



### Άσκηση 1: Ο Μικρός Τσαγκάρης

Στην μακρινή χώρα των Αλγορίθμων ζει ένα μικρός τσαγκάρης. Στο εργαστήριό του έχει δύο μηχανές παραγωγής παπουτσιών, η μία παράγει μόνο δεξιά παπούτσια και η άλλη μόνο αριστερά. Κάθε τύπος δεξιού παπουτσιού μπορεί να γίνει ζευγάρι με ένα μόνο τύπο αριστερού παπουτσιού και αντίστροφα. Λόγω περιορισμένου χώρου στο εργαστήριο, ο τσαγκάρης δεν μπορεί να αποθηκεύει τα παραγόμενα παπούτσια. Έτσι, παίρνοντας ένα παπούτσι από κάθε μηχανή, ο τσαγκάρης έχει τρεις επιλογές:

- Να πετάξει το δεξί παπούτσι και να πάρει ένα επόμενο από τη μηχανή που παράγει δεξιά παπούτσια, προσπαθώντας να το ταιριάξει με το αριστερό παπούτσι που έχει στα χέρια του.
- Να πετάξει το αριστερό παπούτσι και να πάρει ένα επόμενο από τη μηχανή που παράγει αριστερά παπούτσια, προσπαθώντας να το ταιριάξει με το δεξί παπούτσι που έχει στα χέρια του.
- Αν τα παπούτσια είναι ίδιου τύπου, να τα πουλήσει ως ζευγάρι.

Ο τσαγκάρης γνωρίζει την ακολουθία τύπων που παράγει κάθε μηχανή, αλλά δεν μπορεί να την ελέγξει. Μάλιστα γνωρίζει ότι κάθε μηχανή παράγει συνήθως πολλά παπούτσια του ίδιου τύπου διαδοχικά. Γράψτε ένα πρόγραμμα που θα βοηθήσει τον μικρό τσαγκάρη να σχεδιάσει τη στρατηγική του, ώστε να πουλήσει όσο περισσότερα ζευγάρια γίνεται.

**Δεδομένα Εισόδου:** Αρχικά, το πρόγραμμα θα διαβάσει από το standard input δύο θετικούς ακέραιους  $L$  και  $R$  που αντιπροσωπεύουν τα μήκη των ακολουθιών τύπων παπουτσιών που παράγουν οι δύο μηχανές. Στη συνέχεια, το πρόγραμμα θα διαβάσει μία γραμμή με  $2L$  θετικούς ακεραίους  $a_1, A_1, a_2, A_2, \dots, a_L, A_L$  και άλλη μία γραμμή με  $2R$  θετικούς ακεραίους  $b_1, B_1, b_2, B_2, \dots, b_R, B_R$ . Οι αριθμοί αυτοί δηλώνουν ότι η μηχανή αριστερών παπουτσιών θα παράξει  $a_1$  παπούτσια τύπου  $A_1$ , στη συνέχεια  $a_2$  παπούτσια τύπου  $A_2$ , κοκ., μέχρι να σταματήσει παράγοντας  $a_L$  αριστερά παπούτσια τύπου  $A_L$ , και ότι η μηχανή δεξιών παπουτσιών θα παράξει  $b_1$  παπούτσια τύπου  $B_1$ , στη συνέχεια  $b_2$  παπούτσια τύπου  $B_2$ , κοκ., μέχρι να σταματήσει παράγοντας  $b_R$  δεξιά παπούτσια τύπου  $B_R$ . Ένα δεξί και ένα αριστερό παπούτσι, μπορούν να γίνουν ζευγάρι μόνο εάν έχουν τον ίδιο αριθμό τύπου.

**Δεδομένα Εξόδου:** Το πρόγραμμα πρέπει να τυπώνει στο standard output (στην πρώτη γραμμή) το μέγιστο πλήθος ζευγαριών που μπορούν να πουληθούν. Σημειώστε ότι για μεγάλες τιμές των  $a_i$  και  $b_i$ , το μέγιστο πλήθος ζευγαριών (καθώς και κάποια από τα ενδιάμεσα αποτελέσματα που χρειάζονται για τον υπολογισμό του) μπορεί να υπερβαίνουν το  $2^{32}$ .

Περιορισμοί:	Παράδειγμα Εισόδου:	Παράδειγμα Εξόδου:
$1 \leq L, R \leq 100$	3 5	20
$1 \leq A_i, B_i \leq 100$	10 1 6 2 10 1	
Στο 60% των test cases θα έχουμε $1 \leq a_i, b_i \leq 25$	5 1 3 2 10 1 3 2 5 1	
Στο υπόλοιπο 40% των test cases θα έχουμε $1 \leq a_i, b_i \leq 10^{16}$	3 5	21
Όριο χρόνου εκτέλεσης: 1 sec.	10 1 6 2 10 1	
Όριο μνήμης: 64 MB.	5 1 6 2 10 1 6 2 5 1	
	1 1	0
	5000 10	
	50000 100	

## Άσκηση 2: Παιχνίδι Στρατηγικής

Σε ένα νέο ηλεκτρονικό παιχνίδι στρατηγικής, ο παίκτης αναλαμβάνει τον ρόλο του ηγεμόνα μιας αδύναμης χώρας που δέχεται επίθεση από έναν σημαντικά ισχυρότερο αντίπαλο. Αρχικά ο παίκτης ελέγχει όλες τις  $N$  πόλεις της χώρας του. Η χώρα είναι μακρόστενη, έτσι οι πόλεις μπορούν να αναπαρασταθούν ως μία ακολουθία διαδοχικών τετραγώνων που είναι αριθμημένα από το 1 έως το  $N$ , όπως φαίνεται στο Σχήμα 1. Στο τέλος κάθε χρόνου, κάθε πόλη  $i$  πληρώνει  $p_i$  χρυσά νομίσματα στον παίκτη, ως ετήσιο φόρο, εφόσον βέβαια βρίσκεται ακόμη υπό τον έλεγχο του. Ο αντίπαλος είναι σημαντικά ισχυρότερος, και έτσι είναι βέβαιο ότι τελικά θα επικρατήσει. Καθώς η ήττα είναι δεδομένη, στόχος του παίκτη είναι να συλλέξει όσο το δυνατόν περισσότερα χρυσά νομίσματα, πριν ολόκληρη η χώρα τεθεί υπό την κατοχή του αντιπάλου.

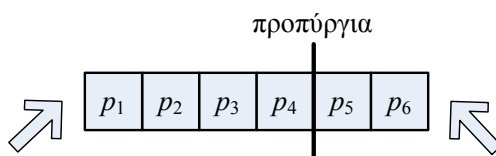
Στην αρχή κάθε έτους λοιπόν, ο παίκτης επιλέγει να κτίσει τα προπύργιά του μεταξύ δύο γειτονικών πόλεων που είναι ακόμη υπό τον έλεγχό του. Κατόπιν, ο αντίπαλος επιλέγει να επιτεθεί είτε από τα δυτικά κατακτώντας όλες τις πόλεις πριν τα προπύργια είτε από τα ανατολικά κατακτώντας όλες τις πόλεις μετά τα προπύργια. Στο τέλος του έτους, ο παίκτης εισπράττει τους φόρους από όλες τις πόλεις που παραμένουν ακόμη υπό τον έλεγχό του. Επίσης, μέχρι το τέλος του έτους, ο αντίπαλος έχει καταφέρει να καταστρέψει τα προπύργια, και μετά σταματά να μάχεται, και προετοιμάζεται για το επόμενο έτος. Το ίδιο σενάριο επαναλαμβάνεται κάθε χρόνο, μέχρι όλη η χώρα να τεθεί υπό τον έλεγχο του αντιπάλου.

Ο παίκτης, που έχει στόχο να συλλέξει όσο το δυνατόν περισσότερα χρυσά νομίσματα, πρέπει να επιλέξει προσεκτικά που θα κτίσει τα προπύργιά του κάθε χρόνο. Είναι γνωστό ότι ο αντίπαλος έχει υπηρεσίες πληροφοριών υψηλού επιπέδου, και ότι επιλέγει πάντα με βέλτιστο τρόπο αν θα επιτεθεί από τα δυτικά ή από τα ανατολικά, προσπαθώντας να ελαχιστοποιήσει το πλήθος των χρυσών νομισμάτων που θα συλλέξει ο παίκτης.

Με βάση το παραπάνω σενάριο, να γράψετε ένα πρόγραμμα που υπολογίζει το μέγιστο πλήθος χρυσών νομισμάτων που μπορεί να συλλέξει ο παίκτης.

**Παράδειγμα:** Ας θεωρήσουμε  $N = 5$  πόλεις, με  $p_1 = 8$ ,  $p_2 = 6$ ,  $p_3 = 2$ ,  $p_4 = 4$ , και  $p_5 = 2$ . Η καλύτερη επιλογή για τον παίκτη είναι να κτίσει τα προπύργιά του μεταξύ των πόλεων 2 και 3. Ο αντίπαλος επιτίθεται από τα δυτικά, κατακτά τις πόλεις 1 και 2, και αφήνει τον παίκτη να συλλέξει 8 χρυσά νομίσματα (από τις πόλεις 3, 4, και 5) στο τέλος του έτους. Το επόμενο έτος, ο παίκτης κτίζει τα προπύργιά του μεταξύ των πόλεων 3 και 4. Ο αντίπαλος επιτίθεται από τα ανατολικά, κατακτά τις πόλεις 4 και 5, και αφήνει τον παίκτη να συλλέξει 2 χρυσά νομίσματα (από την πόλη 3) στο τέλος του έτους. Δεν υπάρχουν πλέον δύο γειτονικές πόλεις υπό τον έλεγχο του παίκτη, μεταξύ των οποίων θα μπορούσε να κτίσει προπύργια. Έτσι το παιχνίδι ολοκληρώνεται εδώ, με συνολικό κέρδος 10 χρυσών νομισμάτων για τον παίκτη.

**Δεδομένα Εισόδου:** Αρχικά, το πρόγραμμα θα διαβάζει από το standard input έναν θετικό ακέραιο αριθμό που αντιστοιχεί στο πλήθος  $N$  των πόλεων. Σε κάθε μία από τις επόμενες  $N$  γραμμές, θα υπάρχει ένας φυσικός αριθμός  $p_i$  που δηλώνει το πλήθος των χρυσών νομισμάτων που πληρώνει η πόλη  $i$  ως ετήσιο φόρο.



**Σχήμα 1.** Παράδειγμα για την Άσκηση 2. Αν ο αντίπαλος επιτεθεί από τα δυτικά (αριστερά), οι πόλεις 5 και 6 θα μείνουν υπό τον έλεγχο του παίκτη, και το κέρδος του για το τρέχον έτος θα είναι  $p_5 + p_6$ . Αν ο αντίπαλος επιτεθεί από τα ανατολικά (δεξιά), οι πόλεις 1, 2, 3 και 4 θα μείνουν υπό τον έλεγχο του παίκτη, και το κέρδος του για το τρέχον έτος θα είναι  $p_1 + p_2 + p_3 + p_4$ .

**Δεδομένα Εξόδου:** Το πρόγραμμα πρέπει να τυπώνει στο standard output (στην πρώτη γραμμή) το μέγιστο πλήθος χρυσών νομισμάτων που μπορεί να συλλέξει ο παίκτης, δεδομένου ότι ο αντίπαλος επιλέγει με βέλτιστο τρόπο την ετήσια επίθεσή του.

**Περιορισμοί:**

$$2 \leq N \leq 500$$

$$1 \leq p_i \leq 30.000$$

Όριο χρόνου εκτέλεσης: 1 sec.

Όριο μνήμης: 128 MB.

*Bonus:* κάποια αρχεία με

$$2000 \leq N \leq 2.500$$

**Παράδειγμα Εισόδου:**

5

8

6

2

4

2

**Παράδειγμα Εξόδου:**

10