

<b>Thời gian làm bài</b>	150 phút (không kể thời gian phát đề)
<b>Ngày thi</b>	16/02/2024 – Ngày thi thử thứ năm
<b>Đề thi gồm</b>	04 câu, 04 trang

**Tổng quan đề thi**

	<b>Tiêu đề</b>	<b>Mã nguồn</b>	<b>Dữ liệu vào</b>	<b>Dữ liệu ra</b>
<b>Câu 1</b>	Xâu hạnh phúc	HAPPYSTR.*	HAPPYSTR.inp	HAPPYSTR.out
<b>Câu 2</b>	Số hai ước	TWODIV.*	TWODIV.inp	TWODIV.out
<b>Câu 3</b>	Số độc lạ	SODOCLA.*	SODOCLA.inp	SODOCLA.out
<b>Câu 4</b>	Tập đoạn thẳng	TAPDT.*	TAPDT.inp	TAPDT.out

**Dấu \*** được thay thế bởi **PAS**, **CPP** hoặc **PY** tương ứng với ngôn ngữ lập trình **Pascal**, **C++** hoặc **Python**

**Câu 1. Xâu hạnh phúc (5 điểm)**

<b>Mã nguồn</b>	<b>Dữ liệu vào</b>	<b>Dữ liệu ra</b>	<b>Thời gian</b>	<b>Bộ nhớ</b>
HAPPYSTR.*	HAPPYSTR.inp	HAPPYSTR.out	1 giây	1024 MB

Xâu  $s$  có độ dài  $n$  và xâu  $r$  là xâu ngược lại của nó. Xâu  $s$  là xâu hạnh phúc nếu với vị trí  $i$  ( $1 < i \leq n$ ) thì khoảng cách trên bảng chữ cái từ kí tự  $s_i$  đến  $s_{i-1}$  bằng khoảng cách trên bảng chữ cái từ kí tự  $r_i$  đến  $r_{i-1}$ .

Khoảng cách trên bảng chữ cái là khoảng cách thứ tự trên bảng chữ cái của hai kí tự (VD: khoảng cách từ ‘a’ đến ‘b’ là 1, khoảng cách từ ‘d’ đến ‘b’ là 2).

**Yêu cầu:** Cho  $n$  xâu, hãy kiểm tra xem mỗi xâu có phải là xâu hạnh phúc hay không?

**Dữ liệu vào:** Nhập từ file HAPPYSTR.inp:

- Dòng đầu tiên gồm số nguyên dương  $n$  ( $1 \leq n \leq 5$ ).
- $n$  dòng tiếp theo chứa xâu  $s$  chỉ gồm các chữ cái in thường ( $1 \leq |s| \leq 10^4$ ).

**Dữ liệu ra:** Ghi ra file HAPPYSTR.out:

- Gồm  $n$  dòng, nếu xâu  $s$  là xâu hạnh phúc thì in ra **YES**, ngược lại in ra **NO**.

**Ví dụ:**

<b>HAPPYSTR.inp</b>	<b>HAPPYSTR.out</b>	<b>Giải thích</b>
2 happy sus	NO YES	Không giải thích gì thêm

**Câu 2. Số hai ước (5 điểm)**

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
TWODIV.*	TWODIV.inp	TWODIV.out	1 giây	1024 MB

Nếu có số tự nhiên  $x$  chia hết cho số tự nhiên  $y$  thì ta nói  $y$  là ước số của  $x$ . Ví dụ: 1, 2, 3, 12 là ước của 12; còn 5, 8 thì không phải là ước của 12.

Kiệt có một danh sách gồm  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$ . Kiệt đặc biệt thích các số tự nhiên có đúng 2 ước nên sẽ thay đổi các phần tử thuộc dãy  $a$  nếu chúng có nhiều hơn 2 ước. Với mỗi phần tử  $a_i$  ( $1 \leq i \leq n$ ) có nhiều hơn 2 ước Kiệt sẽ lặp lại việc thay đổi  $a_i = a_i - 1$  cho đến khi  $a_i$  có đúng 2 ước.

**Yêu cầu:** Cho  $n$  và dãy  $a$ , hãy in ra dãy  $a$  sau khi Kiệt thay đổi.

**Dữ liệu vào:** Đọc từ tệp TWODIV.inp có cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $1 \leq n \leq 2 \times 10^6$ )
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $2 \times 10^4 \leq a_i \leq 2 \times 10^6$ )

**Kết quả:**

- Ghi ra tệp TWODIV.out dãy  $a$  sau khi thay đổi. Các số viết cách nhau một dấu cách.

**Ràng buộc bổ sung:**

- 60% số điểm có  $2 \leq n \leq 2000$ ,  $2 \leq a_i \leq 2000$ ;
- 20% số điểm có  $2000 \leq n \leq 2 \times 10^4$ ,  $2000 \leq a_i \leq 2 \times 10^4$ ;
- 20% số điểm không có giới hạn gì thêm.

**Ví dụ:**

TWODIV.inp	TWODIV.out	Giải thích
3 2 9 16	2 7 13	- 2 có sẵn 2 ước đó là 1 và 2 nên không thay đổi. - 9 có 3 ước là 1, 3, 9 nên cần phải trừ đi 2 lần để được 7 có 2 ước.

**Câu 3. Số độc lạ (5 điểm)**

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
SODOCLA.*	SODOCLA.inp	SODOCLA.out	1 giây	1024 MB

Minh trong những lúc rảnh đã nghiên cứu một số tính chất của các số nguyên dương. Cậu nhận thấy có một số số có tính chất lạ, được cậu gọi là “số độc lạ”. Những số này có một tính chất đặc biệt, đó là tổng các ước của nó trừ chính nó, lớn hơn căn bậc hai của số đó.

Cụ thể hơn, với một số nguyên dương  $N$ , gọi  $f(N)$  là tổng các ước của nó.  $N$  là “số độc lạ” khi  $(f(N) - N) > \sqrt{N}$ .

**Yêu cầu:** Cho số nguyên dương  $N$ , hỏi  $N$  có phải “số độc lạ” không?

**Dữ liệu vào:** Nhập từ file SODOCLA.inp:

- Dòng đầu chứa số nguyên dương  $q$  ( $1 \leq q \leq 10^4$ )
- $q$  dòng tiếp theo, mỗi dòng chứa số nguyên dương  $N$  ( $1 \leq N \leq 10^{12}$ )

**Dữ liệu ra:** Ghi ra file SODOCLA.out:

- Với mỗi truy vấn, in ra **YES** nếu  $N$  là “số độc lạ”, ngược lại in ra **NO**.

**Ràng buộc bổ sung:**

- 50% số điểm có  $N \leq 10^6$ .
- 50% số điểm còn lại không có giới hạn gì thêm.

**Ví dụ:**

SODOCLA.inp	SODOCLA.out	Giải thích
3	NO	- Ở truy vấn thứ nhất, $f(N) = 1 + 2 = 3$
2	YES	- Ở truy vấn thứ hai, $f(N) = 1 + 2 + 4 = 7$
4	YES	- Ở truy vấn thứ ba, $f(N) = 1 + 2 + 3 + 6 = 12$
6		

**Câu 4. Tập đoạn thẳng (5 điểm)**

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
TAPDT.*	TAPDT.inp	TAPDT.out	1,25 giây	1024 MB

Hậu trong lúc rảnh rỗi đã thách đố Minh một bài toán như sau:

Cho 1 số nguyên dương  $n$  và  $k$  đoạn thẳng trên một đường thẳng, đoạn thẳng thứ  $i$  có độ dài  $l_i$  ( $1 \leq l_i \leq n$ ). Đếm số cách đặt  $k$  đoạn thẳng này trên trục  $Ox$  của hệ tọa độ Descartes sao cho:

- Điểm kết thúc của mỗi đoạn thẳng là một số nguyên dương không vượt quá  $n$ .
- Điểm bắt đầu của mỗi đoạn thẳng là một số nguyên dương.
- Hai đoạn thẳng bất kỳ đều giao nhau (có ít nhất một điểm chung).

Hay nói cách khác, bạn cần đếm số tập  $p$  độ dài  $k$  thỏa mãn:

- Với mọi  $1 \leq i \leq k$  thì  $1 \leq p_i \leq n - l_i + 1$ .
- Với mọi  $1 \leq i < j \leq k$ , đoạn thẳng  $[p_i, p_i + l_i - 1]$  và đoạn thẳng  $[p_j, p_j + l_j - 1]$  giao nhau. Hai đoạn thẳng được coi là giao nhau nếu có ít nhất 1 điểm chung.

Minh hiện vẫn chưa tìm ra cách giải. Các bạn hãy giúp Minh nhé!

**Dữ liệu vào:** Nhập từ file TAPDT.inp:

- Dòng đầu chứa 2 số nguyên dương  $n, k$  ( $1 \leq n \leq 5000, 1 \leq k \leq 5 \times 10^5$ ).
- Dòng thứ 2 chứa  $k$  số nguyên dương  $l_1, l_2, \dots, l_k$  ( $1 \leq l_i \leq n$ ).

**Dữ liệu ra:** Ghi ra file TAPDT.out:

- In ra kết quả cần tìm. Vì kết quả có thể rất lớn nên hãy xuất ra kết quả sau khi chia lấy dư cho  $10^9 + 7$ .

**Ràng buộc bổ sung:**

- 20% số điểm có  $k = 2$ .
- 20% số điểm khác có  $1 \leq n, k \leq 5$ .
- 16% số điểm khác có  $1 \leq n, k \leq 50$ .
- 16% số điểm khác có  $1 \leq n, k \leq 200$ .
- 16% số điểm khác có  $1 \leq n, k \leq 5000$ .
- 12% số điểm còn lại không có giới hạn gì thêm.

**Ví dụ:**

TAPDT.inp	TAPDT.out	Giải thích
4 2 2 2	7	- Có 7 tập $p$ thoả mãn, gồm: $\{1; 1\}, \{1; 2\}, \{2; 1\}, \{2; 2\}, \{2; 3\}, \{3; 2\}, \{3; 3\}$ .