234@1 |
| aF1#,2w3E\* | INVALID PASSWORD |

# **BÀI 22. THỐNG KÊ CÁC VỊ TRÍ XUẤT HIỆN**

Viết chương trình để tìm chỉ mục của một phần tử trong một danh sách cho trước. Nếu phần tử xuất hiện nhiều lần trong danh sách, in ra tất cả các chỉ mục của phần tử. Nếu phần tử không xuất hiện trong danh sách, in ra **-1**.

**Input:**

* Dòng đầu là số phần tử trong danh sách (n)
* Dòng thứ hai là n phần tử của danh sách
* Dòng thứ ba là phần tử cần kiểm tra chỉ mục

**Output:**

* Chỉ mục của phần tử cần kiểm tra. Nếu phần tử xuất hiện nhiều lần, mỗi chỉ mục cách nhau bởi một dấu phẩy.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  2 11 13 9 22 10  13 | 2 |
| 11  2 4 5 6 8 11 13 9 22 35 8  8 | 4, 10 |
| 11  2 4 5 6 8 11 13 9 22 35 8  10 | -1 |

# **BÀI 23. SỐ HẠNH PHÚC**

Một số hạnh phúc được xác định bởi quá trình sau đây:

Bắt đầu với bất kỳ số nguyên dương, thay thế số bằng tổng bình phương các chữ số của nó, và lặp lại quá trình cho đến khi số bằng 1.

***Ví dụ 1***: Kiểm tra số 44:

* Lần lặp thứ nhất: 42 + 42 = 16 + 16 = 32.
* Lần lặp thứ hai: 32 + 22 = 9 + 4 = 13.
* Lần lặp thứ ba: 12 + 32 = 1 + 9 = 10.
* Cuối cùng: 12 + 02 = 1 + 0 = 1.

Vậy 44 là một số hạnh phúc.

***Ví dụ 2***: Kiểm tra số 19:

* Lần lặp thứ nhất: 12 + 92 = 82
* Lần lặp thứ hai: 82 + 22 = 68
* Lần lặp thứ ba: 62 + 82 = 100
* Lần lặp cuối cùng: 12 + 02 + 02 = 1

Vậy 19 cũng là một số hạnh phúc

Viết chương trình kiểm tra một số nguyên có phải số hạnh phúc hay không?

**Input:** Số cần kiểm tra

**Output:** In ra YES nếu số kiểm tra là số hạnh phúc, ngược lại in ra NO

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 44 | YES |
| 17 | NO |
| 19 | YES |

# **BÀI 24. LÃI SUẤT NGÂN HÀNG**

Mỗi ngân hàng có các hình thức gửi tiết kiệm như: tiết kiệm không kỳ hạn và có kỳ hạn. Với loại hình tiết kiệm có kỳ hạn, số tiền gửi sẽ được quy định một mức kỳ hạn đi kèm với mức lãi suất cam kết. Ngân hàng sẽ đưa ra nhiều mức kỳ hạn khác nhau để khách hàng dễ dàng lựa chọn theo nhu cầu.

Sau đây là bảng mẫu biểu lãi suất tiết kiệm tại ngân hàng Techcombank:

|  |  |  |  |
| --- | --- | --- | --- |
| **Kỳ hạn (Tháng)** | **Lãi suất (%)** | | |
| **Tiền gửi < 1 tỷ** | **Tiền gửi từ 1-3 tỷ** | **Tiền gửi >= 3 tỷ** |
| 1-2 Tháng | 2.5 | 2.7 | 2.8 |
| 3-6 Tháng | 3.9 | 4.1 | 4.3 |
| 7-12 Tháng | 4.8 | 4.9 | 5.0 |
| 13-36 Tháng | 4.8 | 5.0 | 5.1 |
| > 36 Tháng | 4.7 | 4.9 | 5.0 |

Theo đó, cách tính lãi suất ngân hàng cho trường hợp này như sau:

**Số tiền lãi*=*Số tiền gửi x (lãi suất/12) x số tháng gửi*.***

***Ví dụ:***

Một khách hàng gửi tiết kiệm 50,000,000 VND với kỳ hạn 6 tháng tại Ngân hàng, đối chiếu theo mẫu biểu trên, mức lãi suất là 3.9%. Đến kỳ hạn 6 tháng, số tiền lãi khách hàng nhận được là:

Số tiền lãi = 50,000,000 x (3.9%/12) x 6 = 975,000  VNĐ

Viết chương trình tính số tiền lãi mà khách hàng nhận được khi gửi tiết kiệm có kỳ hạn tại ngân hàng Techcombank.

**Input:**

* Số tiền khách hàng gửi tiết kiệm
* Kỳ hạn gửi tiết kiệm

**Output:**

* Số tiền lãi mà khách hàng nhận được khi kết thúc kỳ hạn gửi.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 50000000  6 | 975000 |
| 1500000000 14 | 87500000 |

# **BÀI 25. PHÉP TOÁN SỐ HỌC CƠ BẢN**

Tạo một lớp Computation với một hàm tạo mặc định nhận tham số đầu vào là số nguyên ***n.***Lớp Computation cho phép thực hiện các phép tính khác nhau trên số nguyên ***n***.  
1 - Tạo một phương thức có tên là Factorial() cho phép tính giai thừa của số nguyên ***n***.

2 - Tạo phương thức có tên là Prime() để kiểm tra ***n*** có phải số nguyên tố hay không. In ra YES nếu ***n*** là số nguyên tố, ngược lại, in ra NO

3 - Tạo phương thức listDiv() để liệt kê tất cả các ước của số nguyên ***n*** đã cho.

Khởi tạo một đối tượng cho lớp Computation và kiểm tra kết qủa của các phương thức đã tạo. Kết quả của mỗi phương thức được viết trên một dòng khác nhau.

**Input:**

* Số nguyên ***n*** cần kiểm tra

**Output:**

* Giai thừa của ***n***
* Kết quả kiểm tra số nguyên tố: **YES/NO**
* Các ước của số n, mỗi số được viết các nhau bởi dấu phẩy.

Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 6 | 720  NO  1,2,3,6 |
| 7 | 5040  YES  1,7 |

# 

# **BÀI 26. CẶP SỐ MAY MẮN**

Một cặp số nguyên dương (x,y) được gọi là cặp số may mắn nếu tồn tại một số nguyên dương k thoả mãn: xk + yk chia hết cho x.y. Ví dụ:

* (2, 4) là một cặp số may mắn vì 23 + 43 = 8 + 64 = 72 chia hết cho 2.4 = 8
* (3, 3) là một cặp số may mắn vì 32 + 32 = 9 + 9 = 18 chia hết cho 3.3 = 9
* (1,2) không phải là một cặp số may mắn vì 1k + 2kluôn luôn là một số lẻ, với mọi k > 0, và không thể chia hết cho 1.2 = 2

Cho một mảng **a** gồm **n** số nguyên dương. Viết chương trình đếm số cặp (i, j) với i < j và (ai, aj) là một cặp số may mắn.

**Input:**

* Dòng đầu tiên chứa độ dài của mảng - ***n*** ( 2 n 105)
* Dòng thứ hai chứa ***n*** số nguyên của mảng ***a***, mỗi phần tử cách nhau bởi khoảng trắng.

**Output:**

* Số cặp số may mắn trong mảng ***a***

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 6  1 2 3 4 5 6 | 1 |
| 4  7 7 7 7 | 6 |