**ĐỀ 5 - Ngày 17 - 8 – 2022**

**Thời gian làm bài 150 phút**

**Bài 1. Giả thuyết Goldbach**

Giả thuyết Goldbach do nhà toán học người Đức Christian Goldbach (1690-1764) nêu ra vào năm 1742 trong một lá thư gửi tới Leonhard Euler, là một trong những bài toán lâu đời và nổi tiếng còn chưa giải được trong lý thuyết số nói riêng và toán học nói chung. Giả thuyết phỏng đoán rằng: Mọi số tự nhiên chẵn lớn hơn 2 có thể biểu diễn bằng tổng của hai số nguyên tố. Trong bài toán này bạn được cho một số tự nhiên chẵn n𝑛, bạn hãy đếm số lượng cặp số nguyên tố a,b (a≤b) mà a+b=n.

**Dữ liệu** • Gồm một dòng chứa một số tự nhiên chẵn n𝑛.

**Kết quả** • Gồm dòng chứa một số là số lượng cặp số nguyên tố a,b mà a≤b và a+b=n

Ví dụ:

|  |  |
| --- | --- |
| **Goldbach.inp** | **Goldbach.out** |
| 10 | 2 |

***Ràng buộc***

* Có 50% test ứng với 50% số điểm có 𝑛≤103;
* Có 50% test còn lại ứng với 50% số điểm có n≤106

**Bài 2. Tìm số k**

Cho xâu chứa các chữ cái thuộc từ A đến Z. Tìm số K nhỏ nhất mà mỗi xâu con K chữ cái liên tiếp bất kỳ chỉ xuất hiện đúng 1 lần trong xâu đã cho.

Dữ liệu:

Dòng đầu chứa số N (1≤*N*≤100)

Dòng tiếp theo chứa N ký tự trong băng chữ cái từ A đến Z.

Kết quả:

Đưa ra một số nguyên K là số tìm được.

Vi dụ

|  |  |
| --- | --- |
| **timk.inp** | **timk.out** |
|  |  |
| 7  ABCDABC | 4 |

**Giải thích đề:**

Nếu k=1 thì có các xâu “A”, “B”, “C”, “D” lặp lại

K=2 thì có xâu “AB”, “BC” lặp lại

K=3 có xâu “ABC” LẶP LẠI

K= 4 là thỏa mãn

# Bài 3. Tên trộm

Ở vương quốc Berland, thời đó, người ta quý nhất là diêm (loại đồ vật tạo ta lửa). Một tên trộm lẻn vào kho diêm của nhà vua Berland, trong kho có *m* hòm đựng diêm. Hòm thứ *i* chứa *ai* bao diêm, với mỗi bao diêm trong hòm này có *bi* que diêm (các bao diêm là giống nhau). Ba lô của tên trộm chỉ chứa được không quá *n* bao diêm. Em hãy giúp hắn chọn lấy các bao diêm để số que diêm hắn lấy được là nhiều nhất có thể. Biết rằng tên trộm không có nhiều thời gian để mở các bao diêm.

Dữ liệu:

* Dòng đầu chứa hai số nguyên *n* (1≤ *n* ≤ 2\*108) và *m* (1 ≤ *m* ≤ 20).
* Dòng thứ *i* trong *m* dòng sau chứa hai số nguyên *ai*và *bi* (1 ≤ *ai* ≤ 108,1≤ *bi* ≤10).

Kết quả:

In ra số que diêm tối đa có thể lấy

Ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thief.inp** | **Thief.out** | **Thief.inp** | **Thief.out** |
| 7 3 5 10 2 5 3 6 | 62 | 3 3 1 3 2 2 3 1 | 7 |

# Bài 4 - Thay thế ký tự

Cho hai xâu ký tự *s* và *t* đều có *n* ký tự là các chữ cái tiếng Anh in thường. Người ta muốn thay thế các ký tự trong hai xâu để chúng giống hệt nhau. Với một phép biến đổi, ta có thể thay đổi một số chữ cái trên 2 xâu. Bạn hãy tính toán số phép biến đổi tối thiểu để hoàn thành việc này.

Chính xác là, Bạn sử dụng các phép biến đổi dạng *R*(*c1, c2*) (trong đó *c1* và *c2* là các chữ cái). Bạn có thể thực hiện một phép biến đổi nào đó với số lần tùy ý để biến đổi một chữ cái *c1* thành một chữ cái *c2* và ngược lại trên cả hai hai xâu *s* và *t*. Bạn cần tìm số phép biến đổi tối thiểu để cho *s* và *t* giống hệt nhau. Thêm nữa, bạn cần in ra chi tiết về các phép biến đổi đó. Xem ví dụ để rõ hơn.

Dữ liệu

* Dòng đầu chứa số nguyên *n* (1 ≤*n* ≤ 105) là độ dài các xâu ký tự.
* Dòng thứ hai chứa *n* chữ cái tiếng Anh in thường, mô tả xâu *s*.
* Dòng thứ ba chứa *n* chữ cái tiếng Anh in thường, mô tả xâu *t*.

Kết quả

* Dòng đầu in ra số nguyên *k* là tổng số phép biến đổi tối thiếu cần thực hiện
* Trong *k* dòng tiếp theo, mỗi dòng in ra một cặp ký tự *c1, c2* cách nhau một dấu cách để mô tả về một phép biến đổi. Các cặp này có thể in theo trật tự bất kỳ. Chú ý, các phép biến đổi có thể không phải duy nhất.

Ví dụ

|  |  |  |  |
| --- | --- | --- | --- |
| Replace.inp | Replace.out | Replace.inp | Replace.out |
| 3  abb  dad | 2 | 8  drpepper  cocacola | 7 |

Giải thích ví dụ 1:

Trong ví dụ 1, bạn có thể dùng 2 phép biến đổi: (*'a', 'd'*) và (*'b', 'a'*). Như vậy các chữ cái đầu sẽ trùng khớp khi ta sẽ thay thế chữ *'a'* bằng *'d'*. Các chữ cái thứ hai sẽ trùng khớp khi ta thay *'b'* bằng *'a'*. Các ký tự thứ ba sẽ trùng khớp khi ta thay thế *'b'* bằng *'a'* và *'a'* bằng *'d'*.