

**SỞ GD&ĐT LONG AN**  
**TRƯỜNG THPT ĐỨC HOÀ**  
**ĐỀ THI CHÍNH THỨC**

**KÌ THI TUYỂN HỌC SINH GIỎI VÒNG TRƯỜNG**  
**Môn thi: TIN HỌC - Cấp THPT**

Thời gian làm bài: 120 phút (Không kể thời gian phát đề)

(Đề thi gồm có 5 bài, 4 trang)

Học sinh tạo thư mục là họ tên và lớp (viết liền không dấu, VD: leevantach\_11T9), lưu các bài làm với tên tương ứng bai1.???, bai2.???, bai3.???, bai4.???, bai5.??? vào thư mục vừa tạo (dấu ??? được thay bằng phần mở rộng của ngôn ngữ lập trình dùng để viết chương trình, VD: bai1.cpp, bai1.pas, bai1.py, ...).

Hãy lập trình giải các bài toán bên dưới.

**Bài 1. (4 điểm) Chữ số trong chuỗi số nguyên**

Xét một chuỗi vô hạn bao gồm tất cả các số nguyên dương theo thứ tự tăng dần:

12345678910111213141516171819202122232425...

**Yêu cầu:** Hãy xử lý  $q$  truy vấn có dạng: Chữ số ở vị trí  $k$  trong chuỗi là gì?

**Ví dụ:** Với  $k = 11$  thì xuất ra 0.

**Dữ liệu vào:** Từ tập tin **bai1.in** gồm:

- Dòng đầu tiên của đầu vào là số nguyên  $q$ : số lượng truy vấn.
- $q$  dòng sau đó mô tả các truy vấn. Mỗi dòng có một số nguyên  $k$ : một vị trí được đánh chỉ số trong chuỗi.

**Dữ liệu ra:** Xuất ra tập tin **bai1.out**  $q$  dòng tương ứng với  $q$  truy vấn.

**Ràng buộc:** Ràng buộc chung:  $1 \leq q \leq 1000$ ;  $1 \leq k \leq 10^{18}$ .

Trong đó: 25% test cases có  $k \leq 10^7$ , 50% test cases có  $k \leq 10^9$ , 75% test cases có  $k \leq 10^{13}$

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

**Ví dụ:**

bai1.in	bai1.out
3	7
7	4
19	1
12	

**Bài 2. (4 điểm) Tổng hai số trong dãy**

Bạn được cho một mảng gồm  $n$  số nguyên.

**Yêu cầu:** Tìm hai giá trị (ở vị trí khác nhau) mà tổng của chúng là  $x$ .

**Dữ liệu vào:** Từ tập tin **bai2.in** gồm:

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $x$ : kích thước của mảng và tổng mục tiêu.
- Dòng thứ hai chứa  $n$  số nguyên  $a_1; a_2; \dots; a_n$ : các giá trị trong mảng.

**Dữ liệu ra:** Xuất ra tập tin **bai2.out**: In ra hai số nguyên: vị trí của các giá trị. Nếu có nhiều cách giải, bạn có thể in ra bất kỳ cách nào. Nếu không có cách giải, in ra **IMPOSSIBLE**.

**Ràng buộc:** Ràng buộc chung:  $1 \leq n \leq 2 \cdot 10^5$ ;  $1 \leq x, a_i \leq 10^9$ .

Trong đó: 25% test cases có  $n \leq 10$ ;  $x \leq 10$ ;  $a_i \leq 10$ ; 50% test cases có  $n \leq 2 \cdot 10^5$ ;  $x \leq 10^{10}$ ;  $a_i \leq 10^{10}$

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

Ví dụ:

bai2.in	bai2.out
4 8 2 7 5 1	2 4

### Bài 3. (4 điểm) Tìm số nguyên tố

Lebuk là một lập trình viên, anh ấy muốn tạo ra một số số nguyên tố cho hệ thống mã hoá của mình. Hãy giúp anh ấy! Nhiệm vụ của bạn là tạo ra tất cả các số nguyên tố giữa hai số cho trước!

**Dữ liệu vào:** Từ tập tin **bai3.in** gồm:

- Số  $t$ : số lượng trường hợp cần kiểm tra trong một dòng đơn.
- $t$  dòng tiếp theo, mỗi dòng chứa hai số nguyên  $L$  và  $U$  cách nhau bởi một dấu cách, trong đó:
  - ★  $L$  - Giới hạn dưới của khoảng.
  - ★  $U$  - Giới hạn trên của khoảng.

**Dữ liệu ra:** Xuất ra tập tin **bai3.out**: Tuần tự mỗi test, các giá trị xuất liên tục mỗi giá trị một dòng.

Đối với mỗi bài kiểm tra, đầu ra phải chứa tất cả các số nguyên tố trong khoảng  $[L; U]$  theo thứ tự tăng dần.

**Ràng buộc:** Ràng buộc chung:  $t \leq 125$ ;  $2 \leq L < U \leq 2147483647$ . Trong đó:

- 25% test cases có  $t \leq 125$ ;  $L < U < 10^6$
- 75% test cases còn lại  $t = 125$ ;  $10^7 < L < U < 10^{11}$

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

Ví dụ:

bai3.in	bai3.out
2	2
2 10	3
3 7	5
	7
	3
	5
	7

### Bài 4. (4 điểm) Phần cơm gà yêu dấu

Ckplex là một lập trình viên trẻ tiềm năng, giống như tất cả các lập trình viên khác, anh ấy thích ăn gà rán "DoubleDay", có thể mua được ở  $n$  cửa hàng khác nhau trong thành phố. Được biết rằng

giá của một phần ăn ở cửa hàng thứ  $i$  bằng  $x_i$  đồng.

Ckplix dự định mua phần ăn yêu thích của mình trong  $q$  ngày liên tiếp. Anh ấy biết, vào ngày thứ  $i$  anh ấy sẽ có thể tiêu  $m_i$  đồng. Bây giờ, đối với mỗi ngày, anh ấy muốn biết anh ấy có thể mua một phần ăn "DoubleDay" ở bao nhiêu cửa hàng khác nhau.

**Dữ liệu vào:** Từ tập tin **bai4.in** gồm:

- ★ Dòng đầu tiên của đầu vào chứa một số nguyên  $n (1 \leq n \leq 100000)$  - số cửa hàng trong thành phố bán combo gà rán yêu thích của Ckplix.
- ★ Dòng thứ hai chứa  $n$  số nguyên  $x_i (1 \leq x_i \leq 10^7)$  - giá của các phần ăn này trong cửa hàng thứ  $i$ .
- ★ Dòng thứ ba chứa một số nguyên  $q (1 \leq q \leq 100000)$  - số ngày mà Ckplix dự định mua đồ ăn.
- ★ Sau đó là  $q$  dòng mỗi dòng chứa một số nguyên  $m_i (1 \leq m_i \leq 10^9)$  - số đồng mà Ckplix có thể tiêu vào ngày thứ  $i$ .

**Dữ liệu ra:** Xuất ra tập tin **bai4.out**: In ra  $q$  số nguyên. Số thứ  $i$  sẽ bằng số cửa hàng mà Ckplix có thể mua một phần đồ ăn vào ngày thứ  $i$ .

**Ràng buộc:**  $1 \leq n, x_i, q \leq 100000$  và  $1 \leq m_i \leq 10^{12}$ . Trong đó:

- 32% *test cases* có  $n \leq 10^3$ .
- 60% *test cases* có  $n \leq 10^5; x_i \leq 10^7; q \leq 10^5; m_i \leq 10^{10}$

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

**Ví dụ:**

bai4.in	bai4.out
5	0
3 10 8 6 11	4
4	1
1	5
	10
	3
	11

**Giải thích:**

- ★ Vào ngày đầu tiên, Ckplix sẽ không thể mua phần ăn ở bất kì cửa hàng nào.
- ★ Vào ngày thứ hai, Ckplix có thể mua phần ăn ở các cửa hàng 1, 2, 3 và 4.
- ★ Vào ngày thứ ba, Ckplix chỉ có thể mua đồ ăn ở cửa hàng số 1.
- ★ Vào ngày cuối cùng, Ckplix có thể mua đồ ăn ở bất kì cửa hàng nào.

### Bài 5. (4 điểm) Kết nối

Bạn được cho một đồ thị vô hướng gồm  $n$  đỉnh. Một số được ghi trên mỗi đỉnh; số trên đỉnh  $i$  là  $a_i$ . Ban đầu không có cạnh nào trong đồ thị.

Bạn có thể thêm một số cạnh vào đồ thị này, nhưng bạn phải trả phí cho chúng. Chi phí để thêm một cạnh giữa các đỉnh  $x$  và  $y$  là  $a_x + a_y$  đồng. Cũng có  $m$  ưu đãi đặc biệt, mỗi ưu đãi được ký hiệu bằng ba số  $x; y$  và  $w$ , có nghĩa là bạn có thể thêm một cạnh kết nối các đỉnh  $x$  và  $y$  và trả  $w$  đồng cho nó. Cần nhớ rằng bạn không nhất thiết phải sử dụng các ưu đãi đặc biệt: nếu có một cặp đỉnh  $x$  và  $y$  có một ưu đãi đặc biệt liên quan đến nó, bạn vẫn có thể kết nối hai đỉnh này bằng cách trả  $a_x + a_y$  đồng cho nó.

Chi phí tối thiểu bạn phải chi để làm cho đồ thị trở nên liên thông là bao nhiêu? *Giải thích: Một đồ thị được gọi là liên thông nếu có thể đi từ bất kỳ đỉnh nào đến bất kỳ đỉnh nào khác chỉ bằng cách sử dụng các cạnh thuộc đồ thị đó.*

**Dữ liệu vào:** Từ tập tin **bai5.in** gồm:

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $m$  ( $1 \leq n \leq 2 \cdot 10^5; 0 \leq m \leq 2 \cdot 10^5$ ) số đỉnh trong đồ thị và số lượng ưu đãi đặc biệt tương ứng.
- Dòng thứ hai chứa  $n$  số nguyên  $a_i : a_1; a_2; \dots; a_n$  ( $1 \leq a_i \leq 10^{14}, \forall i \in \{1; 2; \dots; n\}$ ) - các số được ghi trên các đỉnh.
- Tiếp theo là  $m$  dòng, mỗi dòng chứa ba số nguyên  $x; y$  và  $w$  ( $1 \leq x, y \leq n; 1 \leq w \leq 10^{14}; x \neq y$ ) biểu thị một ưu đãi đặc biệt: bạn có thể thêm một cạnh kết nối đỉnh  $x$  và đỉnh  $y$ , và cạnh này sẽ có giá  $w$  đồng.

**Dữ liệu ra:** Xuất ra tập tin **bai5.out**:

In ra một số nguyên - số đồng tối thiểu bạn phải trả để làm cho đồ thị trở nên liên thông.

**Ràng buộc:** Các ràng buộc với  $n, m, a_i, x, y, w$  như đã mô tả ở dữ liệu vào. Trong đó cụ thể:

- 12.5% số test cases có  $n \leq 10; m \leq 10; a_i \leq 10^2; w \leq 10^2$ .
- 87.5% không giới hạn  $n, m$  và  $a_i \geq 10^{12}; w \geq 10^{12}$ .

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

Một số ví dụ:

bai5.in	bai5.out
3 2 1 3 3 2 3 5 2 1 1	5

bai5.in	bai5.out
4 0 1 3 3 7	16

bai5.in	bai5.out
5 4 1 2 3 4 5 1 2 8 1 3 10 1 4 7 1 5 15	18

**Lý giải:** Trong ví dụ đầu tiên, có thể kết nối 1 với 2 bằng cách sử dụng ưu đãi đặc biệt 2, và sau đó kết nối 1 với 3 mà không cần sử dụng bất kỳ ưu đãi nào.

Trong hai ví dụ tiếp theo, câu trả lời tối ưu có thể đạt được mà không cần sử dụng các ưu đãi đặc biệt.

———— HẾT ————