

Răspunsurile corecte cu explicații

1. b

$(x\%3+x\%6)/8 \times\% 3$ restul maxim 2 $x\%6$ restul maxim 5 $2+5=7$ $7/8 = 0$ ca impartire intreaga!

2. a

3. b

4. b

Acces la al 2-lea student (index 1) și atribuire nota 10: `grupa.studenti[1].nota = 10;`

5. a

Noul șir se obține prin permutare circulară la stânga cu `i` poziții.

6. a

7. a

Problema cere să se facă suma elementelor de pe linii într-o matrice.

8. b

Funcția `transform` calculează transpusa unei matrice.

9. d

Funcția `transform` realizează rotirea unei matrice în sens orar cu 90 grade.

10. d

Funcția `process` calculează subsecvența crescătoare maximală.

11. a

12. a

Numărul submulțimilor formate din k elemente este C_n^k . Rezultă că numărul submulțimilor lui M este $\sum_{k=0}^n C_n^k = 2^n$ (Binomul lui Newton).

13. b

Sunt doar 6 întrucât în funcția de Afișare $i < k$, iar k va avea valoarea $n-1$.

14. d

15. d

Bucula exterioară generează $\log(n)$ iterații. Indexul i ia valorile $n, n/2, n/4 \dots 1$. Pentru fiecare astfel de valoare i , bucla interioară generează $\log(i)$ iterații. Prin urmare, bucla interioară generează pe rând $\log(n), \log(n/2), \log(n/4) \dots 1$ iterații, echivalent cu $\log(n), \log(n)-1, \log(n)-2 \dots 1$ iterații, cu un total de $\log(n) + \log(n)-1 + \log(n)-2 + \dots + 1 = \frac{\log(n)(\log(n)+1)}{2}$. Rezultă complexitate $\mathcal{O}(\log^2(n))$.

16. a

```

int nb_of_triangles(int p){
    int a, b, c;
    int nt = 0;
    for (a = 1; a <= 1 + p / 3; a++)
        for (b = 1 + a; b <= 1 + (p - a) / 2; b++)
        {
            c = p - a - b;
            if (c * c == a * a + b * b)
            {
                nt++;
                cout<<a<<" "<<b<<" "<<c<<endl;
            }
        }
    return nt;
}

```

17. a

În bucla while se calculează câți de 1 sunt în reprezentarea binară a numărului i . Nr adună aceste valori pt fiecare pas, astfel ca Pentru 1 afișează 1 Pentru 2 afișează 2 (1 bit de la 1 și 1 de la 2) Pentru 3 afișează 4 (suma de la 2 + 2 biti de 1 de la 3) Pentru 6 afișează 9 Pentru 19 afișează 40 Pentru 20 afișează 41

Se poate face tabelul cu reprezentarea binară a numerelor de la 1 la 20 și se numără valorile de 1.

18. b**19. a**

Sortat după deadline: (1,2),(1,3),(2,4),(2,5),(3,6),(3,7),(4,8) Executăm task-uri în ordinea lor, fără să depășim deadline-ul:

- Alegem (1,2) → timpul ocupat: 1 • Alegem (1,3) → timpul ocupat: 1 + 1 = 2 • Alegem (2,4) → timpul ocupat: 2 + 2 = 4 • Alegem (2,5) → timpul ocupat: 4 + 2 = 6 • Alegem (3,6) → timpul ocupat: 6 + 3 = 9 (nu putem, deadline 6)
- Alegem (3,7) → timpul ocupat: 6 + 3 = 9 • Alegem (4,8) → timpul ocupat: 9 + 4 = 12

20. b

Se consideră număratoarea în cerc ca o număratoare liniară în mai multe siruri mici, fiecare cu n persoane, obținând un sir cu p persoane (p fiind multiplu de n). Număratoarea se termină atunci când al n -lea spectator (dintr-un sir mic) este selectat, și astfel, următorul spectator care ar trebui să fie selectat va fi un al k -lea din următorul sir mic. Mai exact, p trebuie să fie și multiplu de k , adică $p = \text{cmmmc}(n, k)$. Dintre cei p participanți, au fost selectați exact p / k , deci spectatori neselectați vor fi în număr de $\text{nr} = n - p/k = n - \text{cmmmc}(n,k)/k = n - (n*k/\text{cmmdc}(n,k))/k = n - n/\text{cmmdc}(n,k)$. Vor fi selectați toți dacă $\text{cmmdc}(n,k)=1$.

21. d**22. b**

Adaptare a problemei comis-voiajorului (agent comercial) care trebuie să viziteze n orașe. Cunoscându-se șoselele existente între orașe, să se determine o modalitate (toate modalitățile) prin care comis-voiajorul poate parcurge fiecare oraș o singură dată și se întoarce în orașul de plecare. $\Rightarrow$ graful trebuie să fie hamiltonian.

23. a

Diametrul este dat de nodul/nodurile cu excentricitatea maximă. Cu alte cuvinte, diametrul este cel mai lung dintre cele mai scurte drumuri dintre oricare două noduri din structura S . Cel mai lungi astfel de drumuri minime sunt între nodurile care aparțin grafurilor din extremitățile structurii S , adică G_1 și G_n .

Cum structura S este $G_1 - G_2 - \dots - G_n$, rezultă ca avem un număr de $n - 1$ muchii de la G_1 la G_n , fără a considera muchii din interiorul grafurilor. Cum fiecare graf G_k este complet, orice nod $v \in G_k$ se conectează direct cu nodurile care fac legătura cu grafurile adiacente, G_{k-1} și G_{k+1} . Prin urmare, fiecare graf intermediar G_k are exact o muchie pe drumul minim dintre G_1 și G_n . Astfel, drumul minim între G_1 și G_n este $(n - 1) + (n - 2)$, unde $n - 2$ este numărul de grafuri intermediare G_k din S care sunt pe drumul dintre G_1 și G_n .

Nodurile cu excentricitate maximă sunt toate nodurile din G_1 diferite de nodul prin care se face legătura cu G_2 și toate nodurile din G_n diferite de nodul prin care se face legătura cu G_{n-1} . Practic se mai adaugă două muchii pe drumul minim care conectează nodurile din G_1 și G_n cu nodurile de legătură.

Prin urmare, excentricitatea maximă (diametrul structurii S) este $(n - 1) + (n - 2) + 2 = 2 * n - 1$.

24. d

Funcția process calculează suma valorilor pare ale tabloului.

25. b