

Problema 2018.1.6 – RNG Runlength

Împreună cu echipa de la firmă ați inventat un nou algoritm de generare de numere pseudo-aleatoare. Pentru a valida că generatorul poate fi folosit în algoritmi criptografici (*cryptographically secure*) trebuie să implementați și să rulați o baterie de teste. Unul din aceste teste verifică numărul de apariții pentru secvențe de 1 de diferite lungimi. Pentru ca secvența de biți generată să fie aleatoare, trebuie ca numărul de apariții pentru fiecare lungime de secvență de 1 să aibă o anumită distribuție statistică. Mai precis, trebuie ca numărul de secvențe de un bit 1 să fie mai mare sau egal decât numărul de secvențe de doi biți 1 care trebuie să fie mai mare sau egal decât numărul de secvențe de trei biți 1, șamd.

Cerință

Dându-se un număr n reprezentând numărul de biți generat de RNG și secvența de n biți, să se calculeze numărul de apariții pentru fiecare lungime de secvență de biți 1 și să se decidă dacă generatorul este valid sau nu.

Date de intrare

Pe prima linie se află n , numărul de biți generați. Pe a doua linie se află o secvență continuă de n biți (valori de 0 sau 1), ne-separați prin spații.

Date de ieșire

Programul va afișa în consolă (pe stream-ul *stdout*) pe prima linie o secvență de numere întregi pozitive, separate prin spațiu, reprezentând numărul de apariții pentru fiecare lungime de secvență de biți 1, începând cu numărul de apariții pentru o secvență de un singur bit 1 (delimitat de biți 0) și terminând cu ultimul număr de apariții nenul. Pe a doua linie se va afișa valoarea 1 dacă generatorul este valid sau 0 dacă nu este.

ATENȚIE la respectarea cerinței problemei: afișarea rezultatelor trebuie făcută EXACT în modul în care a fost indicat! Cu alte cuvinte, pe stream-ul standard de ieșire nu se va afișa nimic în plus față de cerința problemei; ca urmare a evaluării automate, orice caracter suplimentar afișat, sau o afișare diferită de cea indicată, duc la un rezultat eronat și prin urmare la obținerea calificativului „Respins”.

Restricții

1. $2 \leq n < 10000$
2. Va exista cel puțin un bit 1 în secvență.
3. **Atenție:** În funcție de limbajul de programare ales, fișierul ce conține codul trebuie să aibă una din extensiile .c, .cpp, .java, sau .m. Editorul web **nu va adăuga automat** aceste extensii și lipsa lor duce la imposibilitatea compilării programului!
4. **Atenție:** Fișierul sursă trebuie numit de candidat sub forma: <nume>.<ext> unde *nume* este numele de familie al candidatului și extensia (*ext*) este cea aleasă conform punctului anterior. Atenție la restricțiile impuse de limbajul Java legate de numele clasei și numele fișierului!

Exemple

Intrare	Ieșire	Explicații
18 10110011011100001	2 2 0 1 0	Secvența are 18 biți. Secvențele de 1 sunt marcate cu roșu. Sunt 2 secvențe de un bit, două de doi biți, niciuna de trei biți și una de patru biți, care este și cea mai lungă. Pentru că sunt mai puține secvențe de 3 biți decât de 4, se afișează 0 pe a doua linie.
24 101100110111000010100100	4 2 1 1	În acest exemplu sunt patru secvențe de lungime 1, două de lungime 2 și una de lungime 3, care este și cea mai lungă. Pentru că se respectă condiția specificată, pe a doua linie se afișează 1.

Timp efectiv de lucru: 120 de minute