**TỔNG HỢP CÁC BÀI TRONG ĐỀ THI KẾT THÚC HỌC PHẦN**

**LẬP TRÌNH VỚI PYTHON – KHOÁ D19**

**MỤC LỤC**

[**BÀI 1. SLICE STRINGS** 1](#_Toc186601547)

[**BÀI 2. HỆ SỐ JACCARD** 2](#_Toc186601548)

[**BÀI 3. DELTA SUBSTITUTION VARIANT** 3](#_Toc186601549)

[**BÀI 4. ĐỘ TƯƠNG TỰ CÔ-SIN** 6](#_Toc186601550)

[**BÀI 5. TỔNG CỘNG DỒN** 8](#_Toc186601551)

[**BÀI 6. TỪ ĐIỂN LỒNG NHAU** 9](#_Toc186601552)

[**BÀI 7. TỪ ĐIỂN** 10](#_Toc186601553)

[**BÀI 8. MONTHLY SALARY** 11](#_Toc186601554)

[**BÀI 9. KHOẢNG CÁCH HAVERSINE** 13](#_Toc186601555)

[**BÀI 10. LỚP TAM GIÁC VUÔNG** 14](#_Toc186601556)

[**BÀI 11. CẶP SỐ THÂN THIẾT** 14](#_Toc186601557)

[**BÀI 12. DANH SÁCH TUPLES** 14](#_Toc186601558)

[**BÀI 13. DELTA DELETION VARIANT** 15](#_Toc186601559)

[**BÀI 13. QUY ĐỔI ĐIỂM** 16](#_Toc186601560)

[**BÀI 14. FIRST SECOND AND LAST OCCURRENCE** 17](#_Toc186601561)

[**BÀI 15. DELTA INSERTION VARIANT** 17](#_Toc186601562)

[**BÀI 16. EUCLIDEAN DISTANCE** 20](#_Toc186601563)

[**BÀI 17. MONTHLY TAX** 22](#_Toc186601564)

[**BÀI 18.** TUYỂN NHÂN VIÊN 24](#_Toc186601565)

[**BÀI 19. TÁCH CHUỖI** 25](#_Toc186601566)

[**BÀI 19. COVID-19 OMICRON SUBSTITUTION VARIANT** 25](#_Toc186601567)

[**BÀI 20. PHẦN DƯ LỚN NHẤT** 28](#_Toc186601568)

[**BÀI 21. FREQUENT WORD** 29](#_Toc186601569)

[**BÀI 22. PHÂN PHỐI NHỊ THỨC** 30](#_Toc186601570)

[**BÀI 22. CHỈ SỐ TF-IDF** 31](#_Toc186601571)

**BÀI 1.** [**SLICE STRINGS**](https://icpc.ptit.edu.vn/student/question/PYCK113)

Cho input là một chuỗi. Viết chương trình trả về 9 dòng với yêu cầu như sau:

1. Trong dòng đầu tiên, in độ dài của chuỗi đã cho.
2. Trong dòng thứ hai, in ba ký tự cuối của chuỗi này.
3. Trong dòng thứ ba, bắt đầu bằng ký tự cuối cùng là vị trí đầu tiên 0, in tất cả các ký tự có vị trí là chẵn của chuỗi theo thứ tự ngược từ cuối chuỗi về đầu chuỗi.
4. Trong dòng thứ tư, in tất cả trừ hai ký tự cuối cùng của chuỗi này.
5. Trong dòng thứ năm, in năm ký tự đầu tiên của chuỗi này.
6. Trong dòng thứ sáu, in tất cả các ký tự của chuỗi này với các chỉ số chẵn (hãy nhớ lập chỉ mục bắt đầu từ 0, vì vậy các ký tự được hiển thị bắt đầu bằng chữ cái đầu tiên).
7. Trong dòng thứ bảy, in tất cả các ký tự của chuỗi này với các chỉ số lẻ (tức là bắt đầu bằng ký tự thứ hai trong chuỗi).
8. Trong dòng thứ tám, in tất cả các ký tự của chuỗi theo thứ tự ngược lại.
9. Trong dòng thứ chín, in ký tự thứ hai tính từ cuối lên của chuỗi này.

**Input.** Là một xâu ký tự

**Output.** Xuất ra màn hình 09 dòng theo yêu cầu đề bài

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| qwertyuiop | 10  iop  piyrw  qwertyui  qwert  qetuo  wryip  poiuytrewq  o |

**BÀI 2.** [**HỆ SỐ JACCARD**](https://icpc.ptit.edu.vn/student/question/PYCK12)

Hệ số Jaccard là một đại lượng dùng để đo mức độ tương tự giữa hai chuỗi bất kỳ. Cho hai chuỗi A và B, hệ số Jaccard giữa A và B, được ký hiệu là J(A, B) và được xác định theo công thức sau:

Cho hai chuỗi A và B, Tính độ đo tương đồng Jaccrad giữa hai chuỗi. Lưu ý:

* + Nếu A và B rỗng, thì J(A, B)=1
  + J(A, B) nằm trong đoạn [0, 1]
  + Lưu ý các kí tự trong chuỗi để tính tương đồng không phân biệt chữ hoa thường, không bao gồm các ký tự như khoảng trắng, dấu chấm (.), dấu phấy (,) và dấu hỏi chấm (?).

**Input.** Hai chuỗi A và B, mỗi chuỗi được viết trên một dòng.

**Output.** Hệ số Jaccrad giữa A và B. Kết quả được in ra với 2 chữ số sau dấu phẩy.

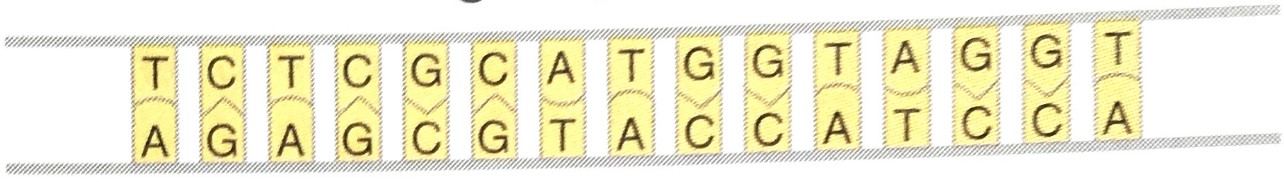
**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| hom nay thi lap trinh python  Lap trinh Python. | 0.91 |
| Hom qua troi mua.  Hom qua mua, ngay mai co nang khong? | 0.50 |

**BÀI 3.** [**DELTA SUBSTITUTION VARIANT**](https://icpc.ptit.edu.vn/student/question/PYCK11)

Biến thể Delta của virus SARS-CoV-2 được phát hiện lần đầu tiên ở Ấn Độ vào tháng 10/2020. Biết rằng, cứ 11 ngày trong quần thể virut xảy ra một đột biến điểm (Nucleotid/ base).

Đây là thuật ngữ của di truyền học phân tử, trong tiếng nước ngoài gọi là Point Mutation (tiếng Anh) dùng để chỉ một loại đột biến gen chỉ xảy ra ở một “điểm” trên phân tử di truyền DNA. “Điểm” này tương ứng với một cặp Nulceotide duy nhất.

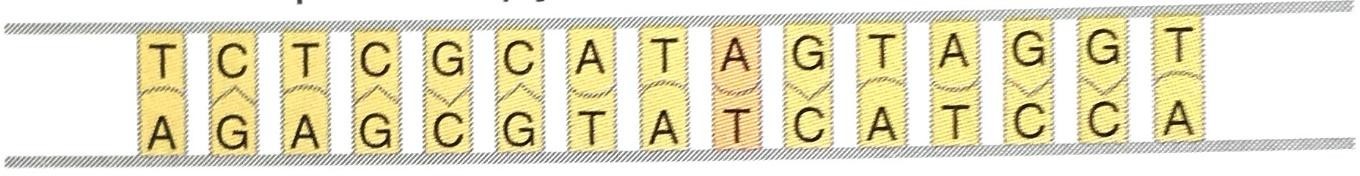


*Hình 1: Bộ gen ban đầu của Virus*

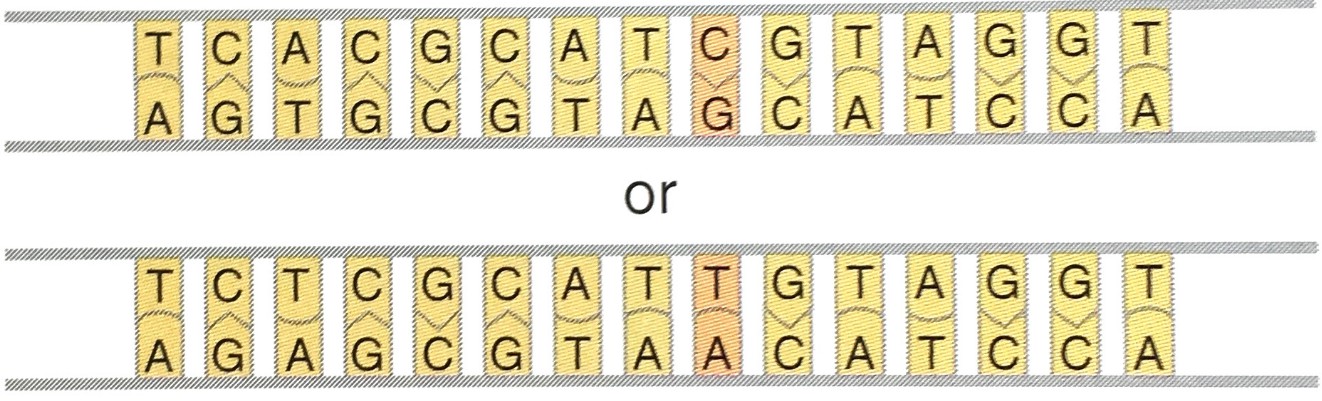
Có ba loại đột biến điểm – Point Mutation là:

* + Đột biến thay thế - Substitution
  + Đột biến xóa – Deletion
  + Đột biến them – Insertion

Đột biến thay thế - Substitution:



Thay thế thế một số cặp Nucleotides AT thành GC hoặc thay đổi thứ tự AT thành TA và GC thành CG:



Biết Bộ gen (hay hệ gen) của virut có chiều dài (kích thước) làm tròn là 30.000 Nucleotides / Bases; các đột biến hoàn toàn ngẫu nhiên và với xác suất là như nhau. Giả sử tất cả các đột biến điểm mới của biến thể Delta đều là đột biến thay thế Substitution và thay thế duy nhất 1 cặp.

Gợi ý:

Một đột biến có thể xảy ra tại 1 cặp bất kỳ trên 30.000 cặp Nucleotides nên xác suất 2 đột biến giống nhau là 1/30.000 ~ 0 coi như các đột biến sinh ra là khác nhau.

* + Sau 11 ngày biến thể Delta ban đầu D0 trên toàn bộ quần thể sẽ tạo ra một đột biến điểm là D01 lúc này ta có 1 đột biến điểm kiểu Substitution.
  + Sau 11 ngày tiếp theo quần thể Delta D0 sẽ tạo ra thêm 1 biến thể mới là D02 và quần thể D01 sẽ tạo ra them 1 đột biến điểm là D012 vậy lúc này ta sẽ có 3 đột biến điểm là D01, D02 và D012….

Viết chương trình tính số lượng đột biến điểm kiểu thay thế Substitution mà biến thể Delta đã có sau ngày 1/10/2020.

**Input:**

Dòng đầu tiên số nguyên N (1 ≤ N ≤ 10) là số tập test.

N dòng tiếp theo, mỗi dòng gồm một số nguyên có giá trị không vượt quá 365 là số ngày tính từ 1/10/2020 thời điểm biến thể Delta bắt đầu xuất hiện.

**Output**

In ra N dòng, mỗi dòng một số nguyên là số lượng đột biến điểm của biến thể Delta. Trường hợp nếu số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra INVALID INPUT

Bộ test nào có giá trị lớn hơn 365 in ra dòng đó là INVALID INPUT

**Ví dụ**

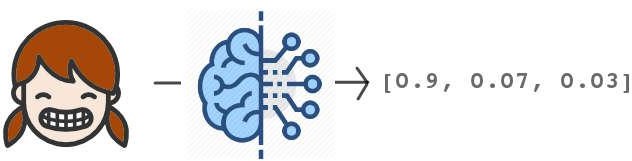
|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 0 |
| 7 | INVALID INPUT |
| 380 | 1 |
| 18 | 1023 |
| 113 | 3 |
| 30 |  |

**BÀI 4. ĐỘ TƯƠNG TỰ CÔ-SIN**

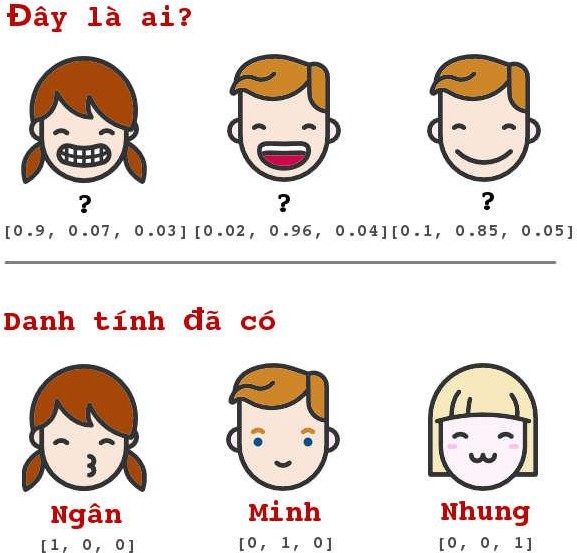
Ứng dụng đo độ tương tự hai vector nhận diện mỗi tấm ảnh đưa vào là khuôn mặt của ai (danh tính) bằng công thức toán tính góc của hai vector:



Trong bài toán xử lý ảnh, đầu ra của giải thuật nhận diện khuôn mặt là 1 vector đặc trưng. Mỗi ảnh trong thư viện ảnh đã có và ảnh cần nhận diện sẽ chạy qua giải thuật trích đặc trưng (deep learning) để có vector đặc trưng tương ứng.



Như vậy ta có vector đặc trưng của mỗi khuôn mặt cần nhận diện. Hình minh họa bên dưới:



Bằng việc so sánh vector đặc trưng của mỗi hình ảnh cần dự đoán Đây là ai? với Danh tính đã có bằng cách tính độ tương tự giữa hai vector dùng cosine như công thức ở mục phía trên: Tính cosine similarity

Bài toán trên đã được số hóa thành:

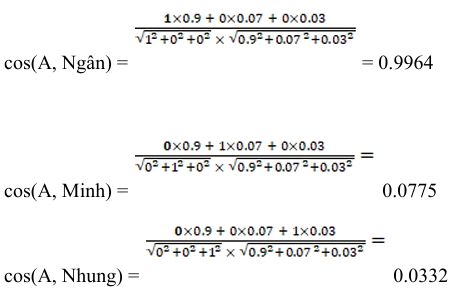
Ngân = [1, 0, 0]

Minh = [0, 1, 0]

Nhung = [0, 0, 1]

Có một hình ảnh A sau khi được mã hóa là A = [0.9, 0.07, 0.03]

So trùng các vector A với các danh tính đã có Ngân, Minh, Nhung:



Kết luận: A là Ngân vì có độ tương đồng lớn nhất

**Yêu cầu:** Viết chương trình tính độ tương đồng cosin của hình ảnh mới với những người trong CSDL và in ra người có độ tương đồng lớn nhất trong hệ thống nhận diện danh tính khi biết vector đặc trưng của mỗi người và một bức ảnh mới A sau khi được mã hóa.

**Input**

Dòng đầu tiên số nguyên N là tổng số dòng trong bộ test (N > 3).

Dòng thứ 2 là số tự nhiên n số chiều của vector đặc trưng cho khuôn mặt (n < 1025)

Dòng thứ 3 là gồm n số là chỉ số cho từng thành phần của vector đặc trưng cho một bức ảnh mới A

N-3 dòng còn lại, mỗi dòng gồm 02 phần:

* String Sn đầu tiên là tên người
* n số tương ứng với n chỉ số cho từng thành phần của vector đặc trưng cho người Sn đó

**Output:**

In ra tên người có vector tương đồng lớn nhất với A và in ra danh sách chỉ số độ tương đồng cosin của hình ảnh mới với những người trong CSDL làm tròn đến số thứ 4 sau dấu phẩy.

Trường hợp số nguyên N < 4 hoặc n > 1024 in ra INVALID INPUT

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 | Ngan [0.9964, 0.0775, 0.0332] |
| 3 |  |
| 0.9 0.07 0.03 |  |
| Ngan 1 0 0 |  |
| Minh 0 1 0 |  |
| Nhung 0 0 1 |  |

[**BÀI**](https://icpc.ptit.edu.vn/student/question/PYCK28) **5. TỔNG CỘNG DỒN**

Cho một danh sách số gồm N phần tử. Viết chương trình tính tổng tích lũy của một danh sách, nghĩa là, sẽ tạo ra một danh sách mới, trong đó phần tử thứ *i* là tổng của *i+1* phần tử đầu tiên từ danh sách ban đầu.

Ví dụ: Một danh sách có 3 phần tử: [1, 2, 3] => danh sách tổng tích lũy mới là: [1, 3, 6] In ra màn hình tích và tổng của các phần tử trong danh sách mới đó.

**Input**

* Dòng đầu tiên là số phần tử của mảng N.
* Dòng thứ hai là N phần tử của mảng, mỗi phần tử cách nhau bởi khoảng trắng.

**Output.** Tổng và tích của danh sách tổng tích lũy, viết trên 1 dòng, cách nhau 1 dấu cách

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1 2 3 | 10 18 |

[**BÀI**](https://icpc.ptit.edu.vn/student/question/PYCK32) **6. TỪ ĐIỂN LỒNG NHAU**

Tên và điểm thi môn Lập trình Python của mỗi sinh viên được lưu vào một từ điển, với key là tên sinh viên, value tương ứng là điểm thi của sinh viên. Lưu trữ chúng trong một từ điển lồng nhau. Với một lớp có N (N <= 70) sinh viên, tìm và in ra tên của K (K<=5) sinh viên có điểm thi cao nhất. In ra màn hình theo thứ tự điểm giảm dần, nếu bằng điểm nhau hãy in ra theo bảng chữ cái tăng dần

**Input**

* Dòng đầu tiên là giá trị N và K, cách nhau bởi một khoảng trắng.
* N dòng tiếp theo là tên và điểm thi của sinh viên

**Output**

* Tên của K sinh viên có điểm cao nhất, được viết trên một dòng duy nhất mỗi tên cách nhau bởi một khoảng trắng.
* Nếu giá trị N và K không thỏa mãn điều kiện, in ra thông báo: INVALID INPUT

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 3  Hung 6  Long 7  Giang 8  Linh 5  Tuan 8  Hoa 9  Mai 5  Ngoc 4  Khanh 9  Ngan 10 | Ngan Hoa Khanh |

**BÀI 7.** [**TỪ ĐIỂN**](https://icpc.ptit.edu.vn/student/question/PYCK34)

Cho một từ điển gồm K phần tử. Viết chương trình tính tổng và tích của các phần tử có giá trị (value) là số trong từ điển đã cho.

Input:

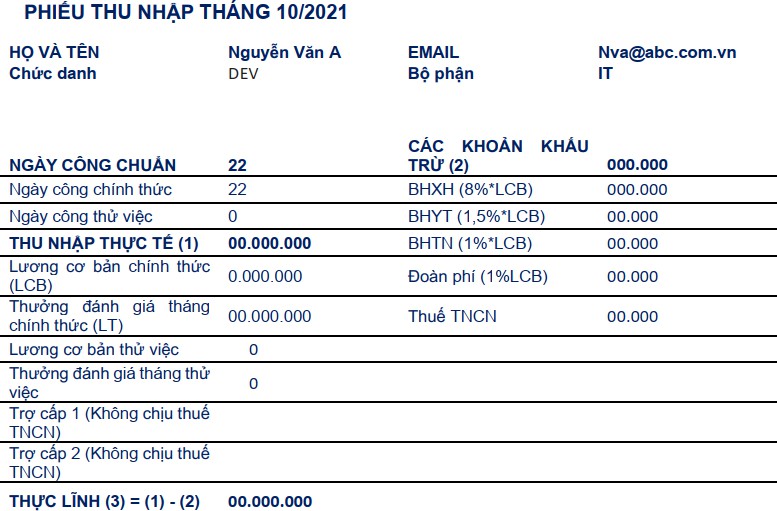
* Dòng đầu tiên là giá trị K (K <= 10)
* K dòng tiếp theo là các phần tử của từ điển, mỗi phần tử viết trên một dòng, lần lượt là key và value tương ứng.

**Output**

* Tổng và tích được viết trên cùng một dòng, cách nhau bởi khoảng trắng.
* Nếu K không thỏa mãn điều kiện thì in ra thông báo INVALID INPUT

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  a 9  b 5  c abc  d 1 | 15 45 |

**BÀI 8.** [**MONTHLY SALARY**](https://icpc.ptit.edu.vn/student/question/PYCK41)

Trên đây là Phiếu thu nhập của một người trong một tháng bao gồm:

* Thu nhập thực tế (1)
* Các khoản khấu trừ (2)
* Thực lĩnh (3)

**Các công thức tính cần chú ý:**

+) Thu nhập thực tế (1) = Lương cơ bản (LCB) + Lương thưởng đánh giá tháng (LT)

+) Các khoản khấu trừ (2) = BHXH + BHYT + BHTN + Đoàn phí + Thuế TNCN

+) Thực lĩnh (3) = Thu nhập thực tế (1) - Các khoản khấu trừ (2)

+) BHXH (Bảo hiểm xã hội) = 8%\*LCB

+) BHYT (Bảo hiểm y tế) = 1,5%\*LCB

+) BHTN (Bảo hiểm thất nghiệp) = 1%\*LCB

+) ĐP (Đoàn phí) = 1%\*LCB

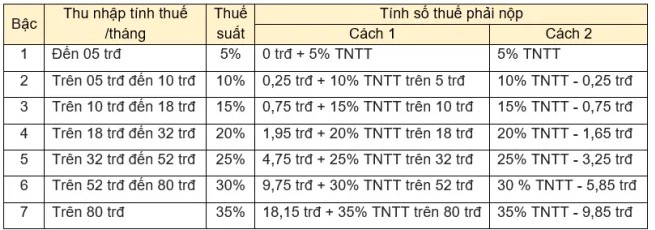
+) Thuế TNCN (Thuế Thu nhập cá nhân) = Thu nhập chịu thuế (TNTT) \* Thuế suất

+) TNTT = Thu nhập thực tế (1) – Các khoản giảm trừ - Tiền đóng bảo hiểm

+) Tiền đóng bảo hiểm (BH) = BHXH + BHYT + BHTN

+) Các khoản giảm trừ = Giảm trừ bản thân (11.000.000 vnđ) + Người phụ thuộc (4.400.000vnđ/ 1 người)

Dựa vào điều kiện của Thu nhập chịu thuế (TNTT) như bảng dưới đây ta sẽ có cách tính Thuế TNCN:

Ví dụ: TNTT = 6.000.000vnđ thì thuế TNCN = 250.000 + 10\*(TNTT – 5.000.000) = 350.000vnd

Viết chương trình in ra số tiền Thực lĩnh của người lao động trong 1 tháng khi biết Thu nhập thực tế (1) và LCB của người đó hang tháng và người này không có người phụ thuộc

**Input:**

Dòng đầu tiên số nguyên N (1 ≤ N ≤ 10) là số tập test.

N dòng tiếp theo, mỗi dòng gồm 2 số m, n > 0.

Trong đó m là Thu nhập thực tế (1) và n là LCB

**Output:**

In ra 1 số là tiền Thực lĩnh hàng tháng (lấy đến phần nguyên)

Trường hợp số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra INVALID INPUT

Bộ test nào có giá trị nhỏ hơn 1000 in ra dòng đó là INVALID INPUT

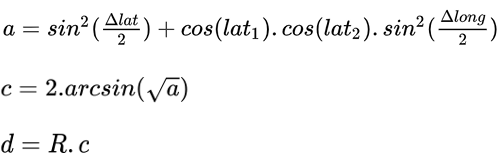
**Ví dụ:**

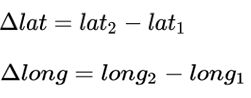
|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  15700000 6300000  13300000 5700000  100000000 7700000  1000 123 | 14773575  12559425  78097475  INVALID INPUT |

**BÀI 9.** [**KHOẢNG CÁCH HAVERSINE**](https://icpc.ptit.edu.vn/student/question/PYCK42)

Khoảng cách Haversine là một đại lượng để xác định khoảng cách địa lý giữa hai địa điểm bất kỳ. Giả sử hai địa điểm P1 và P2 được quy đổi về kinh độ và vĩ độ tương ứng là long1, lat1, long2, lat2. Khoảng cách Haversine d được xác định theo công thức sau:

Trong đó:





R = 6731 (km) (Bán kính Trái Đất)

Hãy khoảng cách Haversine giữa hai địa điểm bất kỳ.

**Input**

Kinh độ và vĩ độ tương ứng của hai địa điểm, mỗi địa điểm được viết trên một dòng, mỗi giá trị cách nhau bởi khoảng trắng và là các số thực (float).

**Output**

Khoảng cách Haversine giữa hai địa điểm, kết quả được in ra với hai số sau dấu phẩy.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 105.96 10.21  107.17 16.79 | 4686.47 |

**BÀI 10.** [**LỚP TAM GIÁC VUÔNG**](https://icpc.ptit.edu.vn/student/question/PYCK52)

Khai báo lớp Tam giác với ba thuộc tính là 3 cạnh của một tam giác. Viết các phương thức phù hợp để kiểm tra xem 3 cạnh đó có tạo thành một tam giác vuông hay không. Nếu có, tính chu vi và diện tích của tam giác đó.

**Input**

3 cạnh của một tam giác, được viết trên một dòng, cách nhau bởi một khoảng trắng.

**Output**

* Nếu 3 cạnh của tam giác không tạo thành một tam giác vuông thì in ra INVALID
* Nếu 3 cạnh tạo thành một tam giác vuông, in ra chu vi và diện tích của tam giác.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 4 5 | 12 6 |
| 5 9 12 | INVALID |

# **BÀI 11.** [**CẶP SỐ THÂN THIẾT**](https://icpc.ptit.edu.vn/student/question/PYCK14)

Hai số tạo thành một cặp số thân thiết khi chúng tuân theo quy luật: Số này bằng tổng tất cả các ước của số kia (trừ chính số đó) và ngược lại.

Ví dụ: Số 220 ngoài bản thân nó ra, nó còn có 11 ước số là 1, 2, 4, 5, 10, 11, 20, 44, 55 và 110. Tổng của 11 ước số này vừa đúng bằng 284.

Ngược lại, số 284 ngoài bản thân nó, nó còn 5 ước số khác là: 1, 2, 4, 71, 142, tổng của chúng cũng vừa đúng bằng 220.

Viết chương trình kiểm tra xem hai số có tạo nên một cặp số thân thiết hay không.

**Input.** Hai số cần kiểm tra, mỗi số cách nhau một khoảng trắng

**Output.** In ra YES nếu hai số kiểm tra tạo nên một cặp số thân thiết, ngược lại in ra NO

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 220 284 | YES |
| 1184 1210 | YES |
| 1234 4321 | NO |

# **BÀI 12.** [**DANH SÁCH TUPLES**](https://icpc.ptit.edu.vn/student/question/PYCK22)

Cho một danh sách A[ ] gồm N tuples, mỗi tuple không quá 10 phần tử. Hãy kết hợp các tuples có phần tử đầu tiên giống nhau.

**Input**

* Dòng đầu là số N (N < 100)
* N dòng tiếp theo là N tuples của mảng A, mỗi tuple được viết trên một dòng

**Output**

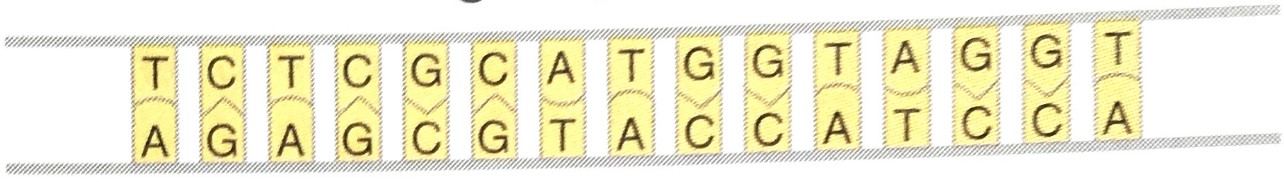
* Mảng gồm các tuples sau khi đã kết hợp.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | [(5, 6, 7, 8), (6, 5, 8, 7)] |
| 5 6 |  |
| 5 7 |  |
| 5 8 |  |
| 6 5 8 7 |  |
| 3 | [(6, 7, 10, 8)] |
| 6 7 |  |
| 6 10 |  |
| 6 8 |  |

**BÀI 13.** [**DELTA DELETION VARIANT**](https://icpc.ptit.edu.vn/student/question/PYCK13)

Biến thể Delta của virus SARS-CoV-2 được phát hiện lần đầu tiên ở Ấn Độ và tháng 10/2020. Biết rằng, cứ 11 ngày trong quần thể virut xảy ra một đột biến điểm (Nucleotid/ base).

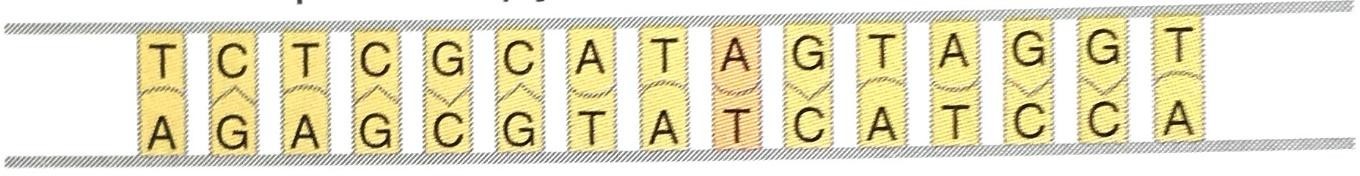
Đây là thuật ngữ của di truyền học phân tử, trong tiếng nước ngoài gọi là Point Mutation (tiếng Anh) dùng để chỉ một loại đột biến gen chỉ xảy ra ở một “điểm” trên phân tử di truyền DNA. “Điểm” này tương ứng với một cặp Nulceotide duy nhất.

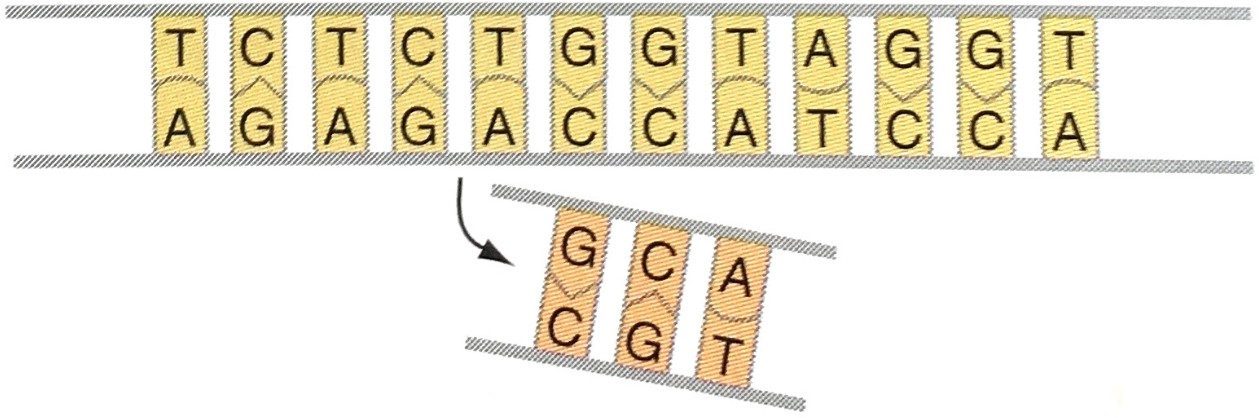
*Hình 1: Bộ gen ban đầu của Virus*

Có ba loại đột biến điểm – Point Mutation là:

* Đột biến thay thế - Substitution
* Đột biến xóa – Deletion
* Đột biến thêm - Insertion

Đột biến xóa - Deletion:



Xóa một số cặp:

Biết Bộ gen (hay hệ gen) của virut có chiều dài (kích thước) làm tròn là 30.000 Nucleotides / Bases; các đột biến hoàn toàn ngẫu nhiên và với xác suất là như nhau. Giả sử tất cả các đột biến điểm mới của biến thể Delta đều là Đột biến xóa – Deletion ở một cặp.

Gợi ý:

Một đột biến có thể xảy ra tại 1 cặp bất kỳ trên 30.000 cặp Nucleotides nên xác suất 2 đột biến giống nhau là 1/30.000 ~ 0 coi như các đột biến sinh ra là khác nhau.

* Sau 11 ngày biến thể Delta ban đầu D0 trên toàn bộ quần thể sẽ tạo ra một đột biến điểm là D01 lúc này ta có 1 đột biến điểm Đột biến xóa - Deletion.
* Sau 11 ngày tiếp theo quần thể Delta D0 sẽ tạo ra thêm 1 biến thể mới là D02 và quần thể D01 sẽ tạo ra them 1 đột biến điểm là D012 vậy lúc này ta sẽ có 3 đột biến điểm là D01, D02 và D012….
* …..

Viết chương trình tính số lượng đột biến điểm kiểu Đột biến xóa - Deletion mà biến thể delta đã có sau ngày 1/10/2020.

**Input:**

Dòng đầu tiên số nguyên N (1 ≤ N ≤ 10) là số tập test.

N dòng tiếp theo, mỗi dòng gồm một số nguyên n có giá trị không vượt quá 365 (0 < n <366) là số ngày tính từ 1/10/2020 thời điểm biến thể Delta bắt đầu xuất hiện.

**Output**

In ra một số nguyên là số lượng đột biến điểm của biến thể Delta.

Trường hợp nếu số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra INVALID INPUT

Bộ test nào có giá trị lớn hơn 365 in ra dòng đó là INVALID INPUT

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | 0 |
| 9 | INVALID INPUT |
| 480 | 1 |
| 19 | 1023 |
| 120 |  |

# **BÀI 13.** [**QUY ĐỔI ĐIỂM**](https://icpc.ptit.edu.vn/student/question/PYCK54)

Điểm thi môn Lập trình Python của một sinh viên bao gồm 4 điểm thành phần: Điểm chuyên cần, Điểm kiểm tra thường xuyên, Điểm bài tập lớn và Điểm thi cuối kỳ. Điểm chuyên cần chiếm 10% tổng điểm, điểm kiểm tra thường xuyên chiếm 10% tổng điểm, điểm bài tập lớn chiếm 20% tổng điểm, điểm cuối kỳ chiếm 60% tổng điểm. Tổng điểm được tính là tổng có trọng số của 4 điểm thành phần trên.

Điểm của sinh viên sẽ được quy đổi ra thang điểm chữ như sau:

|  |  |  |
| --- | --- | --- |
| **Điểm thang 10** | **Điểm chữ** | **Xếp loại** |
| 8.5 - 10 | A | Giỏi |
| 7.0 - 8.4 | B | Khá |
| 5.5 - 6.9 | C | Trung bình |
| 4.0 - 5.4 | D | Trung bình kém |
| Dưới 4.0 | F | Kém |

Viết chương trình khai báo lớp Sinh viên gồm các thông tin: Họ tên, Điểm chuyên cần, Điểm kiểm tra thường xuyên, Điểm bài tập lớn, Điểm thi cuối kỳ.

Đọc thông tin 1 sinh viên từ bàn phím và in ra màn hình 3 thông tin: Họ tên, Điểm chữ và Xếp loại.

**Input**

Gồm 5 dòng lần lượt, mỗi dòng ghi 1 thông tin: Họ và tên, Điểm chuyên cần, Điểm kiểm tra thường xuyên, Điểm bài tập lớn, Điểm thi cuối kỳ.

Họ tên không quá 50 chữ cái. Các giá trị điểm là số thực (float).

**Output**

Ghi ra Họ tên, Điểm chữ và Xếp loại. Mỗi thông tin được in trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Nguyen Van Hung | Nguyen Van Hung |
| 10 | B |
| 7 | Kha |
| 6.5 |  |
| 7 |  |

# **BÀI 14.** [**FIRST SECOND AND LAST OCCURRENCE**](https://icpc.ptit.edu.vn/student/question/PYCK115)

Cho một chuỗi có thể chứa một chữ ‘t’ không phân biệt viết hoa hay viết thường.

In chỉ mục của lần xuất hiện đầu tiên, thứ hai và cuối cùng của t.

Nếu ký tự t chỉ xuất hiện một lần thì chỉ in ra chỉ số của nó trong 1 và nếu chuỗi không chứa ký tự t thì in -1.

**Input.** Dòng đầu tiên là n bộ test.n dòng tiếp theo là các chuỗi.

**Output.** n dòng ghi các chỉ số theo yêu cầu

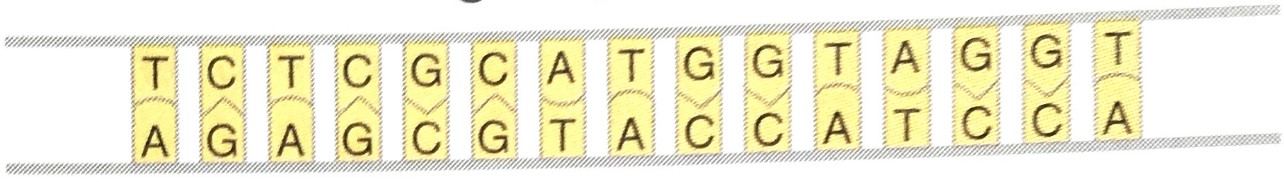
**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | 0 7 9 |
| toiyeuPTIT | 2 3 3 |
| CNTT1 | 2 18 26 |
| Python\_Programing\_Test\_ptit | -1 |
| chucmungnammoi |  |

**BÀI 15.** [**DELTA INSERTION VARIANT**](https://icpc.ptit.edu.vn/student/question/PYCK15)

COVID-19 DELTA INSERTION VARIANT

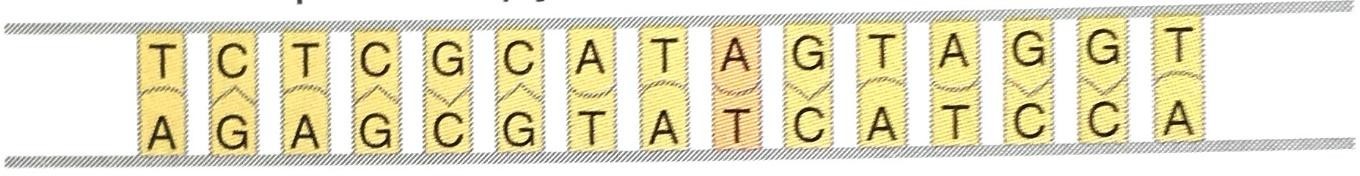
Biến thể Delta của virus SARS-CoV-2 được phát hiện lần đầu tiên ở Ấn Độ và tháng 10/2020. Biết rằng, cứ 11 ngày trong quần thể virut xảy ra một đột biến điểm (Nucleotid/ base).

Đây là thuật ngữ của di truyền học phân tử, trong tiếng nước ngoài gọi là Point Mutation (tiếng Anh) dùng để chỉ một loại đột biến gen chỉ xảy ra ở một “điểm” trên phân tử di truyền DNA. “Điểm” này tương ứng với một cặp Nulceotide duy nhất.

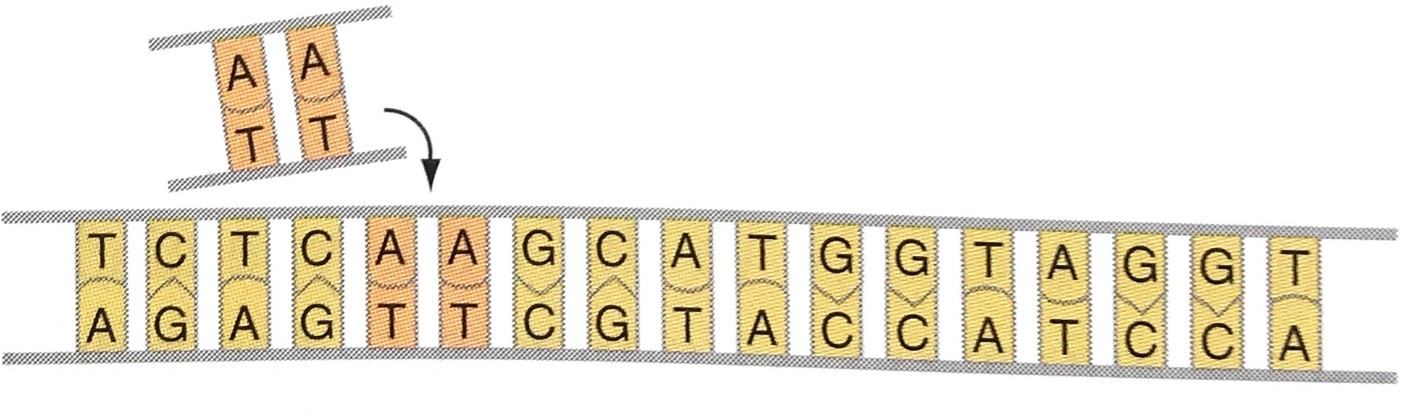
*Hình 1: Bộ gen ban đầu của Virus*

Có ba loại đột biến điểm – Point Mutation là:

* Đột biến thay thế - Substitution
* Đột biến xóa – Deletion
* Đột biến thêm – Insertion Đột biến thêm – Insertion:



Thêm một số cặp:



Biết Bộ gen (hay hệ gen) của virut có chiều dài (kích thước) làm tròn là 30.000 Nucleotides / Bases; các đột biến hoàn toàn ngẫu nhiên và với xác suất là như nhau. Giả sử tất cả các đột biến điểm mới của biến thể Delta đều là Đột biến thêm Insertion ở một cặp.

Gợi ý:

Một đột biến có thể xảy ra tại 1 cặp bất kỳ trên 30.000 cặp Nucleotides nên xác suất 2 đột biến giống nhau là 1/30.000 ~ 0 coi như các đột biến sinh ra là khác nhau.

* Sau 11 ngày biến thể Delta ban đầu D0 trên toàn bộ quần thể sẽ tạo ra một đột biến điểm là D01 lúc này ta có 1 đột biến điểm Đột biến thêm – Insertion.
* Sau 11 ngày tiếp theo quần thể Delta D0 sẽ tạo ra thêm 1 biến thể mới là D02 và quần thể D01 sẽ tạo ra them 1 đột biến điểm là D012 vậy lúc này ta sẽ có 3 đột biến điểm là D01, D02 và D012….
* …..

Viết chương trình tính số lượng đột biến điểm kiểu Đột biến thêm – Insertion mà biến thể delta đã có sau ngày 1/10/2020.

**Input:**

Dòng đầu tiên số nguyên N (1 ≤ N ≤ 10) là số tập test.

N dòng tiếp theo, mỗi dòng gồm một số nguyên có giá trị không vượt quá 365 là số ngày tính từ 1/10/2020 thời điểm biến thể Delta bắt đầu xuất hiện.

**Output**

In ra một số nguyên là số lượng đột biến điểm của biến thể Delta.

Trường hợp nếu số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra INVALID INPUT Bộ test nào có giá trị lớn hơn 365 in ra dòng đó là INVALID INPUT

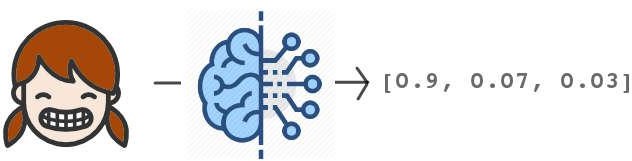
**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1 |
| 17 | 7 |
| 38 | INVALID INPUT |
| 450 | 1023 |
| 113 | 3 |
| 30 |  |

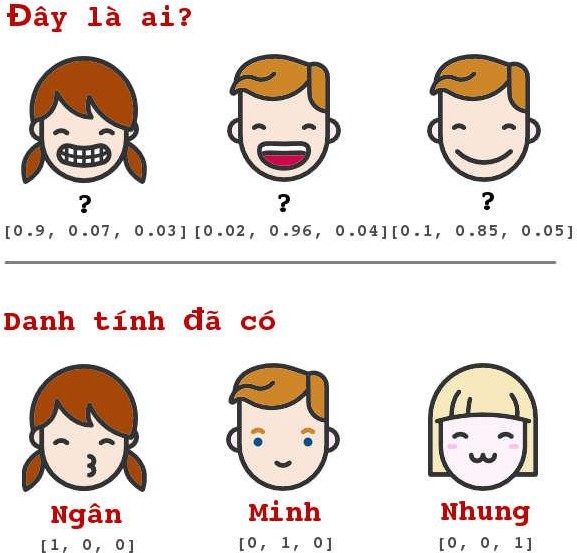
**BÀI 16.** [**EUCLIDEAN DISTANCE**](https://icpc.ptit.edu.vn/student/question/PYCK23)

Ứng dụng khoảng cách Euclide của hai vector nhận diện mỗi tấm ảnh đưa vào là khuôn mặt của ai (danh tính) bằng công thức toán khoảng cách của hai vector n chiều:

Trong bài toán xử lý ảnh, đầu ra của giải thuật nhận diện khuôn mặt là 1 vector đặc trưng. Mỗi ảnh trong thư viện ảnh đã có và ảnh cần nhận diện sẽ chạy qua giải thuật trích đặc trưng (deep learning) để có vector đặc trưng tương ứng.



Như vậy ta có vector đặc trưng của mỗi khuôn mặt cần nhận diện. Hình minh họa bên dưới:



Bằng việc so sánh khoảng cách vector đặc trưng của mỗi hình ảnh cần dự đoán Đây là ai? với Danh tính đã có bằng cách tính khoảng cách giữa hai vector dùng Euclidean như công thức ở mục phía trên:

Tính Euclidean distance

Bài toán trên đã được số hóa thành:

Ngân = [1, 0, 0]

Minh = [0, 1, 0]

Nhung = [0, 0, 1]

Có một hình ảnh A sau khi được mã hóa là A = [0.9, 0.07, 0.03]

So trùng các vector A với các danh tính đã có Ngân, Minh, Nhung:



d(A, Ngân) = = 0.1257



d(A, Minh) = 1.2945



d(A, Nhung) = 1.3251

Kết luận: A là Ngân vì có có khoảng cách là nhỏ nhất.

Viết chương trình tính độ tương đồng cosin của hình ảnh mới với những người trong CSDL và in ra người có độ tương đồng lớn nhất trong hệ thống nhận diện danh tính khi biết vector đặc trưng của mỗi người và một bức ảnh mới A sau khi được mã hóa.

**Input:**

Dòng đầu tiên số nguyên N là tổng số dòng trong bộ test (N > 3).

Dòng thứ 2 là số tự nhiên n số chiều của vector đặc trưng cho khuôn mặt (n < 1025)

Dòng thứ 3 là gồm n số là chỉ số cho từng thành phần của vector đặc trưng cho một bức ảnh mới A

N-3 dòng còn lại bao gồm string Sn đầu tiên là tên người và sau đó là n số tương ứng với n chỉ số cho từng thành phần của vector đặc trưng cho người Sn đó

**Output:**

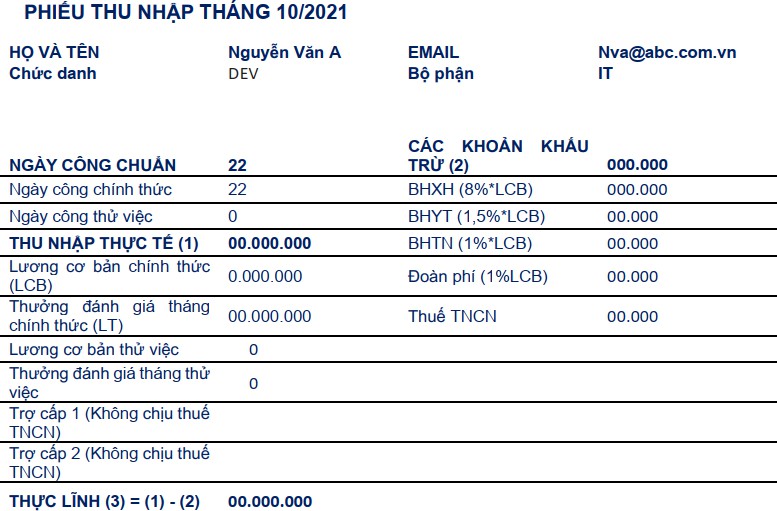
In ra tên người có vector tương đồng lớn nhất với A và in ra danh sách chỉ số độ tương đồng cosin của hình ảnh mới với những người trong CSDL làm tròn đến số thứ 4 sau dấu phẩy.

Trường hợp số nguyên N vượt quá < 4 hoặc n > 1024 in ra INVALID INPUT

Ví dụ

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 | Ngan [0.1257, 1.2945, 1.3251] |
| 3 |  |
| 0.9 0.07 0.03 |  |
| Ngan 1 0 0 |  |
| Minh 0 1 0 |  |
| Nhung 0 0 1 |  |

**BÀI 17.** [**MONTHLY TAX**](https://icpc.ptit.edu.vn/student/question/PYCK45)

****

Trên đây là Phiếu thu nhập của một người trong một tháng bao gồm:

* Thu nhập thực tế (1)
* Các khoản khấu trừ (2)
* Thực lĩnh (3)

Thu nhập thực tế (1) = Lương cơ bản (LCB) + Lương thưởng đánh giá tháng (LT)

Các khoản khấu trừ (2) = BHXH + BHYT + BHTN + ĐP + TNCN

Thực lĩnh (3) = Thu nhập thực tế (1) - Các khoản khấu trừ (2)

BHXH (Bảo hiểm xã hội) = 8%\*LCB

BHYT (Bảo hiểm y tế) = 1,5%\*LCB

BHTN (Bảo hiểm thất nghiệp) = 1%\*LCB

ĐP (Đoàn phí) = 1%\*LCB

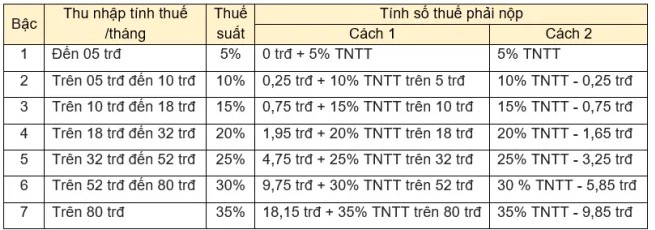
Thuế TNCN (Thuế Thu nhập cá nhân) = Thu nhập chịu thuế (TNTT)\*Thuế suất

TNTT = Thu nhập thực tế (1) – Các khoản giảm trừ - Tiền đóng bảo hiểm

Tiền đóng bảo hiểm (BH) = BHXH + BHYT + BHTN

Các khoản giảm trừ = Giảm trừ bản thân (11.000.000 vnđ) + Người phụ thuộc (4.400.000vnđ/ 1 người)

Dựa vào điều kiện của Thu nhập chịu thuế (TNTT) như bảng dưới đây ta sẽ có cách tính Thuế TNCN:

Ví dụ: TNTT = 6.000.000vnđ thì thuế TNCN = 250.000 + 10\*(TNTT – 5.000.000) = 350.000vnd

Viết chương trình in ra số tiền Bảo hiểm (BH) và Thuế TNCN của một người trong một tháng biết Thu nhập thực tế (1) và LCB của người đó hàng tháng và người này không có người phụ thuộc.

**Input:**

Dòng đầu tiên số nguyên N (1 ≤ N ≤ 10) là số tập test.

N dòng tiếp theo, mỗi dòng gồm 2 số m, n > 0.

Trong đó m là Thu nhập thực tế (1) và n là LCB

**Output:**

In ra 2 số là tiền tiền đóng bảo hiểm (BH) và thuế TNCN (lấy đến phần nguyên) trong một tháng của người Lao động trên

Trường hợp số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra INVALID INPUT

Bộ test nào có giá trị nhỏ hơn 1000 in ra dòng đó là INVALID INPUT

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  15700000 6300000  13300000 5700000  100000000 7700000  1000 123 | 661500 201925 598500 85075 808500 21017025 INVALID INPUT |

**BÀI 18.** [TUYỂN NHÂN VIÊN](https://icpc.ptit.edu.vn/student/question/PYCK56)

Doanh nghiệp X cần tuyển một số nhân viên mới. Bài thi tuyển có hai phần: lý thuyết và thực hành. Sau khi tính điểm trung bình, các thí sinh sẽ được xếp thành 4 loại:

* Nếu điểm dưới 5 -> TRUOT
* Nếu điểm lớn hơn hoặc bằng 5 nhưng nhỏ hơn 8 -> CAN NHAC
* Nếu điểm từ 8 đến 9.5 -> DAT
* Nếu điểm lớn hơn 9.5 -> XUAT SAC

Điểm các bài thi lý thuyết và thực hành đều là số thực trong phạm vi từ 0 đến 10. Tuy nhiên, khi nhập điểm các bài thi, cán bộ tuyển dụng có thể quên mất dấu . phân cách phần nguyên và phần thập phân. Do đó nếu điểm ghi là 78 thì cần được hiểu là 7.8

Hãy sắp xếp danh sách thí sinh đã được xếp loại theo điểm trung bình giảm dần. Nếu có nhiều thí sinh bằng điểm nhau, hãy sắp xếp tên theo chiều tăng dần trên bảng chữ cái.

**Input**

Dòng đầu ghi số thí sinh. Mỗi thí sinh ghi trên 3 dòng lần lượt là:

* Họ và tên (xâu ký tự độ dài không quá 100)
* Điểm lý thuyết
* Điểm thực hành

**Output**

Ghi ra danh sách thí sinh đã sắp xếp, mỗi thí sinh gồm 3 thông tin: họ tên, điểm trung bình (với 2 số phần thập phân) và xếp loại. Mỗi thông tin cách nhau một khoảng trống.

**Ví dụ**

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | Nguyen Thai Binh 6.00 CAN NHAC |
| Nguyen Thai Binh | Phan Van Duc 5.60 CAN NHAC |
| 45 | Le Cong Hoa 4.25 TRUOT |
| 75 |  |
| Le Cong Hoa |  |
| 4 |  |
| 4.5 |  |
| Phan Van Duc |  |
| 56 |  |
| 56 |  |

# **BÀI 19. TÁCH CHUỖI**

Nhập chuỗi S có độ dài không quá 100. Một từ được định nghĩa là một dãy ký tự không có khoảng trống. Hãy tách chuỗi S thành các từ, tìm ra từ có độ dài lớn nhất và nhỏ nhất của S,kèm theo đó là độ dài tương ứng.

**Input.** Chỉ có một dòng ghi xâu S (độ dài không quá 100).

**Output.**

* In ra từ có độ dài lớn nhất và nhỏ nhất, kèm theo là độ dài tương ứng của chúng.
* Kết quả của mỗi từ được viết trên một dòng.
* Nếu có nhiều từ có độ dài bằng nhau, hãy sắp xếp kết quả theo chiều tăng dần của bảng chữ cái.

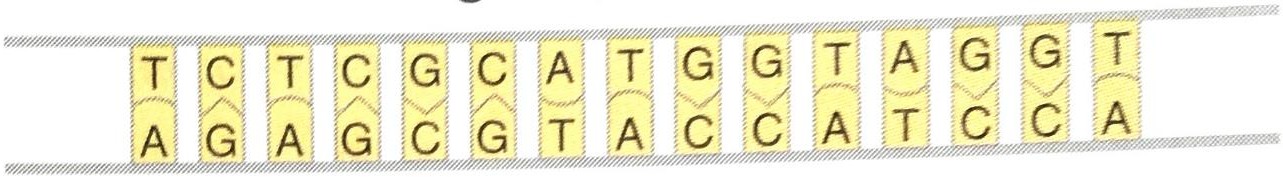
**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Hom nay thi cuoi ky mon Lap trinh Python | Python 6  ky |
| Hom nay troi mua | troi 4  Hom 3  mua 3  nay 3 |

# **BÀI 19. COVID-19 OMICRON SUBSTITUTION VARIANT**

Biến thể Omicron của virus SARS-CoV-2 được phát hiện lần đầu tiên ở Nam Phi vào ngày 24 tháng 11 năm 2021 với khả năng hoạt hóa rất mạnh. Biết rằng, cứ 3 ngày trong quần thể virut xảy ra một đột biến điểm (Nucleotid/ base).

Đây là thuật ngữ của di truyền học phân tử, trong tiếng nước ngoài gọi là Point Mutation (tiếng Anh) dùng để chỉ một loại đột biến gen chỉ xảy ra ở một “điểm” trên phân tử di truyền DNA. “Điểm” này tương ứng với một cặp Nulceotide duy nhất.

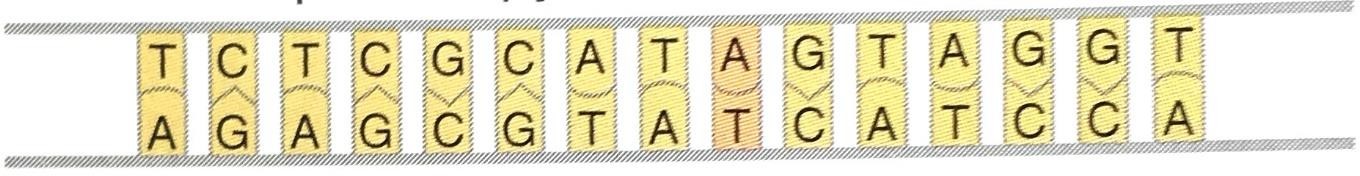


***Hình 1: Bộ gen ban đầu của Virus***

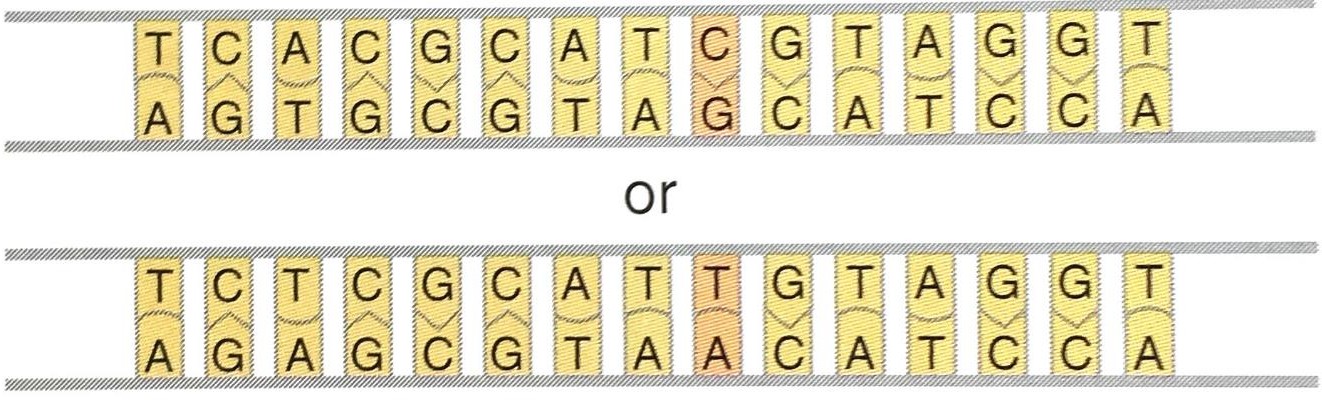
Có ba loại đột biến điểm – Point Mutation là:

* Đột biến thay thế - Substitution
* Đột biến xóa – Deletion
* Đột biến them – Insertion

**Đột biến thay thế - Substitution:**



Thay thế thế một cặp số AT thành GC hoặc thay đổi thứ tự AT thành TA và GC thành CG:



Biết Bộ gen (hay hệ gen) của virut có chiều dài (kích thước) làm tròn là 30.000 Nucleotides / Bases; các đột biến hoàn toàn ngẫu nhiên và với xác suất là như nhau. Giả sử tất cả các đột biến điểm mới của biến thể Omicron đều **là đột biến thay thế Substitution** và thay thế duy nhất 1 cặp.

Gợi ý:

Một đột biến có thể xảy ra tại 1 cặp bất kỳ trên 30.000 cặp Nucleotides nên xác suất 2 đột biến giống nhau là 1/30.000 ~ 0 coi như các đột biến sinh ra là khác nhau.

* Sau 3 ngày biến thể Omicron ban đầu Om0 trên toàn bộ quần thể sẽ tạo ra một đột biến điểm là Om01 lúc này ta có 1 **đột biến điểm kiểu Substitution**.
* Sau 3 ngày tiếp theo quần thể Omicron Om0 sẽ tạo ra thêm 1 biến thể mới là Om02 và quần thể Om01 sẽ tạo ra them 1 đột biến điểm là Om012 vậy lúc này ta sẽ có 3 đột biến điểm là Om01, Om02 và Om012.
* …..

Viết chương trình tính số lượng **đột biến điểm kiểu thay thế Substitution** mà biến thể Omicron đã có sau ngày 24/11/2021.

**Input:**

Dòng đầu tiên số nguyên N (1 ≤ N ≤ 10) là số tập test.

N dòng tiếp theo, mỗi dòng gồm một số nguyên n có giá trị không vượt quá 365

(0< n <366) là số ngày tính từ 24/11/2021 thời điểm biến thể Omicron bắt đầu xuất hiện.

**Output**

In ra một số nguyên là số lượng đột biến điểm của biến thể Omicron.

Trường hợp nếu số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra **INVALID INPUT**

Bộ test nào có giá trị lớn hơn 365 in ra dòng đó là **INVALID INPUT**

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 5  7  380  18  113  30 | 3  INVALID INPUT  63  137438953471  1023 |

# **BÀI 20.** [**PHẦN DƯ LỚN NHẤT**](https://icpc.ptit.edu.vn/student/question/PYCK24)

Cho K danh sách, danh sách thứ i thuộc K có Ni phần tử. Cho hàm f(X) = X2. Chọn từ mỗi danh sách một phần tử sao cho giá trị của biểu thức sau là lớn nhất:

S = (f(X1) + f(X2) + … + f(Xk)) % M

Trong đó *Xi* được chọn từ danh sách thứ *i*. % là phép chia lấy phần dư.

**Lưu ý:** Cần lấy chính xác một phần tử từ mỗi danh sách, không nhất thiết phải lấy phần tử lớn nhất.

**Điều kiện:**

* 1 K 7
* 1 M 1000
* 1 Ni 7

**Input:**

* Dòng đầu là K danh sách và số M, cách nhau bởi khoảng trắng
* K dòng tiếp theo là các danh sách.

**Ouput**

* Giá trị *Smax*
* Nếu các giá trị *K, M, Ni* không thỏa mãn điều kiện, in ra **INVALID INPUT**

**Ví dụ:**

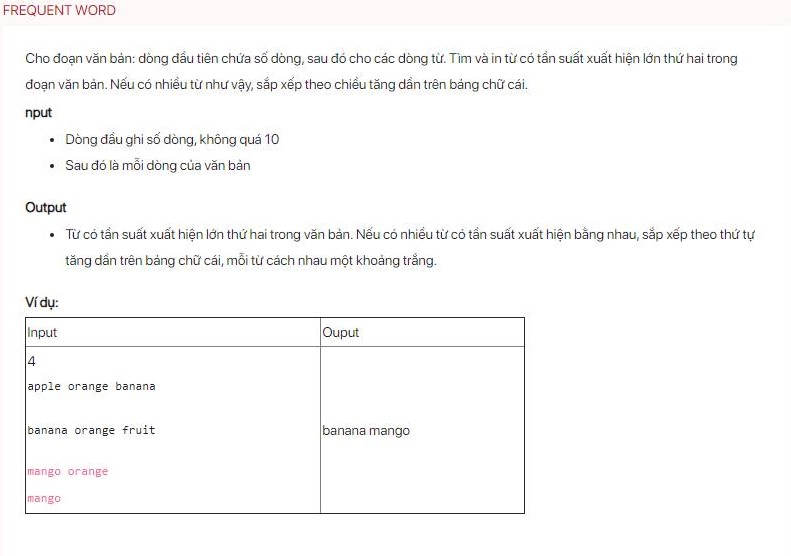
|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 1000  2 5 4  3 7 8 9  5 5 7 8 9 10 | 206 |

**Giải thích:**

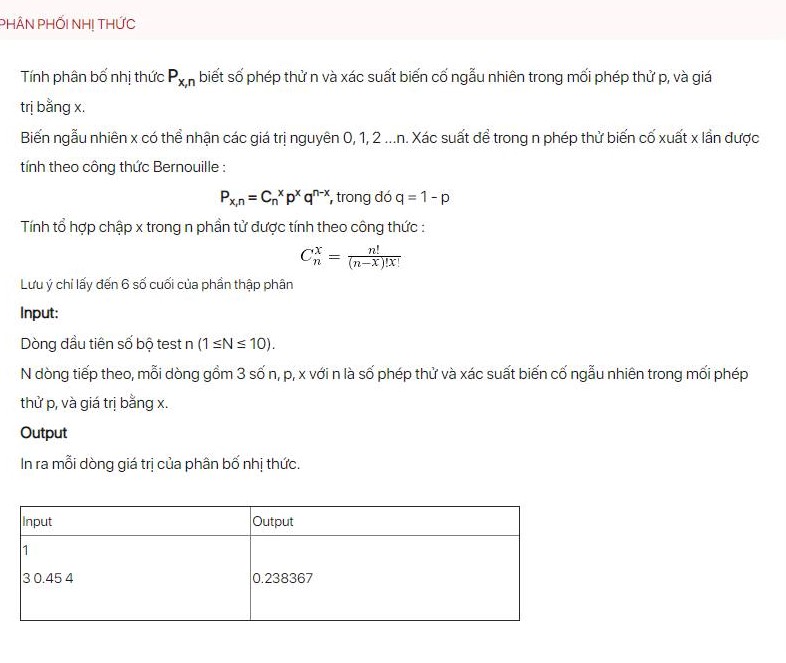
Lấy phần tử 5 từ danh sách thứ nhất, phần tử 9 từ danh sách thứ hai và phần tử 10 từ danh sách thứ ba sẽ thu được giá trị *Smax*:

*Smax* = (52 + 92 + 102) % 1000 = 206

# **BÀI 21. FREQUENT WORD**



# **BÀI 22. PHÂN PHỐI NHỊ THỨC**



# **BÀI 22. CHỈ SỐ TF-IDF**

