

Răspunsurile corecte cu explicații

1. d

```
char sir[] = "admitere";
for (int i = 1; i <= 3; i++)
{
    strcpy(sir + i, sir + 1);
}

cout << sir << endl;
```

2. c

struct punct trebuie urmată de numele unei variabile.

element este un vector de tip `_element`, deci nu poate fi tip de date

```
int x, y;
cin >> x;
y = x;
while (x <= 3)
{
    cout << " ";
    y = y + 1;
    x = x + y;
}
```

4. d

5. a

Se iau valorile din răspunsuri pe rând și se urmărește codul dat pentru fiecare pas al instrucțiunii repetitive.

6. a

În variabila `admitere2021` de tipul structurii examen este definit un vector de candidați. Deci accesul la un element al vectorului (un anumit candidat) se face prin indicele elementului respectiv, cu `admitere2021.c[i]`.

7. b

A 0100 0001 nu se va modifica n 0110 1110 cel mai lung sir de 1. va modifica primii 2 de 1 in 2 de 0 r 0111 0010 se va modifica ultimul 1 e 0110 0101 nu se va modifica a 0110 0001 nu se va modifica p 0111 0000 se va modifica ultima pozitie

Ana - se modifica 2 cifre binare are - se modifica 1 cifră binară apa - se modifica 1 cifră binară

8. d

Condiția verifică relația dintre valorile primului element și a ultimului element din tablou: '4' și '2'. Deoarece condiția verificată este adevărată, se actualizează `s[1]` prin incrementarea valorii sale curente. `s[1]` devine '2'.

9. a

Codul furnizat face suma elementelor de pe ultima coloană.

10. c

Funcția parcurge toate elementele matricii care reprezintă parcelele terenului. Pentru fiecare parcelă se verifică dacă valoarea lucrărilor este cuprinsă în intervalul `[a, b]`. Dacă verificarea condiției este adevărată se decrementează valoarea ce va fi returnată. Inițial, această valoare (variabila locală `ret`) este egală cu numărul elementelor matricii.

11. a

Codul furnizat schimbă elementul de pe diagonala principală cu cel din dreapta sa. Elementul de pe ultima linie și ultima coloană va fi schimbat cu elementul de pe poziția $[0][0]$, care a fost schimbat în 2 la prima trecere prin for.

12. c

Matricea C este produsul cartezian al liniilor din matricile A și B deci nK linii rezultate.

13. c

Numărul de elemente ale mulțimii $\mathbb{C}$ este egal cu $24 \cdot 24 = 576$. Prima parte a condiției filtrează jumătate dintre perechi (rămân doar perechile de elemente ale căror sumă este pară, adică a și b par, sau a și b impar), iar cea de-a doua filtrează jumătate din jumătatea rămasă (rămân doar perechile pentru care măcar una dintre valori este pară, adică a par sau b par). Numărul de elemente care satisfac condiția este egal cu $n/4 = 576/4 = 144$.

14. b

Cuvintele generate care încep cu litera l și se termină cu litera o : lilo, limo, lino, llo, lmo, lno, lmlo, lmno, lno, lmo, lno, lolo, lomo, lono

15. a**16. b****17. a**

O funcție injectivă poate fi asociată cu un aranjament.

18. b

Pentru a rezolva problema trebuie întâi determinată prima suprapunere a celor trei ace ale ceasului. Acest lucru se poate realiza determinând suprapunerile orarului cu minutarul și a minutarului cu secundarul.

Dacă considerăm ca x este numărul de ore din jumătate de zi, adică $n/2$, și y este numărul de minute dintr-o ora, adică m , atunci: - viteza de deplasare a orarului (v_o) într-o ora este: $360/x$ (grade/ora) - viteza de deplasare a minutarului (v_m) într-o ora este: 360 (grade/ora) - viteza de deplasare a secundarului (v_s) într-o ora este: $360 \cdot y$ (grade/ora)

Într-o ora orarul parcurge o distanță d , iar minutarul $d + 360 \Rightarrow d_o = d_m - 360$.

Deoarece $d = v \cdot t$, relația de mai sus devine: $t \cdot 360/x = t \cdot 360 - 360 \Rightarrow t = x/(x-1)$.

Astfel, prima întâlnire a orarului și a minutarului este după $x/(x-1)$ ore.

Deci periodicitatea cu care se întâlnesc orarul și minutarul este $x/(x-1)$. În mod analog se determina periodicitatea cu care se întâlnesc minutarul și secundarul: $1/(y-1)$.

Problema se rezuma la a găsi toate suprapunerile orar-minutar și de a verifica dacă nu cumva coincide cu suprapunerea minutar-secundar.

Ceea ce înseamnă că trebuie determinate toate perechile k_1, k_2 (numere naturale) astfel încât: $k_1 \cdot x/(x-1) = k_2/(y-1)$.

Dacă încercăm pentru toți k_1 începând cu 1 și mai mici decât n , atunci găsim o suprapunere a celor trei ace dacă $x \cdot k_1 \cdot (y-1)$ se divide la $(x-1)$.

Prima valoare k_1 care îndeplinește condiția de mai sus ne da timpul primei suprapuneri, $k_1 \cdot x/(x-1)$, iar numărul total de suprapuneri dintr-o zi este calculat ca fiind $2 \cdot x/(k_1 \cdot x/(x-1))$.

19. c

Suma gradelor tuturor nodurilor unui graf neorientat este egală cu dublul numărului de muchii: $\sum_{i=1}^n d_i = 2 \cdot m$, unde n reprezintă numărul de noduri din graf, d_i gradul nodului i , iar m numărul de muchii din graf.

20. d

2 componente conexe: nodurile pare și cele impare.

21. c

Se formează două subgrafuri complete, cu noduri cu identificatori pari, respectiv impari. Există și câte o muchie între orice noduri cu identificatori de forma $< 2k, 2k + 1 >$ (din subgraful par, respectiv impar), cu mențiunea că nodul 1 (din subgraful impar) și nodul n (din subgraful par) nu formează pereche cu nici un nod din celălalt subgraf: total $\frac{n}{2} - 1$ astfel de muchii. Pentru a izola nodul 1 sau n trebuie tăiate $\frac{n}{2} - 1$ muchii. Pentru a izola un nod $2, 3, \dots, n - 1$ trebuie tăiate $\frac{n}{2}$ muchii. Pentru a izola cele două subgrafuri trebuie tăiate $\frac{n}{2} - 1$ muchii.

22. a

În graful G fiecare vârf trebuie să aibă grad par. Un graf complet cu număr impar de vârfuri îndeplinește această condiție. Evident, un ciclu conține cel puțin 3 vârfuri, deci n trebuie să fie mai mare decât 2.

23. a**24. c**

Funcția realizează numărarea caracterelor litere din șirul de caractere primit prin primul parametru. Al doilea parametru este indexul maxim din șir până la care se face numărarea. Șirul este parcurs în sens descrescător și atunci când caracterul de pe poziția curentă este literă se adună 1 la valoarea returnată, altfel nu se adună nimic.

25. c

La primul apel $n > 0$, deci va intra în primul if, apoi la al doilea if, $n \bmod 10 = 1$, dar 1 nu este mai mare ca 1 (valoarea lui m), deci nu se va intra în acest if. Se va reapele funcția cu valorile $f(1232, 1)$, cum restul împărțirii lui n la 10 este $2 > 1$, se va intra în if și m va deveni 2. Pe ecran se va afișa 2. La apelul următor ($f(123, 2)$) $3 > 2$, se va afișa 3 apoi $m = 3$. La apelurile $f(12, 3)$, apoi $f(1, 3)$ valoarea lui m rămâne 3 întrucât nici 2 nici 1 nu sunt mai mari ca 3. La afișarea finală, m va rămâne 3 pt toate cele 5 afișări întrucât m este transmis prin referință, deci ultima valoare avută se va păstra.