

Olimpiada Națională de Informatică, Etapa județeană, Clasa a VI-a Descrierea soluțiilor

Comisia științifică

16 martie 2025

Problema 1. avion

Propusă de: prof. Șerban Marinel-Paul, Colegiul Național "Emil Racoviță", Iași

Cerința 1 - 50p

Având în vedere că numărul de rânduri NR este par, există același număr de rânduri în fiecare jumătate a avionului.

Este suficient să se compare cu $NR/2$ rândul de pe biletul fiecărui pasager. Dacă rândul este în prima jumătate a avionului ($\text{rândul} \leq NR/2$) se va afișa 1, altfel se va afișa 2.

Cerința 2 - 50p

Pentru fiecare pasager se va compara și de această dată rândul de pe bilet cu mijlocul avionului ($NR/2$). Dacă $\text{rândul} \leq NR/2$ și pasagerul urcă în avion pe scara 1, la suma totală se va adăuga rândul de pe bilet, iar în caz contrar, când pasagerul intră pe scara 2, el va parcurge până la rândul de pe bilet distanța $NR - \text{rândul} + 1$, care se va adăuga la suma totală. Apoi, pasagerul va mai parcurge 1, 2 sau 3 metri în funcție de locul ocupat pe rândul de pe bilet: 1 metru dacă locul este 3 sau 4, 2 metri dacă locul este 2 sau 5 și 3 metri dacă locul este 1 sau 6. Aceste valori se adaugă și ele la suma totală. Având în vedere că fiecare dintre cei n pasageri parcurge 3 metri de la intrarea în avion până la culoarul central, la suma totală se mai adaugă $3 * n$ metri.

Problema 2. mandatar

Propusă de: Nodea Gheorghe-Eugen, Centrul Județean de Excelență Gorj

Cerința 1 - 50p

Se rețin numerele citite din fișierul de intrare în șirul A - tablou unidimensional (vector). Pentru determinarea celui mai mare număr prim din șirul A se folosește un algoritm de verificare a primalității unui număr.

În funcție de implementarea aleasă pentru verificarea primalității se pot obține punctaje gradual, între 20 – 50p ($O(n)$, $O(\sqrt{n})$), ciurul lui Eratostene)

Cerința 2 - 30p

Pe baza șirului A , construim șirul B folosind un algoritm de descompunere în factori primi, unde fiecare element al șirului B_i este cel mai mic număr natural care are aceiași factori primi cu A_i , cu $1 \leq i \leq n$.

Algoritmul de descompunere în factori primi va determina, pentru fiecare element al șirului A , atât elementele șirului B cât și numărul de factori primi al fiecărui element B_i . Se va reține cel mai mare număr al șirului B care are un număr maxim de factori primi.

În funcție de implementarea aleasă pentru descompunerea în factori primi se pot obține punctaje gradual, între 10 – 30p ($O(n)$, $O(\sqrt{n})$), ciurul lui Eratostene sau optimizari ale factorizării)

Observație

Printr-un algoritm de descompunere în factori primi se poate rezolva atât Cerința 1 cât și Cerința 2.

Cerința 3 - 20p

Pentru determinarea lungimii maxime a unei secvențe mandatorii din șirul B se poate folosi o structură *for* ce parcurge numerele mandate. Acestea pot fi: 2, 3, 5, 6 sau 7.

```
for(x = 2; x <= 7; x++) {
    if (x == 4) continue;
    k = 0; ok = 0;
    for(int i=1; i<=n; i++) {
        if(B[i] % x == 0) {
            k++;
            if(B[i] == x) ok = 1;
            if(ok == 1 && k > lgmax)
                lgmax = k;
        }
        else {
            k = 0; ok = 0;
        }
    }
}
```

Echipa

Problemele pentru această etapă au fost pregătite de:

- prof. Arișanu Ana-Maria - Colegiul Național "Mircea cel Bătrân", Râmnicu-Vâlcea
- prof. Georgescu Alice - Colegiul Național "Mihai Viteazul", Ploiești
- prof. Pintescu Alina - Colegiul Național "Gheorghe Șincai", Baia Mare
- prof. Oprîță Petru-Simion - Liceul "Regina Maria", Dorohoi
- prof. Piț-Rada Ionel-Vasile - Colegiul Național "Traian", Drobeta-Turnu Severin
- prof. Pracsiu Dan - Liceul Teoretic "Emil Racoviță", Vaslui
- prof. Șerban Marinel-Paul - Colegiul Național "Emil Racoviță", Iași
- prof. Nodea Gheorghe-Eugen - Centrul Județean de Excelență Gorj
- stud. Gheorghies Petruț-Rareș - Facultatea de Automatică și Calculatoare București
- stud. Răileanu Alin-Gabriel - Universitatea "Alexandru Ioan Cuza", Facultatea de Informatică, Iași