

## Bài tập kỹ thuật lập trình

### Hệ thống NTFS

Trong hệ thống NTFS, bộ nhớ phân phối cho các file theo đơn vị cluster, mỗi cluster là 4KB (tức là 4096 byte). Như vậy dù file của bạn có kích thước là 1 byte nó vẫn chiếm bộ nhớ 4KB trên đĩa từ.

Yêu cầu: cho số nguyên dương  $n$  là kích thước của file trong đơn vị byte. Hãy xác định số KB mà file đó chiếm trên đĩa từ trong hệ thống NTFS.

**Dữ liệu** vào từ tập tin văn bản NTFS.INP gồm một dòng chứa số nguyên  $n$  ( $0 \leq n \leq 10^9$ )

**Kết quả** xuất ra tập tin văn bản NTFS.OUT một số nguyên là số KB mà file đó chiếm trên đĩa từ trong hệ thống NTFS.

Ví dụ

NTFS.INP	NTFS.OUT
4097	8

## Các quả cân

Cho một chiếc cân 2 đĩa và các quả cân có khối lượng  $3^0, 3^1, 3^2, \dots$ . Hãy chọn các quả cân để có thể cân được vật có khối lượng  $N$  ( $N \leq 10^{100}$ ).

Chẳng hạn để cân vật có khối lượng  $N = 11$ , ta chọn và đặt các quả cân lên 2 đĩa như sau:

- Đĩa cân bên trái chứa các quả cân khối lượng  $3^1, 3^2$
- Đĩa cân bên phải chứa vật có khối lượng  $N = 11$  và quả cân khối lượng  $3^0$

**Dữ liệu** đọc từ tập tin văn bản WEIGHTS.INP chứa số nguyên dương  $N$ .

**Kết quả** xuất ra tập tin văn bản WEIGHTS.OUT gồm 2 dòng:

- Dòng đầu ghi số mũ của các quả cân đặt ở đĩa bên trái theo thứ tự tăng dần.
- Dòng thứ hai ghi số mũ của các quả cân đặt ở đĩa bên phải theo thứ tự tăng dần.

Biết rằng vật cần cân luôn đặt ở đĩa cân bên phải.

Ví dụ

WEIGHTS.INP	WEIGHTS.OUT
11	1 2 0

## Dòng chữ tinh ranh

Trong lớp tôi có một cậu tên Hải thường hay thay đổi bài học toán. Cách đây không lâu, cậu ta nghĩ ra một cách tiêu khiển mới, viết một dòng không có dấu cách các lũy thừa 4 của các số nguyên tố từ nhỏ đến lớn như sau:

*16826252401...*

Sau đó cậu ta trở nên thích thú đặt câu hỏi có thể xác định chữ số nào đứng ở vị trí thứ  $k$  trong dòng trên hay không? Bạn hãy giúp Hải xác định chữ số thứ  $k$  trong dãy trên nhé

Dữ liệu vào từ tập tin văn bản **chuso.inp** gồm số nguyên  $k$  ( $1 \leq k \leq 50000$ )

Kết quả xuất ra tập tin văn bản **chuso.out** là chữ số nằm ở vị trí thứ  $k$

Ví dụ:

chuso.inp	chuso.out
5	6
9	4

## Tìm số lớn nhất

Dù mang danh học sinh chuyên Tin song các bạn học sinh lớp chuyên Tin lại chẳng thích thú gì khi thầy Hùng nói về các số nhị phân. Tuy nhiên số nhị phân đóng một vai trò quan trọng trong tin học nên thầy Hùng vẫn bắt học trò của mình giải các bài toán liên quan đến hệ đếm nhị phân. Để thay đổi không khí hôm nay, thầy cho một trò chơi đúng với sở thích của các bạn chuyên Tin và có liên quan đến số nhị phân. Trò chơi như sau:

Với số tự nhiên  $N$ , ban đầu bạn cần đổi ra số nhị phân, sau đó các bạn lần lượt dịch chuyển vòng tròn dãy nhị phân, mỗi lần 1 đơn vị để nhận được các dãy nhị phân có thể. Với mỗi dãy nhị phân (có thể có chữ số 0 vô nghĩa), ta đổi chúng ra thập phân. Người nào dịch chuyển để nhận được số lớn nhất thì được thưởng kẹo.

Ví dụ với  $N = 19$  ta đổi ra dãy nhị phân 10011. Bằng cách dịch chuyển vòng tròn ta được các dãy nhị phân: 11001, 11100, 01110, 00111, 10011. Khi đổi sang thập phân ta được các giá trị 25, 28, 14, 7, 19. Vậy kết quả của trò chơi là 28.

**Yêu cầu** cho số tự nhiên  $N$ . Hãy tìm số lớn nhất  $M$  nhận được của trò chơi trên.

**Dữ liệu** đọc từ tập tin văn bản GAME.INP trong đó chứa duy nhất số  $N \leq 10^9$ .

**Kết quả** ghi ra tập tin văn bản GAME.OUT số  $M$  tìm được.

Ví dụ:

GAME.INP	GAME.OUT
19	28

## MUA VÉ

Để khuyến khích mọi người sử dụng các phương tiện giao thông trong thành phố, ngoài việc bán vé rời từng vé một với giá  $p1$  cho một vé ở hầu hết các nước đều có chế độ bán cả tập vé mỗi tập có  $k$  vé với giá  $p2$  cho mỗi tập.

Bạn đến thành phố tham quan và dự kiến sẽ đi  $n$  chuyến trên các phương tiện giao thông công cộng. Vấn đề đặt ra là nên mua vé như thế nào để tiết kiệm tiền nhất. Dĩ nhiên, bạn không hề có ý nghĩ là sẽ đi lậu vé một lần nào.

**Yêu cầu:** Cho các số nguyên dương  $n, k, p1, p2$ , các số đều có giá trị không vượt quá  $10^9$ . Nếu  $k = 1$  thì  $p1 = p2$ . Hãy tính chi phí tối thiểu cần thiết để mua vé.

**Dữ liệu:** Vào từ file văn bản TICKETS.INP gồm dòng chứa 4 số nguyên  $n, k, p1, p2$ .

**Kết quả:** Đưa ra file văn bản TICKETS.OUT một số nguyên – chi phí tối thiểu tìm được.

**Ví dụ:**

TICKETS.INP
12 10 17 120

TICKETS.OUT
154

### Phép nhân

$6 \times 9 = 42$  là không đúng trong hệ 10 nhưng đúng trong hệ 13. Cho 3 số nguyên  $p$ ,  $q$  và  $r$ . Hãy xác định cơ số  $B$  ( $2 \leq B \leq 16$ ) để biểu thức  $p \times q = r$  là đúng trong hệ cơ số  $B$ . Nếu có nhiều cơ số  $B$  thỏa mãn thì đưa ra giá trị  $B$  nhỏ nhất. Nếu không tồn tại  $B$  thì đưa ra số 0.

**Dữ liệu** đọc từ tập tin văn bản MULT.INP gồm 3 số nguyên  $p$ ,  $q$  và  $r$  ( $1 \leq p, q, r \leq 1000000$ )

**Kết quả** đưa ra tập tin văn bản MULT.OUT số nguyên  $B$ .

Ví dụ

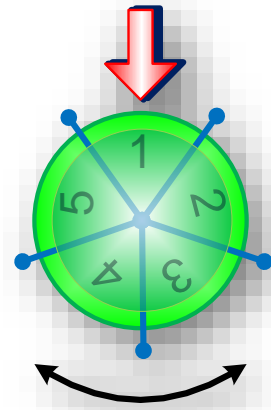
MULT.INP	MULT.OUT
6 9 42	13

## BÁNH XE MAY MẮN

Bánh xe may mắn là một phần của trò chơi truyền hình. Người chơi phải quay một bánh xe lớn theo chiều bất kỳ. Bánh xe được chia thành  $n$  hình cánh quạt bằng nhau, trên mỗi cánh quạt có ghi một số nguyên. Có một mũi tên cố định chỉ vào bánh xe. Khi bánh xe dừng, mũi tên chỉ vào cánh quạt nào thì người chơi nhận được số điểm bằng số ghi trên cánh quạt.

Một bạn trẻ tham gia trò chơi nhận thấy rằng bánh xe quay chậm dần do mũi tên chạm vào mấu của các đường phân chia cánh quạt. Nếu tốc độ quay của bánh xe là  $v$  độ/giây và mũi tên đang chỉ tới cánh quạt  $x$  thì mỗi lần chuyển sang cánh quạt bên cạnh tốc độ quay sẽ giảm đi  $k$  độ/giây. Nếu  $v \leq k$  bánh xe sẽ dừng lại và mũi tên chỉ vào cánh quạt  $x$ .

Biết được cách bố trí các số trên bánh xe người bạn trẻ của chúng ta tính được tốc độ ban đầu cần quay để đạt được số điểm cao nhất. Tốc độ ban đầu của bánh xe là số nguyên và phải nằm trong phạm vi từ  $a$  đến  $b$ .



Ví dụ,  $n = 5$ , các số ghi trên bánh xe theo chiều kim đồng hồ là 1, 2, 3, 4, 5 và mũi tên đang chỉ tới cánh quạt ghi số 1, tốc độ quay ban đầu phải nằm trong phạm vi từ 3 đến 5 độ/giây, với  $k = 2$ , ta có thể quay bánh xe theo chiều kim đồng hồ với tốc độ ban đầu là 3 hoặc 4, khi đó mũi tên sẽ dừng ở hình cánh quạt tiếp theo và số điểm đạt được là 5.

**Yêu cầu:** Cho các giá trị nguyên  $n$ ,  $a$ ,  $b$ ,  $k$  và các số ghi trên bánh xe liệt kê theo chiều kim đồng hồ bắt đầu từ số ở cánh quạt có mũi tên chỉ, mỗi số có giá trị không vượt quá 1000,  $3 \leq n \leq 100$ ,  $1 \leq a \leq b \leq 10^9$ ,  $1 \leq k \leq 10^9$ . Hãy xác định số điểm lớn nhất có thể nhận được.

**Dữ liệu:** Vào từ file văn bản WHEEL.INP:

- Dòng đầu tiên chứa số nguyên  $n$ ,
- Dòng thứ 2 chứa  $n$  số nguyên ghi trên bánh xe liệt kê theo chiều kim đồng hồ bắt đầu từ số ở cánh quạt có mũi tên chỉ,
- Dòng thứ 3 chứa 3 số nguyên  $a$ ,  $b$  và  $k$ .

**Kết quả:** Đưa ra file văn bản WHEEL.OUT một số nguyên – số điểm lớn nhất có thể nhận được.

Ví dụ:

WHEEL.INP					
5					
1	2	3	4	5	
3	5	2			

WHEEL.OUT
5

**Ghi chú:** 50% số điểm dành cho trường hợp  $1 \leq a \leq b \leq 1000$ .

## MẪU NHẬN DẠNG

Mẫu nhận dạng là xâu **P** độ dài  $n$  ( $1 \leq n \leq 9$ ) bao gồm các ký tự từ tập

**{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, ?}**

- Các ký tự từ 0 đến 9 chỉ đại diện cho chính mình.
- Ký tự **a** có thể thay thế bằng ký tự bất kỳ trong tập {0, 1, 2, 3},
- Ký tự **b** có thể thay thế bằng ký tự bất kỳ trong tập {1, 2, 3, 4},
- Ký tự **c** có thể thay thế bằng ký tự bất kỳ trong tập {2, 3, 4, 5},
- Ký tự **d** có thể thay thế bằng ký tự bất kỳ trong tập {3, 4, 5, 6},
- Ký tự **e** có thể thay thế bằng ký tự bất kỳ trong tập {4, 5, 6, 7},
- Ký tự **f** có thể thay thế bằng ký tự bất kỳ trong tập {5, 6, 7, 8},
- Ký tự **g** có thể thay thế bằng ký tự bất kỳ trong tập {6, 7, 8, 9},
- Ký tự **?** có thể thay thế bằng ký tự bất kỳ trong tập {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

**Yêu cầu:** Cho 2 mẫu **P1** và **P2** cùng độ dài. Hãy xác định số lượng xâu đồng thời có thể dẫn xuất từ cả 2 mẫu đã cho.

**Dữ liệu:** Vào từ file văn bản PATTERN.INP:

- Dòng thứ nhất chứa xâu **P1**,
- Dòng thứ 2 chứa xâu **P2**.

**Kết quả:** Đưa ra file văn bản PATTERN.OUT số lượng xâu đồng thời dẫn xuất được.

Ví dụ:

PATTERNS.INP
??
a?

PATTERNS.OUT
40



## GIAI THỪA

Giai thừa của  $n$  là tích các số nguyên từ 1 tới  $n$  và ký hiệu là  $n!$ . Ví dụ,  $4! = 1 \times 2 \times 3 \times 4 = 24$ .

Thầy giáo viết trên bảng giai thừa của một số, sau đó xóa đi một chữ số và thay vào đó là dấu “?”. Kết quả là trên bảng có một xâu  $S$  bao gồm các ký tự số và một ký tự “?”.

**Yêu cầu:** Cho xâu  $S$  có độ dài không quá 100. Hãy xác định chữ số ở vị trí dấu “?” và số nguyên  $n$  có giai thừa tương ứng với xâu đã cho. Trong trường hợp có nhiều  $n$  cùng thỏa mãn, hãy đưa ra kết quả ứng với  $n$  nhỏ nhất.

**Dữ liệu:** Vào từ file văn bản FACTORIAL.INP gồm một dòng chứa xâu  $S$ . Dữ liệu đã cho đảm bảo tồn tại nghiệm.

**Kết quả:** Đưa ra file văn bản FACTORIAL.OUT trên một dòng hai số nguyên: chữ số bị xóa và số  $n$ .

Ví dụ:

FACTORIAL. INP
1241391559253607267086228?047373375038521486354677760000000000

FACTORIAL. OUT
9 48

## TẬP SỐ

Cho số  $n$  ở hệ cơ số 10, có không quá 20 chữ số và không chứa các số 0 không có nghĩa ở đầu. Bằng cách xóa một hoặc một vài chữ số liên tiếp của  $n$  (nhưng không xóa hết tất cả các chữ số của  $n$ ) ta nhận được những số mới. Số mới được chuẩn hóa bằng cách xóa các chữ số 0 vô nghĩa nếu có. Tập số nguyên  $D$  được xây dựng bằng cách đưa vào nó số  $n$ , các số mới khác nhau đã chuẩn hóa và khác  $n$ . Ví dụ, với  $n = 1005$  ta có thể nhận được các số mới như sau:

- Bằng cách xóa một chữ số ta có các số: 5 (từ 005), 105, 105, 100;
- Bằng cách xóa hai chữ số ta có các số: 5 (từ 05), 15, 10;
- Bằng cách xóa 3 chữ số ta có các số: 5 và 1.

Tập  $D$  nhận được từ  $n$  chứa các số  $\{1005, 105, 100, 15, 10, 5, 1\}$ . Trong tập  $D$  này có 3 số chia hết cho 3, đó là các số 1005, 105 và 15.

**Yêu cầu:** Cho số nguyên  $n$ . Hãy xác định số lượng số chia hết cho 3 có mặt trong tập  $D$  được tạo thành từ  $n$ .

**Dữ liệu:** Vào từ file văn bản NUMSET.INP gồm một dòng chứa số nguyên  $n$ .

**Kết quả:** Đưa ra file văn bản NUMSET.OUT một số nguyên – số lượng số chia hết cho 3 tìm được.

Ví dụ:

NUMSET . INP
1005

NUMSET . OUT
3

## Dãy số hamming

Dãy số nguyên dương tăng dần, trong đó ước nguyên tố của mỗi số không quá 5 được gọi là dãy Hamming. Như vậy,  $10 = 2 \times 5$  sẽ là một số trong dãy Hamming, còn  $26 = 2 \times 13$  – không thuộc dãy Hamming.

Phần đầu của dãy Hamming là 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...

**Yêu cầu:** Cho số nguyên  $x$  ( $1 \leq x \leq 10^{18}$ ). Hãy xác định số thứ tự của  $x$  trong dãy Hamming.

**Dữ liệu:** Vào từ file văn bản HAMMING.INP:

- Dòng đầu tiên chứa số nguyên  $t$  – số lượng tests ( $1 \leq t \leq 10^5$ ),
- Mỗi dòng tiếp theo chứa một số nguyên  $x$ .

**Kết quả:** Đưa ra file văn bản HAMMING.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên hoặc thông báo **Not in sequence**.

Ví dụ:

HAMMING.INP
11
1
2
6
7
8
9
10
11
12
13
14

HAMMING.OUT
1
2
6
Not in sequence
7
8
9
Not in sequence
10
Not in sequence
Not in sequence

## Phố may mắn

Người dân thành phố Byteland có rất nhiều điều kiêng kỵ trong cuộc sống. Theo quan điểm của họ, các số 2, 6, 13 và nhiều số khác không mang lại điều may mắn. Trong khi đó các số 3, 5, 7 lại rất được ưa chuộng. Những ngôi nhà có số mà khi phân tích ra thừa số nguyên tố chỉ chứa các thừa số 3, 5, 7 được coi là may mắn và được mua rất nhanh.

Sau một thời gian dài thảo luận, hội đồng thành phố quyết định đánh số tất cả ngôi nhà trên một đường phố mới xây bằng các số may mắn liên tiếp nhau, biến phố đó thành một phố may mắn. Kí hiệu dãy các số may mắn là  $X_1, X_2, X_3, \dots$ . Khi đó, các nhà bên trái mang số  $X_1, X_3, X_5, \dots$  còn các nhà bên phải mang số  $X_2, X_4, X_6, \dots$ . Toàn bộ đường phố không có quá 500 ngôi nhà.

Hãy xác định xem số cho trước có phải là một số nhà ở phố may mắn hay không, nếu đúng thì cho biết nhà đó nằm bên trái hay bên phải của phố.

**Dữ liệu** đọc từ tập tin văn bản STREET.INP gồm nhiều dòng, mỗi dòng chứa 1 số nguyên dương

**Kết quả** xuất ra tập tin văn bản STREET.OUT gồm nhiều dòng, mỗi dòng tương ứng với 1 số ở tập tin dữ liệu vào và chứa 1 trong 3 chữ cái L, R, N phụ thuộc vào số đó tương ứng với nhà bên trái, bên phải hay không phải là số nhà.

Ví dụ

STREET.INP	STREET.OUT
5	L
3	R
4	N

## Dãy số

Xét dãy số nguyên  $a_1, a_2, \dots, a_n, \dots$ , trong đó  $a_1 = 1$ ,  $a_n$  được xác định như sau: đảo ngược thứ tự viết các chữ số của  $a_{n-1}$  (trong hệ cơ số 10) và cộng thêm 2 vào số nhận được.

Phần đầu của dãy số này có giá trị như sau:

<i>Chỉ số</i>	1	2	3	4	5	6	7	8	...
<i>Dãy <math>a</math></i>	1	3	5	7	9	11	13	33	...

***Yêu cầu:*** Cho số nguyên dương  $n$ . Hãy xác định  $a_n$  ( $1 \leq n \leq 10^{12}$ ).

***Dữ liệu:*** Vào từ file văn bản SEQUENCE.INP gồm nhiều tests, mỗi test cho trên một dòng chứa một số nguyên  $n$ .

***Kết quả:*** Đưa ra file văn bản SEQUENCE.OUT, kết quả mỗi test đưa ra trên một dòng.

Ví dụ:

SEQUENCE.INP
1
12

SEQUENCE.OUT
1
77

## Dãy số

Steve trức nhật vì vậy hôm nay bạn đến trường sớm, giặt khăn lau bảng và xóa bảng. Đang xóa bỗng Steve nhận thấy dãy số mà mình đã xóa một phần khá đặc biệt, phần đầu của nó là

**1 2 3 2 3 4 3 4 5 4 5 6 5 6 7**

Đáng tiếc, phần còn lại của dãy đã bị xóa mất. Cuối cùng Steve cũng xóa sạch bảng trước khi trống vào lớp vang lên, nhưng dãy số trên vẫn cứ lờn vờn mãi trong đầu.

Buổi tối, khi lên giường ngủ, Steve lại nghĩ về dãy số này. Steve nhận thấy số 1 xuất hiện 1 lần trong dãy, số 2 xuất hiện 2 lần và lần đầu ở vị trí thứ 2, số 3 xuất hiện 3 lần và lần đầu ở vị trí thứ 3, nhưng số 4 thì lần đầu tiên xuất hiện không ở vị trí thứ 4. Tổng quát hơn số  $k$  sẽ xuất hiện lần đầu ở vị trí thứ mấy trong dãy.

Bạn có thể giúp Steve tìm ra vị trí xuất hiện đầu tiên của số nguyên  $k$  trong dãy không?

**Dữ liệu** vào từ tập tin văn bản **NUM25.INP** gồm 1 dòng chứa số nguyên  $k$  ( $1 \leq k \leq 10^9$ )

**Kết quả** xuất ra tập tin văn bản **NUM25.OUT** là vị trí tìm được

Ví dụ:

NUM25.INP	NUM25.OUT
5	9

## Dãy số

Xét các số tự nhiên vô hạn, ta chọn ra các số bắt đầu từ số 1 để tạo thành một dãy số theo quy luật sau:

- Chọn ra 1 số chia hết cho 1
- Bắt đầu từ số tiếp theo của số được chọn cuối cùng ở lần 1 ta chọn ra 2 số chia hết cho 2
- Bắt đầu từ số tiếp theo của số được chọn cuối cùng ở lần 2 ta chọn ra 3 số chia hết cho 3
- Tương tự ta chọn tiếp 4 số chia hết cho 4
- ...

Như vậy những phần tử đầu của dãy như sau: 1 2 4 6 9 12 16 20 24 28 30 35 40 45 50 54... Phần tử thứ 5 của dãy là 9, phần tử thứ 10 của dãy là 28

**Yêu cầu:** cho số nguyên dương  $n < 10^9$ . Hãy xác định phần tử thứ  $n$  của dãy

**Dữ liệu** đọc từ tập tin văn bản DAYSO.INP gồm số nguyên dương  $n$ .

**Kết quả** xuất ra tập tin văn bản DAYSO.OUT là phần tử thứ  $n$  của dãy

Ví dụ

DAYSO.INP	DAYSO.OUT
5	9
10	28

### Dãy có tổng lớn nhất

Cho dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 1000$ ). Tìm dãy con gồm một hoặc một số phần tử liên tiếp của dãy đã cho với tổng các phần tử trong dãy là lớn nhất.

**Dữ liệu** vào từ tập tin văn bản SUBSEQ.INP gồm 2 dòng

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $n \leq 10^6$ )
- Dòng tiếp theo chứa dãy các số nguyên.

**Kết quả** xuất ra tập tin văn bản SUBSEQ.OUT gồm 3 số nguyên tương ứng với các giá trị: tổng lớn nhất, vị trí bắt đầu, vị trí kết thúc của dãy con trong dãy đã cho.

Ví dụ:

SUBSEQ.INP	SUBSEQ.OUT
8	40 3 6
12 -14 1 23 -6 22 -34 13	



### Nhóm 3

Hãy tìm số lượng nhóm 3 số nguyên  $(i, j, k)$  thỏa mãn các điều kiện:

- $i = j \bmod 2$
- $j = k \bmod 3$
- $0 \leq i \leq j \leq k \leq n$

Ví dụ với  $n = 4$  ta có 5 nhóm  $(0, 0, 0)$ ,  $(0, 0, 3)$ ,  $(0, 2, 2)$ ,  $(1, 1, 1)$  và  $(1, 1, 4)$

**Dữ liệu** đọc từ tập tin văn bản TRIPLES.INP chứa số nguyên  $n$  ( $1 \leq n \leq 10^5$ )

**Kết quả** xuất ra tập tin văn bản TRIPLES.OUT một số nguyên là số lượng nhóm  $(i, j, k)$

Ví dụ

TRIPLES.INP	TRIPLES.OUT
4	4

## Dây dẫn

Cho  $n$  đoạn dây điện ( $1 \leq n \leq 10000$ ). Đoạn thứ  $i$  có độ dài  $l_i$  cm ( $10^2 \leq l_i \leq 10^7$ ). Cần phải cắt các đoạn đã cho thành các đoạn sao cho có được  $k$  đoạn dây bằng nhau. Có thể không cần cắt hết các đoạn dây đã cho. Mỗi đoạn dây bị cắt có thể có phần còn thừa khác 0.

Yêu cầu xác định độ dài lớn nhất của đoạn dây có thể nhận được. Nếu không có cách cắt thì đưa ra số 0.

**Dữ liệu** đọc từ tập tin văn bản WIRES.INP gồm nhiều tests, mỗi test trên một nhóm dòng

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $k$
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa số nguyên  $l_i$

**Kết quả** xuất ra tập tin văn bản WIRES.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên

Ví dụ

WIRES.INP	WIRES.OUT
4 11 802 743 547 539	200

### Tìm khóa

Cho hàm số  $f(x)$  được định nghĩa như sau:

$$f(x) = \begin{cases} x, & x < 10 \\ f(s(x)), & x \geq 10 \end{cases}$$

Trong đó  $s(x)$  là tổng các chữ số của  $x$

Yêu cầu: cho số tự nhiên  $n$ . Hãy tìm giá trị của hàm  $f(n)$

**Dữ liệu** đọc từ tập tin văn bản KEY.INP chứa số nguyên dương  $n$  ( $1 \leq n \leq 1000$ )

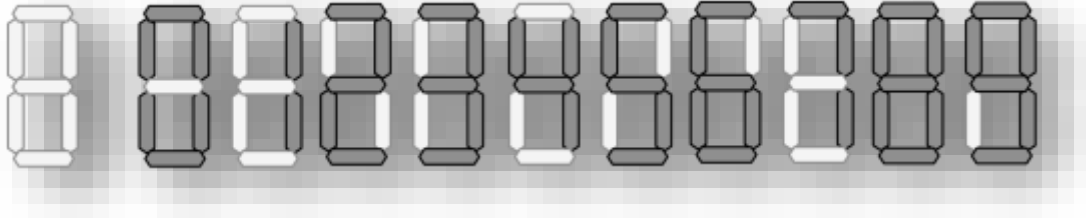
**Kết quả** xuất ra tập tin văn bản KEY.OUT là giá trị của hàm  $f(n)$  tìm được

Ví dụ

KEY.INP	KEY.OUT
4	6

## Hiện số bằng đèn LED

Quà sinh nhật mà Mai nhận được là một máy tính bấm tay màn hình tinh thể lỏng. Màn hình có thể hiển thị được  $n$  chữ số. Có 7 đèn LED tạo thành 7 vạch để hiển thị một chữ số. Mỗi chữ số sẽ tương ứng với một số đèn LED được kích hoạt và vạch tương ứng sẽ có màu đen. Cách hiển thị các số là như sau:



Như vậy, để hiển thị số 0 cần 6 vạch đen, số 1 cần 2 vạch đen, ... Là người ham hiểu biết, Mai tự hỏi không biết số gồm  $n$  chữ số có nghĩa nào nhỏ nhất và lớn nhất có thể hiển thị với đúng  $k$  vạch đen trên màn hình.

Yêu cầu: cho  $n$  và  $k$  ( $1 \leq n \leq 100$ ,  $1 \leq k \leq 700$ ). Hãy xác định số  $n$  chữ số có nghĩa nhỏ nhất và lớn nhất, mỗi số được hiển thị với đúng  $k$  vạch đen trên màn hình.

**Dữ liệu** vào từ tập tin văn bản **LED.inp** gồm một dòng chứa 2 số nguyên  $n$  và  $k$

**Kết quả** xuất ra tập tin văn bản **LED.out** gồm 2 dòng:

- Dòng thứ nhất chứa số nguyên nhỏ nhất
- Dòng thứ hai chứa số nguyên lớn nhất
- Nếu không có nghiệm thì đưa ra thông báo NO SOLUTION

Ví dụ

LED.INP	LED.OUT
5 15	10117 97111

## Trò chơi với dãy số

Hai bạn học sinh trong lúc nhàn rỗi đã nghĩ ra trò chơi sau đây. Mỗi bạn chọn trước một dãy số gồm  $n$  số nguyên. Giả sử dãy số mà bạn thứ nhất chọn là

$$b_1, b_2, \dots, b_n$$

còn dãy số mà bạn thứ hai chọn là

$$c_1, c_2, \dots, c_n$$

Mỗi lượt chơi, mỗi bạn đưa ra một số hạng trong dãy số của mình. Nếu bạn thứ nhất đưa ra số hạng  $b_i$  ( $1 \leq i \leq n$ ), còn bạn thứ hai đưa ra số hạng  $c_j$  ( $1 \leq j \leq n$ ) thì giá của lượt chơi đó sẽ là  $|b_i + c_j|$

Ví dụ: giả sử dãy số bạn thứ nhất chọn là 1, -2 còn dãy số mà bạn thứ hai chọn là 2, 3. Khi đó các khả năng có thể của một lượt chơi là (1, 2), (1, 3), (-2, 2), (-2, 3). Như vậy, giá nhỏ nhất của một lượt chơi trong số các lượt chơi có thể là 0 tương ứng với giá của lượt chơi (-2, 2).

Yêu cầu: hãy xác định giá nhỏ nhất của một lượt chơi trong số các lượt chơi có thể

**Dữ liệu** đọc từ tập tin văn bản SEQGAME.INP

- Dòng đầu tiên chứa số nguyên  $n$  ( $n \leq 10^5$ )
- Dòng thứ hai chứa dãy số nguyên  $b_1, b_2, \dots, b_n$  ( $|b_i| \leq 10^9$ )
- Dòng thứ ba chứa dãy số nguyên  $c_1, c_2, \dots, c_n$  ( $|c_j| \leq 10^9$ )

Các số cách nhau khoảng trắng.

**Kết quả** xuất ra tập tin văn bản SEQGAME.OUT là giá nhỏ nhất tìm được

Ví dụ

SEQGAME.INP	SEQGAME.OUT
2 1 -2 2 3	0

## TẬP SỐ

Cho số  $n$  ở hệ cơ số 10, có không quá 20 chữ số và không chứa các số 0 không có nghĩa ở đầu. Bằng cách xóa một hoặc một vài chữ số liên tiếp của  $n$  (nhưng không xóa hết tất cả các chữ số của  $n$ ) ta nhận được những số mới. Số mới được chuẩn hóa bằng cách xóa các chữ số 0 vô nghĩa nếu có. Tập số nguyên  $D$  được xây dựng bằng cách đưa vào nó số  $n$ , các số mới khác nhau đã chuẩn hóa và khác  $n$ . Ví dụ, với  $n = 1005$  ta có thể nhận được các số mới như sau:

- Bằng cách xóa một chữ số ta có các số: 5 (từ 005), 105, 105, 100;
- Bằng cách xóa hai chữ số ta có các số: 5 (từ 05), 15, 10;
- Bằng cách xóa 3 chữ số ta có các số: 5 và 1.

Tập  $D$  nhận được từ  $n$  chứa các số  $\{1005, 105, 100, 15, 10, 5, 1\}$ . Trong tập  $D$  này có 3 số chia hết cho 3, đó là các số 1005, 105 và 15.

**Yêu cầu:** Cho số nguyên  $n$ . Hãy xác định số lượng số chia hết cho 3 có mặt trong tập  $D$  được tạo thành từ  $n$ .

**Dữ liệu:** Vào từ file văn bản NUMSET.INP gồm một dòng chứa số nguyên  $n$ .

**Kết quả:** Đưa ra file văn bản NUMSET.OUT một số nguyên – số lượng số chia hết cho 3 tìm được.

Ví dụ:

NUMSET . INP
1005

NUMSET . OUT
3

## Kết bạn

Theo quan niệm của người Á đông cổ, mỗi cá nhân khi sinh ra đều ứng với một ngôi sao, được gọi là sao chiếu mệnh. Các hoạt động của cá nhân đều bị chi phối bởi ngôi sao này, kể cả quá trình kết bạn – hẹn hò. Theo thuyết Âm dương – Ngũ hành, hai người chỉ có thể tạo lập mối quan hệ bền vững khi các sao chiếu mệnh của họ không có các thuộc tính tương khắc. Qua hàng ngàn năm quan sát và chiêm nghiệm, các chiêm tinh gia đã ghi nhận được hầu hết các tính chất tương sinh – tương khắc của sao: mỗi sao, ngoài một mã số để nhận diện còn có một giá trị thể hiện khả năng thích nghi của sao gọi là độ thích nghi. Thông qua độ thích nghi, người ta tính khả năng tương hợp của các sao. Khả năng tương hợp của 2 sao được tính bằng tổng 2 độ thích nghi của chúng.

Yêu cầu: cho  $n$  và dãy  $s_1, s_2, \dots, s_n$  là độ thích nghi của các sao. Hãy xác định số lượng  $T$  các cặp sao có khả năng tương hợp bằng  $B$ . Hai cặp  $(s_i, s_j)$  và  $(s_j, s_i)$  được tính là 1.

Ví dụ trong 5 sao với mã số 3, 5, 6, 5, 3 có 4 cặp có khả năng tương hợp là 8

**Dữ liệu** vào từ tập tin văn bản FRIEND.INP

- Dòng đầu tiên ghi 2 số nguyên  $n, B$  ( $2 \leq n \leq 10^5$ )
- Mỗi dòng trong  $n$  dòng tiếp theo ghi một số nguyên là độ thích nghi của một sao, độ thích nghi có trị tuyệt đối bé hơn  $2^{15}$

**Kết quả** xuất ra tập tin văn bản FRIEND.OUT là số nguyên  $T$  tìm được

Ví dụ

FRIEND.INP	FRIEND.OUT
5 8	4
3	
5	
6	
5	
3	

### Đường đi của con ngựa

Có một con ngựa ở ô đầu tiên bên trái của một băng giấy có M ô. Khi di chuyển, con ngựa có thể thực hiện bước nhảy có độ dài (số lượng ô) thuộc tập gồm M giá trị cho trước và chỉ di chuyển về phía bên phải.

									
---	--	--	--	--	--	--	--	--	--

**Yêu cầu:** Tính số cách di chuyển từ ô đầu bên trái đến ô đầu bên phải băng giấy của con ngựa.

**Dữ liệu:** Vào từ file văn bản KNPATH.INP:

- Dòng đầu tiên chứa số nguyên dương N, M được ghi cách nhau bởi dấu cách, trong đó N ( $N \leq 30$ ) là số lượng ô của băng giấy, M ( $M \leq 10$ ) là số loại bước nhảy của con ngựa;
- Dòng tiếp theo chứa M số nguyên dương được ghi cách nhau bởi dấu cách là độ dài của các bước nhảy có thể thực hiện. Không có số nào có giá trị vượt quá N.

**Kết quả:** Ghi ra file văn bản KNPATH.OUT số lượng cách di chuyển tìm được.

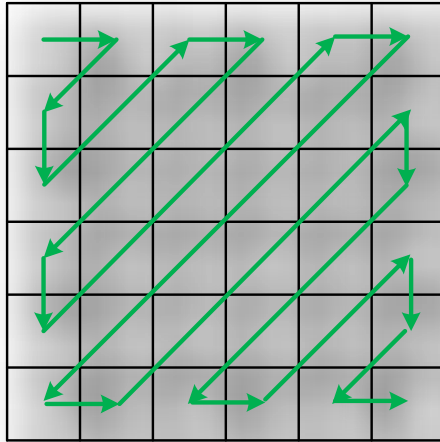
Ví dụ:

KNPATH.INP	KNPATH.OUT
10 2 1 2	89

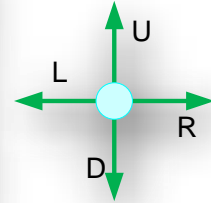


## NHẢY

Xét lưới ô vuông  $n \times n$  ô ( $1 \leq n \leq 10^5$ ), mỗi ô của lưới chứa một số nguyên trong phạm vi từ 1 đến  $n^2$ , các số khác nhau từng đôi một. Các số được điền lần lượt từ 1 trở đi theo đường dích đặc nêu ở hình dưới (với trường hợp  $n = 6$ ).



1	2	6	7	15	16
3	5	8	14	17	26
4	9	13	18	25	27
10	12	19	24	28	33
11	20	23	29	32	34
21	22	30	31	35	36



Một chú thỏ xuất phát từ ô có giá trị 1 ở góc trên trái lần lượt nhảy tới các ô kề cạnh với ô đang đứng. Đường nhảy được mô tả bởi xâu  $S$  chỉ chứa các ký tự thuộc tập  $\{U, D, L, R\}$ , đảm bảo không có bước nhảy nào ra ngoài bảng.

**Yêu cầu:** Cho  $n$  và  $m$  – độ dài xâu  $S$  ( $1 \leq m \leq 3 \cdot 10^5$ ). Hãy tính tổng các số trên đường nhảy.

**Dữ liệu:** Vào từ file văn bản JUMP.INP:

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $m$ ,
- Dòng thứ 2 chứa xâu  $S$ .

**Kết quả:** Đưa ra file văn bản JUMP.OUT một số nguyên – tổng các số trên đường nhảy.

Ví dụ:

JUMP.INP
6 8
DDRRUULL

JUMP.OUT
47

## Bài tập tìm kiếm và sắp xếp

### Tích lớn nhất

**Yêu cầu:** cho một dãy gồm N số nguyên. Tìm 3 số trong dãy có tích lớn nhất

**Dữ liệu** đọc từ tập tin văn bản MAX.INP trong đó

- Dòng đầu chứa N ( $1 \leq N \leq 10^6$ )
- Dòng tiếp theo chứa N số nguyên của dãy, mỗi số có giá trị tuyệt đối không quá 30000.

**Kết quả** ghi ra tập tin văn bản MAX.OUT là tích của 3 số tìm được

Ví dụ

MAX.INP	MAX.OUT
9 3 5 1 7 9 0 9 -3 10	810

### Dãy con tổng 0 dài nhất

Cho một dãy gồm  $N$  số nguyên  $a_1, a_2, \dots, a_N$  ( $|a_i| \leq 10^6$ ). Trong số các dãy con gồm các phần tử liên tiếp của dãy đã cho có tổng các phần tử bằng 0, hãy tìm dãy con gồm nhiều phần tử nhất.

**Dữ liệu:** Vào từ file văn bản ZEROSEQ.INP:

- Dòng đầu tiên chứa số nguyên  $N$  ( $1 \leq N \leq 10^5$ ).
- Dòng thứ hai ghi  $N$  giá trị của dãy, các số cách nhau khoảng trắng

**Kết quả:** Ghi ra file văn bản ZEROSEQ.OUT hai số nguyên  $k, m$  theo thứ tự là số phần tử của dãy tìm được và vị trí bắt đầu của nó trong dãy đã cho. Nếu như không tìm được dãy con thoả mãn điều kiện đầu bài thì hãy ghi hai số -1 -1.

Ví dụ:

ZEROSEQ.INP	ZEROSEQ.OUT
5 2 1 -2 3 -2	4 2

## Ghép số

Cho  $n$  số nguyên dương  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9, n \leq 100)$ . Từ các số này người ta tạo ra một số nguyên mới bằng cách ghép tất cả số đã cho, tức là viết liên tiếp các số đã cho với nhau. Ví dụ, với  $n = 4$  và các số 123, 124, 56, 90 ta có thể tạo ra các số mới sau: 1231245690, 1241235690, 5612312490, 9012312456, 9056124123, ... Có thể dễ dàng thấy rằng với  $n = 4$  ta có thể tạo ra 24 số mới. Trong trường hợp này, số lớn nhất có thể tạo ra là 9056124123.

**Yêu cầu** cho  $n$  và dãy gồm  $n$  số nguyên dương. Hãy xác định số lớn nhất có thể tạo ra khi ghép các số đã cho thành một số mới.

**Dữ liệu** vào từ tập tin văn bản NUMJOIN.INP gồm 2 dòng

- Dòng đầu tiên chứa số nguyên  $n$
- Dòng tiếp theo chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$

**Kết quả** xuất ra tập tin văn bản NUMJOIN.OUT gồm một dòng là số nguyên lớn nhất có thể tạo thành.

Ví dụ

NUMJOIN.INP	NUMJOIN.OUT
4 123 124 56 90	9056124123

## Ước chung lớn nhất

Cho số nguyên  $d$  ( $1 \leq d \leq 10^9$ ) và dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9, n \leq 1000$ ). Hãy tìm một tập gồm nhiều phần tử nhất của dãy có ước chung lớn nhất bằng  $d$ .

**Dữ liệu** vào từ tập tin văn bản GCD.INP trong đó

- Dòng đầu chứa 2 số nguyên  $n$  và  $d$
- Dòng tiếp theo chứa  $n$  giá trị của dãy

**Kết quả** xuất ra tập tin văn bản GCD.OUT

- Trường hợp có lời giải dòng đầu ghi số  $k$  – số các phần tử của tập tìm được, dòng thứ hai ghi các phần tử của tập đó theo thứ tự tăng dần. Nếu có nhiều lời giải hãy đưa ra một lời giải bất kỳ
- Trường hợp không có lời giải xuất ra -1

Ví dụ

GCD.INP	GCD.OUT
4 3	3
6 8 12 9	6 9 12

## Giao hai dãy số

Cho 2 dãy số nguyên  $x_1, x_2, \dots, x_n$  và  $y_1, y_2, \dots, y_n$  tăng nghiêm ngặt ( $x_i < x_{i+1}, y_i < y_{i+1}$ ). Hãy đếm số lượng các phân tử của x xuất hiện trong y.

**Dữ liệu** vào từ tập tin văn bản DAYSO.INP, gồm:

- Dòng đầu chứa số n, các dòng tiếp theo chứa 2n số nguyên ( $n < 500000$ ), trong đó n số đầu là dãy x và n số còn lại là dãy y.

**Kết quả** ghi ra tập tin văn bản DAYSO.OUT chứa một số duy nhất là kết quả tìm được.

Ví dụ:

DAYSO.INP	DAYSO.OUT
5	2
1 2 3 5 7	
2 4 5 6 8	

## Đếm số cặp phân tử

Cho số nguyên  $B$  và dãy số  $a_1, a_2, \dots, a_n (n \leq 10^5)$ . Hãy đếm số cặp  $(a_i, a_j), (i < j)$  sao cho  $a_i + a_j = B$ .

**Dữ liệu** đọc từ tập tin văn bản **PAIRS.INP** gồm 2 dòng:

- Dòng đầu chứa số nguyên dương  $n$  và số nguyên  $B$
- Dòng thứ hai chứa  $n$  số nguyên là các giá trị của dãy

**Kết quả** xuất ra tập tin văn bản **PAIRS.OUT** gồm 1 số nguyên duy nhất là kết quả tìm được

Ví dụ

PAIRS.INP	PAIRS.OUT
5 12 7 1 5 9 3	2

## CÁC ĐỒNG XU

Steve ngồi bên cửa sổ rất lâu quan sát lũ quạ. Chúng là loài chim thông minh, rất thích các vật lóng lánh và hay tha những thứ này về tổ. Hôm nay, không biết kiếm được ở đâu, chúng tha về các đồng xu. Có tất cả  $n$  con quạ. Con quạ thứ  $i$  đã mang về  $a_i$  đồng xu. Nhìn vào vị trí của tổ trên cành Steve hiểu rằng nếu có  $b_i$  thì tổ của quạ thứ  $i$  sẽ bị lật, rơi xuống đất và Steve sẽ nhặt được hết xu trong tổ. Steve đang có trong túi  $m$  đồng xu và có tài ném đầu trúng đó. Bây giờ chính là lúc cái tài lẻ này phát huy tác dụng.

**Yêu cầu:** Hãy xác định số tiền tối đa mà Steve sẽ có được.

**Dữ liệu:** Vào từ file văn bản COINS.INP:

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $m$  ( $1 \leq n \leq 1\,000$ ,  $0 \leq m \leq 1\,000$ ),
- Dòng thứ 2 chứa  $n$  số  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 1\,000$ ,  $i = 1 \div n$ ),
- Dòng thứ 3 chứa  $n$  số  $b_1, b_2, \dots, b_n$  ( $a_i < b_i \leq 1\,000$ ,  $i = 1 \div n$ ).

**Kết quả:** Đưa ra file văn bản COINS.OUT một số nguyên – số tiền tối đa mà Steve sẽ có.

**Ví dụ:**

COINS.INP	
2	3
1	2
4	6

COINS.OUT
6



## Luyện tập dự thi Olympic

Năm 2011 đánh dấu 20 năm hình thành và phát triển của Olympic Tin học sinh viên tại Việt Nam. Để hỗ trợ các bạn sinh viên chuẩn bị tốt cho kỳ thi này, trên website IT-2011 có  $n$  bài tập ( $n \leq 10^5$ ). Các bài tập được đánh số từ 1 đến  $n$ . Mỗi bài tập nhằm rèn luyện một số kỹ năng cho thí sinh, ví dụ như kỹ thuật lập trình, giải thuật, cấu trúc dữ liệu, ...

Nhằm định hướng cho quá trình tự luyện tập được hiệu quả, mỗi bài tập có một yêu cầu tối thiểu về trình độ kỹ năng. Để giải được bài thứ  $i$ , bạn cần có trình độ kỹ năng tối thiểu là  $a_i$ . Điều này có nghĩa là sinh viên có thể giải được bài thứ  $i$  khi và chỉ khi có trình độ kỹ năng bằng hoặc lớn hơn  $a_i$ . Nếu giải được bài thứ  $i$ , trình độ kỹ năng của sinh viên sẽ tăng thêm một lượng là  $b_i$  ( $1 \leq a_i, b_i \leq 10^9$ ). Giả sử ban đầu, trình độ kỹ năng của bạn trước khi làm bài tập là  $c$  ( $0 \leq c \leq 10^9$ ). Các bài tập có thể được làm theo trình tự bất kỳ tùy chọn.

Ví dụ với trình độ kỹ năng ban đầu  $c = 1$ ,  $n = 4$  và các giá trị  $a_i, b_i$  tương ứng là (1, 10), (21, 5), (1, 10), (100, 100), bạn sẽ giải bài 1, sau đó làm bài 3 và cuối cùng làm bài 2. Như vậy bạn sẽ giải được tất cả 3 bài.

**Yêu cầu:** cho các số nguyên  $n, c$  và các cặp giá trị  $(a_i, b_i)$ . Hãy xác định số lượng bài tối đa có thể được giải.

**Dữ liệu:** vào từ tập tin văn bản OLYMPIC.INP có cấu trúc sau:

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $c$
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa 2 số nguyên  $a_i, b_i$

**Kết quả** xuất ra tập tin văn bản OLYMPIC.OUT một số nguyên là số lượng bài tối đa có thể được giải.

Ví dụ

OLYMPIC.INP	
4	1
1	10
21	5
1	10
100	100

OLYMPIC.OUT
3