

D. Διακοσμήσεις κήπων

Όνομα Προβλήματος	Διακοσμήσεις Κήπων
Χρονικό Όριο	7 δευτερόλεπτα
Όριο Μνήμης	1 gigabyte

Κάθε μέρα πηγαίνοντας στο σχολείο και επιστρέφοντας σπίτι, η Detje περπατά κατά μήκος ενός δρόμου με N σπίτια αριθμημένα από 0 έως $N - 1$. Τώρα, στο σπίτι i μένει ο κάτοικος i . Για αλλαγή σκηνικού, οι κάτοικοι αποφάσισαν να αλλάξουν σπίτια μεταξύ τους. Ο κάτοικος που θα μετακομίσει στο σπίτι i , είναι ο a_i (που τώρα ζει στο σπίτι a_i).

Κάθε σπίτι έχει ένα άγαλμα πτηνού στον κήπο. Τα αγάλματα έχουν δυο δυνατές καταστάσεις, είτε με ανοικτά φτερά (σαν να πετούν) είτε με κλειστά (σα να στέκονται στο έδαφος). Οι κάτοικοι δίνουν μεγάλη προσοχή στο πώς πρέπει να εμφανίζονται τα πτηνά τους. Επί του παρόντος, το πτηνό μπροστά από το σπίτι i βρίσκεται στην επιθυμητή κατάσταση του κατοίκου i . Οι κάτοικοι αρνούνται να μετακομίσουν σε ένα σπίτι, εκτός αν το άγαλμά του έχει οριστεί στην επιθυμητή τους κατάσταση. Η Detje θέλει να τους βοηθήσει να τακτοποιήσουν τα αγάλματα των πτηνών ώστε να μπορούν να μετακομίσουν.

Για να το πετύχει, κάνει τα ακόλουθα: όποτε περπατά κατά μήκος του δρόμου (είτε πηγαίνοντας στο σχολείο είτε επιστρέφοντας σπίτι), παρατηρεί τα πτηνά που προσπερνά ένα προς ένα και ενδεχομένως αλλάζει την κατάσταση κάποιων (ανοίγοντας ή κλείνοντας τα φτερά τους). Λόγω του ότι οι μέρες της στο σπίτι και στο σχολείο είναι πολύ φορτωμένες, **δεν θυμάται την κατάσταση των πτηνών που είδε στις προηγούμενες διαδρομές της**. Ευτυχώς, έχει καταγράψει τη σειρά a_0, a_1, \dots, a_{N-1} , οπότε ξέρει ποιος κάτοικος μετακομίζει πού.

Βοηθήστε τη Detje να σχεδιάσει μια στρατηγική που θα την καθοδηγεί ποια πτηνά πρέπει να αλλάξει ώστε να προσαρμόσει τα αγάλματα στις απαιτήσεις των κατοίκων. Επιτρέπεται να διασχίσει τον δρόμο το πολύ 60 φορές, αλλά για να πετύχει υψηλότερη βαθμολογία, θα πρέπει να διασχίσει τον δρόμο λιγότερες φορές.

Υλοποίηση

Αυτό είναι ένα πρόβλημα πολλαπλών εκτελέσεων, δηλαδή το πρόγραμμά σας θα εκτελεστεί πολλές φορές.

Σε κάθε εκτέλεση, θα πρέπει πρώτα να διαβάσετε μια γραμμή με δυο ακεραίους w και N , τον αριθμό της διαδρομής και τον αριθμό των σπιτιών. Στην πρώτη εκτέλεση του προγράμματος σας είναι $w = 0$, στη δεύτερη είναι $w = 1$ κλπ (περισσότερες λεπτομέρειες στη συνέχεια).

Στη δεύτερη γραμμή των δεδομένων εισόδου ακολουθούν N ακέραιοι a_0, a_1, \dots, a_{N-1} , σημαίνοντας ότι ο κάτοικος που μετακομίζει στο σπίτι i μένει τώρα στο σπίτι a_i . Τα a_i σχηματίζουν μια σειρά (*permutation*): όπου, κάθε αριθμός από το 0 έως και το $N - 1$ εμφανίζεται ακριβώς μια φορά στη σειρά a_i . Προσέξτε ότι κάποιοι κάτοικοι μπορούν να επιλέξουν να μην μετακομίσουν, δηλαδή επιτρέπεται $a_i = i$.

Οι κάτοικοι αλλάζουν σπίτια μόνο μια φορά. Δηλαδή για μία συγκεκριμένη δοκιμαστική περίπτωση (test case), η τιμή του N και η σειρά των a_i θα είναι η ίδια για όλες τις εκτελέσεις του προγράμματός σας

Πρώτη Εκτέλεση.

Για την πρώτη εκτέλεση του προγράμματός σας το $w = 0$. Στην εκτέλεση αυτή πρέπει απλά να τυπώσετε έναν μοναδικό ακέραιο W ($0 \leq W \leq 60$), τον αριθμό των φορών που θέλετε η Detje να διασχίσει το δρόμο. Το πρόγραμμα σας πρέπει αμέσως μετά να τερματίσει την εκτέλεσή του. Ακολούθως, το πρόγραμμά σας θα εκτελεστεί ξανά W επιπλέον φορές.

Υπόλοιπες Εκτελέσεις.

Στην επόμενη εκτέλεση του προγράμματος σας θα είναι $w = 1$, στην μεθεπόμενη $w = 2$, κλπ έως την τελευταία εκτέλεση όπου θα είναι $w = W$.

Αφού διαβάσετε τα w , N και τα a_0, a_1, \dots, a_{N-1} , η Detje ξεκινά τη διαδρομή κατά μήκος του δρόμου.

- Αν το w είναι περιττό, η Detje περπατά από το σπίτι της προς το σχολείο και διασχίζει τα σπίτια με τη σειρά $0, 1, \dots, N - 1$.

Το πρόγραμμα σας πρέπει τώρα να διαβάσει μια γραμμή με την τιμή b_0 , ή 0 (κλειστά φτερά) ή 1 (ανοικτά φτερά), την τρέχουσα κατάσταση του αγάλματος μπροστά από το σπίτι 0. Αφού διαβάσετε την τιμή b_0 , πρέπει να τυπώσετε μια γραμμή με την τιμή 0 ή 1, την τιμή δηλαδή που θέλετε να γίνει το b_0 .

Μετά πρέπει να διαβάσετε μια γραμμή με την τιμή του b_1 , την κατάσταση του αγάλματος μπροστά από το σπίτι 1 και τυπώστε τη νέα τιμή του b_1 . Αυτό θα συνεχίσει για κάθε όλα τα N σπίτια. Αφού προσπεράσετε το τελευταίο σπίτι (δηλαδή διαβάσετε και γράψετε το b_{N-1}), **το πρόγραμμα σας πρέπει να τερματιστεί.**

Προσέξτε ότι το πρόγραμμα σας μπορεί να διαβάσει την επόμενη τιμή b_{i+1} αφού γράψει τη νέα τιμή b_i .

- Αν ο w είναι άρτιος, η Detje περπατά από το σχολείο προς το σπίτι της και θα προσπεράσει τα σπίτια με την αντίστροφη σειρά $N - 1, N - 2, \dots, 0$.

Τότε ξεκινάτε το διάβασμα και γράψιμο από το b_{N-1} , μετά το b_{N-2} , κλπ έως το b_0 .

Όταν το $w = 1$, οι τιμές b_0, b_1, \dots, b_{N-1} είναι οι αρχικές καταστάσεις των αγαλμάτων. Όταν $w > 1$, οι τιμές εισόδου b_0, b_1, \dots, b_{N-1} είναι οι τιμές που είχε γράψει το πρόγραμμα σας στην προηγούμενη εκτέλεση του.

Στο τέλος, μετά από την τελευταία εκτέλεση του προγράμματος σας, η τιμή b_i πρέπει να είναι ίση με την αρχική τιμή του b_{a_i} για κάθε i , αλλιώς θα λάβετε την ετυμηγορία Wrong Answer.

Λεπτομέρειες.

Αν το άθροισμα των χρόνων των $W + 1$ ξεχωριστών εκτελέσεων του προγράμματος σας υπερβαίνει το όριο χρόνου, η υποβολή σας θα αξιολογηθεί ως Time Limit Exceeded.

Βεβαιωθείτε ότι κάνετε flush την τυπική έξοδο αφού τυπώσετε κάθε γραμμή, αλλιώς το πρόγραμμα σας θα λάβει ετυμηγορία Time Limit Exceeded. Στην Python, αυτό γίνεται αυτόματα αν χρησιμοποιείται το `input()` για να διαβάζετε γραμμές. Στη C++, το `cout << endl;` κάνει flush, επίπρόσθετα της εκτύπωσης μιας κενής γραμμής. Αν χρησιμοποιείτε `printf`, χρησιμοποιήστε την `fflush(stdout)`.

Περιορισμοί και Βαθμολόγηση

- $2 \leq N \leq 500$.
- Μπορείτε να χρησιμοποιήσετε το πολύ $W \leq 60$ διαδρομές.

Η λύση σας θα αξιολογηθεί με βάση ένα σύνολο ομάδων δοκιμών (test groups), καθεμία από τις οποίες ισοδυναμεί με έναν αριθμό πόντων. Κάθε ομάδα δοκιμών περιέχει ένα σύνολο δοκιμαστικών περιπτώσεων. Για να λάβετε τους πόντους για μια ομάδα δοκιμών, πρέπει να λύσετε επιτυχώς όλες τις δοκιμαστικές περιπτώσεις σε αυτήν την ομάδα.

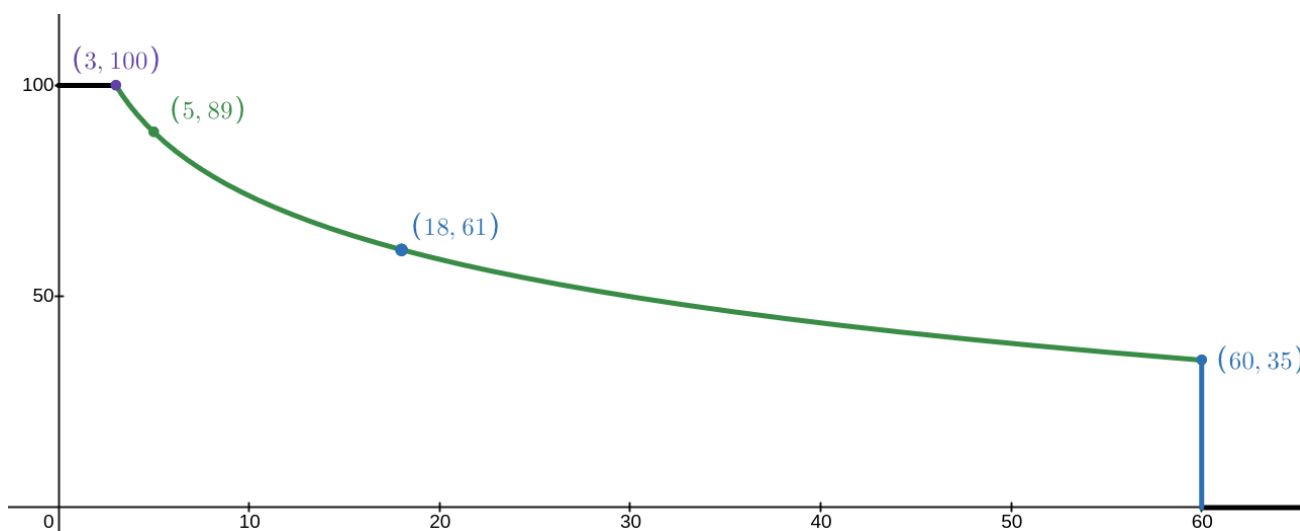
Ομάδα	Μέγιστο σκορ	Περιορισμοί
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Χωρίς άλλους περιορισμούς

Για κάθε ομάδα δοκιμών που το πρόγραμμα σας λύνει σωστά, θα λάβετε μια βαθμολόγηση που βασίζεται στον παρακάτω τύπο:

$$\text{score} = S_g \cdot \left(1 - \frac{1}{2} \log_{10}(\max(W_g, 3)/3)\right),$$

όπου S_g είναι η μέγιστη βαθμολόγηση της ομάδας δοκιμών και W_g είναι η μέγιστη τιμή W που χρησιμοποιήσατε για κάθε δοκιμαστικό αρχείο (test case) της ομάδας δοκιμών. Η βαθμολογία σας στην ομάδα δοκιμών θα στρογγυλοποιηθεί στον πλησιέστερο ακέραιο.

Το διάγραμμα που ακολουθεί δείχνει τη βαθμολογία σαν συνάρτηση του W που το πρόγραμμα σας θα λάβει αν λύσει όλες τις ομάδες δοκιμών με την ίδια τιμή W . Ποιο συγκεκριμένα, για να επιτύχετε βαθμολογία 100 στο πρόβλημα αυτό, πρέπει να λύσετε όλα τα αρχεία δοκιμών με $W \leq 3$.



Εργαλείο Ελέγχου

Ως βοήθεια στη δοκιμή του κώδικα σας, παρέχεται ένα απλό εργαλείο ελέγχου που μπορείτε να κατεβάσετε. Δείτε τα "attachments" στο κάτω μέρος της ιστοσελίδας του προβλήματος στο Kattis. Η χρήση του εργαλείου είναι προαιρετική. Παρατηρείστε ότι ο επίσημος βαθμολογητής στο Kattis είναι διαφορετικός από αυτό το εργαλείο ελέγχου.

Για να χρησιμοποιήσετε το εργαλείο, δημιουργήστε ένα αρχείο εισόδου, όπως "sample1.in", το οποίο πρέπει να ξεκινά με έναν αριθμό N που ακολουθείται από μια γραμμή με N αριθμούς που καθορίζουν τη σειρά (permutation) και άλλη μια γραμμή με N bits (0 ή 1) που καθορίζουν την αρχική κατάσταση των αγαλμάτων. Για παράδειγμα:

```
6
1 2 0 4 3 5
1 1 0 0 1 0
```

Για προγράμματα Python, τρέξετε το `solution.py` (κανονικά τρέχει ως `python3 solution.py`):

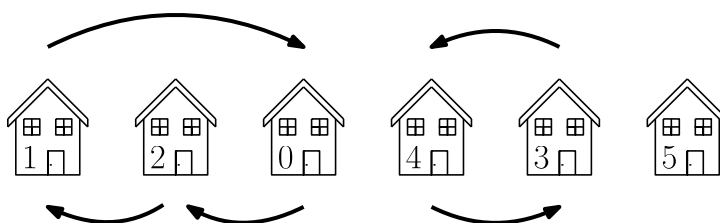
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Για προγράμματα C++, πρώτα μεταγλωττίστε το (π.χ. με `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) και μετά τρέξετε την εντολή:

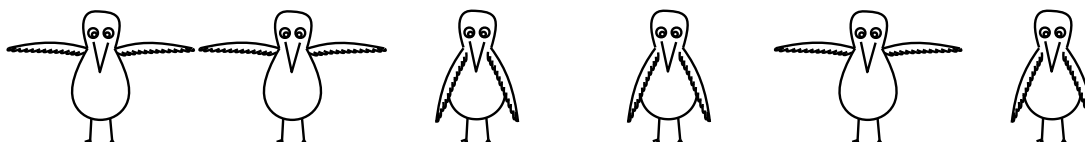
```
python3 testing_tool.py ./solution.out < sample1.in
```

Παράδειγμα

Στο παράδειγμα, μας δίνεται η επόμενη σειρά (permutation) των κατοίκων στα σπίτια:

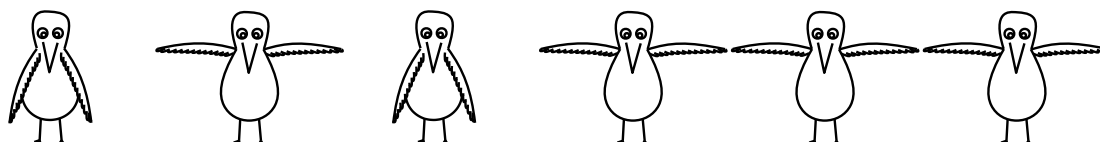


Την πρώτη φορά που το παράδειγμα τρέχει (με $w = 0$), τυπώνει $W = 2$, που σημαίνει ότι η Detje θα περπατήσει κατά μήκος του δρόμου δύο φορές (και το πρόγραμμα θα εκτελεστεί άλλες δύο φορές). Πριν την πρώτη διαδρομή, τα πτηνά στους κήπους εμφανίζονται όπως παρακάτω:



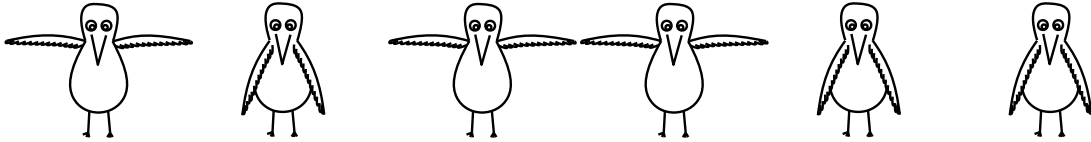
Ακολουθως το πρόγραμμα τρέχει με $w = 1$: δείχνοντας ότι βρισκόμαστε στην πρώτη διαδρομή της Detje. Προσπερνά τα αγάλματα ένα ένα, ξεκινώντας από τα αριστερά και πιθανά, αλλάζει τις καταστάσεις των αγαλμάτων. Το πρόγραμμα πρέπει να τυπώσει την κατάσταση του αγάλματος i πριν δει την κατάσταση του $(i + 1)$ αγάλματος.

Αφού η Detje φτάσει στο σχολείο, η κατάσταση των αγαλμάτων εμφανίζεται ως:



Στην τελική εκτέλεση του προγράμματος (με $w = 2$), η Detje περπατά προς το σπίτι της. Θυμηθείτε ότι σε αυτή την περίπτωση, θα προσπεράσει τα αγάλματα από δεξιά προς τα αριστερά και θα τα επεξεργαστεί με αυτή τη σειρά! Αυτό σημαίνει ότι πρέπει να προσδιορίσει την κατάσταση του αγάλματος i πριν δει το $(i - 1)$.

Τελικά τερματίζει τη διαδρομή της και τα αγάλματα εμφανίζονται ως:



Πράγματι, αυτή είναι η σωστή κατάληξη. Για παράδειγμα, το άγαλμα στην θέση 3 (δηλαδή το τέταρτο από τα αριστερά) έχει ανοικτά φτερά (τώρα $b_3 = 1$), το οποίο είναι σωστό καθώς ο κάτοικος 4 θα μετακομίσει εκεί ($a_3 = 4$) και αρχικά είχε και αυτός άγαλμα με ανοικτά φτερά (αρχικά $b_4 = 1$).

έξοδος βαθμολογητή	η έξοδος σας
0 6	
1 2 0 4 3 5	
	2

έξοδος βαθμολογητή	η έξοδος σας
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

έξοδος βαθμολογητή	η έξοδος σας
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1