



1. ΧΑΛΑΣΜΕΝΟ ΧΡΟΝΟΜΕΤΡΟ

Ο Μίλτος το δελφίνι είναι ένα πολύ ταξιδιάρικο δελφίνι και λατρεύει να διανύει μεγάλες αποστάσεις και να χρονομετρεί τον εαυτό του. Το χρονόμετρο όμως που χρησιμοποιεί έχει δυστυχώς χαλάσει και δείχνει τη μισή ώρα από αυτή που έπρεπε να δείξει κανονικά. Δηλαδή αν ο Μίλτος κάνει μια απόσταση σε 38 δευτερόλεπτα το χρονόμετρο θα δείξει ότι πέρασαν μόνο 19 δευτερόλεπτα.

Ο Μίλτος δεν ξέρει τι να κάνει και στράφηκε σε εσένα να τον βοηθήσεις. Μπορείς να γράψεις ένα πρόγραμμα που να δέχεται σαν είσοδο τις λανθασμένες ενδείξεις του χρονομέτρου και να υπολογίζει ποιες είναι οι σωστές;

Δεδομένα Εισόδου (time.in)

Το πρόγραμμα θα δέχεται σαν είσοδο τις λανθασμένες ενδείξεις του χρονομέτρου στη μορφή ΩΩ:ΛΛ:ΔΔ, όπου ΩΩ, ΛΛ, ΔΔ είναι ένας διψήφιος ακέραιος με $00 \leq \Omega\Omega \leq 99$, $00 \leq \Lambda\Lambda \leq 59$, $00 \leq \Delta\Delta \leq 59$. Δεν θα υπάρχει είσοδος στην οποία θα έχουν περάσει περισσότερες από 50 ώρες.

Δεδομένα Εξόδου (time.out)

Το πρόγραμμα θα τυπώνει ποια είναι η ένδειξη που θα έπρεπε να δείξει το χρονόμετρο στην ίδια μορφή με την ένδειξη της εισόδου.

Παράδειγμα εισόδου (αρχείο "time.in")

12:44:03

Παράδειγμα εξόδου (αρχείο "time.out")

25:28:06



2. ΕΙΣΙΤΗΡΙΑ ΓΙΑ ΤΟ ΕΝΥΔΡΕΙΟ

Το σχολείο σου ετοιμάζει μια εκδρομή για τους μαθητές στο ενυδρείο και εσύ ανέλαβες να πάρεις τηλέφωνο και να ρωτήσεις για την έκπτωση που θα έχετε μιας και θα πάνε τόσα πολλά άτομα. Ο διευθυντής του σχολείου σε έχει πληροφορήσει ότι N άτομα έχουν δηλώσει συμμετοχή. Ο υπεύθυνος του ενυδρείου σου είπε ότι αν πάνε τουλάχιστον A άτομα θα έχετε 10% έκπτωση, αν πάνε

τουλάχιστον B 20%, τουλάχιστον Γ 30% και τουλάχιστον Δ 50%. Η τιμή του κανονικού εισιτηρίου είναι 10 ευρώ.

Μπορείς να βρεις ποιο είναι το ελάχιστο συνολικό ποσό που πρέπει να πληρώσει το σχολείο έτσι ώστε να μπορέσουν να πάνε όλοι οι N μαθητές στο ενυδρείο. Προσοχή, σε ορισμένες περιπτώσεις μπορεί να συμφέρει να αγοραστούν περισσότερα εισιτήρια ώστε να έχετε καλύτερο πακέτο προσφοράς.

Δεδομένα Εισόδου (tickets.in)

Η πρώτη γραμμή θα περιέχει έναν ακέραιο N ($N \leq 10000$). Η δεύτερη γραμμή θα περιέχει τους 4 ακεραίους A , B , Γ και Δ . ($A, B, \Gamma, \Delta \leq 10000$)

Δεδομένα Εξόδου (tickets.out)

Ένας ακέραιος, η ελάχιστη συνολική τιμή.

Παράδειγμα εισόδου (αρχείο "tickets.in")

```
30
10 20 50 100
```

Παράδειγμα εξόδου (αρχείο "tickets.out")

```
240
```

Παράδειγμα εισόδου 2 (αρχείο "tickets.in")

```
95
10 20 50 100
```

Παράδειγμα εξόδου 2 (αρχείο "tickets.out")

```
500
```



3. ΛΗΨΗ ΑΡΧΕΙΩΝ

Κατεβάζεις κάποια αρχεία από το διαδίκτυο και θες να ξέρεις σε πόσο χρόνο θα έχουν κατέβει πλήρως όλα.

Για κάθε αρχείο που κατεβαίνει γνωρίζεις την τρέχουσα ταχύτητα (σε KB/s) και τον χρόνο που απομένει σε κάθε αρχείο βασισμένο σε αυτή την ταχύτητα (σε δευτερόλεπτα). Το άθροισμα των ταχυτήτων όλων των αρχείων είναι ίσο με το bandwidth, το οποίο παραμένει σταθερό και χρησιμοποιείται πλήρως όσο κατεβαίνουν τα αρχεία. Αυτό σημαίνει πως όταν η λήψη ενός αρχείου ολοκληρωθεί, το διαθέσιμο bandwidth θα κατανεμηθεί στα υπόλοιπα αρχεία με αποτέλεσμα να αρχίσουν να κατεβαίνουν πιο γρήγορα. Ο τρόπος με τον οποίο θα κατανεμηθεί δεν επηρεάζει την τελική απάντηση.

Για παράδειγμα, έστω ότι κατεβαίνουν 2 αρχεία:

- 1) Ταχύτητα = 3 KB/s, Απομένουν 57 δευτερόλεπτα
- 2) Ταχύτητα = 2 KB/s, Απομένουν 22 δευτερόλεπτα

Μετά από 22 δευτερόλεπτα, το δεύτερο αρχείο θα έχει ολοκληρωθεί. Το πρώτο θα χρειάζεται όμως 35 δευτερόλεπτα ακόμα, αλλά αυτός ο χρόνος έχει υπολογιστεί με βάση την αρχική του ταχύτητα. Λόγω όμως του διαθέσιμου bandwidth, η ταχύτητα του θα αυξηθεί από 3 σε 5 KB/s. Ο νέος χρόνος που απομένει είναι: $35 * 3/5 = 21$ δευτερόλεπτα. Οπότε ο χρόνος που χρειάζεται για να έχουν κατέβει πλήρως και τα 2 αρχεία είναι $22+21=43$ δευτερόλεπτα.

Δεδομένα Εισόδου (downloads.in)

Η πρώτη γραμμή της εισόδου περιέχει τον αριθμό N ($1 \leq N \leq 100$) που είναι ο αριθμός των αρχείων που κατεβαίνουν. Ακολουθούν N γραμμές οι οποίες περιέχουν 2 ακεραίους T_n και X_n , την ταχύτητα και τον χρόνο αντίστοιχα του n -οστού αρχείου

Δεδομένα Εξόδου (downloads.out)

Το πρόγραμμα πρέπει να τυπώνει σε μια γραμμή έναν αριθμό, το χρόνο (σε δευτερόλεπτα) που χρειάζεται για να κατέβουν όλα τα αρχεία. Ο αριθμός πρέπει να είναι στρογγυλοποιημένος στον πλησιέστερο ακέραιο.

Παράδειγμα εισόδου (αρχείο "downloads.in")

```
2
3 57
2 22
```

Παράδειγμα εξόδου (αρχείο "downloads.out")

```
43
```

4. ΕΙΚΑΣΙΑ ΤΟΥ GOLDBACH

Ένα από τα μεγαλύτερα άλυτα προβλήματα της αρχαιότητας αποτελεί η εικασία του Goldbach, η οποία δηλώνει ότι κάθε άρτιος αριθμός μεγαλύτερος του 2 μπορεί να γραφεί ως άθροισμα δύο πρώτων αριθμών.



Ως πρώτος αριθμός ορίζεται κάθε φυσικός αριθμός μεγαλύτερος του 1, ο οποίος έχει ως διαιρέτες μόνο το 1 και τον εαυτό του. Έτσι οι πρώτοι δέκα πρώτοι αριθμοί είναι: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Καλείστε να κατασκευάσετε πρόγραμμα προκειμένου να επαληθεύσετε την εικασία του Goldbach.

Δεδομένα Εισόδου (goldbach.in)

Το αρχείο εισόδου αποτελείται από έναν άρτιο ακέραιο αριθμό N ($2 < N \leq 1000000$)

Δεδομένα Εξόδου (goldbach.out)

Το αρχείο εξόδου θα αποτελείται από μία μόνο γραμμή με δύο πρώτους αριθμούς P_1 P_2 χωρισμένους από ένα κενό. Θα πρέπει να ισχύει $P_1 + P_2 = N$. Σε περίπτωση που υπάρχουν πολλές λύσεις τυπώστε αυτή με το μικρότερο P_1 .

Παράδειγμα εισόδου (αρχείο "goldbach.in")

14

Παράδειγμα εξόδου (αρχείο "goldbach.out")

3 11