



Problema 2 - mirror

100 puncte

Numim „oglinda” numărului natural nenul a , numărul b , obținut prin modificarea fiecărei cifre din reprezentarea sa binară, de exemplu pentru $a=22_{(10)}=10110_{(2)}$ se obține $01001_{(2)}=9_{(10)}=b$.

Cerințe:

Cunoscându-se numerele naturale N, K și cele N numere naturale nenule, scrieți un program care:

1) Transformă în baza doi termenii șirului dat obținându-se un nou șir format din alipirea cifrelor binare. Din acest șir se vor determina și afișa, separate prin câte un spațiu, toate reprezentările în baza 10 corespunzătoare secvențelor alăturate de exact K cifre binare, parcurse de la stânga la dreapta. Dacă ultima secvență nu are exact K cifre binare, atunci aceasta nu se va mai lua în considerare.

2) Să aplice K transformări asupra șirului inițial, înlocuind la fiecare pas orice termen cu „oglinda” sa. Asupra termenilor care devin zero nu se vor mai efectua alte operații. După efectuarea celor K transformări, să se determine cea mai lungă secvență de numere care au cifra 1 pe aceeași poziție în reprezentarea lor în baza doi. Dacă sunt mai multe astfel de secvențe având lungimea maximă, se va afișa cea mai din stânga.

Date de intrare

Fișierul de intrare `mirror.in` conține pe primul rând numărul C , reprezentând cerința. Pe al doilea rând se află scrise numerele naturale N și K . Pe rândul al treilea sunt cele N numere ale șirului separate prin câte un spațiu.

Date de ieșire

Dacă $C=1$, atunci în fișierul de ieșire `mirror.out` se vor scrie separate prin câte un spațiu, toate numerele cerute în enunț.

Dacă $C=2$, atunci în fișierul de ieșire `mirror.out` se va scrie pe prima linie lungimea maximă a secvenței determinate, iar pe următoarea linie separate prin spațiu, poziția primului și ultimului termen din secvență (prima poziție este 1).

Restricții și precizări

- $1 \leq N \leq 100000$
- $0 \leq K \leq 30$
- Elementele șirului sunt mai mici decât 20000000001 ;
- Pentru 30% din teste cerința va fi $C=1$.

Exemplu:

<code>mirror.in</code>	<code>mirror.out</code>	Explicație																																								
1 4 2 7 8 2 11	3 3 0 1 1 1	$7_{(10)}=111_{(2)}$; $8_{(10)}=1000_{(2)}$; $2_{(10)}=10_{(2)}$; $11_{(10)}=1011_{(2)}$; Șirul format este: 1111000101011 și grupate câte 2 avem numerele: $11_{(2)}=3_{(10)}$; $11_{(2)}=3_{(10)}$; $00_{(2)}=0_{(10)}$; $01_{(2)}=1_{(10)}$; $01_{(2)}=1_{(10)}$; $01_{(2)}=1_{(10)}$;																																								
2 5 1 37 72 101 50 116	3 1 3	După o transformare numerele în baza 2 sunt: <table border="1" style="margin-left: 20px;"> <tr><td></td><td>0</td><td>1</td><td>1</td><td>0</td><td><u>1</u></td><td>0</td><td><-37</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td><u>1</u></td><td>1</td><td><-72</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td><u>1</u></td><td>0</td><td><-101</td></tr> <tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td><-50</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td><-116</td></tr> </table> Cea mai lungă secvență este de lungime 3, fiind formată din numerele 37, 72, 101 care începe pe poziția 1 și se termină pe poziția 3. Mai există încă o astfel de secvență (101, 50, 116) dar se alege cea mai		0	1	1	0	<u>1</u>	0	<-37	0	1	1	0	1	<u>1</u>	1	<-72	0	0	1	1	0	<u>1</u>	0	<-101		0	0	1	1	0	1	<-50	0	0	0	1	0	0	0	<-116
	0	1	1	0	<u>1</u>	0	<-37																																			
0	1	1	0	1	<u>1</u>	1	<-72																																			
0	0	1	1	0	<u>1</u>	0	<-101																																			
	0	0	1	1	0	1	<-50																																			
0	0	0	1	0	0	0	<-116																																			

Timp maxim de execuție/test: 1 secundă pentru Windows și 0,25 secunde pentru Linux
Memorie totală: 32 MB Dimensiunea maximă a sursei: 10 KB



		din stânga.
--	--	-------------

Timp maxim de execuție/test: 1 secundă pentru Windows și 0,25 secunde pentru Linux
Memorie totală: 32 MB Dimensiunea maximă a sursei: 10 KB