1. Indicați posibilitatea evaluării expresiei și eventual rezultatul acestei evaluări:

```
x = (x||!y)&&z;
```

dacă inițial x=2, y=1, z=0;
a) expresia este eronată
b) 1
c) 2
d) 0

2. Ce afișează apelul:

```
printf("%d", sizeof('a'));
```

a) 2
b) 0
c) 1
d) este generată eroare, a trebuie incadrat între ghilimele

3. După o inițializare:

```
char mesaj[ ]={'G','r','e','s','i','t'};
```

precizați care afirmație este corectă:
a) nu este permisă manipularea ulterioară ca șir a variabilei mesaj;
b) inițializarea este eronată deoarece nu conține terminatorul de șir '\0'.
c) inițializarea nu este permisă în cazul masivelor de caractere;
d) declarația este eronată, deoarece lipsește dimensiunea masivului;

4. În programul:

```
void main ( )
{
  int a=1;
  const int c=1;          // varianta 1
  const int *pl=&c;       // varianta 2
  const int *p2=&a;       // varianta 3
   int *p3=&c;            // varianta 4
}
```

Care din variante sunt incorecte:
a) 1+2
b) 2+3
c) 4
d) 1+4

5. Fie declarațiile:

```
int a=5; double x;
```

În urma atribuirii:

```
x=a/2;
```

variabila x va avea valoarea:
a) 0
b) 2.5
c) o valoare reziduală
d) 2

6. Instrucțiunea

```
printf("%d", sizeof( 32767L));
```

a) tipărește 2 (= sizeof(int))
b) tipărește 4 (= sizeof(long))
c) este invalidă (trebuia folosit descriptorul ld)
d) este invalidă (sizeof nu admite operand constant)

7. Fie declarația:

```
int a=17, b=21,c;
```

Ce valoare va avea variabila c în urma atribuirii:

```
c=a&b|c&a;
```

a) 15
b) 17
c) 12
d) 21

8. Se consideră secvența:

```
int a;
double r=sqrt(a);
int x=r==(int)r;
```

variabila x este nenulă dacă:
(a) variabila a este pătrat perfect;
(b) variabila a nu este număr întreg.
(c) variabila a este număr întreg;
(d) variabila a nu este pătrat perfect;

9. Fie declarația:

```
int a=13, b=33, c;
```

Ce valoare va avea variabila c în urma atribuirii

```
c=a|b>>2;
```

a) 13
b) 10
c) 0
d) 11

Nr 42

1. Indicați valorile pe care le vor avea, în ordine, variabilele a,b,c,d, în urma atribuirilor:
   a=b=c=d=2; a+=b+=c+=d;
a) 2,2,2,2
b) 5,4,3,2
c) 8,6,4,2
d) 2,4,6,8

2. Fie declarația:
   int a=7, c;
Ce valoare va avea variabila c în urma atribuirii:
   c=1<<a>>2;
a) 0
b) 32
c) 30
d) 34

3. Fie declarațiile:
   int a=5, x;
In urma atribuirii:
   x=(double)a/2;
variabila x va avea valoarea:
a) 2
b) 0
c) o valoare reziduală
d) 2.5

4. Programul:
#include <stdio.h>
void main() {
   int p=1, i=5; while(i--)
   p*=i; printf("\n%d",p);
}
Afișează:
a) 24
b) 120
c) 0
d) 10

5. Dacă variabila intreagă v conține valoarea -2, secvența:
   if (v) printf("pozitiv"); else printf("negativ");
Afișează
a) negativ pozitiv
b) negativ
c) pozitiv
d) pozitiv negativ

6. Pentru a se asigura corectitudinea apelului
   strcpy(d,s)
trebuie îndeplinită totdeauna condiția:
a) sizeof(d) >= strlen(s);
b) strlen(d) >= strlen(s);
c) sizeof(d) >= strlen(s)+1;
d) sizeof(d) >= sizeof(s);

7. În programul:
void main ( ) {
   char z;
   char * const pc = &z;
   pc[0]='1'; //varianta 1
   pc++;   // varianta 2
   pc="abcd"; // varianta 3
}
Care din variante sunt corecte:
a) 1+3
b) 1+2
c) 1
d) 2+3

8. Fie aplicația:
void main( ) {
   static int a;
}
Variabila a are valoarea:
a) o valoare reziduală
b) depinde de variabilele de mediu
c) INF
d) 0

9. Secvența:
   for (i=19; i>0; i%=7) printf("%d",i);
a) afișează 19;
b) afișează: 19 5 5 5 .... la infinit;
c) conține operatori incorect folosiți;
d) afișează: 19 5;

Nr 59

1. Programul:
```
#include <stdio .h>
void main ( ) {
   int s=0, i=5;
   while(i--) s+=i;
   printf("\n %d ", s/i);
}
```
Afișează:
a) -10
b) 10
c) 0
d) este eronată (împărțire la 0)

2. Care este rezultatul evaluării expresiei
   x = x&&y||z;
în ipoteza că x=2 si y=0; (x,y,z - sunt întregi)
a) 1
b) depinde de valoarea initiala a lui z
c) 0
d) 10

3. Care dintre următoarele constante este invalidă?
a) .1E3
b) -1E3
c) 1.E3
d) E3

4. Se consideră programul:
```
#include <iostream.h>
void main ( ) {
   int a,b,c,d;
   a = b = c = d = 10;
   a += b -= c += d;
   cout << a << ' ' << b << ' ' << c << ' ' << d;
}
```
Indicati valorile afisate (a,b,c,d):
a) 20 20 20 20
b) 0 -10 20 10
c) 0 10 20 30
d) 10 10 10 10

5. Fie secvența:
   int v; if (b=0,1) v = 10; else v = 20;
a) este eronată (constanta flotantă 0,1 eronată)
b) v va conține o valoare reziduală
c) v va avea valoarea 10
d) v va avea valoarea 20

6. Expresia:
   a<b?a:b;
unde a și b sunt doi intregi, evaluează:
a) efectuează o deplasare la stânga
b) este eronată, a<b trebuie inclus între paranteze rotunde
c) maximul dintre a și b
d) minimul dintre a și b

7. Programul:
```
#include <iostream.h>
void main( )
{ static int a[10]; cout << *a++; }
```
a) incrementează incorect un pointer constant
b) afișează 0
c) nu este corect
d) asociază masivului o clasă de memorie neadec

8. Ce afișează apelul:
   printf("%d", sizeof("a"));
a) 2
b) 0
c) este generată eroare, a trebuie încadrat între apostroafe
d) 1

9. Programul:
```
#include <stdio.h>
void main ( ) {
   int x=2|3||4;
   printf("\n %d ",x);
}
```
Afisează:
a) 1
b) 4
c) 3
d) 2