

1. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului:

a) `f(121,1);`

b) `f(120,1);`

c) `f(127,1);`

```
void f(long n, int i)
{
    if(n==0) cout<<i<<" ";
    if(n%3>0)
        f(n/3,i+1);
    cout<<i<<" ";
}
```

2. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(125);**?

```
void f(int n)
{
    cout<<n%10<<" ";
    if(n!=0)
    {
        cout<<n%10<<" ";
        f(n/100);
    }
    cout<<endl<<n<<" ";
}
```

3. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(26);**?

```
void f (int x)
{
    if(x>0)
    {
        cout<<'x';
        f(x/3);
        cout<<'y';
    }
    cout<<endl<<x<<" ";
}
```

4. Pentru subprogramul f definit mai jos, ce se afișează ca urmare a apelului f(15,3);?

```
void f (int n, int x)
{
    if (x>n)
        cout<<"*|";
    else
    {
        f(n, x+3);
        cout<<x%10;
    }
}
```

5. Pentru definiția de mai jos a subprogramului f, ce se afișează ca urmare a apelului f(10);?

```
void f (int b)
{
    if (3<=b)
    {
        f(b-2);
        cout<<"* "<<b<<" ";
    }
    else
        cout<<b<<" ";
}
```

6. Pentru definiția de mai jos a subprogramului f, ce se afișează ca urmare a apelului f(123456);?

```
void f(long n)
{
    if (n!=0) {
        if (n%2==1)
            cout<<n%10;
        f(n/10);
        if (n%2!=1)
            cout<<n%10;
    }
    else
        cout<<endl;
}
```

7. Să se implementeze structura repetitivă din algoritmul de mai jos sub forma unei funcții recursive: (x este parametru formal)

```
citește x (număr natural nenul)
cât timp x>0 execută
| citește y (număr natural)
| dacă x>y atunci
|   scrie x%10
| altfel
|   scrie y%10
| ■
| x←y
| ■
```

8. Să se implementeze structura repetitivă din algoritmul de mai jos sub forma unei funcții recursive: (n și s sunt parametri formali iar s valoarea returnată)

```
citește n (număr natural)
s←10
cât timp n>0 execută
| dacă n%10<s atunci
|   s←n%10
| altfel
|   s←-1
| ■
| n←[n/10]
| ■
scrie s
```

9. Să se implementeze structura repetitivă din algoritmul de mai jos sub forma unei funcții recursive: (n și a sunt parametri formali iar a valoarea returnată)

```
citește a,n
(numere naturale)
pentru i←1,n execută
| dacă i%2=0 atunci
|   a←a-i*i
| altfel
|   a←a+i*i
| ■
scrie a
```

10. Să se implementeze structura repetitivă din algoritmul de mai jos sub forma unei funcții recursive: (a,b și c sunt parametri formali iar c valoarea returnată)

```

,
citește a,b
  (numere naturale nenule)
c ← 0
repetă
  i ← a%2
  j ← b%2
  dacă i+j=0 atunci
    c ← c+1
  ■
  a ← a*i + (1-i) * [a/2]
  b ← b*j + (1-j) * [b/2]
până când i*j=1
scrie c

```

11. Să se implementeze structura repetitivă din algoritmul de mai jos sub forma unei funcții recursive: (a,b și c sunt parametri formali). S-a notat cu $y|x$ faptul că x este divizibil cu y.

```

,
citește a,b,c
  (numere naturale nenule)
dacă a>b atunci
  t ← a; a ← b; b ← t
  ■
cât timp a ≤ b execută
  dacă c|a atunci
    scrie a
  ■
  a ← a+1
  ■

```

12. Să se implementeze structura repetitivă din algoritmul de mai jos sub forma unei funcții recursive: (a,n și j sunt parametri formali iar a valoarea returnată)

```

citește a,n
  (numere naturale)
j ← 3
pentru i=1,n execută
  dacă i%2=0 atunci
    a ← a-j
  altfel
    a ← a+j
  ■
  j ← 7-j
  ■
scrie a

```