

Răspunsurile corecte cu explicații

1. d

```
#include <iostream>
using namespace std;
int main()
{
    //int X=0,Y=1,Z=0;
    //int X=1,Y=0,Z=0;
    //int X=0,Y=0,Z=0;
    //int X=0,Y=1,Z=0;

    /*
    //I. X      False; Y      True; Z      False; F
    //II. X      True; Y      False; Z      False; A
    //III. X      False; Y      False; Z      True; F
    //IV. X      True; Y      True; Z      True; F
    */
    cout<<(X || Z) && (!(X || Y && Z));
    return 0;
}
```

2. b

3. b

4. b

5. a

Funcția process returnează indexul primei apariții a unei valori într-un tablou. Dacă valoarea nu este în tablou atunci se returnează dimensiunea tabloului.

6. b

7. d

Funcția transform realizează inversarea elementrlor tabloului.

8. d

Rulare pas cu pas

9. a

Rulare pas cu pas

10. d

Funcția transform realizează o permutare a elementelor tabloului v.

11. d

Funcția process calculează suma elementelor de pe conturul matricei.

12. c

13. d

combinari de 10 luate cate 3

14. b

Multimi de lungime 1 sunt 3, de lungime 2 sunt 3 formate dintr-un nr par si unul impar, si inca 3 formate din al doilea numar par si cate unul impar, de lungime 3 este unul cu toate elementele impare si alte 3 formate din două numere pare si unul impar.

15. b

Există 625 combinații posibile ($5^4 = 625$), iar numărul maxim de încercări eșuate înainte de a ghici corect codul este $625 - 1 = 624$.

16. c

Rulare pas cu pas

17. b

i) este corecta după a treia iterație vectorul este 8, 5, 3, 2, 4

```
int v[] = 8, 3, 5, 2, 4 ; int n=5; for (int i = 1; i < n; i++) int aux = v[i]; int j = i; while (j > 0 v[j - 1] < aux) v[j] = v[j - 1]; j--; v[j] = aux; for (int k = 0; k < n; k++) cout<<v[k]<< " "; cout<<endl;
```

18. b

Sunt n-1 permutări. $(n-1)! = 5! = 120$

19. b**20. c**

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 conectarea unui graf G_k cu grafurile adiacente, G_{k-1} și G_{k+1} , se face prin același nod, rezultă că fiecare graf intermediar G_k nu adaugă muchii suplimentare pe drumul minim dintre G_1 și G_n . Astfel, drumul minim între G_1 și G_n este $n - 1$.

Nodurile cu excentricitate maximă sunt toate nodurile din G_1 diferite de nodul prin care se face legatura 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 $(n - 1) + 2 = n + 1$.

21. b

i) Fals, vor fi 2 valori nenule pe pozițiile 1 și 3 ii) Fals, va returna 2 iii) Adevarat iv) Fals v) Fals vi) Adevarat

```
include <iostream> using namespace std;
```

```
int fun(int x,int y, int a[]) if (y<=0) return x; else a[-y]=++x; x=a[y]; cout << "apel "; return fun(x,y-1,a);
```

```
int main() int x = 0; int a[] = 0, 0, 0, 0; cout << fun(x, 4, a) << endl; for(int i = 0; i < 4; ++i ) cout << a[i] << " "; cout << endl; cout << x; return 0;
```

22. d

Apelul funcției `process(n, 0)` calculează numărul obținut prin inversarea cifrelor numărului n .

23. b

3, 2, 6, 7, 8, 9

24. b

25 (24 apeluri + 1 apel in care nu se intampla nimic)

25. b