

Πρόβλημα:Αντιστροφή αριθμών

Εκφώνηση

Αντιστρέψτε την σειρά των αριθμών που δίνονται στην είσοδο.

Δεδομένα εισόδου

standard input:

Στην πρώτη γραμμή εισόδου δίνεται ο ακέραιος αριθμός N ($1 \leq N \leq 10000$). Ακολουθούν N γραμμές, η καθεμία από τις οποίες περιέχει έναν αριθμό με μήκος το πολύ 18.

Σημείωση:

Δεν απαιτείται έλεγχος εγκυρότητας των τιμών της εισόδου.

Υπόδειξη:

Μπορείτε να χρησιμοποιήσετε string/char για την αποθήκευση των αριθμών που διαβάζετε από την standard input.

Δεδομένα εξόδου

standard output:

Η έξοδος αποτελείται από N γραμμές με κάθε μία να περιέχει έναν αριθμό.

(Οι αριθμοί πρέπει να τυπώνονται αυτούσια όπως δόθηκαν, π.χ. εάν δόθηκε 42.00, τότε εκτύπωση του 42.0 ή του 42.000 θεωρείται λάθος κ.ο.κ.)

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

10

1

2

3

4

5

6

7

8

9

10

Παράδειγμα εξόδου

10
9
8
7
6
5
4
3
2
1

Παράδειγμα εισόδου 2

3
1
4.2
9.8

Παράδειγμα εξόδου 2

9.8
4.2
1

Πρόβλημα:Επεξεργασία αλφαριθμητικού

Εκφώνηση

Να γραφεί πρόγραμμα, το οποίο διαβάσει αλφαριθμητικά μέχρι να διαβάσει αλφαριθμητικό με μήκος 1. Για κάθε αλφαριθμητικό που διαβάσει θα το τροποποιεί, έτσι ώστε τα πεζά γράμματα να γίνονται κεφαλαία. Κάθε τροποποιημένο αλφαριθμητικό θα πρέπει να εμφανίζεται στην οθόνη.

Δεδομένα εισόδου

standard input:

Κάθε γραμμή περιέχει ένα αλφαριθμητικό με μήκος L ($1 \leq L \leq 100$)

Σημείωση:

Δεν απαιτείται έλεγχος εγκυρότητας των τιμών της εισόδου.

Δεδομένα εξόδου

standard output:

Τυπώστε κάθε τροποποιημένο αλφαριθμητικό με αλλαγή γραμμής με την σειρά που δόθηκαν στην είσοδο.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

```
hElLo  
World  
THIS  
is  
a  
texT  
You  
HAVE  
TO  
mOdifY  
J
```

Παράδειγμα εξόδου

```
HELLO  
WORLD  
IS  
A
```

Επεξήγηση παραδείγματος

Διαβάζουμε τις δύο πρώτες λέξεις, τις τροποποιούμε και τις τυπώνουμε. Η 3η συμβολοσειρά δεν χρειάζεται κάποια τροποποίηση, οπότε δεν την τυπώνουμε. Έπειτα τροποποιούμε την 4η γραμμή και την τυπώνουμε. Τέλος διαβάζουμε την συμβολοσειρά 'a', που έχει μήκος 1, οπότε τυπώνουμε τη νέα τροποποιημένη συμβολοσειρά και τελειώνουμε την εκτέλεση του προγράμματος.

Παράδειγμα εισόδου 2

```
MIYGLAMGFJAOGJEOAGJ  
j
```

Παράδειγμα εξόδου 2

```
J
```

Επεξήγηση παραδείγματος 2

Το πρώτο αλφαριθμητικό δεν χρειάζεται κάποια επεξεργασία, οπότε δεν το τυπώνουμε. Στην συνέχεια τροποποιούμε το δεύτερο αλφαριθμητικό, το

τυπώνουμε και επειδή το μήκος της συμβολοσειράς του είναι ίσο με 1 τελειώνουμε την εκτέλεση του προγράμματος.

Πρόβλημα:Αντιστροφή αλφαριθμητικών

Εκφώνηση

Να γραφεί πρόγραμμα, το οποίο διαβάζει αλφαριθμητικά μέχρι να διαβάσει το αλφαριθμητικό ΤΕΛΟΣ. Για κάθε αλφαριθμητικό που διαβάζει θα πρέπει να το αντιστρέφει και να το εμφανίζει.

Δεδομένα εισόδου

standard input:

Σε κάθε γραμμή δίνεται ένα αλφαριθμητικό με μήκος το πολύ L ($1 \leq L \leq 100$).

Σημείωση:

Δεν απαιτείται έλεγχος εγκυρότητας των τιμών της εισόδου.

Δεδομένα εξόδου

standard output:

Σε κάθε γραμμή τυπώστε το κάθε αλφαριθμητικό αντίστροφα.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

Andreas
Nikos
Giorgos
TELOS

Παράδειγμα εξόδου

saerdnA
sokiN
sogroiG

Πρόβλημα:Αναλογία πινάκων

Εκφώνηση

Να γραφεί πρόγραμμα το οποίο διαβάζει από το πληκτρολόγιο T ($1 \leq T \leq 100$) ακέραια στοιχεία ενός μονοδιάστατου πίνακα και να τα τοποθετεί σε έναν δισδιάστατο $N \times M$ ($1 \leq N \cdot M \leq 100$) πίνακα στήλη προς στήλη.

Για το πρόβλημα αυτό έχει δημιουργηθεί μια βιβλιοθήκη που θα πρέπει να χρησιμοποιηθεί από το πρόγραμμα σου. Οι συναρτήσεις που μπορείς να εκτελέσεις και υποστηρίζονται από τη βιβλιοθήκη είναι 2 διαφορετικές:

- α) `setY(i , j, x)`, η οποία θέτει $Y[i][j] = x$;
- β) `CompareArrays()`, η οποία έλεγχει εάν τοποθετήθηκαν σωστά οι τιμές στον πίνακα Y

Προκειμένου να χρησιμοποιήσετε την βιβλιοθήκη, ο πηγαίος κώδικας θα πρέπει να περιέχει την γραμμή
`#include "CompareArrays.h",`

Σημείωση: Οι δείκτες των πινάκων ξεκινούν από το 0.
Θεωρήστε ότι πάντα $N \cdot M = T$

Δεδομένα εισόδου

standard input:

Στην 1η γραμμή εισόδου δίνονται οι αριθμοί T , N , M .

Ακολουθούν T γραμμές που περιέχουν τις τιμές του πίνακα X .

Σημείωση:

Δεν απαιτείται έλεγχος εγκυρότητας των τιμών της εισόδου.

Δεδομένα εξόδου

Καλέστε την συνάρτηση `CompareArrays()`

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

12 3 4

1
2
3
4
5
6
7
8
9
10
11
12

Παράδειγμα εξόδου

Καλέστε την συνάρτηση CompareArrays()

Επεξήγηση παραδείγματος

Έστω ότι έχουμε τον μονοδιάστατο πίνακα X και τον δισδιάστατο πίνακα Y, τότε

$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

οπότε

$Y = \{ \{1, 4, 7, 10\}$

$\{2, 5, 8, 11\}$

$\{3, 6, 9, 12\} \}$

και τέλος καλούμε την συνάρτηση CompareArrays()

Πρόβλημα: Διακριτά στοιχεία

Εκφώνηση

Σας δίνεται μία ακολουθία από N αριθμούς. Βρείτε πόσοι διακριτοί αριθμοί υπάρχουν.

Δεδομένα εισόδου

standard input:

Στην 1η γραμμή δίνεται ο αριθμός N ($1 \leq N \leq 10000$).

Στις επόμενες N γραμμές που η κάθε μια περιέχει έναν ακέραιο αριθμό.

Σημείωση:

Δεν απαιτείται έλεγχος εγκυρότητας των τιμών της εισόδου.

Δεδομένα εξόδου

standard output:

Μία ΓΡΑΜΜΗ που περιέχει το πλήθος των διακριτών στοιχείων.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

4

1
15
10
1

Παράδειγμα εξόδου

3

Παράδειγμα εισόδου 2

9
45
40
45
94
90
140
314
553
59

Παράδειγμα εξόδου 2

8

Επεξήγηση παραδείγματος 2

Τα διακριτά στοιχεία είναι τα $\{40,45,59,90,94,140,314,553\}$

Παράδειγμα εισόδου 3

23
3
41
748
515
49
5345
525
5252
48
24

41
59
385
543
34582
41
234
49
50
748
230
385
34

Παράδειγμα εξόδου 3

18

Επεξήγηση παραδείγματος 3

Τα διακριτά στοιχεία είναι τα

{3,24,34,41,48,49,50,59,230,234,385,515,525,543,748,5252,5345,34582}

Πρόβλημα:Σχέσεις φιλίας

Εκφώνηση

Έστω F η μήτρα γνωριμίας των μαθητών ενός σχολείου, με

$F = (f_{ij})$, $i, j = 1, 2, \dots, n$

$f_{ij} = 1$, αν οι μαθητές i και j είναι φίλοι

0, διαφορετικά

όπου ΔΕΝ υποτίθεται ότι αν ο i είναι φίλος με τον j τότε και ο j είναι φίλος με τον i και ο καθένας θεωρείται ότι είναι φίλος με τον εαυτό του. Να γραφεί πρόγραμμα που να δέχεται ως είσοδο τις τιμές του πίνακα f_{ij} και να βγάζει στην έξοδο τους τρεις μαθητές με τις περισσότερες φιλίες σε αύξουσα σειρά.

Δεδομένα εισόδου

standard input:

Στην 1η γραμμή δίνεται ο αριθμός N ($3 \leq N \leq 100$).

Στην συνέχεια ακολουθούν N γραμμές, όπου κάθε μία γραμμή περιέχει N αριθμούς που αντιπροσωπεύουν τις τιμές του πίνακα F .

Σημείωση:

Δεν απαιτείται έλεγχος εγκυρότητας των τιμών της εισόδου.

Δεδομένα εξόδου

standard output:

3 γραμμές όπου η κάθε μία περιέχει έναν ακέραιο αριθμό που αντιπροσωπεύει τον i -οστό μαθητή

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

```
3
1 0 0
0 1 0
0 0 1
```

Παράδειγμα εξόδου

```
1
2
3
```

Επεξήγηση παραδείγματος

Κάθε μαθητής είναι φίλος με τον εαυτό του.

Παράδειγμα εισόδου 2

```
4
1 0 0 0
```

0 1 1 0
0 0 1 0
1 1 1 1

Παράδειγμα εξόδου 2

3
2
4

Επεξήγηση παραδείγματος 2

ο 1ος και ο 3ος μαθητής έχουν από ένα 1 φίλο,
ο 2ος μαθητής έχει 2 φίλους
και ο 4ος μαθητής έχει 4 φίλους

Στην έξοδο υπερτερεί ο 3ος μαθητής αντί του 1ου μαθητή διότι $1 < 3$.
Επομένως η έξοδος είναι:

3
2
4

Πρόβλημα: Μάντεψε τον κρυφό αριθμό

Εκφώνηση

Εσύ και ένας φίλος σου παίζετε ένα παιχνίδι "Μάντεψε τον αριθμό". Αρχικά ο φίλος σας σκέφτεται έναν αριθμό N ($1 \leq N \leq 1000$) και ένα αριθμό K ($1 \leq K \leq N$). Στόχος σου είναι να βρεις τον αριθμό X κάνοντας όσο το δυνατόν πιο λίγες ερωτήσεις.

Για το πρόβλημα αυτό έχει δημιουργηθεί μια βιβλιοθήκη που θα πρέπει να χρησιμοποιηθεί από το πρόγραμμα σου. Οι ερωτήσεις που μπορείς να κάνεις και υποστηρίζονται από τη βιβλιοθήκη είναι 2 διαφορετικές:

α) `getN()`, η οποία επιστρέφει τον ακέραιο αριθμό N

β) Ερώτηση σχετικά με τον αριθμό που σκέφτηκες μέσω της συνάρτησης `compare(x)`. Οι πιθανές απαντήσεις στην ερώτηση αυτή είναι:

0, ο αριθμός x είναι ίσος με τον αριθμό K

-1, ο αριθμός x είναι μικρότερος από τον αριθμό K

1, ο αριθμός x είναι μεγαλύτερος από τον αριθμό K

Επίσης όταν το πρόγραμμα ξεκινήσει πρέπει πρώτα να καλέσει την `getN()`. Το πρόγραμμα σου δεν πρέπει να κάνει πάνω από 15 ερωτήσεις συνολικά για να βρει την απάντηση.

Προκειμένου να χρησιμοποιήσετε την βιβλιοθήκη, ο πηγαίος κώδικας θα πρέπει να περιέχει την γραμμή

```
#include "guesslib.h"
```

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Επεξήγηση παραδείγματος

#Πρόγραμμα #Βιβλιοθήκη

Καλεί `getN()` επιστρέφει 100

Καλεί `compare(50)` επιστρέφει -1

Καλεί `compare(75)` επιστρέφει -1

Καλεί `compare(88)` επιστρέφει -1

Καλεί `compare(94)` επιστρέφει 1

Καλεί `compare(91)` επιστρέφει -1

Καλεί `compare(92)` επιστρέφει 0

Εκτύπωση του 92 με αλλαγή γραμμής

Τερματισμός

Πρόβλημα:Μέγιστο στοιχείο μήτρας

Εκφώνηση

Να γίνει πρόγραμμα που να δίνει σαν έξοδο το μέγιστο στοιχείο μήτρας ακεραίων $N \times M$.

Δεδομένα εισόδου

Στην πρώτη γραμμή της εισόδου περιέχονται δύο ακεραίοι N ($1 \leq N \leq 100$), M ($1 \leq M \leq 100$), έπειτα ακολουθούν N γραμμές με κάθε μία να περιέχει M αριθμούς.

Δεδομένα εξόδου

Μία γραμμή που περιέχει έναν ακέραιο, το μέγιστο στοιχείο της μήτρας.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

```
3 3
1 2 3
4 5 6
7 8 9
```

Παράδειγμα εξόδου

```
9
```

Πρόβλημα:Μέγιστο μέσου όρου άρτιων γραμμών

Εκφώνηση

Να γίνει πρόγραμμα που θα υπολογίζει τον μέσο όρο των μέγιστων στοιχείων των άρτιων γραμμών μιας μήτρας ακεραίων $N \times M$.

Δεδομένα εισόδου

Στην πρώτη γραμμή της εισόδου περιέχονται δύο ακεραίοι N ($1 \leq N \leq 100$), M ($1 \leq M \leq 100$), έπειτα ακολουθούν N γραμμές με κάθε μία να περιέχει M ακέραιους αριθμούς.

Δεδομένα εξόδου

Μία γραμμή που περιέχει έναν πραγματικό αριθμό:ακέραιο, το μέγιστο στοιχείο της μήτρας. Ο αριθμός θα πρέπει να είναι στρογγυλοποιημένος στο 3ο δεκαδικό ψηφίο.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

4 3
1 2 3
4 5 6
7 8 9
2 2 3

Παράδειγμα εξόδου

4.50

Παράδειγμα εισόδου 2

6 8
3 9 2 5 6 5 8 8 10
7 6 3 1 5 6 2 3 7
8 9 3 2 10 9 7 1 3
7 5 4 3 8 7 8 10 1
5 5 1 2 8 1 4 8 9
5 9 6 6 1 6 5 7 10

Παράδειγμα εξόδου 2

9.00

Επεξήγηση παραδείγματος 2

2η γραμμή: $\max(7, 6, 3, 1, 5, 6, 2, 3, 7) = 7$

4η γραμμή: $\max(7, 5, 4, 3, 8, 7, 8, 10, 1) = 10$

6η γραμμή: $\max(5, 9, 6, 6, 1, 6, 5, 7, 10) = 10$

Μέσος όρος = $(7 + 10 + 10) / 3 = 9$

Παράδειγμα εισόδου 3

1 8
2 4 7 2 2 5 10 1 3

Παράδειγμα εξόδου 3

0.00

Πρόβλημα:Μέγιστος όρος αρνητικών τιμών

Εκφώνηση

Να γίνει πρόγραμμα το οποίο θα διαβάζει N αριθμούς και θα υπολογίζει το

μέγιστο όρο των αρνητικών τιμών της μήτρας. Σε περίπτωση που δεν υπάρχει τέτοιος όρος τυπώστε NOT FOUND.

Δεδομένα εισόδου

Στην πρώτη γραμμή της εισόδου περιέχεται ένας ακέραιος N ($1 \leq N \leq 100$), στην επόμενη γραμμή ακολουθούν N αριθμοί.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

```
6
3 -1 7 8 45 -10
```

Παράδειγμα εξόδου

```
-10
```

Παράδειγμα εισόδου 2

```
10
1 2 3 4 5 6 7 8 9 10
```

Παράδειγμα εξόδου 2

```
NOT FOUND
```

Πρόβλημα: Ρυθμός αύξησης πληθυσμού

Εκφώνηση

Η πόλη Α έχει N κατοίκους και ρυθμό αύξησης πληθυσμού $a\%$ τον χρόνο.
Η πόλη Β έχει M κατοίκους και ρυθμό αύξησης πληθυσμού $b\%$ τον χρόνο.
Σε πόσα χρόνια ο πληθυσμός της Α θα περάσει τον πληθυσμό της Β?

Δεδομένα εισόδου

Μία γραμμή που περιέχει 4 ακέραιους, N ($1 \leq N \leq 100000$), M ($1 \leq M \leq 100000$), a ($1 \leq a \leq 100$), b ($1 \leq b \leq 100$).

Είναι εγγυημένο ότι στα αρχεία ελέγχου θα ισχύει πάντα $a > b$.

Δεδομένα εξόδου

Μία γραμμή που περιέχει τα χρόνια που ο πληθυσμός της Α θα περάσει τον πληθυσμό της Β.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

60000 80000 20 10

Παράδειγμα εξόδου

4

Επεξήγηση παραδείγματος

Στο τέλος του 1ου χρόνου

Πληθυσμός A: 60000 -> 72000
Πληθυσμός B: 80000 -> 88000

Στο τέλος του 2ου χρόνου
Πληθυσμός A: 72000 -> 86400
Πληθυσμός B: 88000 -> 96800

Στο τέλος του 3ου χρόνου
Πληθυσμός A: 86400 -> 103680
Πληθυσμός B: 96800 -> 106480

Στο τέλος του 4ου χρόνου
Πληθυσμός A: 103680 -> 124416
Πληθυσμός B: 106480 -> 117128

$117128 < 124416$, οπότε η απάντηση είναι 4 χρόνια.

Πρόβλημα:Υπερωρίες

Εκφώνηση

Να γίνει πρόγραμμα στο οποίο ο χρήστης δίνει το πλήθος των ωρών N ($1 \leq N \leq 70$) που εργάστηκε την προηγούμενη εβδομάδα και την ωριαία αντιμισθία σε ευρώ W ($1 \leq W \leq 5$) και το πρόγραμμα επιστρέφει το ποσό που πρέπει να εισπράξει ο χρήστης. Όταν οι ώρες ξεπερνούν τις 40 θεωρούνται υπερωρίες και οι επιπλέον ώρες πληρώνονται προσαυξημένες κατά 50%.

Δεδομένα εισόδου

Μία γραμμή που περιέχει δύο ακεραίους τους N , M .

Δεδομένα εξόδου

Μία γραμμή που περιέχει το ποσό που πρέπει να εισπράξει ο χρήστης.

Περιορισμοί

- Όριο εκτέλεσης: 1 sec.
- Όριο μνήμης: 64MB.

Παράδειγμα εισόδου

40 5

Παράδειγμα εξόδου

200

Επεξήγηση παραδείγματος

Ο χρήστης έχει δουλέψει 40 ώρες με ωρομίσθιο 5€. Υπερωρίες δεν υπάρχουν, οπότε εισπράττει $40 \cdot 5 = 200\text{€}$.

Παράδειγμα εισόδου 2

70 4

Παράδειγμα εξόδου 2

120

Επεξήγηση παραδείγματος 2

Ο χρήστης έχει δουλέψει 70 ώρες με ωρομίσθιο 4€, με 30 ώρες υπερωρίας. Οπότε συνολικά κερδίζει $40 \cdot 4 + 30 \cdot 2 = 120\text{€}$.