„ASPIRAȚII INFORMATICE” – 2023
CLASELE IX – X

1. NUMĂR ECHILIBRAT

Numim număr echilibrat un număr natural pentru care suma cifrelor de pe poziții pare este egală cu suma cifrelor de pe poziții impare. De exemplu numărul 13552 este echilibrat, pentru că 1+5+2 = 8 = 3+5.

Sarcină. Dat fiind un număr natural N, elaborați un program care determină cel mai mic număr echilibrat, strict mai mare decât N.

Date de intrare. Fişierul *NUMAR.IN* va conţine pe o singură linie numărul natural *N*.

Date de ieşire. Fişierul *NUMAR.OUT* va conţine pe prima linie un singur număr natural, care reprezintă cel mai mic număr echilibrat strict mai mare ca *N*.

Restricţii:

- $10 \le N \le 1000000000$;
- Timpul de execuţie nu va depăşi 2 sec.

Exemple:

NUMAR.IN	*NUMAR.OUT*	*Explicații*
99	110	1+0=1
123133	123134	1+3+3=2+1+4

Fişierul sursă va avea denumirea NUMARxxx.PAS sau NUMARxxx.CPP, xxx fiind codul elevului. De exemplu, NUMARU01.PAS.

2. ORDINE

Gigel a primit de ziua lui un joc cu bile. Jocul conţine ***n*** bile numerotate cu numerele naturale distincte de la **1** la ***n***. Jucându-se, Gigel a amestecat bilele astfel încât acum ele nu mai sunt în ordine, ci formează un şir de n numere din intervalul **1**..***n*** plasat pe masă. Ca să le pună înapoi în cutia jocului, Gigel ia de pe masă bilele una câte una, şi le pune în cutie formând un şir final. Gigel ia bilele pe rând dar nu le pune una după alta, în aceeaşi ordine, ci are o regulă pe care o respectă cu strictețe.

Astfel, Gigel încearcă să plaseze fiecare bilă pe care a luat-o de pe masă exact la mijlocul şirului de bile deja format. Dacă acest lucru nu este posibil (şirul are lungime impară), atunci el plasează bila la sfârşitul şirului de bile deja format. După ce toate bilele au fost puse în cutie, Gigel îşi dă seama că nu a notat ordinea în care a luat bilele de pe masă şi, în mod firesc, îşi pune problema dacă nu cumva poate deduce acest lucru din şirul de bile pe care tocmai l-a format.

Sarcină. Cunoscându-se configurația finală a şirului de bile, elaborați un program care determină şirul inițial de bile (modul în care acestea erau aranjate pe masă).

Date de intrare. Fişierul *ORDIN.IN* va conţine:

- pe prima linie un număr natural *n*, reprezentând numărul de bile;
- pe linia a doua şirul din *n* numere naturale cu valori de la 1..*n*, separate prin spațiu, reprezentând valorile de pe bile în şirul final din cutie (de la început până la sfârşit).

Date de ieşire. Fişierul *ORDIN.OUT* va conţine:

- linie formată din *n* numere naturale (x_i), cu valori cuprinse între 1 şi *n*, separate prin câte un spațiu, care reprezintă şirul de bile plasate inițial pe masă.

Restricţii: $1 \le n \le 100$; $1 \le x_i \le n$, $1 \le i \le n$. Timpul de execuţie nu va depăşi 2 sec

Exemple:

ORDIN.IN	ORDIN.OUT
7 1 7 2 5 3 4 6	1 3 7 4 2 6 5
10 4 6 8 1 10 5 7 9 3 2	4 5 6 7 8 9 1 3 10 2

Fişierul sursă va avea denumirea ORDINxxx.PAS sau ORDINxxx.CPP, xxx fiind codul elevului. De exemplu, ORDINU01.PAS.

3. CURSURI

Într-o tabără de vară se programează susținerea unor cursuri în K săli de clasă. Sunt N profesori care și-au exprimat dorința de a participa, fiecare dintre ei specificând intervalul de timp $[a_i, b_i]$ în care își poate susține cursul. Programarea pe săli a profesorilor trebuie să țină cont de faptul că într-o clasă, la un moment dat, nu poate preda decât un singur profesor.

Sarcină. Cunoscând faptul că organizatorii doresc susținerea a cât mai multor cursuri, elaborați un program, care determină:

1) Numărul maxim de cursuri care pot fi programate în cele K săli de clasă, ținând cont de restricția indicată.

2) În dorința de a programa toate cursurile, în cele K săli, organizatorii decid să modifice durata cursurilor, păstrând însă neschimbată ora de început a lor. Astfel, ei hotărăsc ca toate cursurile să dureze un interval egal de timp, care însă nu va depăși durata celui mai lung curs propus inițial de unul dintre cei N profesori. Determinați care poate fi durata maximă pe care o pot avea cursurile în aceste condiții.

Date de intrare. Fișierul *CURS.IN* va conține:

- pe prima linie se găsește o pereche de numere naturale N și K, separate printr-un spațiu, reprezentând numărul profesorilor și numărul de săli de clasă.
- pe următoarele N linii se găsesc perechi de numere naturale a_i și b_i, care reprezintă intervalele de timp în care cei N profesori își susțin cursurile. Numerele în cadrul unei linii sunt separate printr-un spațiu.

Date de ieșire. Fișierul *CURS.OUT* va conține două linii:

- pe prima linie un număr natural, reprezentând numărul maxim de cursuri care pot fi programate în cele K săli de clasă, ținând cont de restricția indicată.
- pe a doua linie un număr natural, reprezentând durata maximă pe care o pot avea cele N cursuri, astfel încât toate să poată fi susținute în cele K săli disponibile.

Restricții:

- $1 \le N \le 1000$; $1 \le K \le 1\,000$; $1 \le a_i < b_i \le 10000$, unde $1 \le i \le N$.
- Timpul de execuție nu va depăși 3 sec.

Exemplu:

CURS.IN	*CURS.OUT*	*Explicații*
5 2 2 8 4 9 1 5 6 18 9 10	4 3	O variantă de programare optimă este următoarea: - în prima sală se vor susține cursurile programate între [1,5] și [6,18]; - în a doua clasă se susține cursul programat între [2,8] și [9,10]. Durata maximă pe care o pot avea toate cursurile este 3. Cursul al cincilea se va mări și se va desfășura între [9,12], celelalte se vor micșora. Cursurile vor fi distribuite în cele două săli astfel: Sala 1: al treilea, al doilea și al cincilea profesor programați între [1,4] respectiv [4,7],[9,12]; Sala 2: primul și al patrulea profesor programați între [2,5] respectiv [6,9].
4 2 5 12 9 18 1 3 1 7	4 4	O variantă de programare optimă este următoarea: - în prima sală se vor susține cursurile programate între [1,3] și [5,12]; - în a doua clasă se susține cursul programat între [1,7] și [9,18]. Durata maximă pe care o pot avea toate cursurile este 4. Cursul al treilea se va mări și se va desfășura între [1,5], celelalte se vor micșora. Cursurile vor fi distribuite în cele două săli astfel: Sala 1: al treilea și primul profesor programați între [1,5] respectiv [5,9]; Sala 2: al patrulea și al doilea profesor programați între [1,5] respectiv [9,13];

Fișierul sursă va avea denumirea *CURSxxx.PAS* sau *CURSxxx.CPP*, *xxx* fiind codul elevului. De exemplu, *CURSU01.PAS*.

4. SPIONAJ

Pentru a realiza dialogul între spioni a fost creat un mecanism de codificare. Mecanismul presupune: la prima etapa se transformă fiecare simbol din mesaj într-un cod binar obținut din codul zecimal ASCII-ext al acestuia, apoi la etapa a doua codurile binare ale simbolurilor se concatenează și se formează un șir de numere în care pe prima poziție se află prima cifră din șirul concatenat (0 sau 1) după care urmează valori, ce reprezintă numărul de cifre binare consecutive.

De exemplu, pentru cuvântul OK obținem: O – simbolul cu codul 79 respectiv în codul binar 01001111, iar K – 75 respectiv în codul binar – 01001011. La etapa a doua mesajul OK se transformă în șirul 01124112112, ca rezultat a numărării cifrelor de 0 și 1 din șirul concatenat 0100111101001011 cu începutul 0 deoarece este prima cifră din acest șir.

Sarcină. Cunoscând șirul final codificat, elaborați un program, care determină mesajul ce a fost codificat.

Date de intrare. Fișierul *SPION.IN* va conține pe prima linie un șir de cifre, reprezentând mesajul codificat.

Date de ieșire. Fișierul *SPION.OUT* va conține pe prima linie mesajul decodificat.

Restricții:

- În mesajul care este codificat sunt folosite doar literele majuscule ale alfabetului latin englez și spațiul liber.
- Lungimea șirului de intrare nu va depăși 1000 de caractere.
- Timpul de execuție nu va depăși 2 sec.

Exemplu:

SPION.IN	*SPION.OUT*	*Explicații*
01124112112	OK	0 este începutul șirului 1 – 0 1 – 1 2 – 00 4 – 1111 1 – 0 1 – 1 2 – 00 1 – 1 1 – 0 2 – 11

Fişierul sursă va avea denumirea *SPIONxxx.PAS* sau *SPIONxxx.CPP*, *xxx* fiind codul elevului. De exemplu, *SPIONU01.PAS*.