

27 mai 2017

Sursa: `codp1.c`, `codp1.cpp`, `codp1.pas`**Problema 1 – anagrama****100 puncte**

Ionică trebuie să rezolve o problemă de informatică ce prelucrează texte. Mai precis are la dispoziție un cuvânt  $C$  format din litere mici și un text  $T$  format din litere mici și spații. Cuvintele în  $T$  sunt separate prin unul sau mai multe spații. Problema pe care trebuie să o rezolve Ionică cere să determine numărul de vocale comune lui  $C$  și primului cuvânt din  $T$  și apoi numărul de cuvinte anagramă ale lui  $C$  în  $T$ . Un cuvânt  $X$  este anagramă pentru un alt cuvânt  $Y$ , dacă  $X$  este format din exact literele lui  $Y$ , eventual în altă ordine.

**Cerință**Cunoscând  $C$  și  $T$  se cere:

1. numărul de vocale comune din  $C$  și primului cuvânt din  $T$
2. numărul de cuvinte anagramă ale lui  $C$  în  $T$

**Date de intrare**

Fișierul de intrare *anagrama.in* conține pe prima linie un număr natural  $p$ . Pentru toate testele de intrare, numărul  $p$  poate avea doar valoarea **1**, **2**.

Pe linia a doua se află  $C$ , iar pe linia a treia se află  $T$ . Fiecare linie se termină cu enter.

**Date de ieșire**

Dacă valoarea lui  $p$  este **1**, se va rezolva numai punctul 1) din cerință.

În acest caz, în fișierul de ieșire *anagrama.out* se va scrie un singur număr natural reprezentând numărul de vocale comune din  $C$  și primului cuvânt din  $T$

Dacă valoarea lui  $p$  este **2**, se va rezolva numai punctul 2) din cerință.

În acest caz, în fișierul de ieșire *anagrama.out* se va scrie un singur număr natural reprezentând numărul de cuvinte anagramă ale lui  $C$  în  $T$ .

**Restricții și precizări**

- $1 \leq \text{lungimea lui } C \leq 100000$
- $1 \leq \text{lungimea lui } T \leq 100000$
- vocalele sunt a, e, i, o, u
- pentru rezolvarea corectă a fiecărei cerințe se acordă 50% din punctaj

**Exemple**

<code>anagrama.in</code>	<code>anagrama.out</code>	Explicație
1 maria marinela se joaca	2	$p = 1$ $C$ este <b>maria</b> Primul cuvânt din $T$ este <b>marinela</b> Sunt 2 vocale comune. a apare de două ori, dar este numărat o singură dată. <b>Atenție! Pentru acest test se rezolvă doar cerința 1).</b>
<code>anagrama.in</code>	<code>anagrama.out</code>	Explicație
2 mara vara aram dica rama aarm rama	4	$p = 2$ Cuvintele anagramă cu $C$ în $T$ sunt aram, rama, aarm și rama. Numărul lor este 4. <b>Atenție! Pentru acest test se rezolvă doar cerința 2).</b>

**Timp maxim de execuție: 0.2 secunde/test.**

**Memorie totală disponibilă 2 MB, din care 1 MB pentru stivă**

**Dimensiunea maximă a sursei: 5 KB.**

27 mai 2017

Sursa: `codp2.c`, `codp2.cpp`, `codp2.pas`**Problema 2 –cuburi****100 puncte**

Ionică are un sac plin cu cuburi, numerotate cu  $1, 2, \dots, N$  și  $M$  cutii numerotate cu  $1, 2, \dots, M$  în care încap  $x_1, x_2, \dots, x_M$  cuburi. Din păcate cutiile nu sunt suficiente pentru a introduce cuburile din sac și mai rămân cuburi neintroduse. În fiecare zi după ce se joacă cu cuburile le introduce în cutii la capacitate maximă și ce rămâne le pune în sac. După multe zile de joacă și așezare a cuburilor în cutii, Ionică se întreabă în câte moduri diferite poate așeza cuburile în cutii știind că ordinea cuburilor dintr-o cutie nu are importanță.

**Cerință**

Cunoscând numerele  $M, N$  și capacitățile cutiilor, se cere:

1. care este numărul maxim de cuburi pe care le poate pune în  $[M/2]$  cutii.

2. în câte moduri diferite se pot așeza cuburile în cutii știind că ordinea cuburilor în cutii nu are importanță.

**Date de intrare**

Fișierul de intrare `cuburi.in` conține pe prima linie un număr natural  $p$ . Pentru toate testele de intrare, numărul  $p$  poate avea doar valoarea **1** sau **2**.

Pe linia a doua se află  $N, M$ , separate printr-un spațiu, iar pe linia următoare numerele  $x_1, x_2, \dots, x_M$  separate prin câte un spațiu.

**Date de ieșire**

Dacă valoarea lui  $p$  este **1**, se va rezolva numai punctul 1) din cerință.

În acest caz, în fișierul de ieșire `cuburi.out` se va scrie un număr ce reprezintă numărul maxim de cuburi pe care le poate pune în  $[M/2]$  cutii.

Dacă valoarea lui  $p$  este **2**, se va rezolva numai punctul 2) din cerință.

În acest caz, în fișierul de ieșire `cuburi.out` se va conține numărul de moduri diferite în care se pot așeza cuburile în cutii. Numarul va fi afișat modulo 60013.

**Restricții și precizări**

- $1 \leq N \leq 1000$ .
- $1 \leq M \leq 1000$ .
- $0 \leq x_1 + x_2 + \dots + x_M \leq N$
- $x$  modulo  $y$  reprezintă restul împărțirii lui  $x$  la  $y$ .
- pentru rezolvarea corectă a fiecărei cerințe se acordă 50% din punctaj

**Exemple**

<code>cuburi.in</code>	<code>cuburi.out</code>	Explicație
1 4 2 1 2	2	$p = 1$ Cutia cea mai mare contine 2 cuburi. <b>Atenție! Pentru acest test se rezolvă doar cerința 1).</b>
<code>cuburi.in</code>	<code>cuburi.out</code>	Explicație
2 4 2 1 2	12	$p = 2$ La cerinta 2 numarul de modalitati de asezare a cuburilor este 12. <b>Atenție! Pentru acest test se rezolvă doar cerința 2).</b>

**Timp maxim de execuție: 0.2 secunde/test.**

**Memorie totală disponibilă 8 MB, din care 2 MB pentru stivă**

**Dimensiunea maximă a sursei: 5 KB.**