

Problema 2018.1.2 – Distribuție biți RNG

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 fiecare secvență posibilă de doi biți: 00, 01, 10 și 11 cât și raportul între numărul de biți de 0 și de 1. Pentru ca secvența de biți să fie aleatoare, trebuie ca numărul de apariții pentru fiecare din cele patru perechi să fie aproximativ egale și în același timp numărul de biți de 0 să fie aproximativ egal cu cei de 1. Mai precis, trebuie ca raporturile R_1 dintre numărul de apariții a perechii care apare de cele mai multe ori și numărul de apariții a perechii care apare de cele mai puține ori, cât și raportul R_2 între numărul de apariții ale celui mai frecvent bit și numărul de apariții ale celui mai puțin frecvent bit să fie mai mici sau egale cu 110%.

Cerință

Dându-se un număr n reprezentând numărul de biți generat de RNG și secvența de n biți, să se calculeze raporturile R_1 și R_2 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 raporturile R_1 și R_2 calculate conform descrierii, valori fracționare cu două zecimale, separate prin spațiu, iar pe a doua linie 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. $2 \leq n \leq 10000$, n este par
2. **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!
3. **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	1.50 1.25 0	Secvența are 18 biți și extragând secvențele de câte doi biți se observă că 00 apare de două ori, 01 apare de două ori, 10 apare de două ori și 11 apare de trei ori. Calculând raportul dintre maxim și minim ($3/2$) obținem 150%. Numărul de biți 1 este 10 și numărul de biți 0 este 8. Raportul

		este $10/8 = 125\%$. Pentru că cel puțin unul din raporturi este mai mare strict decât 110%, se afișează pe a doua linie 0 (secvența nu este aleatoare).
24 101100110111100001010010	1.00 1.00 1	În acest exemplu toate perechile apar de trei ori, deci raportul R_1 este 100% . Numărul de biți 1 este 12 și este egal cu numărul de biți 0, deci și raportul R_2 este 100% . Astfel, pentru că ambele raporturi sunt mai mici sau egale decât 110%, pe a doua linie se afișează 1 (secvența este aleatoare).

Timp efectiv de lucru: 120 de minute