

Test de informatică C/C++

Subiectele 1-10 au un singur răspuns corect.

Subiectele 11 și 12 vor fi rezolvate complet.

Nota finală $N=0.6N1+0.4N2$, unde

$N1$ =punctajul total de la problemele 1-10 +1p din oficiu,

$N2$ =punctajul total de la problemele 11-12 +1p din oficiu.

Timp de lucru - două ore.

1. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile de maximum 4 litere distincte din mulțimea {s,u,p,r,a,c,o,n,d,u,c,t,o,r}. Primele 5 șiruri generate sunt, în această ordine: a, ac, acd, acdn, acdo. Imediat după șirul can se generează:

a. cao b. cand c. cano d. canc

2. Se consideră subprogramul f, definit alăturat.

Valoarea $f(52,3)$ este:

a. 3 b. 52 c. 48 d. 12

```
int f(int x,int y)
{
    if(x < 4*y) return y;
    return f(x,4*y);
}
```

3. O expresie C/C++ a cărei executare determină memorarea în variabila reală x a mediei geometrice a variabilelor x,y este:

a. $x = \text{sqrt}(x*y)$; b. $x = (x+y)/2$; c. $x = 1/x + 1/y$; d. $x = x*y$;

4. Expresia C/C++ $(-10 \leq x) \ \&\& \ !(x > -5) \ || \ (x >= 5) \ \&\& \ !(x > 10)$ are valoarea 1 dacă și numai dacă valoarea memorată de variabila reală x aparține mulțimii:

a. $(-\infty, -10] \cup (10, \infty)$ b. $[-10, -5] \cup [5, 10]$ c. $[-5, 5]$ d. $[-10, 5] \cup [10, \infty)$

5. Se consideră subprogramul f alăturat ($x\%y$ este restul împărțirii numărului natural x la numărul natural nenul y).

Rezultatul executării instrucțiunii:

```
cout << f(21) << ' ' << f(52) << "\n\n";
```

este:

a. 0 0 b. 9 2 c. 7 0 d. 21 0

```
int f(unsigned int x)
{
    if(x%3 == 0)
        return x;
    return f(x/3);
}
```

6. Se dă subprogramul alăturat.

Valoarea afișată la executarea instrucțiunii

```
fs(2468,3579);
```

este:

a. 1111 b. 2468 c. 3579 d. 0

```
int fs(int n, int m)
{
    int tmp,d,s;
    if(n<m)
    { tmp=n; n=m; m=tmp; }
    d=1;
    while(m>0)
    {
        s=m%10; m=m/10; n=n-d*s; d=d*10;
    }
    cout << n << "\n\n";
}
```

7. Se consideră graful orientat cu 7 vârfuri, numerotate de la 1 la 7, reprezentat prin listele de adiacență indicate alături.

Două drumuri sunt distincte dacă diferă prin cel puțin un arc. Numărul de drumuri elementare distincte de la vârful 2 la vârful 3 este:

- a. 0 b. 2 c. 1 d. 3

```
1: 4, 7
2: 1, 5, 6
3: 2
4: listă vidă
5: 3
6: 4, 7
7: 4
```

8. Frunzele arborelui cu rădăcină, având 8 noduri, numerotate de la 1 la 8, reprezentat prin vectorul "de tați" (0, 1, 4, 7, 1, 8, 1, 3) sunt:

- a. 2, 5, 7, 6 b. 2, 5, 6 c. 2, 5, 7 d. 4, 3, 6, 8

9. Indicați șirul afișat pe ecran în urma executării secvenței alăturate.

```
int k='C'-'c', i;
char txt[100];
strcpy(txt, "plasmon");
for(i=0; i<strlen(txt); i++)
    if(strchr("aeiou", txt[i])!=NULL)
        txt[i]=txt[i]+k;
cout << txt << "\n\n";
```

- a. PLASMON b. p1Asm0n c. AO d. PLASMoN

10. Se consideră subprogramul recursiv DIV, definit alături. Pentru a returna valoarea celui mai mare divizor comun al numerelor naturale 123 și 321, apelul subprogramului ar putea fi:

```
int DIV (int x, int y, int z)
{
    if(x%z==0 && y%z==0) return z;
    else return DIV(x,y,z-1);
}
```

- a. DIV(1, 123, 321) b. DIV(123, 321, 123) c. DIV(321, 321, 123) d. DIV(123, 321, 1)

11. Scrieți un program C sau C++ care să numere liniile dintr-un fișier text, indicat ca parametru în linia de comandă.

12. Scrieți un program C sau C++ care să preia valori `double` dintr-un fișier binar, indicat ca parametru în linia de comandă. Se presupune că fișierul binar conține 200 de valori numerice în format `double`, iar programul dumneavoastră trebuie să citească și să stocheze într-un șir ultimele 100 valori.