

1.	Se consideră subprogramul alăturat: Ce valoare are $f(15, 2)$ ? Dar $f(128, 2)$ ? <b>(6p.)</b>	<pre>int f(int a, int b){     if (b&lt;1) return -1;     else         if (a%b==0)             return 1+f(a/b,b);         else             return 0; }</pre>
2.	Se consideră subprogramul $f$ cu definiția alăturată. Ce valoare are $f(1213111, 1)$ ? <b>(4p.)</b>  a. 5                                      b. 3                                      c. 2                                      d. 1	<pre>int f (long n, int k){     if (n!=0)         if (n%10==k)             return 1+f(n/10,k);         else return 0;     else return 0;}</pre>
3.	Se consideră subprogramul cu definiția alăturată. Ce valoare are $f(3, 1)$ ? <b>(4p.)</b>  a. 9                                      b. 6                                      c. 7                                      d. 8	<pre>int f(int n,int y) { if (n!=0)     { y=y+1;       return y+f(n-1,y);     }   else return 0; }</pre>
4.	Pentru funcția $f$ definită alăturat, stabiliți care este valoarea $f(5)$ . Dar $f(23159)$ ? <b>(6p.)</b>	<pre>int f(int n){     int c;     if (n==0) return 9;     else         {c=f(n/10);          if (n%10&lt;c) return n%10;          else return c;         } }</pre>
5.	Pentru funcțiile $f_1$ și $f_2$ definite alăturat, stabiliți care este valoarea lui $f_1(3)$ . Dar $f_2(41382)$ ? <b>(6p.)</b>	<pre>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); }</pre>
6.	Pentru funcțiile $f$ și $g$ definite mai jos, scrieți care este rezultatul returnat la apelul $g(11)$ . Dar rezultatul returnat la apelul $f(6)$ ? <b>(6p.)</b>  <pre>long g(long x) { if (x&gt;9)     return (x/10 + x%10);   else     return x; }</pre>	<pre>long f(int c) { if (c&lt;1)     return 1;   else     return g(c+f(c-1)); }</pre>

7.	<p>Funcția <b>F</b> are definiția alăturată. Ce valoare are <b>F(3)</b>? <b>(4p.)</b></p> <p>a. 1                                      b. 12                                      c. 6                                      d. 10</p>	<pre>int F(int n) {if(n==0    n==1) return 1; else return 2*F(n-1)+2*F(n-2);}</pre>
8.	<p>Funcția <b>F</b> are definiția alăturată. Ce valoare are <b>F(18)</b>? <b>(4p.)</b></p> <p>a. 90                                      b. 171                                      c. 91                                      d. 18</p>	<pre>int F(int x){     if (x&lt;=1) return x;     else return x+F(x-2); }</pre>
9.	<p>Funcția <b>F</b> are definiția alăturată. Ce valoare are <b>F(5)</b>? <b>(4p.)</b></p> <p>a. 5                                      b. 10                                      c. 15                                      d. 6</p>	<pre>int F(int x) {if(x!=0) return x+F(x-1); else return x; }</pre>
10.	<p>Se consideră subprogramul, <b>f</b>, definit alăturat.</p> <ul style="list-style-type: none"> <li>• Ce valoare are <b>f(100)</b>?</li> <li>• Scrieți o valoare pentru <b>x</b> astfel încât <b>f(x)=1</b> <b>(6p.)</b></li> </ul>	<pre>int f(int n) { if(n==0) return 0; else return n%2+f(n/2); }</pre>
11.	<p>Pentru definiția de mai jos a subprogramului <b>f</b>, ce se afișează ca urmare a apelului <b>f(121,1);</b>? <b>(6p.)</b></p> <pre>//C void f(long n, int i) { if(n!=0)   if(n%3&gt;0)     { printf("%d",i); f(n/3,i+1); } }</pre>	<pre>//C++ void f(long n, int i) { if(n!=0)   if(n%3&gt;0)     { cout&lt;&lt;i; f(n/3,i+1); } }</pre>
12.	<p>Pentru definiția de mai jos a subprogramului <b>f</b>, ce se afișează ca urmare a apelului <b>f(12345);</b>? <b>(6p.)</b></p> <pre>//C void f(long n) { printf("%d",n%10);   if(n!=0)     { f(n/100); printf("%d",n%10); } }</pre>	<pre>//C++ void f(long n) { cout&lt;&lt;n%10;   if(n!=0)     { f(n/100); cout&lt;&lt;n%10; } }</pre>
13.	<p>Pentru definiția alăturată a subprogramului <b>f</b>, ce se afișează ca urmare a apelului <b>f(26);</b>? <b>(6p.)</b></p>	<pre>void f (int x) {   if(x&gt;0)   if(x%4==0)     { cout&lt;&lt;'x';   printf("%c",'x');       f(x-1); }   else     { f(x/3);       cout&lt;&lt;'y';   printf("%c",'y');     } }</pre>

14.	Pentru definiția alăturată a subprogramului <code>f</code> , ce se afișează ca urmare a apelului <code>f(15,2);</code> ? <b>(6p.)</b>	<pre>void f (int n, int x) { if(x&gt;n)   cout&lt;&lt;0;   printf("%d",0);   else   if(x%4&lt;=1) f(n,x+1);   else   { f(n,x+3);     cout&lt;&lt;1;   printf("%d",1);   } }</pre>
15.	Pentru subprogramul <code>f</code> definit mai jos, ce se afișează ca urmare a apelului <code>f(3,17)</code> ? <b>(6p.)</b>	<pre>//C void f ( int a, int b) { if(a&lt;=b)   {f(a+1,b-2); printf("%c",'*');}   else printf("%d",b); }  //C++ void f ( int a, int b) { if(a&lt;=b)   { f(a+1,b-2); cout&lt;&lt;'*';}   else cout&lt;&lt;b; }</pre>
16.	Se consideră subprogramul <code>f</code> definit alăturat. Ce se va afișa în urma apelului <code>f(12345);</code> ? <b>(4p.)</b>	<pre>void f(long int n) { if (n!=0)   {if (n%2 == 0)    cout&lt;&lt;n%10;   printf("%d",n%10);    f(n/10);   } }</pre>
	a. 513                      b. 24                      c. 42                      d. 315	
17.	Se consideră subprogramul <code>f</code> , descris alăturat. Ce se va afișa în urma apelului <code>f(3);</code> ? <b>(6p.)</b>	<pre>void f(int n) { if (n!=0)   { if (n%2==0)     cout&lt;&lt;n&lt;&lt;' ';   printf("%d ",n);     f(n-1);     cout&lt;&lt;n&lt;&lt;' ';   printf("%d ",n);   }   else cout&lt;&lt;endl;   printf("\n"); }</pre>
18.	Subprogramul <code>f</code> are definiția alăturată. Ce se va afișa în urma apelului <code>f(12345);</code> ? <b>(4p.)</b>	<pre>void f(long n) {if (n&gt;9)  {cout&lt;&lt;n/100;   printf("%d",n/100);   f(n/10);  } }</pre>
	a. 1231210                      b. 123121                      c. 1234123121                      d. 123	
19.	Funcția <code>f</code> are definiția alăturată. Ce se va afișa în urma apelului <code>f(12345,0);</code> ? <b>(4p.)</b>	<pre>void f(long n, int i) {if (i&lt;n%10)  {cout&lt;&lt;n%10;   printf("%d",n%10);   f(n/10,i+1);  } }</pre>
	a. 54321                      b. 543                      c. 54                      d. 5432	

20.	<p>Considerăm subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului de mai jos?</p> <p><b>f('C');</b> (4p.)</p>	<pre>void f(char c) { if (c&gt;'A') f(c-1);   cout&lt;&lt;c;   printf("%c",c);   if (c&gt;'A') f(c-1); }</pre>
21.	<p>Pentru subprogramul <b>suma</b> definit alăturat, scrieți valoarea expresiei <b>suma(5,4)</b>. (4p.)</p>	<pre>int suma (int a,int b) { if (a==0 &amp;&amp; b==0) return 0;   else if (a==0) return 1+suma(a,b-1);   else return 1+suma(a-1,b); }</pre>
22.	<p>Funcția <b>f</b> are definiția alăturată.</p> <p><b>a)</b> Ce valoare are <b>f(17)</b>? (3p.)</p> <p><b>b)</b> Ce valoare are <b>f(22)</b>? (3p.)</p>	<pre>int f(int n) {if (n&lt;=9) return 0;  if (n%4==0) return 0;  return 1+f(n-3); }</pre>
23.	<p>Funcția <b>f</b> are definiția alăturată:</p> <p><b>a)</b> Ce valoare are <b>f(16)</b>? (3p.)</p> <p><b>b)</b> Scrieți cea mai mare valoare de două cifre pe care o poate avea <b>n</b> astfel încât <b>f(n)</b> să fie egal cu 2. (3p.)</p>	<pre>int f(int n) { if (n&lt;=0) return -1;   if (n%2==0) return 0;   if (n%3==0) return 0;   return 1+f(n-10); }</pre>
24.	<p>Subprogramul <b>afis</b> este definit alăturat. Ce se afișează ca urmare a apelului <b>afis(8)</b>? (4p.)</p>	<pre>void afis (int n) {   cout&lt;&lt;n;   printf("%d",n);   for (int i=n/2;i&gt;=1;i--)     if(n%i==0)afis(i); }</pre>
25.	<p>Subprogramul <b>scrie</b> este definit alăturat. Ce se afișează ca urmare a apelului <b>scrie(2,6)</b>? (6p.)</p>	<pre>void scrie (int x,int y) {   cout&lt;&lt;x&lt;&lt;y;   printf("%d%d",x,y);   if(x&lt;y)   {     scrie(x+1,y-1);     cout&lt;&lt;(x+y)/2;  printf("%d", (x+y)/2);   } }</pre>
26.	<p>Subprogramul <b>f</b> este definit alăturat. Ce valoare are <b>f(8,4)</b>? (4p.)</p>	<pre>int f (int x,int y) {   if(x&lt;y)return 1+f(x+1,y);   if(y&lt;x)return 1+f(y+1,x);   return 1; }</pre>
27.	<p>Ce se afișează ca urmare a apelului <b>p(123)</b>; dacă subprogramul <b>p</b> are definiția alăturată? (6p.)</p>	<pre>void p (int x) {cout&lt;&lt;x;   printf("%d",x);   if(x!=0){p(x/10);     cout&lt;&lt;x%10;   printf("%d",x%10);}}</pre>

28.	Subprogramul <b>f</b> este definit alăturat. Ce se afișează ca urmare a apelului <b>f(1,3);</b> ? (6p.)	<pre>void f (int x,int y) {int i;   for (i=x;i&lt;=y;i++)   {     cout&lt;&lt;i;   printf("%d",i);     f(i+1,y);   } }</pre>
29.	Se consideră subprogramul recursiv <b>f1</b> definit alăturat. Ce se va afișa în urma apelului <b>f1(5);</b> ? (6p.)	<pre>void f1(int x) { if (x&lt;=9) { cout&lt;&lt;x+1;   printf("%d",x+1);   f1(x+2);   cout&lt;&lt;x+3;   printf("%d",x+3); } }</pre>
30.	Subprogramul <b>afis</b> este definit alăturat. Ce se va afișa în urma apelului <b>afis(17);</b> ? (6p.)	<pre>void afis(int x) { if (x&gt;3) { cout&lt;&lt;x-1;   printf("%d",x-1);   afis(x/3);   cout&lt;&lt;x+1;   printf("%d",x+1); } }</pre>
31.	Subprogramul <b>re</b> este definit alăturat. Ce valoarea are <b>re(1)</b> ? Dar <b>re(14)</b> ? (6p.)	<pre>int re(int i) {   if (i&lt;9) return 3+re(i+2);   else     if (i==9) return -2;     else return 1+re(i-1); }</pre>
32.	Se consideră definit subprogramul <b>f</b> . Scrieți două valori naturale, <b>x1</b> și <b>x2</b> ( <b>x1≠x2</b> , <b>x1&lt;12</b> și <b>x2&lt;12</b> ) pentru care <b>f(x1)=f(x2)</b> . (6p.)	<pre>int f(int i) {   if (i&gt;12) return 1;   else return 1+f(i+2); }</pre>
33.	Subprogramul <b>f</b> este definit alăturat. Ce se va afișa în urma executării secvenței de mai jos, în care variabilele <b>a</b> și <b>b</b> sunt de tip întreg? <b>a=3; b=9; f(b,a); f(b,b);</b> (6p.)	<pre>void f(int &amp;a,int b) {   a=a-5;b=a-2;   cout&lt;&lt;a&lt;&lt;b;   printf("%d%d",a,b); }</pre>
34.	Subprogramul <b>scif</b> returnează suma cifrelor unui număr natural transmis ca parametru. Care este valoarea expresiei <b>scif(scif(518)+scif(518))</b> ? <b>a. 14                      b. 10                      c. 28                      d. 1</b> (4p.)	

35.	Considerăm subprogramul $f$ definit alăturat. Ce valoare are $f(7,11)$ ? Dar $f(11,7)$ ? (6p.)	<pre>int f(int x,int y) {     if(x&lt;=y) return x-y;     return f(y-x,x-1)+3; }</pre>
36.	Pentru definiția alăturată a subprogramului $sc$ , stabiliți ce valoare are $sc(10)$ . Dar $sc(901324)$ ? (6p.)	<pre>int sc(long x) {     if(x&lt;10) return x;     return sc(x/10)+x%10; }</pre>
37.	Pentru definiția alăturată a subprogramului $f$ , ce valoare are $f(3)$ ? Dar $f(8)$ ? (6p.)	<pre>int f(int x) {     if(x&lt;=4) return x*x-3;     return f(x-3)+4; }</pre>
38.	Știind că $p$ este un vector cu 3 componente întregi (vector declarat global), stabiliți cu ce trebuie înlocuite $\alpha$ și $\beta$ în definiția subprogramului $G$ alăturat astfel încât în urma apelului $G(0)$ să se afișeze toate numerele de 3 cifre nenule, fiecare număr fiind afișat o singură dată. (6p.)	<pre>void G(int k) {int i;   for(i=1;i&lt;=alpha;i++)   { p[k]=i;     if(beta)G(k+1);     else       cout&lt;&lt;p[0]&lt;&lt;p[1]&lt;&lt;p[2]&lt;&lt;endl;         printf("%d%d%d\n",p[0],p[1],p[2]);   } }</pre>
39.	Subprogramul $f$ are definiția alăturată. Ce valoare are $f(7)$ ? Dar $f(100)$ ? (6p.)	<pre>int f(int x) {if(x%6==0)return x;   else return f(x-1); }</pre>
40.	Subprogramul $f$ are definiția alăturată. Ce valoare are $f(3)$ ? Dar $f(10)$ ? (6p.)	<pre>int f(int x) {if(x==0)return 0;   else return f(x-1)+2; }</pre>
41.	Subprogramul $f$ are definiția alăturată. Ce valoare are $f(6,5)$ ? Dar $f(5,10)$ ? (6p.)	<pre>int f(int x,int y) {if(x==y)return x;   else if(x&lt;y)return f(x+1,y-1);   else return f(x-1,y); }</pre>
42.	Subprogramul $f$ are definiția alăturată. Ce valoare are $f(4)$ ? Dar $f(11)$ ? (6p.)	<pre>int f(int x) {if(x&lt;1)return 1;   else return f(x-3)+1; }</pre>



43.	<p>Funcția <math>f</math> are definiția alăturată. Dacă <math>f(x)</math> are valoarea 10100, care este valoarea lui <math>x</math>? (6p.)</p>	<pre>long f(int n) {     if(n&lt;=0) return 0;     else return f(n-1)+2*n; }</pre>
44.	<p>Se consideră subprogramul <math>f</math> definit alăturat. Ce valoare are <math>f(5)</math>? Dar <math>f(100)</math>? (6p.)</p>	<pre>long f(int n) {     if(n&lt;0) return 0;     else return f(n-2)+n; }</pre>
45.	<p>Se consideră subprogramul <math>f</math> definit alăturat. Ce valoare are <math>f(250)</math>? (4p.)</p> <p>a. 5                                      b. 2                                      c. 3                                      d. 4</p>	<pre>int f(int x) {     if(x%3==0) return 0;     else return 1+f(x/3); }</pre>
46.	<p>Pentru o valoare naturală mai mare decât 1 memorată în variabila globală <math>n</math>, subprogramul recursiv alăturat afișează cel mai mare divizor al lui <math>n</math>, mai mic decât <math>n</math>, la apelul <math>divi(n)</math>. Cu ce trebuie înlocuite <math>\alpha</math> și <math>\beta</math>? (6p.)</p>	<pre>void divi(long i) {     if(<math>\alpha</math>==0)         cout&lt;&lt; <math>\beta</math>;   printf("%ld", <math>\beta</math>);     else divi(i-1); }</pre>
47.	<p>Valorile memorate de componentele tabloului <math>v</math>, cu indicii de la 0 la 5, sunt, în această ordine: 973, 51, 75, 350, 350, 15. Se consideră subprogramul <math>t</math> cu definiția alăturată. Care dintre următoarele expresii are valoarea 2? (4p.)</p> <p>a. <math>tablou(0,v)+tablou(3,v)</math>                                      b. <math>tablou(1,v)+tablou(4,v)</math>  c. <math>tablou(4,v)+tablou(5,v)</math>                                      d. <math>tablou(3,v)+tablou(4,v)</math></p>	<pre>int t (int i, int v[]) {     if(i==0) return 0;     if(<math>v[i] \neq v[i-1]</math>) return t(i-1,v);     return 1; }</pre>
48.	<p>Care este antetul corect al subprogramului <math>cifre</math>, care primește prin intermediul primului parametru, <math>x</math>, un număr natural și furnizează prin intermediul celui de-al doilea parametru, <math>y</math>, suma cifrelor numărului natural <math>x</math>? (4p.)</p> <p>a. <math>void\ cifre(int\ x,\ int\ \&amp;y)</math>                                      b. <math>int\ cifre(int\ x)</math>  c. <math>void\ cifre(int\ x,\ int\ y)</math>                                      d. <math>void\ cifre(int\ \&amp;x,\ int\ y)</math></p>	
49.	<p>Se consideră subprogramul <math>f</math>, definit alăturat. Ce valoare are <math>f(4)</math>? Dar <math>f(9)</math>? (6p.)</p>	<pre>int f(int n) {     if (n==0) return 0;     if(n%2==1) return n-f(n-1);     return f(n-1)-n; }</pre>

50.	Se consideră subprogramul $f$ , definit alăturat. Pentru ce valori ale lui $n$ aparținând intervalului $[10, 20]$ se obține la apel $f(n) = 0$ ? (6p.)	<pre>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); }</pre>
51.	Se consideră subprogramul $f$ definit alăturat. Ce se afișează la apelul $f(1)$ ? (6p.)	<pre>void f(int i) {     if (i&lt;=5){         cout&lt;&lt;i&lt;&lt;" ";   printf("%d ",i);         f(i+1);         cout&lt;&lt;i/2&lt;&lt;" ";   printf("%d ",i/2);     } }</pre>
52.	Se consideră subprogramul $f$ , definit alăturat. a) Ce valoare are $f(25)$ ? b) Dar expresia $f(1)+f(5)+f(15)$ ? (6p.)	<pre>int f(int n) { if (n&gt;20) return 0;   else return 5+f(n+5); }</pre>
53.	Se consideră subprogramul $f$ , definit alăturat. Ce valoare are $f(12,3)$ ? Dar $f(21114,1)$ ? (6p.)	<pre>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); }</pre>
54.	Se consideră subprogramul $f$ , definit alăturat. Ce se va afișa la apelul $f(38)$ ? (6p.)	<pre>void f(int x) {     if (x) {         if (x%3==0){             cout&lt;&lt;3;   printf("3");             f(x/3);         }         else{             f(x/3);             cout&lt;&lt;x%3;   printf("%d",x%3);         }     } }</pre>
55.	Se consideră subprogramul $f$ , definit alăturat. Ce valoare are $f(88,1)$ ? Dar $f(3713,3)$ ? (6p.)	<pre>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); }</pre>



56.	Se consideră subprogramul $f$ , definit alăturat. Ce valoare are $f(12,5)$ ? Dar $f(261,31)$ ? (6p.)	<pre>int f(int a,int b) {     if(a&lt;10)         return b;     return f(a/10,b)*10+b+1; }</pre>
57.	Se consideră subprogramul $f$ , definit alăturat. Ce valoare are $f(3,13)$ ? Dar $f(1000,2009)$ ? (6p.)	<pre>int f(int a,int b) {     if(2*a&gt;=b)         return 0;     if(b%a==0)         return b-a;     return f(a+1,b-1); }</pre>
58.	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 $f(1)$ ? Dar $f(4)$ ? (6p.)	<pre>int f(int x) {if(x&gt;=1)     return (a[x]+f(x-1))%10; else     return 0; }</pre>
59.	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? Dar dacă înainte de apel $x$ are valoarea 52? (6p.)	<pre>void f(int n,int &amp;a) {int c;   if(n!=0){     c=n%10;     if(a&lt;c) a=c;     f(n/10,a);   } }</pre>
60.	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 $f(2,1)$ ? Dar $f(3,3)$ ? (6p.)	<pre>int f(int b,int i) {if(i&gt;=1)     return f(b,i-1)*b+a[i]; else return 0; }</pre>
61.	Se consideră subprogramul $f$ , definit alăturat. Ce se va afișa în urma apelului $f(5,0)$ ? (6p.)	<pre>void f(int i,int j) { if(j&lt;=9) f(i,j+1);   printf("%d*%d=%d\n",i,j,i*j);/   cout&lt;&lt;i&lt;&lt;' '*&lt;&lt;j&lt;&lt;'='&lt;&lt;i*j&lt;&lt;endl; }</pre>
62.	Se consideră funcția $f$ , definită alăturat. Ce se afișează ca urmare a executării secvenței de mai jos în care variabilele $a$ și $b$ sunt de tip <code>int</code> ? <code>a=4; b=18;</code> <code>printf("%d",f(a,b));/ cout&lt;&lt;f(a,b);</code> <code>printf("%d %d",a,b);/ cout&lt;&lt;a&lt;&lt;b;</code> (6p.)	<pre>int f( int &amp;a, int &amp;b) { while (a !=b)     if (a&gt;b) a=a-b;     else b=b-a;   return a;} </pre>

63.	Se consideră subprogramul recursiv definit alăturat. Ce valoare are <code>functie(1)</code> ? Dar <code>functie(4)</code> ? (6p.)	<pre>int functie(int x) {     if (x&lt;=0) return 3 ;     else return functie(x-3)*4 ; }</pre>
64.	Se consideră subprogramul recursiv definit alăturat. Ce valoare are <code>alfa(6)</code> ? Pentru ce valoare a parametrului <code>u</code> , <code>alfa(u)</code> are valoarea 25 ? (6p.)	<pre>int alfa(int u) {if (u==0) return 3;   else return alfa(u-1)+3*u-2; }</pre>
65.	Se consideră subprogramul recursiv definit alăturat. Ce valori vor fi afișate pe ecran în urma apelului <code>gama(6)</code> ? (6p.)	<pre>void gama(int n) {int i;   if(n&gt;=3)     {for(i=3;i&lt;=n;i++)       printf("%d ",n);   cout&lt;&lt;n&lt;&lt;" ";       gama(n-3);     } }</pre>
66.	Se consideră subprogramul recursiv definit alăturat. Ce valori vor fi afișate pe ecran în urma apelului <code>beta(15)</code> ? (6p.)	<pre>void beta(int n) {if (n!=1)   {printf("%d ",n);   cout&lt;&lt;n&lt;&lt;" ";    if (n%3==0) beta(n/3);    else beta(2*n-1);}   else printf("%d",1);   cout&lt;&lt;1; }</pre>
67.	Se consideră subprogramul recursiv definit alăturat. Ce valoare are expresia <code>bac(10,4)</code> ? Care este cea mai mică valoare de 2 cifre a lui <code>u</code> pentru care funcția <code>bac(u,2)</code> are valoarea 1? (6p.)	<pre>int bac(int u, int x) {if (u&lt;x) return 0;   if (x==u) return 1;   if (u%x==0) return 0;   return bac(u,x+1); }</pre>
68.	Ce valoare va avea variabila întreagă <code>x</code> , în urma apelului <code>F(1,x)</code> , știind că, înainte de apel, variabila <code>x</code> are valoarea 0, iar subprogramul <code>F</code> este definit alăturat? (6p.)	<pre>void F(int i, int &amp;x) { if (i &lt;= 10)   {   if(i % 2) x = x + 2;       else    x = x - 1;       F(i + 1, x);   } }</pre>
69.	Scrieți ce se va afișa în urma executării subprogramului alăturat, la apelul <code>F(57)</code> ? (6p.)	<pre>void F(int x) { if(x != 0)   { F(x/2);     cout &lt;&lt; x%2;   printf("%d",x%2);   } }</pre>
70.	Ce se va afișa în urma executării subprogramului alăturat, la apelul <code>F(56)</code> ? (6p.)	<pre>void F(int x) { if(x)   { F(x/2);     cout &lt;&lt; x%10;   printf("%d",x%10);   } }</pre>

71.	Se consideră funcția <b>Suma</b> , definită alăturat. Ce valoare are <b>Suma(3)</b> ? Dar <b>Suma(8)</b> ? (6p.)	<pre> int Suma(int x) {     if(x == 1) return 0;     if(x%2==0) return Suma(x-1)+(x-1)*x;     return Suma(x-1) - (x-1)*x; }                     </pre>
72.	Ce valoare are <b>F(2758)</b> , pentru funcția <b>F</b> definită alăturat? (4p.)	<pre> int F(int x) {     if(x == 0) return 0;     if(x%10%2 == 0) return 2 + F(x/10);     return 10 - F(x/10); }                     </pre> <p>a. 0                      b. 20                      c. 12                      d. 4</p>
73.	Pentru definiția alăturată a subprogramului <b>sub</b> , scrieți ce valoare are <b>sub(3)</b> . Dar <b>sub(132764)</b> ? (6p.)	<pre> long sub(long n) {     if (n!=0)         if(n%2!=0) return n%10*sub(n/10);         else return sub(n/10);     else return 1; }                     </pre>
74.	Pentru definiția alăturată a subprogramului <b>sub</b> , scrieți ce valoare are <b>sub(4)</b> . Dar <b>sub(132764)</b> ? (6p.)	<pre> int sub(long n) {     if (n!=0)         if(n%2!=0)             return n%10+sub(n/10);         else return sub(n/10);     else return 0; }                     </pre>
75.	Pentru definiția alăturată a subprogramului <b>sub</b> , scrieți ce valoare are <b>sub(4)</b> . Dar <b>sub(123986)</b> ? (6p.)	<pre> int sub(long n) {     if (n!=0)         if(n%2!=0) return 1+sub(n/10);         else return sub(n/10);     else return 0; }                     </pre>
76.	Pentru definiția alăturată a subprogramului <b>f</b> , scrieți ce valoare are <b>f(8)</b> . Dar <b>f(1209986)</b> ? (6p.)	<pre> int f(long x) {     int y,z;     if (x==0) return x;     else {y=x%10;           z=f(x/10);           if(y&gt;z) return y;           else return z;     } }                     </pre>
77.	Pentru definiția alăturată a subprogramului <b>sub</b> , ce valoare are <b>sub(9)</b> ? Dar <b>sub(132764)</b> ? (6p.)	<pre> int sub(long n) {     if (n!=0)         if(n%2!=0)             return n%10+sub(n/10);         else return sub(n/10);     else return 0; }                     </pre>

78.	Pentru definiția alăturată a subprogramului $f$ , scrieți ce valoare are $f(0,0)$ . Dar $f(525,5)$ ? (6p.)	<pre> int f(int x,int y) { if(x==0) return 0;   else     if(x%10==y)       return f(x/10,y)+1;     else return f(x/10,y); } </pre>
79.	Pentru definiția alăturată a subprogramului $f$ , stabiliți ce valoare are $f(2)$ . Dar $f(123)$ ? (6p.)	<pre> int f(int x) { if(x==0) return 0;   else     if(x%2==0) return 1+f(x/10);     else return 2+f(x/10); } </pre>
80.	Pentru definiția alăturată a subprogramului $f$ , stabiliți ce valoare are $f(2)$ . Dar $f(123)$ ? (6p.)	<pre> int f(int x) { if(x==0) return 0;   else     if(x%2==0) return 3+f(x/10);     else return 4+f(x/10); } </pre>
81.	Pentru definiția alăturată a subprogramului $f$ , stabiliți ce valoare are $f(1)$ ? Dar $f(100)$ ? (6p.)	<pre> int f(int x) { if(x==0) return 1;   else     return 1+f(x-1); } </pre>
82.	Pentru definiția alăturată a subprogramului $f$ , scrieți ce valoare are $f(51)$ . Dar $f(100)$ ? (6p.)	<pre> int f(int x) { if(x==50) return 1;   else     return 2+f(x-1); } </pre>
83.	Se consideră subprogramul $f$ , definit alăturat. Ce se afișează la apelul $f(4)$ ? (6p.)	<pre> void f(int n) {   cout&lt;&lt;"*";   printf("*");   if(n&gt;2)   {     f(n-1);     cout&lt;&lt;"#";   printf("#");   } } </pre>
84.	Se consideră subprogramul $f$ , definit alăturat. Ce se afișează la apelul $f(4)$ ? (6p.)	<pre> void f(int n) {   if(n&gt;0)   {     cout&lt;&lt;n;   printf("%d",n);     f(n-1);     cout&lt;&lt;n;   printf("%d",n);   } } </pre>

85.	Se consideră subprogramul $f$ , definit alăturat. Ce valoare are $f(1)$ ? Dar $f(4)$ ? <b>(6p.)</b>	<pre> int f(int n) {     if (n==0) return 1;     else if (n==1) return 2;     else return f(n-1)-f(n-2); } </pre>
86.	Se consideră subprogramul $f$ , definit alăturat. Ce valoare are $f(0)$ ? Dar $f(4)$ ? <b>(6p.)</b>	<pre> long f(int n) {     if (n==0) return 0;     else return n*n+f(n-1); } </pre>
87.	Se consideră subprogramul $f$ , definit alăturat. Ce se afișează la apelul $f('a')$ ? <b>(6p.)</b>	<pre> void f(char c) {     if (c != 'e')     {         f(c+1);         cout&lt;&lt;c;   printf("%c",c);     } } </pre>
88.	Subprogramul recursiv alăturat este definit incomplet. Scrieți expresia care poate înlocui punctele de suspensie astfel încât, în urma apelului, $f(12)$ să se afișeze șirul de valori:  12 6 3 1 1 3 6 12. <b>(6p.)</b>	<pre> void f(int i) { if ( . . . )   { printf("%d ",i);   cout&lt;&lt;i&lt;&lt;' ';     f(i/2);     printf("%d ",i);   cout&lt;&lt;i&lt;&lt;' ';   } } </pre>
89.	Se consideră subprogramul $f$ definit alăturat. Ce se va afișa în urma apelului $f(14)$ ? <b>(6p.)</b>	<pre> void f(int x) {     if (x&lt;=10)         cout&lt;&lt;0&lt;&lt;" "; printf("%d ",0);     else     {f(x-2);      cout&lt;&lt;x&lt;&lt;" "; printf("%d ",x);     } } </pre>
90.	Subprogramul $f$ este definit alăturat. Ce se afișează ca urmare a apelului $f(1,3)$ ? <b>(6p.)</b>	<pre> void f (int x,int y) { if (x&lt;y){x=x+1;f(x,y);   y=y-1;f(x,y); } else   cout&lt;&lt;x&lt;&lt;y; printf("%d%d",x,y); } </pre>
91.	Funcția recursivă $f$ este astfel definită încât $f(1)=8$ , iar $f(n+1)=2*f(n)-4$ pentru orice $n$ natural nenul.  <b>a)</b> Ce valoare are $f(5)$ ? <b>(3p.)</b>  <b>b)</b> Care este cea mai mare valoare pe care o poate lua $x$ astfel încât $f(x) < 1000$ ? <b>(3p.)</b>	

92.	<p>Funcția <math>f</math> are definiția alăturată. Scrieți cinci valori de apel pe care le poate avea <math>n</math> astfel încât, pentru cele 5 apeluri corespunzătoare acestor valori, să se obțină 5 valori ale funcției, distincte două câte două. (6p.)</p>	<pre>int f(int n) {     if (n&lt;=9) return 0;     if (n%5==0) return 0;     return 1+f(n-3); }</pre>
93.	<p>Funcția <math>f</math> are definiția alăturată. Scrieți 4 valori de apel pe care le poate avea <math>n</math> astfel încât, pentru cele 4 apeluri, corespunzătoare acestor valori, să se obțină 4 valori, distincte două câte două. (6p.)</p>	<pre>int f(int n) {if (n&lt;=9) return 0;   if (n%4==0) return 0;   return 1+f(n-3); }</pre>
94.	<p>Subprogramul recursiv alăturat este definit incomplet. Care dintre următoarele expresii poate înlocui punctele de suspensie astfel încât, în urma apelului, subprogramul <math>f</math> să returneze suma primelor două cifre ale numărului primit prin intermediul parametrului <math>x</math>?  <b>Exemplu:</b> în urma apelului <math>f(2318)</math> valoarea returnată este 5. (4p.)</p> <p>a. <math>x \leq 100</math>                      b. <math>x \leq 99</math>                      c. <math>x = 99</math>                      d. <math>x \neq 0</math></p>	<pre>int f(int x){     if (...)         return x%10 + x/10;     else         return f(x/10); }</pre>
95.	<p>Se consideră subprogramul recursiv alăturat, definit incomplet. Cu ce valoare trebuie înlocuite punctele de suspensie, pentru ca funcția să returneze cifra minimă a numărului natural nenul transmis prin intermediul parametrului <math>x</math>? (4p.)</p> <p>a. -1                      b. 1                      c. 9                      d. 0</p>	<pre>int Min(int x){     int c;     if (x==0) return ...;     else {         c=Min(x/10);         if (c &lt; x%10) return c;         else return x%10;     } }</pre>
96.	<p>Se consideră subprogramul recursiv alăturat, <math>S</math>, definit incomplet. Cu ce expresie pot fi înlocuite punctele de suspensie astfel încât, în urma apelului <math>S(2)</math>, să se afișeze 3 caractere <math>*</math>? (4p.)</p> <p>a. <math>x &gt; 1</math>                      b. <math>x &gt; 2</math>                      c. <math>x \geq 3</math>                      d. <math>x &gt; 0</math></p>	<pre>void S(int x) { cout&lt;&lt;'*';   if (...) {       cout&lt;&lt;'*';       S(x-1);   } }</pre>
97.	<p>Ce afișează subprogramul <math>F</math>, descris alăturat, la apelul <math>F(5)</math>? (6p.)</p>	<pre>void F(int x) {     cout&lt;&lt;x;   printf("%d",x);     if(x&gt;=3)         F(x-2);     cout&lt;&lt;x-1;   printf("%d",x-1); }</pre>