

Histograma de erori procentuale

Studiați o problemă de minimizare a unor funcții și în urma rulării mai multor algoritmi doriți să vedeți cum s-au comparat aceștia cu soluțiile optime cunoscute. Pentru a realiza acest lucru vreți să vedeți cu cât la sută diferă soluțiile fiecărui algoritm față de optimul cunoscut. După acest pas, vreți să faceți o histograma pe procente de eroare, adică să vedeți câți algoritmi au oferit o valoare cu eroarea între 0% și 5%, apoi 5% și 10% etc.

Cerință

Scrieți un program ce primește valoarea minimă optimă și apoi multiple valori (rezultatul fiecărui algoritm). Calculați eroarea relativă, în procent față de minimul optim. Calculați histograma erorilor având intervalele cu o rezoluție de 5% și apoi afișați-o.

Intervalele sunt:

- [0%-5%)
- [5%-10%)
-
- [95%-100%)
- 100% și peste

Date de intrare

Se va citi de la tastatură (fluxul *stdin*), de pe prima linie, un număr rațional reprezentând soluția optimă cunoscută. De pe a doua linie, se vor citi multiple numere raționale (citite până la *end of file*), separate prin spațiu. Liniile sunt terminate cu caracterul *newline* (tasta *Enter*).

Date de ieșire

Programul va afișa pe ecran (*stream*-ul standard de ieșire), histograma cu erorile relative procentuale, în formatul:

```
[0% - 5%) = a
[5% - 10%) = b
.....
[95% - 100%) = c
100%+ = d
```

Unde *a*, *b*, *c* și *d* sunt numărul de erori procentuale găsite în intervalul respectiv.

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 și precizări

1. Algoritmii nu pot găsi valori mai mici decât optimul cunoscut și deci orice eroare va fi de 0% sau mai mare.
2. Numărul de algoritmi testați nu va depăși 100.
3. Formula erorii relative este $E_r = (valoare_masurata - valoare_nominala) / valoare_nominala$
4. **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 de compilare a programului!
5. **Atenție:** Fișierul sursă trebuie numit de candidat sub forma: <nume>.<ext> unde nume este

numele de familie al candidatului și extensia este cea aleasă conform punctului anterior.

6. **Atenție** la restricțiile impuse de limbajul Java legate de numele clasei și numele fișierului!

Exemplu

Intrare	Ieșire
5 5 5 5	[0% - 5%) = 3 [5% - 10%) = 0 [10% - 15%) = 0 [15% - 20%) = 0 [20% - 25%) = 0 [25% - 30%) = 0 [30% - 35%) = 0 [35% - 40%) = 0 [40% - 45%) = 0 [45% - 50%) = 0 [50% - 55%) = 0 [55% - 60%) = 0 [60% - 65%) = 0 [65% - 70%) = 0 [70% - 75%) = 0 [75% - 80%) = 0 [80% - 85%) = 0 [85% - 90%) = 0 [90% - 95%) = 0 [95% - 100%) = 0 100%+ = 0
Explicație Toți algoritmi au obținut valoarea optimă și deci erori de 0% față de optimul cunoscut.	
Intrare	Ieșire
10 10 15 20	[0% - 5%) = 1 [5% - 10%) = 0 [10% - 15%) = 0 [15% - 20%) = 0 [20% - 25%) = 0 [25% - 30%) = 0 [30% - 35%) = 0 [35% - 40%) = 0 [40% - 45%) = 0 [45% - 50%) = 0 [50% - 55%) = 1 [55% - 60%) = 0 [60% - 65%) = 0 [65% - 70%) = 0 [70% - 75%) = 0 [75% - 80%) = 0 [80% - 85%) = 0

	$[85\% - 90\%) = 0$ $[90\% - 95\%) = 0$ $[95\% - 100\%) = 0$ $100\%+ = 1$
Explicație Un algoritm a găsit valoarea optimă. Alt algoritm a găsit o valoare cu 50% mai mare decât optimul si deci se va încadra în intervalul de [50% - 55%). Al treilea algoritm a produs dublul soluției optime și deci o eroare de +100%.	

Timp de lucru: 120 de minute