



27^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Θέμα 1^ο: Παρέες αριθμών

[30 Μονάδες]

Λέμε ότι δύο φυσικοί αριθμοί είναι στην ίδια **παρέα** όταν έχουν το ίδιο πλήθος άσων (1) στη δυαδική τους αναπαράσταση. Για παράδειγμα, το 5 και το 17 είναι στην ίδια παρέα γιατί $5 = 101_{(2)}$ και $17 = 10001_{(2)}$, άρα και οι δύο αυτοί αριθμοί έχουν δύο άσους στη δυαδική τους αναπαράσταση. Αντίθετα, το 17 και το 42 ανήκουν σε διαφορετικές παρέες.

Πρόβλημα

Να γραφεί ένα πρόγραμμα σε μια από τις γλώσσες του IOI το οποίο θα διαβάζει μία ακολουθία N αριθμών και στη συνέχεια θα βρίσκει το πλήθος των μελών της **μεγαλύτερης** παρέας που μπορεί να σχηματιστεί από όρους της ακολουθίας. Η ακολουθία ενδέχεται να περιέχει όρους που είναι ίσοι και, στην περίπτωση αυτή, θα πρέπει να συμπεριλάβετε όλους τους ίσους όρους στην ίδια φυσικά παρέα.

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **aces.in** είναι αρχεία κειμένου αποτελούμενα από δύο γραμμές: Στην πρώτη γραμμή δίνεται ακριβώς ένας ακέραιος: το πλήθος N των αριθμών της ακολουθίας ($1 \leq N \leq 1.000.000$). Στη δεύτερη γραμμή δίνονται N ακέραιοι αριθμοί A_i ($0 \leq A_i \leq 1.000.000.000$), χωρισμένοι ανά δύο με ένα κενό διάστημα: οι όροι της ακολουθίας.

Αρχεία Εξόδου:

Τα αρχεία εξόδου με όνομα **aces.out** είναι αρχεία κειμένου αποτελούμενα από ακριβώς μία γραμμή που περιέχει ακριβώς έναν ακέραιο αριθμό: το πλήθος των μελών της μεγαλύτερης παρέας αποτελούμενης από όρους της ακολουθίας.

Παράδειγμα Αρχείων Εισόδου - Εξόδου

aces.in	aces.out	<i>εξήγηση</i>
6 1 2 4 3 2 5	4	Στη μεγαλύτερη παρέα ανήκουν: 1, 2 (δύο φορές) και 4 που έχουν έναν άσο στη δυαδική τους αναπαράσταση.

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα newline.

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 16 MB.

Σελίδα 1 από 7

Copyright **ΕΠΥ** 2014-15. Σπύρου Τρικούπη 20, 106 83 Αθήνας, Spirou Trikoupi 20, PC 106 83, Athens, Greece

☎ +30-210-3300999, 📠 +30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr



Θέμα 2^ο: Εφημερίδες

[30 Μονάδες]

Ο Πέτρος μοιράζει τον ημερήσιο τύπο. Πρέπει να επισκεφθεί N σημεία πώλησης, τοποθετημένα σε μία κυκλική διαδρομή. Γνωρίζει ότι το i -οστό σημείο πώλησης του αφήνει κέρδος A_i . Αν όμως αρχίσει να επισκέπτεται τα N σημεία με τη σειρά, τότε σε κάποιες περιοχές ο τύπος θα φτάσει έγκαιρα σε όλα τα σημεία πώλησης, ενώ σε κάποιες άλλες περιοχές ο τύπος θα αργήσει να φτάσει.

Ο Πέτρος αποφασίζει λοιπόν να ξεκινάει από κάποιο σημείο πώλησης i , να αφήνει εκεί τις εφημερίδες και στη συνέχεια, κινούμενος κυκλικά, να προσπερνάει $K-1$ σημεία και να πηγαίνει στο επόμενο σημείο για να αφήσει την εκεί ζήτηση. Αφήνει δηλαδή εφημερίδες κάθε K σημεία, αρχίζοντας από το i . Το μόνο πρόβλημα είναι ότι έτσι του είναι δύσκολο να θυμάται πού έχει πάει και πού όχι. Αποφασίζει λοιπόν να σταματάει όταν φθάσει σε ένα σημείο όπου έχει ήδη αφήσει εφημερίδες. Έτσι, είναι πιθανό όταν σταματήσει να μην έχει επισκεφθεί όλα τα σημεία πώλησης.

Πρόβλημα

Να γραφεί ένα πρόγραμμα σε μια από τις γλώσσες του IOI το οποίο να διαβάσει τις τιμές του N και του K και τα κέρδη A_i και να βρίσκει το μεγαλύτερο συνολικό κέρδος που μπορεί να έχει ο Πέτρος, ακολουθώντας αυτή τη διαδικασία.

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **efimerides.in** είναι αρχεία κειμένου αποτελούμενα από δύο γραμμές: Στην πρώτη γραμμή δίνονται δύο ακέραιοι χωρισμένοι μεταξύ τους με ένα κενό διάστημα: οι τιμές των N και K ($2 \leq K \leq N \leq 1.000.000$). Στη δεύτερη γραμμή δίνονται N ακέραιοι αριθμοί A_i ($1 \leq A_i \leq 1.000.000$), χωρισμένοι ανά δύο με ένα κενό διάστημα: οι όροι της ακολουθίας. Το άθροισμα όλων των A_i δε θα υπερβαίνει το 1.000.000.000.

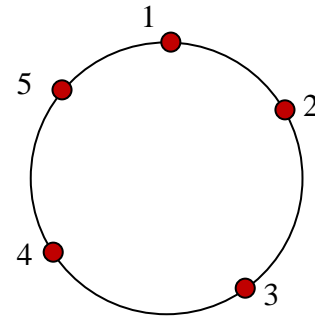
Αρχεία Εξόδου:

Τα αρχεία εξόδου με όνομα **efimerides.out** είναι αρχεία κειμένου αποτελούμενα από ακριβώς μία γραμμή που περιέχει ακριβώς έναν ακέραιο αριθμό: το μέγιστο δυνατό κέρδος του Πέτρου.



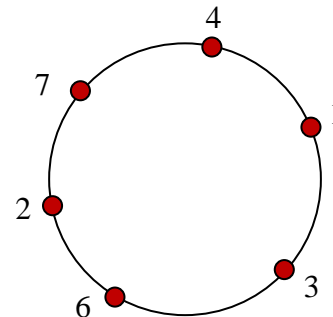
Παραδείγματα Αρχείων Εισόδου - Εξόδου 1^ο

efimerides.in	efimerides.out
5 2 1 2 3 4 5	15



2^ο

efimerides.in	efimerides.out
6 3 4 1 3 6 2 7	10



Εξήγηση παραδειγμάτων:

Στο πρώτο παράδειγμα, απ' όπου κι αν ξεκινήσει ο Πέτρος θα επισκεφθεί όλα τα σημεία πώλησης. Επομένως το συνολικό του κέρδος θα είναι ίσο με το συνολικό άθροισμα. Μια δυνατή διαδρομή με αυτό το συνολικό κέρδος είναι: $1 + 3 + 5 + 2 + 4 = 15$.

Στο δεύτερο παράδειγμα το σημείο από το οποίο ξεκινάει έχει σημασία. Αν π.χ. ξεκινήσει από το 2, θα έχει κέρδος $2 + 1 = 3$, ενώ αν ξεκινήσει από το 4, θα έχει κέρδος $4 + 6 = 10$. Το μέγιστο δυνατό κέρδος είναι 10 και μπορεί να το πάρει με άλλους, διαφορετικούς τρόπους, π.χ. $7 + 3 = 10$.

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα newline.

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 16 MB.



Θέμα 3^ο: Σουπερμάρκετ

[40 Μονάδες]

Ένα σουπερμάρκετ δίνει δωροεπιταγές στους πελάτες του με τον παρακάτω τρόπο. Ο πελάτης βάζει τα προϊόντα με μια σειρά στον ιμάντα του ταμείου και η ταμίας, όπως τα χρεώνει, όταν φτάσει στο προϊόν που βρίσκεται σε θέση πολλαπλάσια του αριθμού **K** δίνει μια δωροεπιταγή στον πελάτη ίσης αξίας με την τιμή αυτού του προϊόντος. Π.χ., για $K=3$, δίνει δωροεπιταγές ίσες με την αξία του τρίτου προϊόντος, του έκτου, του ένατου, κ.ο.κ.

Ο Θανάσης έχει βάλει τα **N** προϊόντα που έχει αγοράσει στον ιμάντα με τυχαία σειρά και γνωρίζει ότι η τιμή του i -οστού προϊόντος (με τη σειρά που τα έχει βάλει) είναι **A_i**. Τότε, πληροφορείται για τις δωροεπιταγές και μαθαίνει την τιμή του **K**. Φυσικά, θα ήθελε να μεγιστοποιήσει τη συνολική αξία των δωροεπιταγών που θα πάρει. Δεν προλαβαίνει όμως να αναδιατάξει όλα τα προϊόντα του. Το μόνο που προλαβαίνει να κάνει είναι να τα διατρέξει μία φορά, ξεκινώντας από αυτό που είναι πλησιέστερα στο ταμείο, και (αν θέλει) να μετακινήσει κάποια προϊόντα στο τέλος της σειράς. Μπορεί όμως να κάνει το πολύ **M** τέτοιες μετακινήσεις και δεν μπορεί να μετακινήσει το ίδιο προϊόν δύο φορές.

Πρόβλημα

Να γραφεί ένα πρόγραμμα σε μια από τις γλώσσες του IOI το οποίο να διαβάζει τις τιμές των **N**, **M** και **K**, τις αξίες των προϊόντων **A_i** που θέλει να αγοράσει ο Θανάσης και να βρίσκει τη μέγιστη δυνατή συνολική αξία δωροεπιταγών που μπορεί να κερδίσει.

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **supermarket.in** είναι αρχεία κειμένου αποτελούμενα από δύο γραμμές: Στην πρώτη γραμμή δίνονται τρεις ακέραιοι χωρισμένοι ανά δύο με ένα κενό διάστημα: οι τιμές των **N**, **M** και **K** ($1 \leq K \leq N \leq 100.000$, $0 \leq M \leq 500$). Στη δεύτερη γραμμή δίνονται **N** ακέραιοι αριθμοί **A_i** ($1 \leq A_i \leq 10.000.000$), χωρισμένοι ανά δύο με ένα κενό διάστημα: οι τιμές των προϊόντων με την αρχική σειρά. Το άθροισμα όλων των **A_i** δε θα υπερβαίνει το 1.000.000.000.

Αρχεία Εξόδου:

Τα αρχεία εξόδου με όνομα **supermarket.out** είναι αρχεία κειμένου αποτελούμενα από ακριβώς μία γραμμή που περιέχει ακριβώς έναν ακέραιο αριθμό: τη μέγιστη δυνατή συνολική αξία των δωροεπιταγών που θα πάρει ο Θανάσης.



Παράδειγμα Αρχείων Εισόδου - Εξόδου 1^ο

supermarket.in	supermarket.out
5 1 2 10 2 6 4 8	14

2^ο

supermarket.in	supermarket.out
5 2 2 10 1 1 1 10	11

Εξήγηση παραδειγμάτων:

Στο πρώτο παράδειγμα, ο Θανάσης μπορεί να επιλέξει να μετακινήσει στο τέλος το πρώτο προϊόν (αξίας 10) και έτσι η τελική σειρά με την οποία η ταμίας θα χρεώσει τα προϊόντα του να είναι 2, 6, 4, 8, 10. Επομένως, η συνολική αξία δωροεπιταγών που θα πάρει θα είναι $6+8=14$. Αυτό είναι το μέγιστο δυνατό σε αυτή την περίπτωση.

Στο δεύτερο παράδειγμα, παρά το γεγονός ότι μπορεί να μετακινήσει δύο προϊόντα, ο Θανάσης δεν έχει κανένα τρόπο να βάλει τα προϊόντα του με τέτοια σειρά ώστε να πάρει δωροεπιταγές για τα δύο «δεκάρια». Ένας τρόπος να πάρει το μέγιστο δυνατό είναι να μετακινήσει έναν από τους «άσους» στο τέλος και έτσι η τελική σειρά να είναι 10, 1, 1, 10, 1. Τότε, η συνολική αξία δωροεπιταγών που θα πάρει θα είναι $1+10=11$.

Βαθμολογία:

Σε τουλάχιστον 25% των περιπτώσεων ελέγχου θα είναι $M < N \leq 100$. Επίσης, σε όλες τις περιπτώσεις ελέγχου θα ισχύει (τουλάχιστον) ένα από τα παρακάτω:

- $N \leq 500$ και $M \leq 500$
- $N \leq 1.000$ και $M \leq 300$
- $N \leq 10.000$ και $M \leq 100$
- $N \leq 100.000$ και $M \leq 10$

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα newline.

Μέγιστος χρόνος εκτέλεσης: 2 sec.

Μέγιστη διαθέσιμη μνήμη: 64 MB.



ΝΑ ΑΠΑΝΤΗΣΕΤΕ ΚΑΙ ΣΤΑ ΤΡΙΑ ΘΕΜΑΤΑ

ΔΙΑΡΚΕΙΑ ΕΞΕΤΑΣΗΣ ΤΕΣΣΕΡΙΣ (4) ΩΡΕΣ

ΚΑΛΗ ΣΑΣ ΕΠΙΤΥΧΙΑ

Ακολουθούν χρήσιμες οδηγίες !

Διαβάστε τις ακόλουθες παρατηρήσεις προσεκτικά!

- ✓ Ερωτήσεις που αφορούν τις παρατηρήσεις αυτές δεν θα απαντηθούν. Η πιστή τήρηση των αναφερόμενων οδηγιών είναι απαραίτητη.
- ✓ Οι αναφερόμενοι σε κάθε θέμα χρόνοι είναι ενδεικτικοί. Η επιτροπή μπορεί να τους αυξομειώσει προκειμένου να επιτύχει καλύτερη κλιμάκωση της βαθμολογίας.

1. Στην αρχή του πηγαίου κώδικά σας, θα πρέπει να χρησιμοποιήσετε τις επικεφαλίδες, ανάλογα με το πρόβλημα πχ.:

```
/*  
USER: username  
LANG: C  
TASK: aces  
*/
```

για κώδικα σε C

```
/*  
USER: username  
LANG: C++  
TASK: aces  
*/
```

για κώδικα σε C++

```
(*  
USER: username  
LANG: PASCAL  
TASK: aces  
*)
```

για κώδικα σε PASCAL

2. Έλεγχος τιμών δεν απαιτείται. Οι τιμές των αρχείων ελέγχου είναι πάντα έγκυρες.
3. Το σύστημα αξιολόγησης «τρέχει» σε **Linux**. Σας προτείνουμε να δοκιμάζετε τις λύσεις σας στο σύστημα. Έχετε δικαίωμα πολλαπλών υποβολών μέχρι το τέλος του διαγωνισμού. Μετά από κάθε υποβολή θα λαμβάνετε την αξιολόγηση της λύσης σας, σε τμήμα των Αρχείων Ελέγχου.

Σελίδα 6 από 7



4. Οι επιλογές του μεταγλωττιστή που χρησιμοποιούνται για τη βαθμολόγηση είναι οι εξής:
 - C: `gcc -std=c99 -O2 -DCONTEST -s -static -lm`
 - C++: `g++ -O2 -DCONTEST -s -static -lm`
 - Free Pascal: `fpc -O2 -dCONTEST -XS`
 5. Το Linux ξεχωρίζει μεταξύ κεφαλαίων και πεζών γραμμάτων. Ελέγξτε ότι τα ονόματα των αρχείων εισόδου και εξόδου είναι γραμμένα με μικρά (πεζά) γράμματα.
 6. Τα προγράμματά σας πρέπει να επιστρέφουν ως κωδικό εξόδου το μηδέν.
 7. Για προγραμματισμό σε C και C++ η συνάρτηση `main()` πρέπει πάντα να τερματίζει με τις εντολές `"return(0);"` ή `"exit(0);"`.
 8. Οι προγραμματιστές σε Pascal πρέπει να χρησιμοποιούν την εντολή `"halt"` μόνο με κωδικό εξόδου το μηδέν (μόνο δηλαδή με την μορφή `"halt;"` ή `"halt(0);"`).
 9. Το πρόγραμμα αξιολόγησης θα εξετάσει την τιμή που επιστρέφει το πρόγραμμά σας. Εάν η τιμή αυτή δεν είναι μηδέν, τότε το πρόγραμμα δεν θα βαθμολογηθεί για το συγκεκριμένο test.
 10. Κανένας άλλος χαρακτήρας εκτός του χαρακτήρα νέας γραμμής (newline) (χαρακτήρας 0A στο ASCII εκφρασμένο στο δεκαεξαδικό σύστημα αρίθμησης, \n για προγραμματιστές C ή C++, \$0A για προγραμματιστές Pascal) δεν θα υπάρχει μετά τον τελευταίο αριθμό κάθε γραμμής των αρχείων εισόδου και εξόδου. Δηλαδή, κάθε γραμμή των αρχείων εισόδου και εξόδου, συμπεριλαμβανομένης και της τελευταίας, τερματίζεται με τον χαρακτήρα νέας γραμμής όπως ορίστηκε παραπάνω.
- ✓ Κάθε απόπειρα κακόβουλης εισόδου ή ακόμα και εξερεύνησης του συστήματος, εκτός της παρεχόμενης διεπαφής, θα εντοπίζεται και θα επιβάλλονται κυρώσεις.

Με τη συνεργασία:

Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών, Εθνικού Μετσόβιου Πολυτεχνείου, Πανεπιστημίου Αιγαίου, Πανεπιστημίου Ιωαννίνων, Πανεπιστημίου Πατρών, Πανεπιστημίου Παιραιώς, ΤΕΙ Αθήνας,

