

## RECURSIVITATE – Fișă de lucru – Execuții tip BACALAUREAT

1. Se consideră subprogramul **ceFace**, definit mai jos:

```
int ceFace (int m, int n)
{
    if(n==0) return 1;
    return 1+ ceFace (n, m%n);
}
```

Valoarea lui **ceFace(210, 49)** este:

- a. 1            b. 2            c. 3            d. 4

2. Se consideră subprogramul **ceFace**, definit mai jos:

```
int ceFace (int a, int b)
{
    if(a>=b) return (a-b)+ ceFace (a/b, b);
    return 1;
}
```

Scrieti două numere naturale din intervalul deschis (20,30), care pot fi memorate în variabilele întregi **x<sub>1</sub>**, respectiv **x<sub>2</sub>**, astfel încât valoarea lui **ceFace(x<sub>1</sub>,3)** să fie **29**, iar valoarea lui **ceFace(3,x<sub>2</sub>)** să fie **1**.

3. Se consideră subprogramele recursive **C1** și **C2**, definite mai jos:

```
int C1 (int a, int b)
{
    if(a==b) return a;
    else if(a>b) return C1(a-b, b);
    else return C1(a, b-a);
}
int C2 (int a, int b)
{
    if(b==0) return a;
    else return C2(b, a%b);
}
```

La apel, returnează valoarea celui mai mare divizor comun al celor două numere naturale nenule primite ca parametri:

- a. numai **C1**    b. numai **C2**    c. atat **C1** cat si **C2**    d. nici **C1**, nici **C2**

4. Scrieti ce se afișează în urma apelului **ceFace (12345)**, pentru subprogramul **ceFace**, definit mai jos:

```
void ceFace (int x)
{
    cout<<"*";
    if(x>0)
    {
        cout<<x;
        ceFace (x/100);
        cout<<"*";
    }
}
```

5. Se consideră subprogramul **ceFace**, definit mai jos. Scrieți ce se afișează în urma apelului de mai jos:  
**ceFace (9)**;

## RECURSIVITATE – Fișă de lucru – Execuții tip BACALAUREAT

```
void ceFace (int x)
{
    cout<<x;
    if(x>0)
    {
        if(x%2==0) cout<<'*';
        ceFace (x/2);
    }
}
```

6. Se consideră subprogramul **ceFace**, definit mai jos. Scrieți valorile **ceFace(3,4)** și **ceFace(2,20)**.

```
int ceFace (int x, int y)
{
    if(x*2>y) return x;
    return ceFace (x*2, y);
}
```

7. Se consideră subprogramele **f1** și **f2**, definite mai jos:

```
int f1 (int x, int y)
{
    return x*y;
}
int f2 (int x, int y)
{
    if (y==1) return x;
    else return x * f2(x, y-1);
}
```

La apel, pentru parametrii **x=2** și **y=3**, returnează produsul **xy**:

- a. atât **f1**, cât și **f2**      b. numai **f1**      c. numai **f2**      d. nici **f1**, nici **f2**

8. Se consideră subprogramul **ceFace**, definit mai jos. Ce se afișează în urma apelului **ceFace (3)**?

```
void ceFace (int n)
{
    if (n!=0)
    {
        ceFace (n-1);
        cout<<n;
    }
}
```

- a. 12      b. 123      c. 321      d. 3210

9. Se consideră subprogramul **ceFace**, definit mai jos. Indicați ce valoare are **ceFace (15)**.

```
int ceFace (int n)
{
    if (n<10) return ceFace (n+1) + 3;
    else if (n==10) return 7;
    else return ceFace (n-2) - 1;
}
```

- a. 1      b. 7      c. 8      d. 10

## RECURSIVITATE – Fișă de lucru – Execuții tip BACALAUREAT

10. Se consideră subprogramul **ceFace**, definit mai jos. Scrieți două valori naturale distincte din intervalul **[1,50]** pe care le poate avea variabila întreagă **x**, astfel încât **ceFace(30,x)** să aibă valoarea **5**.

```
int ceFace (inta, int b)
{
    if (b==0) return a;
    else return ceFace(b, a%b);
}
```

11. Se consideră subprogramul **f**, definit mai jos. Scrieți valorile **ceFace(5,5)** și **ceFace(10,21)**.

```
int ceFace(int a, int b)
{
    if (a==b) return 0;
    if (b/a==0) return a+b;
    return ceFace (a+2, b-3);
}
```

12. Subprogramul **ceFace** este definit mai jos. Scrieți instrucțiunea prin care se poate apela subprogramul pentru a afișa, în ordine strict descrescătoare, toți divizorii pozitivi proprii ai numărului **2015** (divizori naturali diferiți de **1** și de **2015**).

```
void ceFace (int n, int d)
{
    if (d<n/2) ceFace (n, d+1);
    if (n%d==0) cout<<d<<' ';
}
```

13. Subprogramul **ceFace** este definit mai jos. Scrieți ce se afișează în urma apelului **ceFace ('d')**;

```
void ceFace (char c)
{
    if(c>='a')
    {
        cout<<c;
        ceFace (c-1);
    }
}
```

14. Subprogramul **ceFace** este definit mai jos. Scrieți ce valoare are **ceFace(105,42)**.

```
int ceFace(int x, int y)
{
    if(x==y) return 1;
    else {    if(x<y)    {
                x=x+y; y=x-y; x=x-y;
            }
            return 1+ ceFace (x-y,y);
        }
}
```

## RECURSIVITATE – Fișă de lucru – Execuții tip BACALAUREAT

15. Subprogramul **ceFace** este definit mai jos. Scrieți ce se afișează în urma apelului **ceFace (154678,3)**;

```
void ceFace (long a, int b)
{
    if(a*b!=0)
    if(a%2==0)
    {
        cout<<a%10;
        ceFace (a/10, b-1);
    }
    else
    {
        ceFace (a/10, b+1);
        cout<<a%10;
    }
}
```

16. Subprogramul **ceFace** este definit mai jos. Indicați cea mai mare valoare din intervalul [20,25] pe care o poate avea variabila întregă x, astfel încât, în urma apelului **ceFace (x,1)**, să se afișeze numerele 2 4 16.

```
void ceFace (int n, int p)
{
    if(n>0)
    {
        if(n%2==1) cout<<p<<" ";
        ceFace (n/2, p*2);
    }
}
```

a. 21            b. 22            c. 23            d. 24

17. Subprogramul **ceFace** este definit mai jos. Indicați ce se afișează în urma apelului **ceFace(2016)**;

```
void ceFace (int n)
{
    if (n!=0)
    {
        cout<<n;
        ceFace (n/10);
    }
}
```

a. 6102            b. 2202012016            c. 2016201202            d. 20162012020