

Mirel COȘULSCHI

Mihai GABROVEANU

Practica programării în C

Mirel COȘULSCHI

Mihai GABROVEANU

Practica programării în C



EDITURA UNIVERSITARIA
Craiova, 2014

Referenți științifici:

Prof.univ.dr. Ion Iancu

Lect.univ.dr. Petre Băzăvan

Copyright © 2014 Universitaria

Toate drepturile sunt rezervate Editurii Universitaria.

Descrierea CIP a Bibliotecii Naționale a României**COȘULSCHI, MIREL**

Practica programării în C / Mirel Coșulschi, Mihai
Gabroveau. - Craiova : Universitaria, 2014

Bibliogr.

ISBN 978-606-14-0841-2

I. Gabroveau, Mihai Laurențiu

004.43 C

Prefață

Apariția în anul 1972 a limbajului *C* a constituit un pas important în evoluția limbajelor de programare de nivel înalt. Proiectat și implementat de către Dennis Ritchie la AT&T Bell Laboratories acest limbaj a devenit unul dintre cele mai populare limbaje de programare din toate timpurile. Scopul inițial pentru care a fost realizat a fost acela de a rescrie nucleul sistemului de operare UNIX, rescriere ce a fost finalizată în anul 1973.

Popularitatea acestui limbaj a crescut exponențial începând cu anul 1978 când Brian Kernighan și Dennis Ritchie publică cartea *The C Programming Language* (reeditată în 1988 [16]). În anul 1989 American National Standards Institute (ANSI) publică primul standard pentru acest limbaj, *ANSI C* întâlnit și sub numele de *C89*. Ulterior acest standard a suferit o serie de actualizări, *C11*¹ fiind cea mai recentă versiune de standard și a fost realizată în decembrie 2011.

Având în vedere faptul că sintaxa acestui limbaj o regăsim la foarte multe dintre limbajele de programare moderne precum *C++*, *Java*, *JavaScript*, *C#*, *PHP*, putem afirma că deprinderea abilităților de a programa în limbajului *C* constituie o bază solidă către învățarea acestor limbaje.

Această carte se adresează elevilor, studenților, precum și tuturor persoanelor interesate de programarea în limbajul *C* și constituie o abordare pragmatică de a-l învăța (de tip *learning by doing*). Practic, cititorul va regăsi o veritabilă culegere de probleme rezolvate sau propuse spre rezolvare în limbajul *C*. Acolo unde am considerat că este cazul, soluțiile sunt însoțite de explicații, prezentarea conceptelor teoretice implicate, algoritmi în pseudo-cod precum și diverse comentarii legate de performanțele acestora.

Abordarea este una graduală pornind de probleme simple, continuând apoi cu probleme din ce în ce mai complexe. Multe dintre aceste probleme au fost inspirate din cerințele înaintate candidaților pentru angajare la multe companii din domeniul *IT*. În cele cinci capitole ale cărții se regăsesc probleme legate de algoritmi elementari (operații cu vectori și matrici, polinoame, șiruri de caractere, prelucrarea parametrilor din linia de comandă), algoritmi recursivi, algoritmi de sortare, lucru cu fișiere și operații cu structuri de date alocate dinamic.

La finalul lucrării am inclus o listă de referințe bibliografice pe care le recomandăm cititorilor în vederea aprofundării tainelor programării în limbajul *C* și nu numai.

Activitatea celui de-al doilea autor a fost parțial finanțată prin contractul POSDRU 159/1.5/S/133255, proiect strategic ID 133255 (2014), co-finanțat din *Fondul Social European* prin *Programul Operațional Sectorial Dezvoltarea Resurselor Umane* 2007-2013.

Craiova, iulie, 2014

Mirel Coșulschi
Mihai Gabroveanu

¹Ultima versiune disponibilă public a standardului C99 se poate consulta la adresa <http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1256.pdf>.

Capitolul 1

Algoritmi elementari

Problema 1.1 *Să se scrie un program care să folosească două variabile întregi, cărora să li se atribuie valori prin program, să se adune aceste variabile, să se calculeze media aritmetică, iar rezultatul să se depună într-o variabilă de tip real și să se afișeze.*

Soluție: Programul ce calculează media aritmetică este următorul:

Listing 1.1: media1.c

```
#include <stdio.h>

void main(void) {
    int a, b;
    int suma;
    float media;

    a = 8; b = 7;
    suma = a + b;
    media = suma / 2.0;
    printf("Media = %f\n", media);
}
```

Observația 1.1 *Deoarece operatorul '/' realizează împărțirea întreagă în cazul în care ambii operanzi sunt numere întregi, am împărțit suma la numărul real 2.0 pentru a forța realizarea împărțirii cu virgulă. Mai exact, pentru variabila **suma** având valoarea 15, dacă în locul expresiei **suma/2.0**, am fi avut **suma/2**, rezultatul ar fi fost 7 în loc de 7.5.*

□

Problema 1.2 *Să se modifice programul anterior ce calculează suma și media aritmetică a două numere întregi astfel încât valorile acestora să se citească de la tastatură.*

Soluție:

Listing 1.2: media2.c

```
#include <stdio.h>

void main(void) {
    int a, b;
    int suma;
```

```

float media;

printf("a = ");
scanf("%d", &a);
printf("b = ");
scanf("%d", &b);

suma = a + b;
media = suma / 2.0;

printf("Media = %f\n", media);
}

```

□

Problema 1.3 *Să se scrie un program de calcul a sumei și mediei a două numere reale și să se afișeze rezultatele:*

- fără format;
- cu format.

Soluție:

Listing 1.3: media3.c

```

#include <stdio.h>

void main(void) {
    float x, y;
    float suma, media;

    printf("x = "); scanf("%f", &x);
    printf("y = "); scanf("%f", &y);

    suma = x + y;
    media = suma / 2;

    printf("Media = %f \n", media);
    printf("Media = %2.4f \n", media);
}

```

□

Problema 1.4 *De pe mediul de intrare (de la tastatură) se citește valoarea unui unghi exprimat în grade. Să se calculeze și să se afișeze unghiul în radiani.*

Problema 1.5 *Să se calculeze și să se afișeze valoarea distanței dintre două puncte, A și B, din plan dându-se coordonatele acestora, $A(x_1, y_1)$ și $B(x_2, y_2)$.*

Soluție: Reamintim formula de calcul a distanței euclidiene dintre două puncte din plan:

$$d(A(x_1, y_1), B(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Pe baza acesteia, avem următorul program de calcul:

Listing 1.4: dist2pct.c

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

void main(void) {
    float x1, y1, x2, y2;
    float d;

    printf("x1 = "); scanf("%f", &x1);
    printf("y1 = "); scanf("%f", &y1);
    printf("x2 = "); scanf("%f", &x2);
    printf("y2 = "); scanf("%f", &y2);

    d = sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
    printf("Distanța dintre (%g,%g) și (%g,%g) este %g \n", x1, y1, x2, y2, d);
}
```

□

Problema 1.6 *Se citesc de la tastatură trei valori întregi a, b, c . Să se afișeze, în ordine crescătoare, valorile variabilelor, fără modificarea conținutului acestora.*

Soluție: După ce se citesc cele trei valori, programul va compara valorile acestora și va afișa una din următoarele secvențe:

```
a b c
a c b
c a b
c b a
b a c
b c a
```

Listing 1.5: perm3.c

```
#include <stdio.h>

void main(void) {
    int a, b, c;

    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);
    printf("c = "); scanf("%d", &c);

    if (a < b)
        if (b < c)
            printf("%d %d %d\n", a, b, c);
        else
            if (a < c)
                printf("%d %d %d\n", a, c, b);
            else
                printf("%d %d %d\n", c, a, b);
    else
        if (a < c)
```

```

    printf("%d %d %d\n", b, a, c);
else
    if (b < c)
        printf("%d %d %d\n", b, c, a);
    else
        printf("%d %d %d\n", c, b, a);
}

```

□

Problema 1.7 Să se scrie un program de rezolvare, cu discuție, a ecuației de gradul I ($ax + b = 0$), valorile lui a și b fiind citite de la tastatură (a, b valori reale).

Indicație: Un algoritm simplu ce rezolvă ecuația de gradul I, $ax + b = 0, a, b \in \mathbb{R}$ este algoritmul 1. Acesta se bazează pe rezolvarea matematică generală a ecuației de gradul I:

1. dacă $a = 0$, atunci ecuația devine $0 \cdot x + b = 0$. Această egalitate este satisfăcută doar dacă $b = 0$, relația fiind adevărată pentru orice valoare a lui $x \in \mathbb{R}$. În acest caz ecuația se numește *ecuație compatibil nedeterminată*. Pentru cazul în care $b \neq 0$ egalitatea nu poate fi satisfăcută pentru nici o valoare a lui $x \in \mathbb{R}$. Spunem în acest caz că *ecuația este incompatibilă*.
2. dacă $a \neq 0$, ecuația are o singură soluție, $x = -\frac{b}{a}$.

Alg. 1 Algoritm pentru rezolvarea ecuației de gradul I

```

1: Input { $a, b$ }
2: if ( $a = 0$ ) then
3:     if ( $b = 0$ ) then
4:         Output { 'Ecuatie compatibil nedeterminata' }
5:     else
6:         Output { 'Ecuatie incompatibila' }
7:     end if
8: else
9:      $x \leftarrow -\frac{b}{a}$ 
10:    Output { 'Solutia este:',  $x$  }
11: end if

```

Problema 1.8 Se citesc de la tastatură trei numere reale a, b și c reprezentând coeficienții unei ecuații de gradul al II-lea, $ax^2 + bx + c = 0$. Să se realizeze un program ce calculează rădăcinile acesteia.

Soluție: Rezolvarea ecuației de gradul al II-lea subsumează și rezolvarea ecuației de gradul I (dacă $a = 0$ atunci ecuația devine $bx + c = 0$). Programul ce determină valorile rădăcinilor unei ecuații de gradul al doilea cu coeficienți reali este:

Listing 1.6: ecgr2sol.c

```

#include <stdio.h>
#include <math.h>

void main(void) {
    float a, b, c;

```

```

float x1, x2, x;
float delta, preal, pimag;

printf("a = "); scanf("%f", &a);
printf("b = "); scanf("%f", &b);
printf("c = "); scanf("%f", &c);

if (a == 0) {
    printf("Ecuatie de gradul I\n");
} else {
    delta = b*b - 4*a*c;
    if (delta >= 0) {
        printf("Radacini reale: ");
        x1 = (-b + sqrt(delta))/(2*a);
        x2 = (-b - sqrt(delta))/(2*a);
        printf("x1 = %f, x2 = %f\n", x1, x2);
    } else {
        printf("Radacini complexe: ");
        delta = -delta;
        preal = -b/(2*a);
        pimag = sqrt(delta)/(2*a);
        printf("x1 = %f + i*%f x2 = %f - i*%f \n", preal, pimag, preal, pimag);
    }
}
}

```

□

Problema 1.9 Realizați un program ce rezolvă următorul sistem de ecuații cu coeficienți reali:

$$\begin{cases} ax + by = c \\ mx + ny = p \end{cases} \quad (1.1)$$

Indicație: Pentru a obține soluția sistemul de ecuații 1.1 se calculează următorii determinanți:

$$\begin{aligned} D &= \begin{vmatrix} a & b \\ m & n \end{vmatrix} = an - bm \\ D_x &= \begin{vmatrix} c & b \\ p & n \end{vmatrix} = cn - bp \\ D_y &= \begin{vmatrix} a & c \\ m & p \end{vmatrix} = ap - cm \end{aligned}$$

Pentru $D \neq 0$, soluțiile sistemului de ecuații sunt date de următoarele formule:

$$x = \frac{D_x}{D}, y = \frac{D_y}{D}. \quad (1.2)$$

Problema 1.10 Să se realizeze un program pentru calculul lui $n!$ ($n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$), unde $n \in \mathbb{N}$ și $0 < n < 13$.

Soluție: Având în vedere că valoarea lui $n!$ crește foarte repede:

$1! = 1$, $2! = 2$, $3! = 2 \times 3 = 6$, $4! = 24$, $5! = 120$, $6! = 720$
 $7! = 5040$, $8! = 40320$, $9! = 362880$, $10! = 3628800 \dots$

vom utiliza pentru reținerea produsului o variabilă de tip *unsigned long int* ce ne permite păstrarea unor valori mult mai mari (maxim $2^{32} - 1 = 4.294.967.295$) spre deosebire de tipul *int* care ne-ar fi permis memorarea unei valori maxime de $2^{15} - 1 = 32.767$.

Observația 1.1 *Considerațiile anterioare sunt valabile pentru variabile având tipul int ce sunt păstrate pe 2 octeți, respectiv variabile având tipul long păstrate pe 4 octeți de memorie.*

Prezentăm în continuare două variante echivalente de implementare:

Varianta 1 - Implementare utilizând o instrucțiune repetitivă **for**:

Listing 1.7: calculnfactv1.c

```
#include <stdio.h>

void main(void) {
    int n,i;
    unsigned long produs;

    printf("n = "); scanf("%d", &n);
    produs = 1;
    for (i = 1; i <= n; i++)
        produs = produs * i;

    printf("%d! = %lu\n", n, produs);
}
```

Varianta 2 - Implementare utilizând o instrucțiune repetitivă **while**:

Listing 1.8: calculnfactv2.c

```
#include <stdio.h>

void main(void) {
    int n,i;
    unsigned long produs;

    printf("n = "); scanf("%d", &n);
    produs = 1;
    i = 1;
    while (i <= n) {
        produs = produs * i;
        i++;
    }

    printf("%d! = %lu\n", n, produs);
}
```

□

Operații cu secvențe

Problema 1.11 *Să se calculeze suma primelor n numere naturale ($n \in \mathbb{N}^*$).*