

**Laviniu Aurelian BĂDULESCU**



**Laviniu Aurelian BĂDULESCU**

**LIMBAJUL C**  
**PRIN TESTE GRILĂ DE PROGRAMARE**

**- monografie -**



**EDITURA UNIVERSITARIA**  
**Craiova, 2018**

**Referenți științifici:**

*Prof.univ.dr.ing. Emil PETRE*

*Universitatea din Craiova, Facultatea de Automatică, Calculatoare și Electronică*

*Prof.univ.dr.ing. Cătălin Cosmin IONETE*

*Universitatea din Craiova, Facultatea de Automatică, Calculatoare și Electronică*

© Laviniu Aurelian Bădulescu, 2018.

---

**Descrierea CIP a Bibliotecii Naționale a României**

**BĂDULESCU, LAVINIU AURELIAN**

**Limbajul C prin teste grilă de programare / Laviniu Aurelian Bădulescu.** - Craiova : Universitaria, 2018

Conține bibliografie

ISBN 978-606-14-1376-8

004

*Coperta 1: Danemarca, Copenhaga, Muzeul Thorvaldsens, lângă bustul lui Fibonacci*

## INTRODUCERE

*“C++ and Java, say, are  
presumably growing faster  
than plain C, but I bet C  
will still be around.”*

(Dennis Ritchie)

În februarie 2018, conform datelor de pe site-ul <https://www.tiobe.com/tiobe-index/>, în clasamentul limbajelor de programare în funcție de popularitate, locul II era ocupat de limbajul de programare C (în creștere cu 3.41%), după locul I ocupat de limbajul de programare Java (în scădere cu 1.69%). Următoarele opt locuri erau ocupate, în ordine, de: C++, Python, C#, Visual Basic .NET, PHP, JavaScript, Delphi și Ruby. Iată de ce, se impune în continuare studierea limbajului de programare C, iar cartea de față vine în ajutorul celor care doresc să facă acest lucru într-un mod plăcut, pe baza testelor grilă.

Prezenta monografie conținând teste grilă de programare este destinată tuturor celor care doresc să-și fixeze și aprofundeze competențele și cunoștințele de programare în limbajul C prin parcurgerea unor baterii de teste grilă care acoperă detaliat și cvasiexhaustiv problematica limbajului de programare C, precum și noțiuni de programarea calculatoarelor și de limbaje de programare, în general.

Ea poate fi utilizată de studenții de la facultățile tehnice, care în marea lor majoritate studiază acest limbaj, pentru pregătirea, fixarea și aprofundarea competențelor și cunoștințelor de programare, atât a celor prezentate la curs, cât și a celor practice exersate în timpul orelor de seminar sau de laborator.

Prin organizarea sa în peste 100 de baterii de teste grilă, cartea se dorește un instrument util pentru pregătirea de către studenți a examenului de la finalul disciplinei de programarea calculatoarelor.

În același timp, cartea de față poate fi utilizată de oricine dorește să învețe și să aprofundeze limbajul de programare C, de cei care nu au studiat la școală limbajul C și vor să îl învețe, sau de elevi din învățământul preuniversitar (ciclul gimnazial și liceal), unde se studiază acest limbaj ca variantă cu limbajele Pascal sau C++. Cartea poate fi utilizată de cadre didactice din învățământul universitar sau preuniversitar în pregătirea unor lecții aplicative sau de testare a cunoștințelor și competențelor de programare în limbajul C.

Imediat după parcurgerea unui test, cititorul poate verifica corectitudinea răspunsurilor folosind tabelul de la finalul cărții, de la capitolul *Răspunsuri*.

Aștept cu interes sugestiile de îmbunătățire și eventualele erori descoperite în parcurgerea materialului de față pe adresa de mail: *badulescu@automation.ucv.ro*.

Lavinii Aurelian Bădulescu

## Test 1

**T1.1.** Limbajul de programare este riguros definit?

- a. nu
- b. da
- c. doar limbajele de programare de nivel înalt
- d. doar limbajele de programare cod-mașină

**T1.2.** Ce afișează programul următor dacă se citește de la tastatura 100 ?

```
#include <stdio.h>
main() {
    int n, t = 0;
    scanf("%d", &n);
    while ( n != 0 ){
        t = t * 10 + n % 10;
        n = n / 10;
    }
    printf("%d", t);
}
```

- a. 001                      b. 1                      c. 0                      d. 111

**T1.3.** Controlul execuției programului (control flow) specifică

...

- a. modul cum se execută programul
- b. ordinea în care sunt efectuate calculele
- c. lansarea în execuție a programului sub controlul utilizatorului
- d. cele trei structuri de control

**T1.4.** Cu ce valoare trebuie să apelăm funcția  $f()$  astfel încât aceasta să returneze 10? Funcția  $g(t)$  inversează șirul  $t$ .

```
void f(int n, char t[]) {
    int j, m;
    if ( ( m = n ) < 0 )
        n = -n;
```

```
j = 0;
do {
    t[j++] = n % 10 + '0';
} while ( ( n /= 10 ) > 0 );
if ( m < 0 )
    t[j++] = '-';
t[j] = '\\0';
g(t);
}
```

a. -10    b. 10abc    c. +10    d. 100

**T1.5.** Fie declarația `int v[25], *pv;`. Care dintre cele patru variante următoare de atribuire nu are ca efect faptul că `pv` va indica adresa primului element al tabloului `v`?

a. `pv = &v[0];`            b. `pv = &v;`  
c. `pv = v[0]`              d. `pv = v;`



## Test 2

**T2.1.** Ce este vocabularul?

- a. totalitatea regulilor de scriere corectă a programelor
- b. semnificația construcțiilor corecte din punct de vedere sintactic
- c. codul programului
- d. totalitatea cuvintelor care pot fi folosite într-un limbaj de programare

**T2.2.** Ce afișează programul următor dacă se introduce 0xA<Enter>?

```
#include <stdio.h>
main() {
    int t;
    scanf("%i", &t);
    printf("%d", t);
}
```

- a. 0XA            b. 0xa            c. 10            d. A

**T2.3.** O expresie precum  $k = 1$  devine o instrucțiune atunci când ....

- a. se adaugă un egal:  $k == 1$
- b. este plasată într-un program
- c. este urmată de caracterul punct și virgulă “;”
- d. este folosită la inițializare

**T2.4.** Ce returnează funcția următoare la apelul

```
f("abc\n\t ")?
int f(char t[]) {
    int i;
    for ( i = strlen(t) - 1; i >= 0; i-- )
        if ( t[i] != '\n' && t[i] != ' ' && t[i] != '\t'
    )
        break;
    t[i + 1] = '\0';
```

```
    return i + 1;  
}
```

a. cba    b. abcnt    c. abc    d. 3

**T2.5.** Fie `int v[25], *pv;`. Care dintre cele patru variante următoare de atribuire nu este permisă?

a. `pv = &v[0];`    b. `pv = &v;`    c. `pv = v;`    d. `v = pv;`