

I. Lập trình với các cấu trúc dữ liệu cơ bản

1.1. Các bài tập với dữ liệu kiểu số nguyên

BÀI 1.1.1: TỔNG CHỮ SỐ

Viết chương trình tính tổng chữ số của một số không quá 9 chữ số.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên tương ứng

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra trên một dòng giá trị tổng chữ số tương ứng

Ví dụ:

Input	Output
2	10
1234	2
1000001	

BÀI 1.1.2: BẮT ĐẦU VÀ KẾT THÚC

Viết chương trình kiểm tra một số nguyên dương bất kỳ (2 chữ số trở lên, không quá 9 chữ số) có chữ số bắt đầu và kết thúc bằng nhau hay không.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương tương ứng cần kiểm tra

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra YES hoặc NO, tương ứng với bộ dữ liệu vào

Ví dụ:

Input	Output
2	YES
12451	NO
1000012	

BÀI 1.1.3: BÔI SỐ CHUNG NHỎ NHẤT

Viết chương trình tính bội số chung nhỏ nhất của hai số nguyên dương không quá 7 chữ số.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra trên một dòng giá trị bội số chung nhỏ nhất của hai số đó

Ví dụ

Input	Output
2	60
30 20	98754568
222222 8888888	

BÀI 1.1.4: SỐ CÓ TỔNG CHỮ SỐ CHIA HẾT CHO 10

Viết chương trình kiểm tra một số có thỏa mãn tính chất tổng chữ số của nó chia hết cho 10 hay không.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng một số nguyên dương, ít nhất 2 chữ số nhưng không quá 9 chữ số.

Kết quả:

Mỗi bộ test viết ra YES hoặc NO tùy thuộc kết quả kiểm tra.

Ví dụ

Input	Output
3	NO
3333	YES
555555	YES
123455	

BÀI 1.1.5: SỐ ĐẸP 1

Một số được coi là đẹp nếu nó là số thuận nghịch, tổng chữ số là số nguyên tố và tất cả các chữ số đều lẻ. Bài toán đặt ra là đếm xem trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số đẹp như vậy.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không vượt quá 9 chữ số.

Kết quả:

Với mỗi bộ test viết ra số lượng các số thuận nguyên tố tương ứng.

Ví dụ

Input	Output
3	4
23 199	0
2345 6789	311
222222 99999999	

BÀI 1.1.6: SỐ ĐẸP 2

Một số được coi là đẹp nếu nó có tính chất thuận nghịch và tổng chữ số chia hết cho 10. Bài toán đặt ra là cho trước số chữ số. Hãy đếm xem có bao nhiêu số đẹp với số chữ số như vậy.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số chữ số tương ứng cần kiểm tra (lớn hơn 1 và nhỏ hơn 10)

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra số lượng số đẹp tương ứng

Ví dụ

Input	Output
2	1
2	90
5	

BÀI 1.1.7: TRÒ CHƠI ĐOÁN SỐ

Trong lúc rảnh rỗi, hai bạn sinh viên quyết định chơi trò đoán số giống học sinh cấp 1. Mỗi bạn nghĩ ra hai con số nguyên không âm sau đó viết ra tổng và hiệu của chúng (cũng là các số nguyên không âm). Công việc của bạn kia là xác định hai con số ban đầu. Ở một số lượt chơi, một bạn có thể cố tình đưa ra một cặp giá trị không thể là tổng và hiệu của hai số nguyên nào cả.

Viết chương trình giúp tính toán nhanh ra kết quả cho bài toán trên.

Dữ liệu vào:

Dòng đầu là số bộ test, không quá 200.

Mỗi dòng sau chứa hai số nguyên không âm s và d lần lượt là giá trị tổng và hiệu hai số. Cả hai số s và d đều không quá 10^4 .

Kết quả: Ghi ra màn hình

Với mỗi bộ dữ liệu, đưa ra hai số ban đầu, số lớn viết trước, cách nhau một khoảng trống. Nếu không thể có cặp số như vậy thì in ra “impossible”

Ví dụ

Input	Output
2	30 10
40 20	impossible
20 40	

BÀI 1.1.8: MÁY BÁN HÀNG TỰ ĐỘNG

Khi mua hàng bằng máy bán hàng tự động, người mua sẽ trả bằng một số tiền chẵn lớn hơn hoặc bằng giá của sản phẩm. Máy sẽ tính toán để trả lại số tiền thừa cho người mua. Giả sử trong máy chỉ có ba mệnh giá tiền là 1 dollar, 5 dollar và 10 dollar với quy ước mỗi lần trả chỉ được phép dùng ít hơn 5 tờ 1 dollar và ít hơn 2 tờ 5 dollar.

Hãy viết chương trình tính số tiền mỗi loại mà máy bán hàng tự động phải trả lại cho người mua.

Dữ liệu vào:

Dòng đầu tiên là số bộ test, mỗi bộ test ghi trên một dòng hai số nguyên không âm là giá của sản phẩm và tổng số tiền người mua đưa vào. Cả hai giá trị này đều không vượt quá 10^5 .

Kết quả:

Với mỗi bộ test, viết ra biểu diễn số tiền cần trả của máy bán hàng tự động theo mẫu trong bộ test ví dụ dưới đây. (Chú ý: giữa các số và các dấu luôn có đúng một khoảng trống, cả với dấu =, dấu * hoặc dấu +)

Ví dụ cho Input và Output:

<u>Input</u>	<u>Output</u>
3	$28 = 2 * 10 + 1 * 5 + 3 * 1$
72 100	$163 = 16 * 10 + 0 * 5 + 3 * 1$
37 200	$45 = 4 * 10 + 1 * 5 + 0 * 1$
5 50	

BÀI 1.1.9: (*) BIỂU DIỄN SỐ BẰNG QUE DIÊM

Một trong những cách biểu diễn số khá phổ biến trong các đồng hồ điện tử là sử dụng que diêm. Các ký tự số sẽ được biểu diễn như sau:

1 2 3 4 5 6 7 8 9 0

Với một số lượng que diêm cho trước, hãy các định số nhỏ nhất và số lớn nhất mà bạn có thể biểu diễn được.

Chú ý:

- Bạn không được phép để thừa que diêm nào khi xếp.
- Các số biểu diễn không được bắt đầu bằng số 0

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test viết trên một dòng một số nguyên duy nhất không lớn hơn 100 là số que diêm bạn có.

Kết quả:

Với mỗi bộ test, output đưa ra hai số nguyên theo thứ tự là số nhỏ nhất và số lớn nhất có thể biểu diễn bởi số que diêm cho bởi input (mỗi số cách nhau một khoảng trống).

Ví dụ cho Input và Output:

Input	Output
4	7 7
3	6 1 1 1
6	8 7 1 1
7	10 8 7 1 1 1 1 1
15	

BÀI 1.1.10: ĐẾM SỐ CHÍNH PHƯƠNG TRONG ĐOẠN

Viết chương trình đếm trong một đoạn giữa hai số nguyên có bao nhiêu số chính phương.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không quá 9 chữ số.

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra trên một dòng giá trị số các số chính phương đếm được.

Ví dụ:

Input	Output
2	10
23 199	34
2345 6789	

BÀI 1.1.11: SỐ THUẦN NGUYÊN TỔ

Một số được coi là thuần nguyên tố nếu nó là số nguyên tố, tất cả các chữ số là nguyên tố và tổng chữ số của nó cũng là một số nguyên tố. Bài toán đặt ra là đếm xem trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số thuần nguyên tố.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không vượt quá 9 chữ số.

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra số lượng các số thuần nguyên tố tương ứng.

Ví dụ

Input	Ouput
-------	-------

2	1
23 199	15
2345 6789	

BÀI 1.1.12: SỐ CÓ TỔNG CHỮ SỐ CHIA HẾT CHO 10

Viết chương trình kiểm tra một số có thỏa mãn tính chất tổng chữ số của nó chia hết cho 10 hay không.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng một số nguyên dương, ít nhất 2 chữ số nhưng không quá 9 chữ số.

Kết quả:

Mỗi bộ test viết ra YES hoặc NO tùy thuộc kết quả kiểm tra.

Ví dụ:

Input	Output
3	NO
3333	YES
555555	YES
123455	

BÀI 1.1.13: SỐ ĐẸP 3

Một số được coi là đẹp nếu nó là số thuận nghịch, tổng chữ số là số nguyên tố và tất cả các chữ số đều lẻ. Bài toán đặt ra là đếm xem trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số đẹp như vậy.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không vượt quá 9 chữ số.

Kết quả:

Với mỗi bộ test viết ra số lượng các số thuần nguyên tố tương ứng.

Ví dụ

Input	Output
3	4
23 199	0
2345 6789	311
222222 99999999	

BÀI 1.1.14:

Hãy viết chương trình tìm số các số tự nhiên N thỏa mãn đồng thời những điều kiện dưới đây ($N \leq 2^{31}$):

- N là số có K chữ số ($K \leq 15$).
- N là số nguyên tố.
- Đảo ngược các chữ số của N cũng là một số nguyên tố.
- Tổng các chữ số của N cũng là một số nguyên tố.
- Mỗi chữ số của N cũng là các số nguyên tố ;
- Thời gian thực hiện chương trình không quá 1sec.

Dữ liệu vào (Input) cho bởi file data.in theo khuôn dạng:

- Dòng đầu tiên ghi lại số tự nhiên M là số lượng các test ($M \leq 100$).
- M dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test bao gồm một số K . Hai số được viết cách nhau một vài khoảng trống.

Kết quả ra (Output): ghi lại M dòng trong file ketqua.out, mỗi dòng ghi lại bộ hai số N, X . Trong đó X là số các số có N chữ số thỏa mãn yêu cầu của bài toán. Ví dụ dưới đây minh họa cho file input và output của bài toán.

<u>Input.in</u>	<u>Output.out</u>	
5	2	0
2	3	8
3	4	15
4	5	46
5	7	359
7		

BÀI 1.1.15:

Trong ngày đầu tiên phát hành các số điện thoại di động, công ty viễn thông dự định khuyến mại cho N khách hàng đăng ký trước nhất các số điện thoại loại 1, M khách hàng kế tiếp số điện thoại loại 2 và K khách hàng cuối cùng các số điện thoại loại 3. Các số điện thoại loại 1, loại 2 và loại 3 có tính chất sau:

- Số loại 3 (Loại 3): là các số điện thoại mà sáu số cuối cùng của nó tạo thành một số thuận nghịch có sáu chữ số. Ví dụ số : 0913.104401.
- Số loại 2 (Loại 2): là các số điện thoại Loại 3 có tổng sáu số cuối cùng của nó là một số chia hết cho 10. Ví dụ số : 0913.104401.
- Số loại 1 (Loại 1): là các số điện thoại Loại 2 có tổng sáu số cuối cùng của nó không chứa bất kỳ số 0 nào. Ví dụ số : 0913.686686.

Bài toán được đặt ra là cho trước một phương án N, M, K, hãy trả lời “YES” nếu công ty thực hiện được, trả lời “NO” nếu công ty không thực hiện được.

Input: Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test là một bộ 3 số N, M, K được ghi trên một dòng. Các số được ghi cách nhau một vài khoảng trống.

Output: Với mỗi bộ test, viết ra trên một dòng giá trị “YES” hoặc “NO” tương ứng với phương án thực hiện được, hoặc phương án không thực hiện được.

Ví dụ cho Input và Output:

INPUT	OUTPUT
5	NO
100 100 200	NO
50 150 200	YES
100 50 300 120	YES
50 500	NO
140 50 700	

BÀI 1.1.16:

Số N nguyên hệ cơ số ACM là những số nguyên thông thường sử dụng các ký hiệu từ 0, 1, ..., 9 làm ký hiệu của hệ đếm (Ví dụ số 719_{ACM}). Nguyên tắc chung để đổi một số $A = (a_1, a_2, \dots, a_N)$ ở hệ cơ số ACM sang số ở hệ cơ số 10 được thực hiện như sau:

$$K_{10} = \sum_{i=1}^N a_i * (i!), \text{ trong đó } a_i \text{ chữ số tại vị trí thứ } i \text{ của ở hệ cơ số ACM.}$$

Ví dụ:

$$A = 719_{ACM} = 9.(1!) + 1.(2!) + 7.(3!) = 53_{10}.$$

Nhiệm vụ của bạn là viết một chương trình đọc một số nguyên ở hệ cơ số ACM rồi đổi số đó thành số hệ cơ số 10.

Dữ liệu vào:

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên không lớn hơn 100 là số lượng các bộ dữ liệu. Những dòng tiếp theo chứa các bộ dữ liệu. Mỗi bộ dữ liệu được viết trên một dòng. Mỗi dòng viết một số nhỏ hơn 2^{32} là số ở hệ cơ số ACM.

Dữ liệu ra:

Với mỗi bộ dữ liệu, ghi ra trên một dòng một số được chuyển đổi.

Ví dụ dữ liệu vào	Ví dụ dữ liệu ra
6	53
719	1
1	7
15	8
110	8
102	0
8	

1.2. Các bài tập về mảng và ma trận

BÀI 1.2.1: SỐ CẶP BẰNG NHAU TRONG DÃY

Viết chương trình đếm các cặp số bằng nhau liên tiếp trong dãy số nguyên.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test có hai dòng:

- Dòng đầu ghi số phần tử của dãy, không quá 30
- Dòng tiếp theo ghi các phần tử của dãy, mỗi phần tử cách nhau một khoảng trống. Các phần tử không quá 100.

Kết quả: Ghi ra màn hình

Mỗi bộ test viết ra trên một dòng giá trị tổng chữ số tương ứng

Ví dụ:

Input	Output
2	1
4	6
1 3 3 4	
12	
1 2 3 3 3 3 4 4 5 5 5 1	

BÀI 1.2.2: ĐẾM CÁC SỐ LỚN HƠN SỐ ĐỨNG TRƯỚC TRONG DÃY

Cho một dãy số nguyên dương có n phần tử ($2 \leq n \leq 50$). Hãy liệt kê số các phần tử trong dãy không nhỏ hơn các số đứng trước nó (tính cả phần tử đầu tiên).

Dữ liệu vào:

- Dòng 1 ghi số bộ test

- Với mỗi bộ test: dòng đầu tiên ghi số n. Dòng tiếp theo ghi n số nguyên dương của dãy (các số không vượt quá 1000).

Kết quả:

Ghi ra màn hình trên một dòng số các phần tử thỏa mãn.

Ví dụ:

<u>Input</u>	<u>Output</u>
2	5
7	3
3 5 6 8 4 2 9	
15	
9 8 123 7 11 14 18 21 399 10 5 4 1 2 3	

BÀI 1.2.3: ĐOÁN SỐ TIẾP THEO

An và Bình chơi trò chơi số học đơn giản. Dãy số mà An đưa ra là $A = \{1, 1, 3, 4, 5, 9, 7, 16, 9, \dots\}$ và đồ Bình tìm ra số tiếp theo trong dãy đó. Rất nhanh chóng, Bình tìm được số tiếp theo là số 25. Bình đố lại An, nếu cho trước một số k không quá 100, hãy tính số đứng vị trí đó trong dãy đã cho (thứ tự trên dãy tính từ 1). Bạn hãy giúp An tính ra kết quả trên.

Dữ liệu vào:

Dòng đầu là số bộ test, không quá 20. Mỗi bộ test ghi trên một dòng số nguyên dương k.

Kết quả:

Với mỗi bộ test, đưa ra trên một dòng giá trị ở vị trí k của dãy.

Ví dụ:

<u>C.IN</u>	<u>Kết quả</u>
3	1
1	4
4	25
10	

BÀI 1.2.4: TỔNG HAI ĐA THỨC

Cho hai đa thức $P(x)$ -bậc n và $Q(x)$ -bậc m có các hệ số nguyên, n và m không quá 100. Hãy viết chương trình tính tổng hai đa thức trên.

Dữ liệu vào:

Dòng đầu tiên chứa số nguyên N là số bộ dữ liệu ($1 \leq N \leq 10$).

Mỗi bộ dữ liệu gồm 4 dòng:

- Dòng 1 ghi số n là bậc của P . Dòng 2 ghi $n+1$ số nguyên tương ứng là hệ số của P từ P_0 đến P_n
- Dòng 3 ghi số m là bậc của Q . Dòng 4 ghi $m+1$ số nguyên tương ứng là hệ số của Q , từ Q_0 đến Q_m

Kết quả: Ghi ra màn hình

Với mỗi bộ dữ liệu vào, in ra kết quả trên hai dòng: Dòng 1 ghi bậc của đa thức tổng. Dòng 2 ghi lần lượt các hệ số của đa thức tổng, tính từ 0.

Ví dụ:

D.IN	Kết quả
1 3 1 2 3 4 5 1 1 2 -2 3 3	5 2 3 5 2 3 3

BÀI 1.2.5: TÌM CÁC VỊ TRÍ BẰNG NHAU CỦA HAI MA TRẬN

Cho hai ma trận vuông A và B chỉ gồm số nguyên dương cấp n . Hãy viết chương trình tìm ra các vị trí bằng nhau trong hai ma trận (vị trí $[i,j]$ được coi là bằng nhau nếu $A[i,j]=B[i,j]$).

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: Dòng đầu tiên ghi số n ; n dòng tiếp theo ghi ma trận A ; n dòng tiếp theo ghi ma trận B

Kết quả (ghi ra màn hình):

Với mỗi bộ test ghi ra một số nguyên dương m là số vị trí bằng nhau. m dòng tiếp theo ghi hai giá trị chỉ số của từng cặp (cách nhau một khoảng trống). (Chú ý: các chỉ số đều tính từ 0 đến $n-1$).

Ví dụ:

Input	Output
1	2
2	0 1
1 1	1 1
1 2	
9 1	
4 2	

BÀI 1.2.6: SỐ FIBONACI THỨ N

Dãy Fibonacci được định nghĩa như sau: $F(0) = 0$, $F(1)=1$, ..., $F(n)=F(n-1)+F(n-2)$. Cho trước số nguyên dương n (không quá 45), hãy tính số Fibonacci thứ n .

Dữ liệu vào:

Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng số nguyên dương n .

Kết quả (ghi ra màn hình):

Với mỗi bộ test ghi ra giá trị số Fibonacci thứ n .

Ví dụ:

Input	Output
3	1
1	13
7	55
10	

BÀI 1.2.7: TÍCH MA TRẬN VỚI CHUYỂN VỊ CỦA NÓ

Cho ma trận A chỉ gồm các số nguyên dương cấp $n \times m$. Hãy viết chương trình tính tích của A với ma trận chuyển vị của A.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: Dòng đầu tiên ghi hai số n và m là bậc của ma trận a; n dòng tiếp theo, mỗi dòng ghi m số của một dòng trong ma trận A.

Kết quả (ghi ra màn hình):

Với mỗi bộ test ghi ra ma trận tích tương ứng, mỗi số cách nhau đúng một khoảng trống.

Ví dụ:

Input	Output
1	5 11
2 2	11 25
1 2	
3 4	

BÀI 1.2.8: SỐ XUẤT HIỆN NHIỀU LẦN NHẤT TRONG DÃY

Cho một dãy số nguyên dương không quá 100 phần tử, các giá trị trong dãy không quá 30000. Hãy xác định xem số nào là số xuất hiện nhiều lần nhất trong dãy. Chú ý: trong trường hợp nhiều số khác nhau cùng xuất hiện số lần bằng nhau và là lớn nhất thì in ra tất cả các số đó theo thứ tự xuất hiện trong dãy ban đầu.

Dữ liệu vào:

Dòng đầu là số bộ test, không quá 20.

Mỗi bộ test gồm hai dòng. Dòng đầu ghi số phần tử của dãy, dòng tiếp theo ghi các phần tử của dãy.

Kết quả: Ghi ra màn hình

Với mỗi bộ test, đưa ra số xuất hiện nhiều lần nhất trong dãy đã cho.

Ví dụ:

Input	Output
2	1
10	1 2 3 4 5 6 7 8 9 0
1 2 3 1 2 3 1 2 3 1	
10	
1 2 3 4 5 6 7 8 9 0	

BÀI 1.2.9: MA TRẬN ĐƠN VỊ

Một ma trận vuông A cấp n chỉ gồm các số nguyên dương. A được gọi là ma trận đơn vị nếu tất cả các phần tử trong A đều là 0 hoặc 1. Viết chương trình kiểm tra xem một ma trận có đối xứng hay không.

Dữ liệu vào:

Dòng đầu tiên ghi số bộ test (không quá 10). Với mỗi bộ test: Dòng đầu tiên ghi số n là bậc của ma trận A; n dòng tiếp theo, mỗi dòng ghi n số của một dòng trong ma trận A.

Các giá trị đều không quá 100.

Kết quả (ghi ra màn hình):

Với mỗi bộ test ghi ra màn hình chữ YES nếu đó đúng là ma trận đơn vị, NO nếu ngược lại.

Ví dụ:

Input	Output
2	NO
2	YES
1 1	
1 2	
4	
1 1 0 0	
1 0 0 1	
0 0 1 1	
1 0 1 0	

BÀI 1.2.10: TỔNG TAM GIÁC

Số tam giác thứ n là tổng của n số tự nhiên đầu tiên, $T(n) = 1 + \dots + n$. Nó cũng chính là số lượng điểm trong một mảng tam giác có cạnh là n , ví dụ như với $T(4)$ thì:

X
X X
X X X
X X X X

Hãy viết chương trình tính tổng có trọng số của số tam giác:

$$W(n) = \sum_{k=1}^n k \cdot T(k+1)$$

Dữ liệu vào:

Dòng đầu tiên chứa số nguyên N là số bộ dữ liệu ($1 \leq N \leq 1000$).

Mỗi bộ dữ liệu gồm 1 dòng chứa duy nhất 1 số nguyên n ($1 \leq n \leq 300$) là số điểm trên cạnh của tam giác.

Kết quả: Ghi ra màn hình

Với mỗi bộ dữ liệu vào, in ra trên một dòng số thứ tự của bộ dữ liệu (1 đến N), một dấu cách, giá trị của n trong bộ dữ liệu, một dấu cách và tổng có trọng số $W(n)$ của những số tam giác tương ứng với n .

Ví dụ:

D.IN	Kết quả
4	1 3 45
3	2 4 105
4	3 5 210
5	4 10 2145
10	

BÀI 1.2.11: ĐỖ XE TỐI ƯU

Khi mua sắm trên khu Long Street, Michael thường đỗ xe của mình ở một số vị trí ngẫu nhiên và đi bộ vào cửa hàng. Bạn hãy giúp Michael chọn một chỗ đỗ xe để khoảng cách phải đi bộ khi mua hàng là nhỏ nhất.

Long Street có thể coi như là một đường thẳng mà tất cả những điểm mua hàng là các điểm có tọa độ nguyên.

Dữ liệu vào:

Dòng đầu tiên chứa một số nguyên $1 \leq t \leq 100$ là số lượng bộ test. Mỗi bộ test gồm 2 dòng, dòng đầu tiên ghi số cửa hàng n mà Michael muốn qua mua hàng, $1 \leq n \leq 20$ và dòng thứ hai ghi n số nguyên là tọa độ các điểm này trên phố Long Street, $0 \leq x_i \leq 99$.

Kết quả:

Với mỗi bộ test, in ra màn hình trên một dòng khoảng cách nhỏ nhất phải đi bộ với chỗ đỗ xe tối ưu.

Ví dụ cho Input và Output:

Input	Output
2	152
4	70
24 13 89 37	
6	
7 30 41 14 39 42	

BÀI 1.2.12:

Ta định nghĩa một từ là dãy các kí tự không chứa khoảng trống (space), dấu tab, dấu xuống dòng (' $\backslash n$ '), dấu về đầu dòng (' $\backslash r$ ') và dấu kết thúc dòng (' $\backslash 0$ '). Tần suất xuất hiện của từ W trong tập văn bản D_1 và D_2 , ký hiệu là $P(W)$ được tính theo công thức:

$$P(W) = \frac{N_1(W) + N_2(W)}{N(D_1) + N(D_2)}; \text{ trong đó } N_i(W) \text{ là số lần xuất hiện từ } W \text{ trong } D_i, N(D_i) \text{ là tổng}$$

số từ của tập văn bản D_i ($i=1, 2$).

Cho hai file văn bản *data1.in* và *data2.in*. Sử dụng CTDL Mảng, hãy tìm tập các từ và tần xuất xuất hiện của mỗi từ hoặc xuất hiện trong *data1.in* hoặc xuất hiện trong *data2.in*. Tập các từ tìm được ghi lại trong file *Ketqua.out* theo khuôn dạng:

- Dòng đầu tiên ghi lại số tự nhiên K là số từ W tìm được theo yêu cầu của bài toán.
- K dòng kế tiếp, mỗi dòng ghi lại một từ W và tần xuất xuất hiện $P(W)$ thỏa mãn yêu cầu của bài toán. W và $P(W)$ được viết cách nhau bởi một vài khoảng trống.

Ví dụ với file *data1.in* và *data2.in* dưới đây sẽ cho ta file *Ketqua.out* của bài toán.

data1.in	data2.in	Ketqua.out
AB AC AD AE AF AB AC AD AE AF	AB AC AD AH AK AB AC AD AH AK	3 AB 0.2 AC 0.2 AD 0.2

BÀI 1.2.13:

Ta định nghĩa một từ là dãy các kí tự không chứa khoảng trống (space), dấu tab, dấu xuống dòng ('\\n'), dấu về đầu dòng ('\\r') và dấu kết thúc dòng ('\\0'). Tần xuất xuất hiện của từ W trong tập văn bản D_1 và D_2 , ký hiệu là $P(W)$ được tính theo công thức:

$$P(W) = \frac{N_1(W) + N_2(W)}{N(D_1) + N(D_2)}; \text{ trong đó } N_i(W) \text{ là số lần xuất hiện từ } W \text{ trong } D_i, N(D_i) \text{ là tổng}$$

số từ của tập văn bản D_i ($i=1, 2$).

Cho hai file văn bản *data1.in* và *data2.in*. Sử dụng CTDL Mảng, hãy tìm tập các từ và tần xuất xuất hiện của các từ xuất hiện trong file *data1.in* nhưng không xuất hiện trong file *data2.in*. Tập các từ tìm được ghi lại trong file *Ketqua.out* theo khuôn dạng:

- Dòng đầu tiên ghi lại số tự nhiên K là số từ W tìm được theo yêu cầu của bài toán.
- K dòng kế tiếp, mỗi dòng ghi lại một từ W và tần xuất xuất hiện $P(W)$ thỏa mãn yêu cầu của bài toán. W và $P(W)$ được viết cách nhau bởi một vài khoảng trống.

Ví dụ với file *data1.in* và *data2.in* dưới đây sẽ cho ta file *Ketqua.out* của bài toán.

data1.in	data2.in	Ketqua.out
AB AC AD AE AF AB AC AD AE AF	AB AC AD AH AK AB AC AD AH AK	7 AB 0.2 AC 0.2 AD 0.2 AE 0.1 AF 0.1 AH 0.1 AK 0.1

BÀI 1.2.14:

Ta định nghĩa một từ là dãy các kí tự không chứa khoảng trống (space), dấu tab, dấu xuống dòng ('\\n'), dấu về đầu dòng ('\\r') và dấu kết thúc dòng ('\\0'). Tần xuất xuất hiện của từ W trong tập văn bản D_1 và D_2 , ký hiệu là $P(W)$ được tính theo công thức:

$$P(W) = \frac{N_1(W) + N_2(W)}{N(D_1) + N(D_2)}; \text{ trong đó } N_i(W) \text{ là số lần xuất hiện từ } W \text{ trong } D_i, N(D_i) \text{ là tổng}$$

số từ của tập văn bản D_i ($i=1, 2$).

Cho hai file văn bản data1.in và data2.in. Sử dụng CTDL Mảng, hãy tìm tập các từ và tần xuất xuất hiện của mỗi từ đồng thời trong cả hai tập data1.in và data2.in. Tập các từ tìm được ghi lại trong file Ketqua.out theo khuôn dạng:

- Dòng đầu tiên ghi lại số tự nhiên K là số từ W tìm được theo yêu cầu của bài toán.
- K dòng kế tiếp, mỗi dòng ghi lại một từ W và tần xuất xuất hiện $P(W)$ thỏa mãn yêu cầu của bài toán. W và $P(W)$ được viết cách nhau bởi một vài khoảng trống.

Ví dụ với file data1.in và data2.in dưới đây sẽ cho ta file Ketqua.out của bài toán.

data1.in	data2.in	Ketqua.out
AB AC AD AE AF AB AC AD AE AF	AB AC AD AH AK AB AC AD AH AK	2 AE 0.1 AF 0.1

BÀI 1.2.15:

Cho tập các số tự nhiên ghi lại trong file data.in theo từng dòng; mỗi dòng ghi lại nhiều nhất 10 số; hai số được viết cách nhau một vài khoảng trống; mỗi số tự nhiên có thể xuất hiện nhiều lần ở những vị trí khác nhau trong file. Ta gọi tần xuất xuất hiện số tự nhiên X trong file data.in là $P(X) = \frac{N(X)}{K}$; trong đó N(X) là số lần xuất hiện số tự nhiên X trong file data.in, K là số các số tự nhiên trong file data.in. *Sử dụng CTDL mảng*, hãy viết chương trình tìm số vừa nguyên tố vừa thuận nghịch X có P(X) lớn nhất đầu tiên tìm được trong file data.in. Ví dụ với file data.in được cho dưới đây, ta tìm được số X = 131 vừa là số nguyên tố, vừa là số thuận nghịch với tần xuất xuất hiện lớn nhất là $P(X) = 3/30 = 0.1$.

data.in

10	131	171	13	37	27	30	23	77	444
10	131	171	20	23	77	23	27	77	444
10	131	171	20	23	77	23	27	77	444

BÀI 1.2.16:

Cho tập các số tự nhiên trong file data.in được ghi theo từng dòng, mỗi dòng ghi nhiều nhất 5 số, hai số được viết cách nhau một vài khoảng trống. Biết rằng, mỗi số tự nhiên trong file data.in hoặc là số nguyên tố, hoặc là số thuận nghịch và có thể xuất hiện nhiều lần ở những vị trí khác nhau trong file. *Sử dụng cấu trúc dữ liệu mảng*, hãy viết chương trình tách tập các số và đếm số lần xuất hiện của mỗi số trong file data.in thành 3 file ketqua1.out, ketqua2.out, ketqua3.out thỏa mãn những yêu cầu dưới đây.

- File ketqua1.out ghi lại các số nguyên tố nhưng không là số thuận nghịch cùng với số lần xuất hiện của nó trong file data.in;
- File ketqua2.out ghi lại các số thuận nghịch nhưng không là nguyên tố cùng với số lần xuất hiện của nó trong file data.in;
- File ketqua3.out ghi lại các số vừa là số nguyên tố vừa là số thuận nghịch cùng với số lần xuất hiện của nó trong file data.in;
- Khuôn dạng của các file kết quả được qui định như sau:
 - Dòng đầu tiên của mỗi file ghi lại số các số của mỗi file kết quả;

- Những dòng kế tiếp mỗi dòng ghi lại một số cùng với số lần xuất hiện của nó trong file data.in. Hai số được viết cách nhau một vài khoảng trống.

Ví dụ dưới đây minh họa cho các file data.in, ketqua1.out, ketqua2.out và ketqua3.out.

Data.in	Ketqua1.out	Ketqua2.out	Ketqua3.out
10007 10009 10801 10901 13831	2	2	2
10007 10009 10801 10901 34543	10007 4	10801 4	13831 2
10007 10009 10801 10901 13831	10009 4	10901 4	34543 2
10007 10009 10801 10901 34543			

BÀI 1.2.17:

Cho một hình vuông gồm 5 x 5 hình vuông đơn vị như hình vẽ sau:

3	5	1	1	1
5	0	0	3	3
1	0	3	4	3
1	3	4	2	1
1	3	3	1	3

Hãy điền các chữ số từ 0 đến 9 vào mỗi hình vuông đơn vị sao cho những điều kiện sau được thỏa mãn:

- Đọc từ trái sang phải theo hàng ta nhận được năm số nguyên tố có năm chữ số;
- Đọc từ trên xuống dưới theo cột ta nhận được năm số nguyên tố có năm chữ số;
- Đọc theo hai đường chéo chính từ trái sang phải ta nhận được hai số nguyên tố có năm chữ số;
- Tổng các chữ số trong mỗi số nguyên tố đều bằng S cho trước (trong ví dụ trên S=11).

Input: Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test viết trên một dòng một bộ đôi hai số S, T tương ứng với tổng các chữ số trong mỗi số nguyên tố và giá trị ô vuông trên cùng góc trái.

Output: Với mỗi bộ test, output đưa ra một số duy nhất là số các lời giải của bài toán. Mỗi lời giải của bài toán là một hình vuông gồm 5x5 hình vuông đơn vị. Hai lời giải được xem là khác nhau nếu tồn tại một ô vuông đơn vị ở cùng một vị trí có giá trị khác nhau.

Ví dụ cho Input và Output của bài toán:

INPUT	OUTPUT
3	5
11 3	0
11 9	0
11 8	

BÀI 1.2.18:

Trò chơi Rubik trên mặt phẳng có thể được mô tả bằng dãy các số từ 1 đến 8 và được chia thành hai hàng. Ta qui ước trạng thái khởi đầu S của Rubik là trạng thái trong hình 1. Người ta đã chứng minh được rằng, từ trạng thái khởi đầu S ta có thể dịch chuyển thành trạng thái T bất kỳ trên các thao tác A, B, C dưới đây:

A : Đổi hàng trên cho hàng dưới (Hình 2).

B : Tạo nên một hoán vị vòng trên mỗi hàng (Hình 3).

C : Quay 90 độ 4 ô ở giữa (Hình 4)

Hình 1. Trạng thái khởi đầu S

1	2	3	4
8	7	6	5

Hình 2. Thao tác A				Hình 3. Thao tác B				Hình 4. Thao tác C			
8	7	6	5	4	1	2	3	1	7	2	4
1	2	3	4	5	8	7	6	8	6	3	5

Yêu cầu: Cho trước một trạng thái kết thúc T. Hãy chuyển S thành T sao cho số các phép A, B, C thực hiện là ít nhất.

Input: Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test được tổ chức trên một dòng là giá trị trạng thái kết thúc T.

Output: Với mỗi bộ test, viết ra trên một dòng số các bước A, B, C ít nhất.

INPUT	OUTPUT
2	7
2 6 8 4 5 7 3 1	0
1 2 3 4 5 6 7 8	

1.3. Các bài tập về xâu ký tự

BÀI 1.3.1: PHÉP CỘNG SỐ NGUYÊN

Viết chương trình cộng hai số nguyên dương bất kỳ (không quá 256 chữ số).

Dữ liệu vào:

Dòng 1 ghi số bộ test. Mỗi bộ test gồm 2 dòng, mỗi dòng ghi một số nguyên dương

Kết quả (ghi ra màn hình):

Với mỗi bộ test ghi ra một số nguyên dương là tổng hai số đã cho (số này cũng không quá 256 chữ số).

Ví dụ:

Input	Output
3	112
12	10100
100	121212121257800190
1212	
8888	
1212121212121212	
45678978	

BÀI 1.3.2: DANH SÁCH ĐIỆN THOẠI NHẤT QUÁN

Cho một danh sách các số điện thoại, hãy xác định danh sách này có số điện thoại nào là phần trước của số khác hay không? Nếu không thì danh sách này được gọi là nhất quán.

Giả sử một danh sách có chứa các số điện thoại sau:

- Số khẩn cấp: 911
- Số của alice: 97 625 999
- Số của Bob: 91 12 54 26

Trong trường hợp này, ta không thể gọi cho Bob vì tổng dài sẽ kết nối bạn với đường dây khẩn cấp ngay khi bạn quay 3 số đầu trong số của Bob, vì vậy danh sách này là không nhất quán.

Dữ liệu vào:

Dòng đầu tiên chứa một số nguyên $1 \leq t \leq 40$ là số lượng bộ test. Mỗi bộ test sẽ bắt đầu với số lượng số điện thoại n được ghi trên một dòng, $1 \leq n \leq 10000$. Sau đó là n dòng, mỗi dòng ghi duy nhất 1 số điện thoại. Một số điện thoại là một dãy không quá 10 chữ số.

Kết quả:

Với mỗi bộ dữ liệu vào, in ra “YES” nếu danh sách nhất quán và “NO” trong trường hợp ngược lại.

Ví dụ cho Input và Output:

Input	Output
2	NO
3	YES
911	
97625999	
91125426	
5	
113	
12340	
123440	
12345	
98346	

BÀI 1.3.3: CHUẨN HÓA XÂU HỌ TÊN

Một xâu họ tên được coi là viết chuẩn nếu chữ cái đầu tiên mỗi từ được viết hoa, các chữ cái khác viết thường. Các từ cách nhau đúng một dấu cách và không có khoảng trống thừa ở đầu và cuối xâu. Hãy viết chương trình đưa các xâu họ tên về dạng chuẩn.

Dữ liệu vào:

Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng xâu ký tự họ tên, không quá 80 ký tự.

Kết quả (ghi ra màn hình):

Với mỗi bộ test ghi ra xâu ký tự họ tên đã chuẩn hóa.

Ví dụ:

Input	Output
3	Nguyen Van Nam
nGuYEN vAN naM	Tran Trung Hieu
tRan TRUNG hiEU	Vo Le Hoa Binh
vO le hOA bINh	

1.4. Các bài tập về file và cấu trúc

BÀI 1.4.1: ĐẾM TỪ KHÁC NHAU

Viết chương trình đếm số từ khác nhau của một file văn bản.

Dữ liệu vào: File: A.IN

Gồm nhiều dòng, chỉ bao gồm các chữ cái.

Kết quả: Ghi ra màn hình

Viết ra số từ khác nhau trong file.

Ví dụ:

A.IN	Kết quả
Xin chao cac ban	8
Xin moi cac ban tap trung hoc tap	

BÀI 1.4.2: ĐẾM SỐ NGUYÊN TỔ KHÁC NHAU

Cho một file văn bản chỉ bao gồm các số nguyên không quá 9 chữ số. Viết chương trình đếm xem trong file đó có bao nhiêu số nguyên tố khác nhau.

Dữ liệu vào: File: B.IN

Gồm nhiều dòng, chỉ bao gồm các số nguyên, cách nhau một hoặc một vài khoảng trống.

Kết quả: Ghi ra màn hình

Viết ra số các số nguyên tố khác nhau.

Ví dụ:

B.IN	Kết quả
11 13 14 16 18 21 13 23 99 88 77 66 13 13 13 13 23 23 23 13 24 25 26 27 28 97	4

BÀI 1.4.3: TÌM TỪ THUẬN NGHỊCH DÀI NHẤT TRONG FILE VĂN BẢN

Cho một file văn bản bất kỳ. Hãy tìm ra từ thỏa mãn tính chất **thuận nghịch có độ dài lớn nhất** trong file đó và cho biết từ đó **xuất hiện bao nhiêu lần**. Nếu có nhiều từ cùng có độ dài lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện trong file ban đầu.

Dữ liệu vào: File C.IN

Gồm một đoạn văn bản bất kỳ. Không quá 1000 từ.

Kết quả (ghi ra màn hình):

- Ghi ra trên một dòng từ thuận nghịch có độ dài lớn nhất và số lần xuất hiện của nó.
- Nếu có nhiều từ cùng có độ dài lớn nhất thì các từ được liệt kê theo thứ tự xuất hiện ban đầu.

Ví dụ:

C.IN	KẾT QUẢ
------	---------

AAA BAABA HDHDH ACBSD SRGTDH DDDDS DUAHD AAA AD DA HDHDH AAA AAA AAA AAA DDDAS HDHDH HDH AAA AAA AAA AAA AAA AAA AAA AAA DHKFKH DHDHDD HDHDHD DDDHHH HHHDDD TDTD	HDHDH 3
---	---------

BÀI 1.4.4: TÌM TỪ DÀI NHẤT TRONG FILE

Cho một file văn bản bất kỳ. Hãy tìm ra từ có độ dài lớn nhất trong file. Nếu có nhiều từ khác nhau có độ dài bằng nhau và bằng giá trị lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện trong file dữ liệu vào (nhưng một từ dù xuất hiện nhiều lần cũng chỉ được liệt kê một lần).

Dữ liệu vào: File D.IN

Gồm một đoạn văn bản bất kỳ. Không quá 1000 từ.

Kết quả:

Ghi ra màn hình từ dài nhất và độ dài của nó, cách nhau một khoảng trống. Nếu có nhiều từ như vậy thì liệt kê lần lượt các từ theo thứ tự xuất hiện trong file ban đầu.

Ví dụ:

<u>D.IN</u>	<u>KẾT QUẢ</u>
Tiet hoc cuoi cung da ket thuc. Mon hoc Tin hoc co so 2 da ket thuc. Cac ban co gang on tap tot de thi dat ket qua cao. Chuc cac ban ngay cang gat hai duoc nhieu thanh cong tren con duong da chon	thuc. 5 nhieu 5 thanh 5 duong 5

BÀI 1.4.5: TÌM TỪ THUẬN NGHỊCH DÀI NHẤT TRONG FILE VĂN BẢN

Cho một file văn bản bất kỳ. Hãy tìm ra từ thỏa mãn tính chất **thuận nghịch có độ dài lớn nhất** trong file đó và cho biết từ đó **xuất hiện bao nhiêu lần**. Nếu có nhiều từ cùng có độ dài lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện trong file ban đầu.

Dữ liệu vào: File E.IN

Gồm một đoạn văn bản bất kỳ. Không quá 1000 từ.

Kết quả (ghi ra màn hình):

- Ghi ra trên một dòng từ thuận nghịch có độ dài lớn nhất và số lần xuất hiện của nó.
- Nếu có nhiều từ cùng có độ dài lớn nhất thì các từ được liệt kê theo thứ tự xuất hiện ban đầu.

Ví dụ:

E.IN	KẾT QUẢ
AAA BAABA HDHDH ACBSD SRGTDH DDDDS DUAHD AAA AD DA HDHDH AAA AAA AAA AAA DDDAS HDHDH HDH AAA AAA AAA AAA AAA AAA AAA AAA DHKFKH DHDHDD HDHDHD DDDHHH HHHDDD TDTD	HDHDH 3

BÀI 1.4.6: SẮP XẾP THÍ SINH

Hãy sắp xếp danh sách thí sinh đã có trong file theo tổng điểm giảm dần.

Mỗi thí sinh gồm các thông tin:

- Mã thí sinh: là một số nguyên, tự động tăng. Tính từ 1.
- Tên thí sinh, ngày sinh
- Điểm môn 1, điểm môn 2, điểm môn 3

Dữ liệu vào: File: F.IN

Dòng đầu chứa số thí sinh. Mỗi thí sinh viết trên 3 dòng:

- Dòng 1: Tên thí sinh
- Dòng 2: Ngày sinh
- Dòng 3,4,5: 3 điểm thi tương ứng. Các điểm thi đều đảm bảo hợp lệ (từ 0 đến 10).

Kết quả: Ghi ra màn hình

In ra danh sách thí sinh đã sắp xếp theo tổng điểm giảm dần. Nếu 2 thí sinh bằng điểm nhau thì thí sinh nào xuất hiện trước trong file sẽ viết trước. Mỗi thí sinh viết trên một dòng gồm: mã, tên, ngày sinh và tổng điểm. Các thông tin cách nhau đúng 1 khoảng trống. Điểm tổng được làm tròn đến 1 số sau dấu phẩy.

Ví dụ

F.IN	Kết quả
3 Nguyen Van A 12/12/1994 3.5 7.0 5.5 Nguyen Van B 1/9/1994 7.5 9.5 9.5 Nguyen Van C 6/7/1994 4.5 4.5 5.0	2 Nguyen Van B 1/9/1994 26.5 1 Nguyen Van A 12/12/1994 16.0 3 Nguyen Van C 6/7/1994 14.0

BÀI 1.4.7: SẮP XẾP MẶT HÀNG

Hãy sắp xếp danh sách các mặt hàng đã có trong file theo lợi nhuận giảm dần.

Mỗi mặt hàng gồm các thông tin:

- Mã mặt hàng: là một số nguyên, tự động tăng. Tính từ 1.
- Tên mặt hàng, nhóm hàng: là các xâu ký tự
- Giá mua, giá bán: là các số thực (không quá 9 chữ số)

Dữ liệu vào: File: H.IN

Dòng đầu chứa số mặt hàng. Mỗi mặt hàng viết trên 4 dòng:

- Dòng 1: Tên mặt hàng
- Dòng 2: Nhóm hàng
- Dòng 3: Giá mua.
- Dòng 4: Giá bán

Kết quả: Ghi ra màn hình

In ra danh sách mặt hàng đã sắp xếp theo lợi nhuận giảm dần (lợi nhuận tính bằng giá bán trừ đi giá mua). Nếu 2 mặt hàng lợi nhuận bằng nhau thì mặt hàng nào xuất hiện trước trong file sẽ viết trước. Mỗi mặt hàng viết trên một dòng gồm: mã, tên, nhóm hàng và lợi nhuận. Các thông tin cách nhau đúng 1 khoảng trống.

Ví dụ

D.IN	Kết quả
3 May tính SONY VAIO Dien tu 16400 17699 Tu lanh Side by Side Dien lanh 18300 25999 Banh Chocopie Tieu dung 27.5 37	2 Tu lanh Side by Side Dien lanh 7699 1 May tinh SONY VAIO Dien tu 1299 3 Banh Chocopie Tieu dung 10.5