**Bài 1. Chia hết** (2*.0 điểm*)

# Cho bốn số nguyên dương *a, b, x, y (ab; a, b, x, y )*

# **Yêu cầu:** Đếm số lượng số nguyên dương thuộc đoạn [a;b] chia hết cho x hoặc chia hết cho y.

# **Dữ liệu vào:** Ghi vào file **Bai1.inp** gồm một dòng duy nhất chứa 4 số nguyên dương a, b, x, y và các số cách nhau ít nhất một khoảng trắng.

# **Dữ liệu ra:** Ghi ra file **Bai1.out**

# Một dòng duy nhất ghi số lượng số nguyên dương đếm được.

# **Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Bai1.inp** | **Bai1.out** | ***Giải thích*** |
| 2 15 3 5 | 7 | a=2, b=15, x=3, y=5 các số nguyên dương thuộc đoạn [2;15] chia hết cho 3 hoặc 5 là 3, 5, 6, 9, 10, 12, 15 nên số lượng là 7 |

**Ràng buộc: \*** *50% số test (ab ≤ )*

*\* 30% số test (ab)*

*\* 20% số test (ab)*

**Bài 2. Đếm hình vuông** (*2.0 điểm*)

Dũng có *n* đoạn thẳng. Bạn ấy thấy rằng một số đoạn thẳng có cùng chiều dài, do đó có thể xếp thành những hình vuông từ những đoạn thẳngg có độ dài bằng nhau đó.

Bạn Dũng muốn biết hình vuông lớn nhất mà có thể tạo ra bằng cách sử dụng những đoạn thẳng đó như thế nào. Vì số lương các đoạn thẳng lớn nên khó làm điều đó bằng tay. Em là một học sinh giỏi Tin học em hãy lập trình giúp bạn Dũng biết diện tích của hình vuông lớn nhất có thể được tạo thành từ các đoạn thẳng đã cho.

**Yêu cầu:** Tìm ra diện tích lớn nhất của hình vuông được tạo thành và số lượng hình vuông có diện tích lớn nhất ấy.

**Dữ liệu vào:** Ghi vào file **Bai2.inp** gồm một số nguyên dương T (0<T≤ 103) biểu thị số bộ dữ liệu kiểm tra, mỗi bộ dữ liệu gồm:

* Dòng đầu chứa số nguyên *n* biểu thị số đoạn thẳng (1*≤ n*≤103)
* Dòng tiếp theo chứa *n* số nguyên dương Ai (0< Ai ≤ ) biểu thị độ dài của mỗi đoạn thẳng.

**Dữ liệu ra:** Ghi ra file **Bai2.out**. Ứng với mỗi bộ dữ liệu vào, in ra 1 dòng gồm 2 số nguyên được ngăn cách bởi 1 kí tự trống là diện tích lớn nhất của hình vuông và số lượng hình vuông có diện tích lớn nhất như vậy. Nếu không có hình vuông nào được tạo thành thì in ra -1.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Bai2.inp** | **Bai2.out** | ***Giải thích*** |
| 3  7  5 3 2 3 6 3 3  9  2 2 2 9 2 2 2 2 2  4  3 3 3 4 | 9 1  4 2  -1 | **Test 1:** Trong dãy có 4 phần tử bằng 3 tạo thành 1 hình vuông. Diên tích của của hình vuông là: 3\*3=9.  **Test 2:** Trong dãy có 8 phần tử bằng 2 tạo thành hình 2 vuông. Diện tích hình vuông là: 2\*2=4. |

**Ràng buộc:** *\* Có 60% test với 1 <T ≤ 100 và n ≤ 103*

*\* Có 40% số test với T =1 và n ≤ 105*

**Bài 3. Xâu con** (*2.0 điểm*)

Một xâu gọi là xâu nhị phân nếu chỉ chứa hai kí tự “0” hoặc “1” . Xâu V gọi là xâu con của W nếu xâu V có độ dài khác 0 và gồm các kí tự liên tiếp trong xâu W. Ví dụ: xâu “010” có các xâu con là “0”, “1”, “01”, “010”.

**Yêu cầu:** Cho trước một giá trị K, hãy đếm xem có bao nhiêu xâu con chứa đúng K ký tự “1”.

**Dữ liệu vào:** Ghi vào file **Bai3.inp** có cấu trúc như sau:

- Dòng đầu tiên ghi số nguyên dương K (0 ≤ K < 106);

- Dòng thứ hai chứa một xâu nhị phân có độ dài không quá 106).

**Dữ liệu ra:** Ghi ra file **Bai3.out** gồm một số nguyên duy nhất là kết quả tìm được.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Bai3.inp** | **Bai3.out** | ***Giải thích*** |
| 2  01010 | 4 | Có 4 xâu chứa 2 ký tự “1” là: “101” ; “0101” ; “1010” ; “01010”. |
| 2  1111 | 3 |  |

**Ràng buộc: \****60% số test với 1*≤ *K* ≤ 102

*\* 20% số test với 10*≤*K* ≤ 254.

*\* 20% số test với K* < 106*.*

**Câu 4.** **THU HOẠCH NĂNG LƯỢNG** (2.0 điểm)

Một nhà khoa học đang nghiên cứu một dải năng lượng bí ẩn trải dài qua nhiều điểm, mỗi điểm được đánh dấu bằng một giá trị năng lượng Ai (có thể là dương hoặc âm). Họ muốn chọn một đoạn liên tục trong dải này để thu thập năng lượng tối đa.

Nhà khoa học có một thiết bị thu năng lượng hình chữ nhật với kích thước 2×k (với k ≥ 2 tùy ý). Thiết bị này có thể thu thập năng lượng từ bất kỳ đoạn nào trên dải, nhưng phải đảm bảo thiết bị nằm hoàn toàn trong phạm vi của dải.

**Yêu cầu:** Hãy giúp nhà khoa học tìm ra vị trí đặt thiết bị sao cho tổng năng lượng thu được là lớn nhất.

**Dữ liệu vào:** Từ tệp **ENERGY.INP** có cấu trúc như sau:

- Dòng 1 chứa số nguyên dương N (4 ≤ N ≤ 106) – số lượng điểm trên dải năng lượng.

- Dòng thứ hai chứa N số nguyên A1, A2,…, AN (≤ 109 với 1 ≤ i ≤ N) – giá trị năng lượng tại mỗi điểm.

**Dữ liệu ra:** Ghi vào tệp **ENERGY.OUT** gồm một số nguyên duy nhất là tổng năng lượng lớn nhất có thể thu được từ một đoạn mà thiết bị có thể bao phủ.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **ENERGY.INP** | **ENERGY.OUT** | **Giải thích** |
| 6  -4 3 -2 -6 7 2 | 2 | Đoạn thu được năng lượng cao nhất  3 -2  7 -6  Tổng năng lượng thu được từ đoạn này là:  3+ (−2) + 7 + (−6) = 2. |

**Ràng buộc:**

*Subtask 1: - Có 30% điểm tương ứng với trường hợp 4 ≤ N ≤ 300.*

*Subtask 2: - Có 30% điểm tương ứng với trường hợp 300 < N ≤ 5000.*

*Subtask 3: - Có 40% điểm tương ứng với trường hợp 5000 < N ≤ 106.*

**Câu 5. MÃ SỐ** (2.0 điểm)

Công ty XYZ muốn ghép các mã số của khách hàng thành một chuỗi số lớn nhất có thể để sử dụng trong chiến dịch quảng cáo. Các mã số sẽ được sắp xếp lại và ghép nối theo thứ tự đặc biệt để tạo thành số lớn nhất có thể.

**Yêu cầu**: Cho n mã số, hãy sắp xếp và ghép các mã số này lại sao cho chuỗi số kết quả là lớn nhất.

**Dữ liệu vào**: Trong tệp văn bản **MASO.INP** gồmdòng:

* Dòng đầu tiên chứa số nguyên dương n – số lượng mã số.
* Dòng thứ hai chứa n số nguyên dương m1, m2,...., mn  đại diện cho các mã số. Các mã số cách nhau một dấu cách.

**Dữ liệu ra**: Tệp văn bản **MASO.OUT** ghi ra số nguyên lớn nhất có thể tạo thành bằng cách sắp xếp và ghép nối các mã số. Nếu tất cả các mã số đều là 0, chỉ in ra 0.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| MASO.INP | MASO.OUT | Giải thích |
| 312 907 91 | 9190712 | Có thể sắp xếp và ghép các mã số lại để tạo ra số lớn nhất là 9190712. |

**Ràng buộc:**

#### Subtask 1: - Có 40% số điểm có 1 ≤ n ≤ 10, 1 ≤ mi ≤ 103

#### Subtask 2: - Có 30% số điểm có 1 ≤ n ≤ 100, 103 < mi ≤ 105

#### Subtask 3: - Có 30% số điểm có 1 ≤ n ≤ 10000, 105 < mi ≤ 1019