

FIȘA 8. Recursivitate

Pentru fiecare dintre următorii itemi, alegeți litera corespunzătoare răspunsului corect.

1. Pentru definiția alăturată a subprogramului `ex`, stabiliți ce se afișează la apelul `ex(120)`?

```
void ex(int x)
{if(x!=0) {
  cout<<x%10;
  ex(x/10);
}
}
```

- a. 021                                      b. 012                                      c. 120                                      d. 21

2. Pentru definiția alăturată a subprogramului `ex`, stabiliți ce se afișează la apelul `ex(120)`?

```
void ex(int x)
{  cout<<x%10;
  if(x!=0) ex(x/10);
}
```

- a. 0120                                      b. 021                                      c. 210                                      d. 0210

3. Pentru definiția alăturată a subprogramului `ex`, stabiliți ce se afișează la apelul `ex(2,10)`?

```
void ex(int i,int j)
{ if(i<=j) {
  cout<<i;
  ex(i+1,j-1);
  if(i%2!=0)
  cout<<9-j;
}
}
```

- a. 2345620                                      b. 246357                                      c. 2345646                                      d. 234520

4. Se știe că variabila `a` de tip `int` memorează valoarea 0. Pentru definiția alăturată a subprogramului `ex`, ce valoare va avea variabila `a` în urma executării apelului `ex(10542,1921,a)`?

```
void ex(int n, int m, int &z)
{ int c;
  if(n+m>0){  c=n%10;
              if(m%10>c)c=m%10;
              z=z*10+c;
              ex(n/10,m/10,z);
  }
}
```

- a. 12500                                      b. 24811                                      c. 11248                                      d. 2481

5. Considerând subprogramul `f`, definit alăturat, stabiliți ce valoare are expresia `f(10,30)`?

```
int f(int x,int y)
{ if (x>y) return 0;
  else if (x%2==0) return 1+f(x+1,y);
  else return f(x+1,y);
}
```

- a. 20                                      b. 11  
c. 10                                      d. 15

6. Fie subprogramul `f` definit alăturat. Care este valoarea expresiei `f(4,12)`?

```
int f(int x,int y)
{ if (x>y) return 0;
  else if (x%2!=0) return 1+f(x+1,y);
  else return f(x+1,y);
}
```

- a. 3                                      b. 2                                      c. 4                                      d. 5

7. Fie subprogramul `f` definit alăturat și `a` o variabilă de tip întreg. Dacă în urma apelului `f(a)` valoarea returnată de funcție a fost 153, atunci valoarea variabilei `a` a fost:

```
int f(int n)
{
  if (n==0) return 0;
  else return n+f(n-1);
}
```

- a. 18                                      b. 31                                      c. 20                                      d. 17

8. Fie subprogramul  $f$  definit alăturat și  $a$  o variabilă de tip întreg. Dacă în urma apelului  $f(f(a))$  valoarea returnată de funcție a fost 55, atunci valoarea variabilei  $a$  a fost:
- ```
int f(int n)
{ if (n==0) return 0;
  else return n+f(n-1);
}
```
- a. 10                      b. 4                      c. 3                      d. 5
9. Ce valoare va returna  $f(23951)$ , pentru funcția  $f$  definită alăturat?
- ```
int f(int n){
if (n==0) return 0;
else
{int c= f(n/10);
if (n%10>c)
return n%10;
else return c;
}}
```
- a. 2                      b. 3                      c. 5                      d. 9
10. Ce valoare trebuie transmisă prin intermediul parametrului  $n$  la apelul funcției  $f$  alăturate, astfel încât valoarea returnată de funcție să fie 7?
- ```
int f(long n)
{ if (n==0) return 0;
  else
  if ((n/10)%2==1) return n%10+f(n/10);
  else return f(n/10);
}
```
- a. 3258                      b. 1528                      c. 3972                      d. 3472
11. Ce valoare va fi returnată la apelul  $E(4)$ ?
- ```
int E(int n)
{if(n==0 || n==1) return 1;
else return 2*E(n-1)+E(n-2);}
```
- a. 17                      b. 15                      c. 21                      d. 9
12. Ce valoare trebuie să primească la apel parametrul formal  $n$ , pentru ca funcția alăturată să returneze valoarea 21?
- ```
int ex(int n)
{if(n==0) return 0;
else return n+ex(n-1);}
```
- a. 7                      b. 8                      c. 4                      d. 6
13. Ce valoare va fi returnată la apelul  $f(20)$ ?
- ```
int f(int n){
if (n<=1) return n; else return f(n-2)+n;}
```
- a. 210                      b. 110                      c. 0                      d. 20
14. Fie subprogramul definit alăturat. Ce se afișează în urma apelului  $P(3)$ ?
- ```
void P(int x){
if (x)P(x-1);
cout<<x;
}
```
- a. 3                      b. 123                      c. 3210                      d. 0123
15. Fie subprogramul  $P$  definit mai jos. Ce se afișează în urma apelului  $P(3)$ ?
- ```
void P(int x){
if (x) if (x%2==0) {
cout<<x;
P(x/2);}
else{
P(x-1);
cout<<x;}
```
- a. 321                      b. 231                      c. 213                      d. 123
16. Pentru funcția recursivă alăturată stabiliți care este valoarea returnată în urma apelului  $f(4)$ ?
- ```
int f ( int x)
{ if(x<=1) return x+1;
else return f(x-1)+2*f(x-2);}
```
- a. 16                      b. 15                      c. 4                      d. 11

17. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul  $f(1, 5)$ .

```
void f(int i, int j)
{if(i<=j)
  if(i+j<5){
    cout<<i;
    f(i+1,j);}
  else {
    cout<<j;
    f(i,j-1);}
}
```

- a. 54132                      b. 12345                      c. 51423                      d. 54321

18. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul  $f(5, 1)$ .

```
void f(int i, int k)
{if(k<=4){
  cout<<i*k;
  f(i-1,k+1);
}}
```

- a. 1248                      b. 5898                      c. 1234                      d. 5488

19. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul  $f(5, 1)$ .

```
void f(int n, int k)
{if(k<=n){
  cout<<n-k;
  f(n,k+1);
}}
```

- a. 12345                      b. 01234                      c. 43210                      d. 54321

20. Se consideră funcția alăturată  $f1$ . În urma apelului  $f1(7)$ , se va afișa:

```
void f1(int i)
{
  if(i<=9)
  { cout<<i+1;
    f1(i+2);
    cout<<3*i;}
}
```

- a. 89                      b. 821                      c. 8102721                      d. 810

21. Ce se va afișa în urma apelului  $f1(13)$ ?

```
void f1(int k)
{ if(k>3)
  {cout<<k-1;
   f1(k/2);
   cout<<k+2;}}
```

- a. 125815                      b. 122414                      c. 1268                      d. 1255

22. Fie  $k$  o variabilă globală de tip întreg,  $n$  și  $m$  două numere naturale și subprogramul  $p$  alăturat. Știind că variabilei  $k$  i s-a atribuit valoarea 0 exact înaintea apelului  $p(n, m)$ , stabiliți care este valoarea variabilei  $k$  după încheierea executării subprogramului.

```
int p(int n, int m)
{k++;
 if(n==0) return 0;
 else
  return m+p(n-1,m);}
```

- a.  $m$                       b.  $n$                       c.  $n-1$                       d.  $n+1$

23. Se consideră funcția alăturată. Care va fi valoarea returnată de funcție după apelul  $f(5)$ ?

```
int f(int n)
{if(n==1 || n==2) return 1;
 else return f(n-1)+f(n-2);
}
```

- a. 4                      b. 6                      c. 5                      d. 8

Pentru fiecare dintre următoarele enunțuri scrieți programul C++ corespunzător.

1. Șirul de numere 1, 2, 3, 1, 3, 1, 0, 4, ... este definit prin relația alăturată, în care s-a notat cu  $x \% y$  restul împărțirii numărului natural  $x$  la numărul natural  $y$ .

$$f_n = \begin{cases} n, & \text{pentru } n \leq 3 \\ (f_{n-1} * f_{n-2} + f_{n-3} + f_{n-2}) \% n, & \text{pentru } n \geq 4 \end{cases}$$

a) Descrieți în limbaj natural o metodă eficientă (ca timp de executare și spațiu de memorare) prin care se poate determina elementul din șir aflat pe o poziție dată,  $n$ . Explicați în ce constă eficiența metodei alese (4-8 rânduri).

b) Scrieți definiția completă a unui subprogram  $F$  care are un singur parametru,  $n$ , prin intermediul căruia primește o valoare naturală cu cel mult 3 cifre, astfel încât prin instrucțiunea `cout<<F(n);`

să se afișeze valoarea celui de-al  $n$ -lea termen din șir, utilizând algoritmul descris la punctul a).

**Exemplu:** Instrucțiunea `cout<<F(7);` va afișa valoarea 0.

2. Se consideră șirul  $f: 1, 5, 2, 13, 10, \dots$  generat după regula de mai jos, în care s-a notat cu  $[x]$  partea întreagă a numărului  $x$ :

$$f_n = \begin{cases} 1 & \text{pentru } n = 1 \\ 2 * f_{[n/2]} + 3 & \text{pentru } n > 1, n \text{ par} \\ 2 * f_{[n/2]} & \text{pentru } n > 1, n \text{ impar} \end{cases}$$

a) Scrieți un program C/C++ care citește un număr natural  $x$  (cu cel mult 9 cifre) apoi afișează pe ecran mesajul **DA** dacă  $x$  este termen al șirului dat, respectiv mesajul **NU** în caz contrar. Se va utiliza un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** pentru  $x=10$  se va afișa **DA**

b) Descrieți metoda utilizată și explicați în ce constă eficiența ei.

3. Realizați următoarele cerințe utilizând limbajul C sau C++:

a) Scrieți definiția completă a unui subprogram recursiv `sub`, care primește prin intermediul parametrului  $n$  un număr natural cu cel mult 8 cifre și returnează prin intermediul celui de-al doilea parametru `max`, cea mai mare cifră din scrierea lui  $n$ ;

b) Scrieți un program care citește de la tastatură un număr natural  $nr$  cu cel mult 8 cifre și care, folosind subprogramul `sub`, afișează pe ecran cea mai mare cifră a numărului  $nr$  și de câte ori apare aceasta în scrierea sa.

Exemplu, pentru  $nr=26361$ , se afișează: 6 2.

4. Realizați următoarele cerințe utilizând limbajul C sau C++:

a) Scrieți definiția completă a unui subprogram recursiv `sub` care returnează 1 dacă un număr natural mai mic decât 30000, dat prin intermediul parametrului  $nr$ , este prim și returnează 0 în caz contrar.

b) Scrieți un program care citește din fișierul `bac.txt` un număr natural  $n$  ( $n < 1000$ ) și un șir de  $n$  numere naturale mai mici decât 30000, separate prin caracterul spațiu; programul determină și afișează pe ecran secvența de lungime maximă de termeni ai șirului, aflați pe poziții consecutive și care sunt numere prime, separați prin câte un caracter spațiu, folosind apeluri ale subprogramului de la punctul a).

Exemplu: Dacă fișierul `bac.txt` conține valorile: 8 4 7 3 6 11 17 101 45 se afișează 11 17 101

5. Scrieți un subprogram recursiv cu un singur parametru  $n$ , număr natural cu cel mult 9 cifre, care să afișeze pe ecran numai cifrele impare ale lui  $n$ , în ordinea în care ele apar în număr, ca în exemplul de mai jos. Dacă numărul are numai cifre pare, nu se va afișa nimic.

De exemplu pentru  $n=29385567$  se va afișa 93557.

Pentru o implementare iterativă a subprogramului scrisă corect se acordă numai 7p.