

Model Subiect AF - NOTA 6 – Timp 50 mins

La o ferma din Brasov un fermier are n vaci cu m culori fiecare.

Aceste vaci pot avea 0 sau mai multe pete din fiecare culoare. Indexandu-se culorile cu valori intregi incepand cu 0, sa se citeasca din **fișier** toate vacile pe care le are acest fermier alaturi de cate pete are fiecare vacuta si de ce culoare.

Exemplu de date de intrare:

6 (numarul de vaci)

7 (numarul de culori)

3 0 2 8 1 5 4 (numarul de pete de fiecare culoare pe care il are prima vacuta)

2 1 4 6 9 2 1

0 0 2 3 0 2 9

0 1 1 8 3 2 5

2 3 4 1 10 0 0

13 5 21 34 8 3 55

(0.5pct oficiu)

0.25 citirea corecta din fisier a intrarii

- a) (1pct) Sa se scrie o functie care valideaza n si m , unde n este numarul de vaci care trebuie sa fie minim 4 si maxim 100, iar m este numarul de culori posibile, care este in intervalul $[4, 100]$. Functia valideaza aceste valori, dar si inputul din fisier.
- b) (1.5pct) Daca continutul fisierului este valid, sa se determine *indexul culorii* cea mai intalnita printre vacile fermierului.

Exemplu: Cea mai intalnita culoare dintre vacile din exemplu este culoarea cu indexul 6 avand cea mai mare suma totala a petelor (egala cu 74).

- c) (1.75pct) Luand in calcul un index de culoare " k ", sa se sorteze descrescator vacutele pe baza acestei culori. Sa se afiseze in consola sirul vacutelor sortate.

Exemplu: Daca indexul este 6, atunci se vor sorta descrescator in functie de numarul petelor cu culoarea de pe indexul 6.

Observatie! Nu se poate utiliza `std::sort`, in rest, oricare varianta de sortare este valida.

- d) (1pct) Sa se verifice daca exista vacute pentru care numarul petelor din fiecare culoare de pe vacutele respective respecta secventa lui Fibonacci ($f(x) = f(x-1) + f(x-2)$).

Exemplu: Vacuta cu indexul 3 respecta secventa lui Fibonacci deoarece numarul petelor se poate organiza astfel: 0 1 1 2 3 5 8. De asemenea, si vacuta cu indexul 5: 3 5 8 13 21 34 55

Sa se scrie intr-un fisier de iesire indexul (indecsii) vacutelor care au aceasta proprietate indeplinita.

Model Subiect AF - NOTA 10 – Timp 80 mins

Un investitor va cere sa il ajutati cu o aplicatie care ii poate oferi informatii legate de piata de criptomonede. Acesta va ofera ca date de intrare pentru aplicatia voastra un fisier care contine un numar (n) de criptomonede alese de el, alaturi de preturile pentru fiecare dintre aceste criptomonede in ultimele (x) zile (n si x citite tot din fisier).

Exemplu de date de intrare:

4 (numarul de criptomonede diferite)

5 (numarul de zile precedente pentru care primim preturile)

200 210 205 190 180

89 80 90 95 100

21 21 21 22 21

1890 1887 1900 1930

- a) Sa se determine care a fost cea mai profitabila criptomoneda procentual in ultimele x zile daca am fi investit din prima zi si am fi retras in ultima zi. Sa se afiseze in consola a cata criptomoneda a fost alaturi de procentul de profit.

Pentru datele de intrare din exemplu cea mai profitabila criptomoneda ar fi fost a doua criptomoneda cu 12.35% profit. Se va afisa "Criptomoneda 1 cu 12.35% profit".

- b) Sa se determine care a fost intervalul de 24 de ore in care se inregistreaza pierderea cea mai mare (ca valoare) pentru fiecare criptomoneda in parte. Sa se afiseze in consola pentru fiecare criptomoneda in parte intervalul respectiv alaturi de cati bani am fi pierdut daca am fi cumparat o criptomoneda la inceputul intervalului si am fi vandut-o la finalul sau.

Exemplu: Pentru prima criptomoneda a patra zi (index 3) a fost cea mai nonprofitabila ca valoare si pierderea ar fi fost de 15 "bani" (s-a sczut de la 205 la 190). Se va afisa "De la ziua cu indexul 2 la ziua cu indexul 3 am fi avut pierderea de 15"

La a doua criptomoneda a doua zi (index 1) a fost cea mai nonprofitabila cu pierdere de 9 "bani". Se va afisa "De la ziua cu indexul 0 la ziua cu indexul 1 am fi avut pierderea de 9"

- c) Sa se ordoneze criptomonedele crescator in functie de diferenta dintre numarul de zile cu trend crescator si numarul de zile cu trend descrescator. Se doreste o sortare eficienta, fara utilizare de std::sort.

Exemplu: Pentru prima criptomoneda avem 1 zi cu trend crescator si 3 zile cu trend descrescator => - 2. A doua criptomoneda are 3 zile cu trend crescator si 1 zi cu trend descrescator => 2.

- d) Pe baza fluctuatiei din aceste zile, sa se genereze random ce pret ar putea avea fiecare criptomoneda pentru urmatoarele 3 zile si sa se scrie intr-un fisier aceste valori. Fluctuatia este determinata de diferenta dintre cea mai mare valoare si cea mai mica valoare (spre exemplu pentru prima criptomoneda fluctuatia este $30 = 210 - 180$). Fluctuatia va putea fi influentata

ulterior de catre fiecare zi generata in parte adica daca urmatoarea zi generata random ar putea creste fluctuatia si mai mult urmatoarea valoare generata va avea intervalul si mai mare. Generarea noilor valori se va face utilizand rand din std sau o alta metoda de generare viabila.

Exemplu: Pentru prima criptomoneda determinam fluctuatia ca fiind 30. Prima valoare generata va putea fi din intervalul [150, 210]. Sa presupunem ca prima valoare generata a fost 160. Aceasta valoare creste fluctuatia la 50 (210 - noua valoare 160), prin urmare urmatoarea valoare generata va putea fi din intervalul [110, 210].

Barem

1. Citirea datelor de intrare/afisare corecta si conform cerintelor 1p
2. Subpunct a 1.5p
3. Subpunct b 1.5p
4. Subpunct c 2.5p (sortare ineficienta reduce punctajul la 1.5p)
5. Subpunct d 2.5p

Observatii!

1pct se distribuie catre scriere modulara a aplicatiei (subprograme), denumire corecta a variabilelor/subprograme, transmitere corecta de parametrii, eficienta per total a aplicatiei, compilare ok.