

Bacalaureat 2007

7. Pentru definiția alăturată a subprogramului **ex**, stabiliți ce se afișează la apelul **ex(120)** ?
- ```
void ex(int x)
{ if(x!=0) {
 printf("%d",x%10); / cout<<x%10;
 ex(x/10);
 }
}
```
- a. 021                      b. 012                      c. 120                      d. 21
8. Pentru definiția alăturată a subprogramului **ex**, stabiliți ce se afișează la apelul **ex(120)** ?
- ```
void ex(int x)
{ printf("%d",x%10); /cout<<x%10;
  if(x!=0) ex(x/10);
}
```
- a. 0120 b. 021 c. 210 d. 0210
1. Pentru un vector **x** declarat global, vector cu 10 componente întregi, stabiliți care sunt valorile memorate de componentele tabloului **x** în urma apelului **ex(0,9)**.
- ```
void ex(int i,int j)
{ if(i<=j)
 { x[i]=i;x[j]=j; ex(i+1,j-1);
 if(i%2!=0) {x[i]=j;x[j]=9-x[i];}
 }
}
```
- a. (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)                      b. (1, 9, 3, 7, 5, 6, 4, 8, 2, 10)  
c. (0, 8, 2, 6, 4, 5, 3, 7, 1, 9)                      d. (9, 1, 7, 3, 5, 4, 6, 2, 8, 0)
8. Pentru definiția alăturată a subprogramului **ex**, stabiliți ce se afișează la apelul **ex(2,10)** ?
- ```
void ex(int i,int j)
{ if(i<=j) {
    printf("%d",i); / cout<<i;
    ex(i+1,j-1);
    if(i%2!=0)
      printf("%d",9-j); / cout<<9-j;
  }
}
```
- a. 2345620 b. 246357 c. 2345646 d. 234520
6. 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(10542,1821,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
7. Pentru definiția alăturată a subprogramului **f**, stabiliți ce valoare returnează apelul **f(30,4)** ?
- ```
int f(int x,int y)
{
  if (x<y) return 0;
  else return y+f(x-y,y);
}
```
- a. 28 b. 30 c. 32 d. 7
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
8. 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

6. 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:
- a. 10 b. 4 c. 3 d. 5
3. Ce valoare va returna $f(23951)$, pentru funcția f definită alăturat?
- a. 2 b. 3 c. 5 d. 9
1. Ce valoare trebuie transmisă prin intermediul parametrului n la apelul funcției f alăturate, astfel încât valoarea returnată de funcție să fie 7?
- a. 3258 b. 1528 c. 3972 d. 3472
6. Pentru funcțiile $f1$ și $f2$ definite alăturat, stabiliți care este rezultatul returnat la apelul $f2(305)$?
- a. 126 b. 3 c. 127 d. 8
6. Pentru funcțiile $f1$ și $f2$ definite mai jos, stabiliți care este rezultatul returnat la apelul $f1(6)$?
- ```
long f2(long x){ if (x%2==0) return f2(x/2); else return x; }
```
- ```
long f1(int c) { if (c<1) return 1; else return f2(c*f1(c-1)); }
```
- a. 720 b. 16 c. 45 d. 360
5. 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
5. 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
5. 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
8. Care este rezultatul tipărit la apelul  $f(\text{"examenul"}, 0)$ ?
- ```
void f(char s[256],int i)
{if (i<strlen(s))
{f(s,i+1);
if (strchr("aeiou",s[i])!=0)
cout<<s[i]; | printf("%c",s[i]);
}
}
```
- a. eaeu b. examenul c. ueae d. e
3. Se consideră subprogramul F , definit alăturat. Care este valoarea expresiei $F(11)$?
- ```
int F(int x){
if(x%2==0) return x/2;
else return F(F(x-1));}
```
- a. 2                      b. 5                      c. 1                      d. 4

6. Știind că  $p$  este un vector cu 3 componente întregi (vector declarat global),  $M$  este mulțimea tuturor cifrelor nenule, iar subprogramul `tipar` afișează valorile elementelor  $p[0]$ ,  $p[1]$  și  $p[2]$ , cu ce trebuie înlocuite simbolurile  $\alpha$ ,  $\beta$  și  $\gamma$  în definiția funcției  $G$  astfel încât în urma apelului  $G(0)$  să se afișeze toate elementele produsului cartezian  $M \times M \times M$ ?
- a.  $\alpha=0 \ \beta=10 \ \gamma=3$       b.  $\alpha=1 \ \beta=3 \ \gamma=9$       c.  $\alpha=1 \ \beta=9 \ \gamma=3$       d.  $\alpha=1 \ \beta=9 \ \gamma=2$
4. Se consideră subprogramul  $F$  definit alăturat. De câte ori se afișează valoarea 0 dacă se apelează  $F(9)$ ?
- a. 9      b. 2      c. 3      d. 12
7. Fie subprogramul definit alăturat. Ce se afișează în urma apelului  $P(3)$ ?
- a. 3      b. 123      c. 3210      d. 0123
1. 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; | printf("%d",x);
        P(x/2);}
    else{
        P(x-1);
        cout<<x; | printf("%d",x);}
}
```
- a. 321 b. 231 c. 213 d. 123
1. Pentru funcția recursivă alăturată stabiliți care este valoarea returnată în urma apelului $f(4)$?
- a. 16 b. 15 c. 4 d. 11
6. Câte caractere * se afișează în urma apelului `stea(3)`?
- ```
void stea (int x)
{int i;
 if(x>0){stea(x-1);
 for(i=1;i<=x;i++)
 printf("*"); | cout<<"*";
 }}
```
- a. 6      b. 3      c. 9      d. 12
5. 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){
    printf("%d",i); | cout<<i;
    f(i+1,j);}
  else {
    printf("%d",j); | cout<<j;
    f(i,j-1);}
}
```
- a. 54132 b. 12345 c. 51423 d. 54321
7. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul $f(5, 1)$.
- ```
void f(int i, int k)
{if(k<=4){
 printf("%d",i*k); | cout<<i*k;
 f(i-1,k+1);
}}
```
- a. 1248      b. 5898      c. 1234      d. 5488

7. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul `f(5,1)`.
- ```
void f(int n, int k)
{ if (k <= n) {
    printf("%d", n-k); | cout << n-k;
    f(n, k+1);
  } }
```
- a. 12345 b. 01234 c. 43210 d. 54321
6. Se consideră funcția alăturată `f1`. În urma apelului `f1(7)`, se va afișa:
- ```
void f1(int i)
{
 if (i <= 9)
 { cout << i+1; / printf("%d", i+1)
 f1(i+2);
 cout << 3*i; / printf("%d", 3*i)
 } }
```
- a. 89                      b. 821  
c. 8102721                      d. 810
2. Ce se va afișa în urma apelului `f1(13)`?
- ```
void f1(int k)
{ if (k > 3)
{ cout << k-1; / printf("%d", k-1);
  f1(k/2);
  cout << k+2; / printf("%d", k+2); } }
```
- a. 125815 b. 122414 c. 1268 d. 1255
2. 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`
3. 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
5. 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 (m==0) return 0;
 else return n + p(n,m-1);
}
```
- a. `m`                      b. `m-1`                      c. `m+1`                      d. `n`
5. Considerând funcția recursivă alăturată, precizați ce se va afișa pe ecran în urma apelului `f(2)`?
- ```
void f(int a)
{ int i;
  printf("%d", a); | cout << a;
  for (i=0; i<a; i++) f(i); }
```
- a. 2011 b. 1010 c. 2100 d. 2010
4. Fie subprogramul `f` definit alăturat. În urma executării secvenței `int x=1; f(x);` se vor afișa valorile:
- ```
void f(int & b)
{ b=b+1; if (b<4) f(b);
 printf("%d", b); | cout << b;
}
```
- a. 432                      b. 234                      c. 111                      d. 444
3. Se consideră subprogramul `numar` având definiția alăturată. Care este valoarea returnată de funcție pentru apelul `numar(5)`?
- ```
int numar(int n)
{
  if (n <= 0) return 0;
  else return 2*n+numar(n-1);
}
```
- a. 15 b. 30 c. 10 d. 20

4. Care dintre următoarele subprograme afișează în ordine inversă (începând cu cifra unităților) cifrele unui număr natural, primit ca parametru?
- `void numar(int a)
{if(a>9)numar(a%10);
cout<<a/10; | printf("%d",a/10);}`
 - `void numar(int a)
{cout<<a%10; | printf("%d",a%10);
if(a>9)numar(a/10);}`
 - `void numar(int a)
{cout<<a/10; | printf("%d",a/10);
if(a>9)numar(a%10);}`
 - `void numar(int a)
{if(a>9)numar(a/10);
cout<<a%10; | printf("%d",a%10);}`
5. Fie funcția `numara` prezentată mai jos:
- ```
int numara(int x,int y)
{if (y==0) return 0;
 else if (x%y==0) return numara(x,y-1)+1;
 else return numara(x,y-1);}
```
- Care este apelul corect al funcției `numara` pentru a verifica dacă un număr natural `n` este prim?
- `if (numara(n,n)==2)`  
`cout<<"prim"; | printf("prim");`
  - `if (numara(2,n)==2)`  
`cout<<"prim"; | printf("prim");`
  - `if (numara(n,2)==0)`  
`cout<<"prim"; | printf("prim");`
  - `if (numara(n,n/2)==2)`  
`cout<<"prim"; | printf("prim");`
3. Se consideră subprogramul `f` având definiția alăturată. Care este valoarea returnată de subprogramul `f` după apelul: `f(10,2)`?
- ```
int f(int x, int y)
{
  if (x*y<=0) return 0;
  else return 1 + f(x-y,y);
}
```
- 3
 - 5
 - 1
 - 4
5. Pentru subprogramul alăturat, apelul `func(4)` are ca rezultat:
- ```
int func(int x) {
 if(x<=0) return -1;
 else return 1/func(x-1)-2*func(x-2);
}
```
- 1
  - 5
  - 6
  - 0
8. Fie vectorul `v` cu indicii de la 0 la 7, vector ce reține, în ordine, următoarele valori întregi: 88, 777, 789, 976, 998, 7856, 7887, 8979. Care este apelul corect al subprogramului alăturat astfel încât valoarea returnată să fie 5?
- ```
int f(int poz,int v[],int c)
{ if(poz<0) return c;
  else { int nr=v[poz];
        while(nr) {
          if(nr%10<c)
            c=nr%10;
          nr=nr/10; }
        return f(poz-1,v,c); }
}
```
- `printf("%d",f(5,v,9)) | cout<<f(5,v,9)`
 - `printf("%d",f(5,v,0)) | cout<<f(5,v,0)`
 - `printf("%d",f(4,v,9)) | cout<<f(4,v,9)`
 - `printf("%d",f(3,v,9)) | cout<<f(3,v,9)`
2. Fie subprogramul cu definiția alăturată. Care dintre următoarele instrucțiuni determină afișarea valorii -7?
- ```
int f (int n)
{
 if(n<=0) return -1;
 else
 return f(n-1)-2*f(n-2)+3*f(n-3);
}
```
- `printf("%d",f(3)) | cout<<f(3)`
  - `printf("%d",f(6)) | cout<<f(6)`
  - `printf("%d",f(5)) | cout<<f(5)`
  - `printf("%d",f(4)) | cout<<f(4)`

1. Fie subprogramul cu definiția alăturată. Care dintre următoarele instrucțiuni determină afișarea valorii -9?
- ```
int f (int n)
{
    if(n<=0) return -1;
    else
        return f(n-1)-2*f(n-2)+3*f(n-3);
}
```
- a. `printf("%d",f(4)) | cout<<f(4)` b. `printf("%d",f(6)) | cout<<f(6)`
c. `printf("%d",f(5)) | cout<<f(5)` d. `printf("%d",f(7)) | cout<<f(7)`
4. Se consideră următorul subprogram recursiv:
- ```
void f(int x)
{if(x>0) {
 f(x/10);
 if(x%10%2!=0)
 cout<<x%10; | printf("%d",x%10);
 }
}
```
- Ce se va afișa în urma apelului `f(123)`?
- a. 12      b. 123      c. 13      d. 31
7. Se consideră următorul subprogram recursiv:
- ```
void f(int x)
{if(x>8) f(x/8);
cout<<x%8; | printf("%d",x%8) ;}
```
- Ce se va afișa în urma apelului `f(1234)`?
- a. 2300 b. 2322 c. 123 d. 2388
3. Se consideră următorul subprogram recursiv:
- ```
void p(int i,int n)
{int j;
if(i<=n)
{p(i+1,n);
for(j=1;j<=i;j++)
cout<<j; | printf("%d",j);
cout<<endl; | printf("\n");
}}}
```
- Ce se va afișa în urma apelului `p(1,3)`?
- a. 123      b. 123      c. 321      d. 1  
23      12      32      12  
3      1      3      123
6. Se consideră subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului `f(1,3)`?
- ```
void f(int i,int n)
{if(i<=n)
{cout<<"*"; | printf("*");
f(i+1,n);
cout<<"#"; | printf("#");}
}
```
- a. ##### b. ##### c. ##### d. #####
8. Se consideră următorul subprogram recursiv:
- ```
long f(int x,int n)
{if(n==0) return 1;
else if (n%2==0) return f(x,n/2)*f(x,n/2);
else return x*f(x,n-1);}
```
- Ce valoare va întoarce subprogramul după apelul `f(2,10)`?
- a.  $10^2$       b.  $2^{10}$       c. 20      d.  $2^9$
2. Pentru definiția alăturată a subprogramului `f`, ce se va afișa în urma apelului `f(14663)`?
- ```
int f(int n)
{
    if(n==0) return 0;
    if(n%2) return 1+f(n/10);
    return f(n/10);
}
```
- a. 5 b. 0 c. 2 d. 3

1. Pentru definiția alăturată a subprogramului f , ce valoare se va returna la apelul $f(14625)$?
- ```
int f(int n)
{
 if(n==0) return -1;
 int m=f(n/10);
 if(n%2||m>n%10) return m;
 return n%10;
}
```
- a. -1                      b. 2                      c. 6                      d. 1
2. Pentru definiția alăturată a subprogramului  $f$ , ce se va returna la apelul  $f(20400)$  ?
- ```
int f(int n){
    if(n!=0){
        if(n%10!=0) return f(n/10);
        return 1+f(n/10);
    }
    return 0;
}
```
- a. 6 b. 5 c. 0 d. 3
5. Pentru definiția alăturată a subprogramului f , ce se va afișa la apelul $f(27524)$?
- ```
void f(int n){
 if(n){
 if(n%2){
 printf("%d",n%10);/cout<<n%10;
 f(n/10);
 }
 else{f(n/10);
 printf("%d",n%10);/cout<<n%10;
 }
 }
}
```
- a. 27524                      b. 75422                      c. 57224                      d. 42572
3. Pentru definiția alăturată a subprogramului  $f$ , ce se va afișa la apelul  $f(245284003)$  ?
- ```
int f(long n){
    if(n==n%10) return n ;
    return f(n/10);
}
```
- a. 3 b. 0 c. 2 d. 9
4. Știind că tabloul b este declarat prin `int b[5]={1,2,3,4,5}`; stabiliți ce se afișează în urma apelului $f(b,m,m-1)$, a funcției f cu definiția alăturată, în care m este egal cu 5?
- ```
void f(int a[5],int n,int i)
{if (i>=0)
{cout<<a[i]%10;/printf("%d",a[i]%10);
f(a,n,i-1);
if (a[i]<9)
cout<<i; /printf("%d",i); }
}
```
- a. 1234554321                      b. 5432154321                      c. 1234512345                      d. 5432101234
6. Se consideră definiția alăturată a subprogramului  $f$ . Ce se va afișa în urma apelului  $f(13)$  ?
- ```
void f(int n)
{ if (n>0) f(n/10);
printf("%d",n%10);/cout<<n%10;
}
```
- a. 013 b. 310 c. 31 d. 13
7. De câte ori se apelează funcția f în timpul executării atribuirii $x=f(f(999))$, cu x întreg?
- ```
int f(int n)
{if (n==0) return 0;
else return 1+f(n/10);}
```
- a. 4                      b. 6                      c. 2                      d. 5
6. Se consideră subprogramul recursiv cu definiția alăturată. În urma apelului  $p(n, 2)$  funcția returnează valoarea 1 dacă și numai dacă:
- ```
int p(unsigned int n, unsigned int x)
{if (n<x) return 0;
if (x==n) return 1;
else
if (n%x==0) return 0;
else return p(n,x+1);
}
```
- a. n este un număr natural impar b. n este un număr natural neprim
c. n este un număr natural par d. n este un număr natural prim
5. Se consideră funcția recursivă cu definiția alăturată:
Pentru ce valoare a parametrului x în urma apelului $f(x)$ se va returna valoarea 25 ?
- ```
int f(int x)
{if (x==0) return 0;
else return f(x-1)+2*x-1;
}
```
- a. 5                      b. 3                      c. 10                      d. 15

8. Care este valoarea returnată de funcția cu definiția alăturată în urma apelului  $f(4)$  ?
- ```
int f(int x)
{
    if (x<=0) return 3 ;
    else return f(x-1)*2 ;
}
```
- a. 48 b. 16 c. 24 d. 3
3. Se consideră subprogramul recursiv cu definiția alăturată :
- ```
void p(int n)
{ if (n!=1)
 { printf("%d ",n); / cout<<n<<" ";
 if(n%2==0) p(n/2);
 else p(3*n+1);
 }
 else printf("%d",1); / cout<<1;
}
```
- Ce valori vor fi afișate pe ecran în urma apelului  $p(10)$  ?
- a. 5 16 8 4 2 1                      b. 10 5 16 8 4 2 1  
c. 10 5 16 8 4 2                      d. 10 5 4 2 1
5. Se consideră subprogramul recursiv cu definiția alăturată:
- ```
void g(int n)
{int i;
  if(n>=1)
  {for(i=1;i<=n;i++)
    printf("%d ",n); / cout<<n<<" ";
    g(n-1);
  }
}
```
- Ce valori vor fi afișate pe ecran în urma apelului $g(4)$?
- a. 4 4 4 4 3 3 3 2 2 1 b. 4 1 2 3 4 4 3 2 1 4
c. 1 1 1 1 2 2 2 3 3 4 d. 4 3 2 1 3 2 1 2 1 1
6. Care trebuie să fie valoarea variabilei întregi v în apelul $F(v, s)$ dacă valoarea inițială a variabilei întregi s este 0, iar valoarea acesteia în urma executării subprogramului F , definit alăturat, este 4301?
- ```
void F(long int n, long int &s)
{
 if (n > 0)
 {
 s = s * 10 + n % 10;
 F(n / 10, s);
 }
}
```
- a. 1043                      b. 1304                      c. 4301                      d. 1034
5. Ce valoare va avea variabila întregă  $s$ , în urma apelului  $Suma(1, s)$ , știind că variabila globală  $n$  are valoarea 10, variabila  $s$  are înainte de apel valoarea 0, iar subprogramul  $Suma$  este definit alăturat?
- ```
void Suma(int i, int &s)
{ if (i <= n)
  {
    if(i % 2) s = s + 2;
    else s = s - 1;
    Suma(i + 1, s);
  }
}
```
- a. 8 b. 5 c. 7 d. 6
1. Câte caractere „*” se vor afișa în urma executării subprogramului alăturat, la apelul $Scrie(1,1)$?
- ```
void Scrie(int i, int j)
{if(i <= 3)
 if(j <= i)
 {
 printf("*");|cout<<'*';
 Scrie(i, j+1);
 }
 else Scrie(i+1, 1);
}
```
- a. 5                      b. 6                      c. 3                      d. 10
6. Ce se va afișa în urma apelului  $F(123)$ , pentru funcția  $F$  definită alăturat:
- ```
int F(int n)
{
    if ( n == 0 || F(n / 10) )
    {
        cout << n % 10;
        return 1;
    }
    return 0;
}
```
- a. 0123 b. 123 c. 321 d. 3210

8. Ce valoare va avea în urma apelului `x(5639)` variabila globală `i` de tip întreg, dacă înainte de apel `i=0` și funcția `x` este definită alăturat?
- ```
int X(long int n)
{
 i++;
 if (n == 0) return 0;
 else return X(n / 10) + X(n / 100);
}
```
- a. 15                      b. 7                      c. 8                      d. 16
3. Se consideră funcția alăturată. Ce se va afișa pe ecran la apelul `f(13)`?
- ```
void f(int x)
{
    if (x>1) f(x/2);
    printf("%d",x%2); | cout<<x%2;
}
```
- a. 0101 b. 1100 c. 1101 d. 1011
6. Ce se va afișa pe ecran la apelul `p(630,2,1)`?
- ```
void p(int x,int y,int k)
{if(x>1)
 if(x%y==0)
 {if(k==1)
 printf("%d ",y); | cout<<y<<" ";
 p(x/y,y,k+1);
 }
 else p(x,y+1,1);
}
```
- a. 2                      b. 2 3 3 5 7                      c. 2 3 5 7                      d. 2 1 3 2 5 1 7 1
2. Pentru definiția alăturată a subprogramului `p`, stabiliți ce returnează funcția la apelul `p(9876543)`?
- ```
int p(long n)
{if (n)
    if(n%2==0) return 1+p(n/10);
    else return p(n/10);
    else return 0;
}
```
- a. 0 b. 7 c. 3 d. 42
5. Ce se afișează în urma apelului `f(247)`, dacă funcția `f` are următoarea definiție:
- ```
void f(int n)
{ if (n!=0)
 { f(n/10);
 cout<<n%10<<" "; / printf("%d ", n%10);
 }
}
```
- a. 7 4 2                      b. 4 2 7                      c. 2 7 4                      d. 2 4 7
5. Fie funcția recursivă:
- ```
void f(int n)
{ if (n)
    { cout<<n%10<<" "; / printf("%d ", n%10);
      f(n/10);}
}
```
- Ce afișează funcția la apelul `f(247)`?
- a. 7 4 2 b. 2 4 7 c. 2 7 4 d. 4 2 7
3. Fie funcția recursivă:
- ```
int f(int i, int j)
{ if (i==1 || j==1) return 1;
 else if (i<j) return f(i,j-1)+f(i-1,j-1);
 else return f(i-1,j)+f(i-1,j-1);
}
```
- Ce returnează funcția la apelul `f(2,3)`?
- a. 5                      b. 3                      c. 4                      d. 2

6. Se consideră funcția recursivă:
- ```
int f(int n)
{ if (n==0) return 1;
  else if (n%2==0) return f(n/10);
    else return 0;
}
```
- Care dintre instrucțiunile următoare afișează 1?
- `cout<<f(2212); / printf("%d", f(2212));`
 - `cout<<f(10); / printf("%d", f(10));`
 - `cout<<f(9426); / printf("%d", f(9426));`
 - `cout<<f(2426); / printf("%d", f(2426));`
6. Pentru subprogramul `rec` cu definiția alăturată, ce se va afișa în urma apelului `rec(35,4)`?
- ```
void rec(unsigned x, unsigned y)
{ if (x>y) rec(x/y, y);
 printf("%u", x%y); / cout<<(x%y); }
```
- 302
  - 203
  - 100011
  - 83
4. Se definește funcția alăturată; ce se afișează în urma apelului `f(5)`?
- ```
void f(int n)
{ if (n>0)
  { f(n-1);
    cout<<n<<" "; / printf("%d ", n); } }
```
- 5 4 3 2 1
 - 5 4 3 2 1 0
 - 5
 - 1 2 3 4 5
3. Fie funcția `f` definită alăturat. Ce se afișează în urma apelului `f(523)`?
- ```
void f(int x)
{ cout<<x%10; / printf("%d", x%10);
 if (x>9)
 f(x/10);
 cout<<x%10; / printf("%d", x%10); }
```
- 325523
  - 325325
  - 325
  - 523523
6. Se consideră subprogramul `f` cu definiția alăturată. Ce se afișează în urma apelului `f(6)`?
- ```
void f(int x)
{ if (x>=0)
  { cout<<x; / printf("%d", x);
    f(x-1);
    if (x%2==0)
      cout<<x; / printf("%d", x);
  } }
```
- 65432100246
 - 65432106420
 - 0123456
 - 6543210
2. Se consideră subprogramul `f` cu definiția alăturată și o variabilă `y` de tip `int`. Ce se va afișa în urma executării secvenței de program următoare:
- ```
y=0; f(4, y);
cout<<y; / printf("%d", y);
```
- ```
void f(int x, int&k)
{ if (x>1)
  { k=k+1;
    f(x-1, k);
  } }
```
- 4
 - 0
 - 5
 - 3
2. Se consideră subprogramul recursiv alăturat. Ce se afișează la apelul `tip(4)`?
- ```
void tip(int i)
{ printf("X");
 if (i>0)
 if (i%2==0) tip(i/2);
 else tip(i-1);
}
```
- XXXXXX
  - XXXX
  - XX
  - XXX
7. Pentru o valoare naturală mai mare decât 1 memorată în variabila globală `n`, subprogramul recursiv alăturat afișează cel mai mare divizor al lui `n`, mai mic decât `n`, la apelul `divi(n)`. Cu ce expresii trebuie completate punctele de suspensie?
- ```
void divi(long i)
{ if (... ==0)
  printf("%ld", ...);
  else divi(i-1); }
```
-
- ```
void divi(long i)
{ if (... ==0) cout<<...;
 else divi(i-1); }
```
- `n%i=0` și `i`
  - `n%(i-1)` și `i-1`
  - `n%(i-1)=0` și `i`
  - `n%i` și `i-1`

3. În subprogramul recursiv alăturat se consideră că  $v$  este un vector declarat global, format din numere întregi. Pentru care dintre vectorii  $v$  următori, expresia  $ddd(0,18)$  returnează valoarea 9?
- a. (1,2,1,2,3,1,2,3,4,1,1,2,3,4,1,2,3,1,2)  
b. (1,0,1,0,1,0,1,0,1,1,1,1,0,1,0,1,0,1,0)  
c. (1,2,1,2,3,1,2,3,4,0,4,3,2,1,3,2,1,2,1)  
d. (1,1,1,1,1,1,1,1,1,0,2,2,2,2,2,2,2,2,2)
4. În subprogramul recursiv alăturat se consideră că vectorul de numere întregi  $v$  este declarat global. Pentru care dintre vectorii  $v$  următori, la apelul  $star(0)$ , se afișează 7 asteriscuri?
- a. (1,4,3,2,1,6,5,4,3,10)  
b. (1,2,3,4,5,6,7,8,9,10)  
c. (3,2,1,4,3,6,7,2,9,2)  
d. (10,9,8,7,6,5,4,3,2,1)
2. Pentru definiția alăturată a subprogramului  $f$ , stabiliți ce valoare returnează apelul  $f(8,10)$ ?
- a. 50  
b. 45  
c. 40  
d. 55
6. Pentru definiția alăturată a subprogramului  $f$ , stabiliți ce valoare returnează apelul  $f(6,3)$ ?
- a. 3  
b. 1  
c. 2  
d. 4
1. Pentru definiția alăturată a subprogramului  $f$ , stabiliți ce valoare returnează apelul  $f(7)$ ?
- a. 1  
b. -3  
c. -4  
d. 4
7. Ce se afișează ca urmare a apelului  $f(2)$ , dacă subprogramul  $f$  are declarația alăturată?
- a. 1122  
b. 112  
c. 121  
d. 1121

```
float ddd(int i,int j)
{
 if (v[i]==v[j]) return i;
 else return ddd(i+1,j-1);
}
```

```
void star(int i)
{
 if(i<10){
 printf("*"); | cout<<'*';
 if(v[i]==i+1)star(i+2);
 else star(i+1);
 }
}
```

```
long f(int x, int y)
{if (x==y || x==0) return 1;
 else
 return f(x,y-1) + f(x-1,y-1);
}
```

```
long f(int n, int k)
{long s=0,i;
 if (n==k || k==1) return 1;
 if (n<k) return 0;
 for(i=1;i<=k;i++) s = s + f(n-k,i);
 return s;
}
```

```
long f(int n)
{if (n==0) return 1;
 else
 if (n==1) return 4;
 else return f(n-1) - f(n-2);
}
```

```
void f(int n)
{ int j;
 if (n>0)
 for (j=1;j<=n;j++)
 {cout<<j; / printf("%d",j);
 f(n-1);
 }
}
```