

1. Pentru definiția alăturată a subprogramului `f`, stabiliți ce valoare are `f(23461)`. (4p.)
- ```

int f(int x)
{ if(x<10){if(x%2!=0) return 0;
 return x;
}
 if(x%2!=0)
 return f(x/10);
 return f(x/10)+x%10; }

```
2. Pentru definiția de mai jos a subprogramului `f`, ce se afișează ca urmare a apelului `f(121,1)`? (6p)
- ```

//C
void f(long n, int i)
{ if(n!=0)
  if(n%3>0)
    { printf("%d",i); f(n/3,i+1); }
}

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

```
3. Pentru definiția de mai jos a subprogramului `f`, ce se afișează ca urmare a apelului `f(12345)`? (6p.)
- ```

//C
void f(long n)
{ printf("%d",n%10);
 if(n!=0)
 { f(n/100); printf("%d",n%10); }
}

//C++
void f(long n)
{ cout<<n%10;
 if(n!=0)
 { f(n/100); cout<<n%10;}
}

```
4. Pentru definiția alăturată a subprogramului `f`, ce se afișează ca urmare a apelului `f(26)`? (6p.)
- ```

void f (int x)
{
  if(x>0)
  if(x%4==0)
    { cout<<'x'; | printf("%c",'x');
      f(x-1); }
  else
  { f(x/3);
    cout<<'y'; | printf("%c",'y');
  }
}

```
5. Pentru subprogramul `f` definit mai jos, ce se afișează ca urmare a apelului `f(3,17)`? (6p.)
- ```

//C
void f (int a, int b)
{ if(a<=b)
 {f(a+1,b-2); printf("%c",'*');}
 else printf("%d",b);
}

//C++
void f (int a, int b)
{ if(a<=b)
 { f(a+1,b-2); cout<<'*';}
 else cout<<b;
}

```
6. Se consideră subprogramul `f` definit alăturat. Ce se va afișa în urma apelului `f(12345)`? (4p.)
- ```

void f(long int n)
{ if (n!=0)
  {if (n%2 == 0)
    cout<<n%10; | printf("%d",n%10);
    f(n/10);
  }
}

```
7. Pentru definiția alăturată a subprogramului `f`, ce se afișează ca urmare a apelului `f(15,2)`? (6p.)
- ```

void f (int n, int x)
{ if(x>n)
 cout<<0; | printf("%d",0);
 else
 if(x%4<=1) f(n,x+1);
 else
 { f(n,x+3);
 cout<<1; | printf("%d",1);
 }
}

```
- 8.

Se consideră subprogramul alăturat:  
Ce valoare are  $f(128, 2)$ ?

(6p.)

```
int f(int a, int b){
 if (b<1) return -1;
 else
 if (a%b==0)
 return 1+f(a/b,b);
 else
 return 0; }
```

9.

Se consideră subprogramul  $f$ , descris alăturat. Ce se va afișa în urma apelului  $f(3)$ ? (6p.)

```
void f(int n)
{ if (n!=0)
 { if (n%2==0)
 cout<<n<<' '; | printf("%d ",n);
 f(n-1);
 cout<<n<<' '; | printf("%d ",n);
 }
 else cout<<endl; | printf("\n");
}
```

10.

1. Se consideră subprogramul  $f$  cu definiția alăturată. Ce valoare are  $f(1213111, 1)$ ?

(4p.)

```
int f (long int n, int k){
 if (n!=0)
 if (n%10==k)
 return 1+f(n/10,k);
 else return 0;
 else return 0;}
```

a. 5

b. 3

c. 2

d. 1

11.

1. Se consideră subprogramul cu definiția alăturată. Ce valoare are  $f(3, 1)$ ?

(4p.)

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

a. 9

b. 6

c. 7

d. 8

12.

2. Care este valoarea expresiei  $f(23159)$  pentru funcția  $f$ , cu definiția alăturată? (6p.)

```
int f(int n){
 int c;
 if (n==0) return 9;
 else
 {c=f(n/10);
 if (n%10<c) return n%10;
 else return c;
 }
}
```

13.

1. Se consideră subprogramul  $P$ , definit alăturat. Știind că valoarea variabilei întregi  $a$  este înainte de apel 4, care este valoarea ei după revenirea din apelul  $P(a)$ ?

```
void P(int &x)
{ x=x+5; }
```

(4p.)

a. 10

b. 4

c. 9

d. 5

14.

1. Fie subprogramul  $fct$  definit alăturat, parțial. Inițial, variabilele întregi  $a$ ,  $b$  și  $c$  au valorile  $a=8$ ,  $b=31$  și  $c=9$ , iar după apelul  $fct(a, b, c)$ , valorile celor trei variabile sunt  $a=9$ ,  $b=31$  și  $c=39$ . Care poate fi antetul subprogramului  $fct$ ?

```
void fct(....)
{ x=x+1;
 y=y-1;
 z=x+y;
}
```

(4p.)

a. void fct(int &x,int &y,int &z)

b. void fct(int x,int &y,int &z)

c. void fct(int x,int y,int z)

d. void fct(int &x,int y,int &z)

15.

2. Pentru funcțiile `f1` și `f2` definite alăturat, stabiliți care este valoarea lui `f2(41382)`? (6p.)
- ```

long f1(int c)
{ if (c%2==1) return 1;
  else return 2;
}

long f2(long n)
{ if (n==0) return 0;
  else return f1(n%10)+f2(n/10);
}

```
16. 2. Pentru funcțiile `f` și `g` definite mai jos, stabiliți care este rezultatul returnat la apelul `f(6)`? (6p.)
- ```

long g(long x)
{ if (x>9)
 return (x/10 + x%10);
 else
 return x;
}

long f(int c)
{ if (c<1)
 return 1;
 else
 return g(c+f(c-1));
}

```
1. Funcția `F` are definiția alăturată. Ce valoare are `F(3)`? (4p.)
- ```

int F(int n)
{if(n==0 || n==1) return 1;
  else
  return 2*F(n-1)+2*F(n-2);}

```
- a. 1 b. 12 c. 6 d. 10
17. 1. Subprogramul `f` are definiția alăturată. Ce se va afișa în urma apelului `f(12345)`? (4p.)
- ```

void f(long n)
{if (n>9)
 {cout<<n/100; | printf("%d",n/100);
 f(n/10);
}
}

```
- a. 1231210              b. 123121              c. 1234123121              d. 123
18. 1. Funcția `F` are definiția alăturată. Ce valoare are `F(18)`? (4p.)
- ```

int F(int x){
  if (x<=1) return x;
  else return x+F(x-2);
}

```
- a. 90 b. 171 c. 1 d. 18
19. 1. Funcția `F` are definiția alăturată. Ce valoare are `F(5)`? (4p.)
- ```

int F(int x)
{if(x!=0) return x+F(x-1);
 else
 return x;
}

```
- a. 5                      b. 10                      c. 15                      d. 6
- Se consideră subprogramul, `f`, definit alăturat. Ce valoare are `f(100)`? (6p.)
- ```

int f(int n)
{ if(n==0) return 0;
  else return n%2+f(n/2);
}

```
20. 1. Considerăm subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului de mai jos? (4p.)
- ```

void f(char c)
{ if (c>'A') f(c-1);
 cout<<c; | printf("%c",c);
 if (c>'A') f(c-1);
}

```
- `f('C');`
21. 2. Pentru subprogramul `suma` definit alăturat, scrieți valoarea expresiei `suma(5,4)`. (4p.)
- ```

int suma (int a,int b)
{ if (a==0 && b==0) return 0;
  else if (a==0) return 1+suma(a,b-1);
  else return 1+suma(a-1,b);
}

```
22. 2. Funcția `f` are definiția alăturată. (3p.)
- ```

int f(int n)
{if (n<=9) return 0;
 if (n%4==0) return 0;
 return 1+f(n-3);
}

```
- a) Ce valoare are `f(17)`? (3p.)
- b) Ce valoare are `f(22)`? (3p.)
- 23.

2. Funcția  $f$  are definiția alăturată:
- a) Ce valoare are  $f(16)$ ? (3p.)
- b) Scrieți cea mai mare valoare de două cifre pe care o poate avea  $n$  astfel încât  $f(n)$  să fie egal cu 2. (3p.)
- ```

int f(int n)
{ if (n<=0) return -1;
  if (n%2==0) return 0;
  if (n%3==0) return 0;
  return 1+f(n-10);
}

```
24. Subprogramul `afis` este definit alăturat. Ce se afișează ca urmare a apelului `afis(8)`? (4p.)
- ```

void afis (int n)
{ cout<<n; | printf("%d",n);
 for (int i=n/2;i>=1;i--)
 if (n%i==0) afis (i);
}

```
25. Subprogramul `scrie` este definit alăturat. Ce se afișează ca urmare a apelului `scrie(1,7)`? (6p.)
- ```

void scrie (int x,int y)
{cout<<x<<y; | printf("%d%d",x,y);
  if (x<y)
  {scrie(x+1,y-1);
   cout<<(x+y)/2; | printf("%d", (x+y)/2);
  }
}

```
26. Subprogramul f este definit alăturat. Ce valoare are $f(8,4)$? (4p.)
- ```

int f (int x,int y)
{ if (x<y) return 1+f(x+1,y);
 if (y<x) return 1+f(y+1,x);
 return 1;
}

```
27. Ce se afișează ca urmare a apelului `p(123)`; dacă subprogramul `p` are definiția alăturată? (6p.)
- ```

void p (int x)
{cout<<x; | printf("%d",x);
  if (x!=0) {p(x/10);
   cout<<x%10; | printf("%d",x%10);}}

```
28. 2. Subprogramul f este definit alăturat. Ce se afișează ca urmare a apelului `f(1,4)`? (6p.)
- ```

void f (int x,int y)
{int i;
 for (i=x;i<=y;i++)
 { cout<<i; | printf("%d",i);
 f(i+1,y);
 }
}

```
29. 2. Se consideră subprogramul recursiv `f1` definit alăturat. Ce se va afișa în urma apelului `f1(5)`? (6p.)
- ```

void f1(int x)
{ if (x<=9)
  { cout<<x+1; | printf("%d",x+1);
    f1(x+2);
    cout<<x+3; | printf("%d",x+3);
  }
}

```
30. 2. Subprogramul `afis` este definit alăturat. Ce se va afișa în urma apelului `afis(17)`? (6p.)
- ```

void afis(int x)
{ if (x>3)
 { cout<<x-1; | printf("%d",x-1);
 afis(x/3);
 cout<<x+1; | printf("%d",x+1);
 }
}

```
31. 2. Subprogramul `re` este definit alăturat. Ce valoare are `re(1)`? Dar `re(14)`? (6p.)
- ```

int re(int i)
{
  if (i<9) return 3+re(i+2);
  else
  if (i==9) return -2;
  else return 1+re(i-1);
}

```
32. 2. Se consideră definit subprogramul f . Scrieți două valori naturale, x_1 și x_2 ($x_1 \neq x_2$, $x_1 < 12$ și $x_2 < 12$) pentru care $f(x_1) = f(x_2)$. (6p.)
- ```

int f(int i)
{
 if (i>12) return 1;
 else return 1+f(i+2);
}

```
- 33.

2. Subprogramul `f` este definit alăturat. Ce se va afișa în urma executării secvenței de mai jos?  
`a=3; b=9; f(b, a); f(b, b);`  
 (6p.)
- ```
void f(int &a, int b)
{
  a=a-5; b=b-a-2;
  cout<<a<<b; | printf("%d%d", a, b);
}
```
34. 2. Considerăm subprogramul `f` definit alăturat. Ce valoare are `f(11, 7)`?
 (6p.)
- ```
int f(int x, int y)
{
 if(x<=y) return x-y;
 return f(y-x, x-1)+3;
}
```
35. 2. Pentru definiția alăturată a subprogramului `sc`, stabiliți ce valoare are `sc(901324)`.  
 (6p.)
- ```
int sc(long x)
{
  if(x<10) return x;
  return sc(x/10)+x%10;
}
```
36. Subprogramul `f` are definiția alăturată. Ce valoare are `f(7, 2)`? Dar `f(35, 2)`?
 (6p.)
- ```
int f(int x, int y)
{
 if(x%y==0) return y;
 else return f(x, y+1);
}
```
37. Subprogramul `f` are definiția alăturată. Ce valoare are `f(7)`? Dar `f(100)`?  
 (6p.)
- ```
int f(int x)
{
  if(x%6==0) return x;
  else return f(x-1);
}
```
38. Subprogramul `f` are definiția alăturată. Ce valoare are `f(3)`? Dar `f(10)`?
 (6p.)
- ```
int f(int x)
{
 if(x==0) return 0;
 else return f(x-1)+2;
}
```
39. Subprogramul `f` are definiția alăturată. Ce valoare are `f(4)`? Dar `f(11)`?  
 (6p.)
- ```
int f(int x)
{
  if(x<1) return 1;
  else return f(x-3)+1;
}
```
40. Subprogramul `f` are definiția alăturată. Ce valoare are `f(5, 10)`?
 (6p.)
- ```
int f(int x, int y)
{
 if(x==y) return x;
 else if(x<y) return f(x+1, y-1);
 else return f(x-1, y);
}
```
41. 2. Se consideră subprogramul `f`, definit alăturat. Câte valori ale lui `n` aparținând intervalului `[10, 20]` au proprietatea că la apelul `f(n)` se obține valoarea 0?  
 (6p.)
- ```
int f(unsigned int n)
{
  if (n==0) return 0;
  else if(n%2==0) return n%10+f(n/10);
  else return f(n/10);
}
```
42. 2. Se consideră subprogramul `f`, definit alăturat. Câte valori ale parametrului `n` aparținând intervalului `[1, 100]` au proprietatea că la apelul `f(n)` se obține o valoare nenulă?
 (6p.)
- ```
int f(unsigned int n)
{
 if (n>20) return 0;
 else return 5+f(n+5);
}
```
43. Funcția `f` are definiția alăturată. Dacă `f(x)` are valoarea 10100, care este valoarea lui `x`?  
 (6p.)
- ```
long f(int n)
{
  if(n<=0) return 0;
  else return f(n-1)+2*n;
}
```

1. Valorile memorate de componentele tabloului unidimensional v , cu indicii de la 0 la 5, sunt, în această ordine: 420, 48, 635, 628, 837, 93. Care este apelul corect al subprogramului `tablou` alăturat pentru ca, în urma executării apelului, valoarea variabilei întregi x să fie 3? (4p.)
- ```
int tablou (int i,int v[])
{int n;
 if(i<0) return 0;
 else {
 n=v[i];
 while(n) {
 if(n%2)
 return 1+tablou(i-1,v);
 n=n/10;
 }
 return tablou(i-1,v);
 }
}
```
- a. `x=tablou(3,v);`  
b. `x=tablou(4,v);`  
c. `x=tablou(5,v);`  
d. `x=tablou(0,v);`
44. 1. Valorile memorate de componentele vectorului  $v$ , cu indicii de la 0 la 5, sunt, în această ordine: 973, 51, 871, 350, 691, 15. Care este apelul corect al subprogramului `tablou` alăturat pentru ca, în urma executării apelului, valoarea variabilei întregi  $x$  să fie 3? (4p.)
- ```
int tablou (int i,int v[], int c)
{int n;
 if(i<0) return 0;
 else {
  n=v[i];
  while(n) {
   if(n%10==c)
    return 1+tablou(i-1,v,c);
   n=n/10;
  }
  return tablou(i-1,v,c);
 }
}
```
- a. `x=tablou(4,v,5);`
b. `x=tablou(5,v,5);`
c. `x=tablou(5,v,1);`
d. `x=tablou(0,v,4);`
45. 2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(21114,1)$? (6p.)
- ```
int f(int n,int c)
{
 if(n==0) return 0;
 if(n%10==c) return n%100+f(n/10,c);
 return f(n/10,c);
}
```
- a. `x=tablou(4,v,5);`  
b. `x=tablou(5,v,5);`  
c. `x=tablou(5,v,1);`  
d. `x=tablou(0,v,4);`
46. 2. Se consideră subprogramul  $f$ , definit alăturat. Ce valoare are  $f(3713,3)$ ? (6p.)
- ```
int f(int n,int c){
 if(n==0)
 return 0;
 if(n%10==c)
 return f(n/10,c)*10+c;
 return f(n/10,c);
}
```
- a. `x=tablou(4,v,5);`
b. `x=tablou(5,v,5);`
c. `x=tablou(5,v,1);`
d. `x=tablou(0,v,4);`
47. 2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(261,31)$? (6p.)
- ```
int f(int a,int b)
{
 if(a<10)
 return b;
 return f(a/10,b)*10+b+1;
}
```
- a. `x=tablou(4,v,5);`  
b. `x=tablou(5,v,5);`  
c. `x=tablou(5,v,1);`  
d. `x=tablou(0,v,4);`
48. 2. Se consideră tabloul unidimensional  $a$  definit global, ce memorează elementele  $a_1=12$ ,  $a_2=35$ ,  $a_3=2$ ,  $a_4=8$  și subprogramul  $f$ , definit alăturat. Ce valoare are  $f(4)$ ? (6p.)
- ```
int f(int x)
{if(x>=1)
 return (a[x]+f(x-1))%10;
 else
 return 0;}
```
- a. `x=tablou(4,v,5);`
b. `x=tablou(5,v,5);`
c. `x=tablou(5,v,1);`
d. `x=tablou(0,v,4);`
49. 2. Se consideră subprogramul f , definit alăturat. Ce se va afișa în urma apelului $f(5,0)$? (6p.)
- ```
void f(int i,int j)
{ if(j<=9) f(i,j+1);
 printf("%d*%d=%d\n",i,j,i*j);/
 cout<<i<<' '*<<j<<'='<<i*j<<endl;
}
```

50. 2. Se consideră subprogramul  $f$ , definit alăturat. Ce valoare are  $f(398, 2008)$ ? (6p.)
- ```
int f(int a,int b){
    if(2*a>=b)
        return 0;
    if(b%a==0)
        return b-a;
    return f(a+1,b-1);
}
```
51. 2. Considerăm subprogramul f , definit alăturat. Care va fi valoarea variabilei globale x după apelul $f(4962, x)$, dacă înainte de apel, x are valoarea 0? (6p.)
- ```
void f(int n,int &a)
{int c;
 if(n!=0){c=n%10;
 if(a<c) a=c;
 f(n/10,a);
 }
}
```
52. 2. Se consideră tabloul unidimensional  $a$  definit global, ce memorează elementele  $a_1=1, a_2=2, a_3=0$  și subprogramul  $f$ , definit alăturat. Ce valoare are  $f(3, 3)$ ? (6p.)
- ```
int f(int b,int i)
{ if(i>=1)
  return f(b,i-1)*b+a[i];
  else return 0;}

```
53. Valorile memorate de componentele tabloului unidimensional v , cu indicii de la 0 la 5, sunt, în această ordine: 183, 212, 453, 18, 42, 83. Care dintre următoarele apeluri ale subprogramului tab , cu definiția alăturată, **NU** returnează un număr par? (4p.)
- ```
int tab (int i,int v[], int c)
{ int n;
 if(i<0) return 0;
 else
 { n=v[i];
 while(n!=0)
 { if(n%10==c%10)
 return 1+tab(i-1,v,c%10);
 n=n/10;
 }
 }
 return tab(i-1,v,c%10);
}

```
- a.  $tab(4, v, 218)$     b.  $tab(5, v, 72)$     c.  $tab(4, v, 123)$     d.  $tab(5, v, 71)$
54. Valorile memorate de componentele tabloului  $v$ , cu indicii de la 0 la 5, sunt, în această ordine: 183, 212, 453, 18, 42, 83. Care este apelul corect al subprogramului  $t$  alăturat pentru ca, în urma executării apelului, valoarea variabilei întregi  $x$  să fie 2? (4p.)
- ```
int t (int i,int v[], int c)
{int n;
  if(i<0) return 0;
  else
  { n=v[i];
    while(n>9)
    { if(n%10==c)
      return 1+t(i-1,v,c);
      n=n/10;
    }
  }
  return t(i-1,v,c);
}

```
- a. $x=t(2, v, 2)$; b. $x=t(2, v, 5)$; c. $x=t(5, v, 2)$; d. $x=t(2, v, 8)$;