

## Bài 1: Tổng của $N$ phép chia.

Còn nhỏ nhưng Tý rất thích lập trình. Tý đã tìm đọc nhiều sách hay về các ngôn ngữ lập trình khác nhau và các kỹ thuật lập trình. Tý nhận ra có sự khác nhau giữa các ngôn ngữ. Với phép chia lấy phần nguyên khi chia  $N$  cho  $i$  ( $N$  và  $i$  là hai số nguyên), trong ngôn ngữ lập trình Pascal thì viết  $N \text{ div } i$ , trong ngôn ngữ lập trình Basic thì viết  $N \setminus i$ , trong C++ thì viết  $N / i$ . Nhưng trong toán học thì chỉ có một cách viết là:  $\left\lfloor \frac{N}{i} \right\rfloor$ . Sau khi hiểu rõ phép chia lấy phần nguyên Tý quyết định lập trình tìm giá trị của biểu thức:

$$S = \left\lfloor \frac{N}{1} \right\rfloor + \left\lfloor \frac{N}{2} \right\rfloor + \left\lfloor \frac{N}{3} \right\rfloor + \cdots + \left\lfloor \frac{N}{N-1} \right\rfloor + \left\lfloor \frac{N}{N} \right\rfloor$$

Bạn chọn ngôn ngữ nào để viết, còn Tý thì Tý chọn ngôn ngữ lập trình Pascal.

**Dữ liệu vào** từ tệp văn bản **SUMDIV.INP** gồm một dòng ghi số nguyên dương  $N$  ( $1 \leq N \leq 10^9$ ).

**Kết quả** ghi ra tệp văn bản **SUMDIV.OUT** gồm một dòng ghi số  $S$  tìm được.

Ví dụ:

SUMDIV.INP	SUMDIV.OUT
5	10

*Có 60% tests,  $N \leq 1000000$ .*

## Bài 2: Tìm kiếm văn bản.

Tý viết chương trình tạo ra một số nguyên có  $N$  chữ số và ghi vào một tệp văn bản. Sau đó mở tệp đó trong một chương trình soạn thảo văn bản, chọn (đánh dấu) một số chữ số đầu tiên (bên trái) và chơi trò chơi cùng với chức năng tìm kiếm văn bản *Find – Next* của chương trình soạn thảo văn bản đó. Nếu chương trình tìm thấy văn bản đã chọn ở vị trí không trùng với văn bản đã chọn thì dấu chọn sẽ di chuyển đến vị trí tìm thấy đó, nếu không thì dấu chọn sẽ không di chuyển.

Bảng sau mô tả ví dụ cho việc chọn và sự di chuyển dấu chọn sau khi thực hiện việc tìm kiếm bằng chức năng *Find – Next*.

<i>Trước khi thực hiện việc tìm kiếm</i>	<i>Sau khi thực hiện việc tìm kiếm</i>
12312312312	12312312312
12312312312	12312312312

12312312312	12312312312
12312312312	12312312312
12312312312	12312312312
12312312312	12312312312

Và Tý muốn biết có bao nhiêu chữ số (bên trái sang) cần chọn để sau khi dùng chức năng *Find – Next* thì dấu chọn sẽ di chuyển. Các chữ số được đánh số từ trái sang phải và bắt đầu từ 1, các chữ số được chọn để tìm luôn bắt đầu từ vị trí 1. Hãy cùng Tý tìm lời giải cho bài toán với trò chơi trên.

**Dữ liệu vào** từ tệp văn bản **FINDTEXT.INP** gồm một dòng ghi số nguyên  $x$  có  $N$  chữ số ( $1 \leq N \leq 10^7$ ).

**Kết quả** ghi ra tệp văn bản **FINDTEXT.OUT** gồm một dòng ghi hai số:

- Số thứ nhất là số chữ số lớn nhất mà Tý có thể chọn từ bên trái của  $x$  sao cho sau khi thực hiện chức năng *Find – Next* thì dấu chọn sẽ di chuyển.
- Số thứ hai là vị trí đầu tiên mà dấu chọn di chuyển đến trong trường hợp dấu chọn di chuyển. Trong trường hợp ngay sau khi chọn chữ số đầu tiên của  $x$  và thực hiện việc tìm bằng *Find – Next* mà dấu chọn không di chuyển thì ghi ra 0 và 1, các số ghi cách nhau một dấu cách.

Ví dụ:

FINDTEXT.INP	FINDTEXT.OUT	FINDTEXT.INP	FINDTEXT.OUT
12312312312	5 7	1234567890	0 1

- Có 30% tests,  $0 < n < 1000$ ;
- Có 30% tests,  $999 < n < 10000$ ;
- Có 40% tests,  $9999 < n < 10000000$ .

### Bài 3: Hái nấm.

Tý được mẹ giao nhiệm vụ đến thăm bà nội. Từ nhà mình đến nhà bà nội, Tý phải đi qua một khu rừng có rất nhiều nấm. Khu rừng được chia thành một lưới ô vuông kích thước  $N \times N$ , các hàng được đánh số từ trên xuống dưới bắt đầu từ 1, còn các cột được đánh số từ trái sang phải, bắt đầu từ 1. Ô giao của hàng  $i$ , cột  $j$  có tọa  $(i, j)$ . Trên mỗi ô có một số nguyên là giá trị của loại nấm trong ô đó. Tý xuất phát tại ô  $(1, 1)$  để đến nhà bà nội tại ô  $(N, N)$ . Một cách nhanh nhất là đi theo hướng sang phải hay

xuống dưới (từ ô  $(i, j)$  có thể đến ô  $(i, j+1)$  hoặc ô  $(i+1, j)$ ). Tranh thủ trên đường đi Tý sẽ hái nấm cho bà. Tuy nhiên có thể hái nhiều nấm và mất không quá nhiều thời gian Tý cho phép mình có thể đi không quá  $K$  lần sang các ô theo hướng ngược lại, tức là đi theo hướng sang trái hay lên trên (từ ô  $(i, j)$  có thể đến ô  $(i, j-1)$  hoặc ô  $(i-1, j)$ ). Khi đi qua một ô Tý hái được một lượng nấm có giá trị ghi trên ô (cả ô  $(1, 1)$ ). Một ô Tý có thể đến nhiều lần, mỗi lần đến Tý chỉ được hái đúng một lượng nấm bằng giá trị ghi trong ô. Và tất nhiên là Tý không được vượt ra khỏi phạm vi khu rừng.

Hãy tìm tổng lượng nấm lớn nhất mà Tý hái được trên hành trình từ ô  $(1, 1)$  đến ô  $(N, N)$  theo cách đi mô tả trên.

**Dữ liệu** vào từ tệp văn bản **MUSHROOM.INP** gồm:

- Dòng đầu tiên ghi 2 số nguyên dương  $N$  và  $K$ .
- $N$  dòng tiếp theo, mỗi dòng ghi  $N$  số nguyên mô tả khu rừng.

**Kết quả** ghi ra tệp văn bản **MUSHROOM.OUT** gồm một dòng ghi một số là lượng nấm lớn nhất mà Tý thu được trên hành trình từ ô  $(1, 1)$  đến ô  $(N, N)$  theo cách đi mô tả trên.

Ví dụ:

MUSHROOM.INP	MUSHROOM.OUT
3 1 1 1 0 1 1 0 1 1 1	7
MUSHROOM.INP	MUSHROOM.OUT
4 4 1 1 1 0 1 0 1 0 1 1 1 0 0 0 1 1	15

**Giải thích:**

*Ví dụ 1:* tương ứng với đường đi  $(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3)$ .

*Ví dụ 2:* tương ứng với đường đi  $(1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3) \rightarrow (3,2) \rightarrow (3,1) \rightarrow (2,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3) \rightarrow (4,3) \rightarrow (4,4)$

**Giới hạn:**  $2 \leq N \leq 1000$ , tất cả các số nguyên đã cho đều thuộc đoạn  $[-1000, 1000]$ ,  $0 \leq K \leq 100$ .

- 20% test có  $K = 0$ .
- 20% test có  $2 \leq N \leq 5, 1 \leq K \leq 5$ .
- 20% test có  $2 \leq N \leq 200, 1 \leq K \leq 100$ .