

## Teste grila propuse pentru examenul de licenta 1999 la specializarea Informatica Economica, disciplina *Limbaje evaluate de programare*

1. Ce erori apar în descrierea:

```
struct copil { int virsta; char prenume[10]; };
static struct persoana {
    char marca;
    struct copil;
    struct persoana *f; } s;
```

a) cuvânt cheie eronat; b) pointer în structura (neadmis); c) referire recursiva a unei structuri; d) nume de sablon absent; e) definirea incompleta a unui câmp;

2. #include <stdio.h>

#include <conio.h>

void main( )

```
{
    int i=3;
    while( --i ) printf(" %d", i);
    getch();
}
```

Programul de mai sus afiseaza:

a) 2 1 0; b) 2 1. c) 3 3; d) 3 2; e) 3 2 1;

3. Fiind date doua clase :

```
class ex1 { int i; } ;
```

```
class ex2 : public ex1 {int j; } ;
```

Atunci în functia :

```
void main ( ) { ex1 ob1; ex2 ob2 ;ob2=ob1;}
```

Atribuirea ob2 = ob1 este posibila ?

a) nu, deoarece sunt obiecte de tipuri diferite; b) nu, deoarece nu sunt permise atribuirii între obiecte, aflate în relatii de mostenire; c) da, deoarece ob1 este obiect de baza si ob2 este obiect derivat; d) nu, deoarece trebuie folosit operatorul cast, nedefinit aici; e) da, deoarece ob2 este mai larg si-l poate contine pe ob1.

4. Secventa:

```
nota=5;
```

```
switch(nota)
```

```
{
    case 4: printf("restanta");break;
```

```
case 5: printf("promovat, dar...");
```

```
case 6:7:8: printf("mai invata!"); break;
```

```
default: printf("nimic implicit, totul cu
```

```
munca!");break;
```

```
}
```

a) afiseaza: promovat, dar... b) foloseste incorect case; c) afiseaza: promovat, dar...mai invata! d) afiseaza: nimic implicit, totul cu munca! e) contine break, care este incompatibil cu default.

5. Programul:

```
#include<stdio.h>
```

```
void main()
```

```
{ auto j; printf("\n j=%d", j); }
```

a) afiseaza zero, deoarece j a fost initializat automat; b) afiseaza o valoare reziduala, oarecare; c) este eronat, deoarece lipseste tipul lui j; d) programul contine erori de sintaxa; e) nu afiseaza nimic, deoarece pâna la initializare variabila j nu exista.

6. Marcati raspunsurile corecte privitoare la valoarea returnata în urma apelului:

```
strncmp("1234567890", "12345", strlen("12345"));
```

a) zero; b) o valoare negativa; c) 5; d) apelul este eronat deoarece compara doua constante; e) false, deoarece se compara doua adrese.

7. În urma macrodefinitiei #define PATRAT(x) (x)\*(x), apelul PATRAT(3++) genereaza:

a) o expresie cu valoarea 9; b) o expresie cu valoarea 12; c) erori în compilare; d) o expresie cu valoarea 16; e) o expresie cu alte valori.

8. Programul

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a=3,b=5;
```

```
if(a=b) printf("\n egale");
else printf("\n diferite");
}
```

a) imprima diferite; b) imprima egale; c) contine erori de sintaxa; d) imprima true; e) imprima false.

```
9. int f( int a = 0, int b = 0, int c=0){ return
a+2*b+3*c; }
void main( ) { printf("\n%d",f(1,1) ); getch(
);}
```

Programul de mai sus:

a) afiseaza 0; b) este eronat sintactic; c) afiseaza un numar depinzând de valoarea lui b; d) afiseaza 7; e) afiseaza 3.

10. Precizati care din urmatoarele cuvinte nu pot constitui identificatori în limbajul C:

a) Suma; b) then; c) an\_72; d) #suma; e) \_acesta.

11. Dupa o initializare:

```
char mesaj[]={ 'G','r','e','s','i','t' }; precizati ca-
re afirmatie este corecta:
```

a) initializarea nu este permisa în cazul masivelor de caractere; b) nu este permisa manipularea ulterioara ca sir, a variabilei mesaj; c) declaratia este eronata, deoarece lipseste dimensiunea masivului; d) initializarea este eronata deoarece nu contine terminatorul de sir '\0'; e) initializarea contine o lista incorecta de initializatori (apostrof în loc de ghilimele ).

12. Daca ps este rezultatul unei declaratii char \*ps; atunci apelul gets(ps);

a) este incorect, deoarece gets() nu este functie; b) este incorect, deoarece gets() nu poate primi parametri de intrare; c) este incorect, deoarece ps nu a fost încarcat în prealabil; d) este incorect, deoarece nu se receptioneaza pointerul returnat de gets(); e) este incorect, deoarece prototipul lui gets() este altul.

13. Afisarea unei variabile de tip double se face cu:

a) %d (prescurtare de la double); b) %lf (prescurtarea de la long float); c) %e (prescurtare de la exponential); d) alt descriptor de format, decât cele mentionate; e) %f (prescurtare de la float, double neavând descriptor propriu).

14. Apelul printf("%d", scanf("%s", sir));

a) citeste o variabila si afiseaza ce a citit; b) citeste o variabila si afiseaza zero; c) citeste un sir si afiseaza lungimea lui; d) este incorect sintactic; e) citeste o variabila si afiseaza 1 sau -1 sau -2;

15. Structura struct s { int a,\* b; struct s c; }; este:

a) eronata, deoarece contine o referire recursiva; b) eronata, deoarece contine o structura în structura; c) corecta; d) eronata, deoarece contine un pointer la int; e) eronata, deoarece nu s-au declarat variabile de tipul structurii.

16. Fie declaratia: short int x=2, y=3; float z = 1; precizati valoarea expresiei sizeof( x < y ?x :z );

a) 1; b) 2; c) 8; d) 4; e) 3

17. O functie independenta declarata friend în domeniul private dintr-o clasa are acces:

a) la toti membrii clasei; b) doar la membrii public ai clasei; c) la membrii public si protected; d) la membrii private; e) la toti membrii clasei, dar îi poate doar consulta, nu si modifica.

18. Marcati variantele care indica valorile corecte, atribuite elementelor unui masiv static x, de 5 elemente, prin secventa: for(i=0;i<5;) x[i++]=i;

a) 1,2,3,4,5 b) 0,1,2,3,4 c) 3,2,1,0,0 d) 0,0,0,0,0 e) atribuirea este incorecta

19. Declaratia char \* pa=(char\*)10519;

a) foloseste facilitatea de "cast" neadmisa pentru pointeri; b) face dubla indirectare pentru pointer, pa fiind deja pointer la char; c) modifica continutul zonei pointate, fara ca

pointerul sa fi fost încarcat cu adresa unei zone constante; d) este corecta; e) foloseste impropriu adrese impare pentru variabile pointer.

**20.** O expresie compusa din mai multi operatori de atribuire, se evalueaza:

a) de la stînga la dreapta; b) la compilare; c) la preprocesare; d) nu se admit mai multe atribuirii în aceeași expresie; e) de la dreapta la stînga.

**21.** struct sablon{int a; float b;} s[5], \*p1, \*p2;

p1=&s[1];p2=s+2;

La afisarea expresiei p2-p1 se obtine:

a) un numar ce depinde de implementarea tipului int, pe 2 sau 4 baid; b) 6; c) 2; d) sizeof(struct sablon); e) 1.

**22.** #include <conio.h>

#include <iostream.h>

class cls

{ public: void \*operator new( size\_t dim)

{ cls \*po = new cls[dim];

cout << "\n Aloca obiect"; return po;

}

};

void main( ) { cls \*po= new cls, \*pv= new cls[5]; getch(); }

În exemplul anterior, suprascrierea operatorului new afiseaza "Aloca obiect":

a) de doua ori, cîte o data pentru fiecare pointer; b) de zero ori, caci nu se foloseste niciodata în acest context varianta suprascrisa; c) o data, caci pentru vectori se foloseste varianta initila a lui new; d) de sase ori, cîte o data pentru fiecare obiect alocat; e) de zero ori, caci operatorul new nu se suprascrie.

**23.** Dupa o declaratie de forma short int \*i; expresia i+sizeof(short int) prin evaluare produce o adresa care fata de adresa initiala continuta în i,

a) este cu 2B mai mare (marimea unui int); b) este cu 8B mai mare; c) este cu 2B mai

mica; d) este cu 4B mai mare; e) este cu 4B mai mica.

**24.** Functia:

int max( int a, int b, int c )

{

static int Max;

Max = a; if( Max < b) Max = b; if( Max < c)

Max = c;

return( Max );

}

a) este o constructie ambigua datorita definirilor max, Max; b) selecteaza un element dintr-un grup de trei; c) alege elementul minim dintre cei trei parametri; d) returneaza mereu aceeași valoare statica; e) contine trei erori sintactice.

**25.** Evaluati rezultatele continute în variabilele întregi x si y, în urma executiei atribuirilor:  $y=3+2*(x=7/2)$ ;

a) atribuirea este nepermisa în aceasta forma; b) 3, 10; c) 3, 9; d) 3.5, 10; e) alte valori decât cele mentionate.

**26.** Programul

void main( ) { putchar(getchar( ) ); }

a) are parametri obligatorii absenti într-un apel de functie; b) este eronat, deoarece nu se receptioneaza valoarea de retur a functiei; c) este corect în totalitate; d) are un parametru incorect într-un apel de functie; e) este eronat, deoarece lipseste o directiva obligatorie de preprocesare.

**27.** Fie data urmatoarea structura de clase:

class B { /\*.....\*/ };

class D1:B { /\*.....\*/ };

class D2:B { /\*.....\*/ };

class M1:D1, public D2 { /\*.....\*/ };

class M2:virtual D1, virtual public D2

{ /\*.....\*/ };

Precizati care afirmatii sunt corecte:

a) clasa M2 nu va mosteni nici un obiect de tip B; b) clasa M1 va mosteni un singur obiect de tip B; c) clasa M1 nu va mosteni nici un obiect de tip B; d) clasa M2 va mos-

teni doua obiecte de tip B; e) clasa M1 va mosteni doua obiecte de tip B.

**28.** Dupa o declaratie de forma `char *pc;` operatorul `sizeof(pc)`:

a) întoarce lungimea zonei de memorie ocupata de tipul pointat (1B); b) întoarce lungimea zonei de memorie ocupata de pointer (2 sau 4 B); c) este folosit ilegal în acest context; d) este utilizat pentru determinarea lungimii (size) unui sir; e) întoarce zero, deoarece pointerul nu a fost încarcat cu o adresa.

**29.** Functia:

```
long f(int x, unsigned int y) {long r=1;
while (y) if(y%2) r*=x, y--; else y/=2, x*=x;
return r;}
```

returneaza: a) restul împartirii lui y la 2; b) x la puterea y, x si y sunt pozitivi; c) restul împartirii lui y la x; d) y la puterea x; e) alta valoare.

**30.** Fie declaratiile:

```
typedef short int a[2][7];
typedef a *B;
void main( ) { B b;}
```

Fata de o valoare initiala a lui b, b+1 înseamna o adresa mai mare cu:

a) 1 bait; b) 2 baiti; c) 4 baiti; d) 14 baiti; e) 28 baiti.

### Problema

Se considera o retea de  $n$  magazine pentru desfacerea de mobila. Cunoscand volumul valoric al desfacerilor zilnice pentru o luna, ale fiecarui magazin sa se scrie functiile pentru determinarea vanzarilor la un magazin  $i$  dat si, respectiv, pentru calculul volumului de vanzari ale intregii retele, pentru o zi  $j$ , data.