

Pentru nici una din problemele care se vor rezolva la disciplina
Programarea Calculatoarelor
NU SE FOLOSESC VARIABLE GLOBALE

ELEMENTE INTRODUCTIVE ALE LIMBAJULUI C

1. PRIMUL PROGRAM ÎN C

Programele pe care le scriem în **C** lucrează cu variabile, constante, funcții și instrucțiuni.

În variabile stocăm diferite informații care se pot modifica pe parcursul execuției programului (de aici și numele de **variabilă**), constantele sunt valori care nu se modifică pe parcursul execuției programului.

Funcțiile sunt porțiuni de cod (similare rutinelor din limbajul de asamblare) care pot face o prelucrare bine determinată a unor informații (de exemplu putem avea o funcție care, având la dispoziție lungimile laturilor unui triunghi, să calculeze aria triunghiului). Pentru a putea scrie o funcție trebuie să specificăm informațiile pe care le are la dispoziție o funcție pentru a face o anumită prelucrare. Aceste informații le vom numi în continuare date de intrare pentru funcție sau parametri de intrare ai funcției (sunt similare argumentelor funcțiilor din matematică, ele fiind din domeniul funcției). De asemenea, trebuie să cunoaștem ce rezultat avem după ce funcția face toate calculele (trebuie specificat codomeniul funcției). De fapt, pentru a putea scrie sau a folosi o funcție trebuie să cunoaștem ce fel de rezultat așteptăm de la funcție (Valoarea rezultată este de tip întreg sau real?). Pe măsură ce înaintăm în cunoașterea limbajului, vom vedea că pot exista funcții care nu calculează vreun rezultat.

Programele scrise în **C** respectă următorul principiu: nu lucrează decât cu lucruri cunoscute.

Din acest motiv înainte de a folosi o variabilă trebuie să specificăm ce fel de variabilă este. Acest lucru se face indicând tipul (întreg sau real) variabilei (sau variabilelor cu care lucrăm).

Exemple:

a) dacă vrem să lucrăm cu o variabilă de tip întreg pe care să o denumim **a** vom scrie:

```
int a;
```

b) dacă vrem să lucrăm cu o variabilă de tip real în simplă precizie care să aibă numele **ma**:

```
float ma;
```

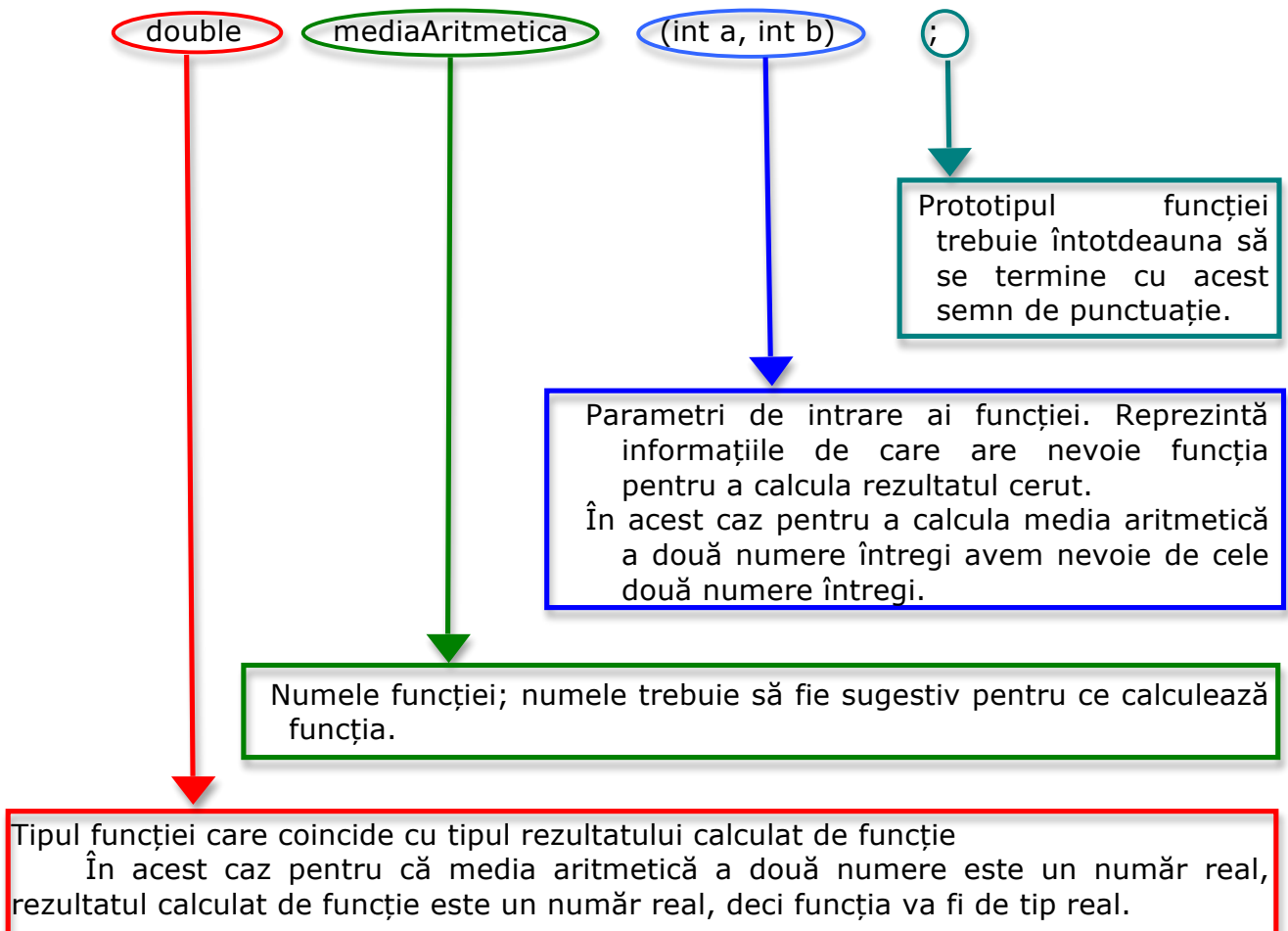
Pentru o variabilă de tip real în dublă precizie vom avea:

```
double ma;
```

În cazul unei funcții, indicarea modului în care vom lucra cu funcția respectivă înseamnă să indicăm informațiile pe care funcția trebuie să le primească la intrare (care se numesc parametri de intrare) și tipul rezultatului pe care îl calculează funcția. De exemplu dacă vom lucra cu o funcție care calculează media aritmetică a două numere întregi vom scrie prototipul (modelul) funcției astfel:

```
double mediaAritmetica(int a, int b);
```

În care:



TEMA 1

Problema nr. 1.1.

Să se scrie un program care citește de la tastatură trei numere întregi, folosește o funcție pentru a calcula maximul lor și afișează, în main, valoarea maximului.

Problema nr. 1.2.

Să se scrie un program care citește de la tastatură cinci numere întregi (nu se vor folosi vectori), folosește o funcție pentru a determina media lor aritmetică și afișează, în main, rezultatul.

Problema nr. 1.3.

Fie următoarea funcție:

$$f(x) = \begin{cases} x^2 + 3x + 5, & x \in (-\infty, 2] \\ 3x, & x \in (2, 8) \\ e^x + 2, & x \in [8, \infty) \end{cases}$$

Să se scrie o funcție **C** care calculează valoarea funcției *f* pentru o valoare a lui *x* dată.

Să se scrie un program care citește o valoare reală de la tastatură, calculează (apelând funcția scrisă) valoarea funcției în punctul dat și afișează (din main) rezultatul.

Problema nr. 1.4.

Să se scrie o funcție **heron** care calculează aria triunghiului folosind formula lui Heron.

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

unde p este semiperimetrul triunghiului, iar a , b și c sunt lungimile celor trei laturi ale triunghiului.

Să se scrie un program prin care se citesc de pe dispozitivul de intrare 3 numere întregi (care reprezintă lungimile laturilor unui triunghi), apelează funcția **heron** și afișează (din main) rezultatul obținut.

Observație: pentru calculul radicalului se folosește funcția **sqrt** al cărui prototip se găsește în fișierul header **math.h**, iar la link-editare trebuie inclusă biblioteca matematică.

Problema nr. 1.5.

Să se scrie o funcție care calculează, folosind algoritmul lui Euclid (**prin împărțiri repetate**), cmmdc a două numere întregi pozitive primite ca parametri.

Să se scrie un program care citește de la tastatură două numere întregi pozitive, folosește funcția scrisă pentru calculul cmmdc și afișează (din main) rezultatul obținut.

Problema nr. 1.6.

Să se scrie un program care citește de la tastatură trei numere întregi și determină dacă aceste numere pot reprezenta lungimile laturilor unui triunghi. În caz afirmativ, programul trebuie să determine și tipul triunghiului (isoscel sau dreptunghic sau echilateral sau oarecare). (Rezolvarea va cuprinde funcții pentru determinarea calității de triunghi, triunghi echilateral, triunghi isoscel sau triunghi dreptunghic).

TEMA 2

Problema nr. 2.1.

Se citesc de pe dispozitivul de intrare 3 numere reale care reprezintă coeficienții unei ecuații de gradul 2. Să se determine rădăcinile acestei ecuații. În cazul în care ecuația nu poate fi rezolvată se vor afișa mesaje corespunzătoare.

Problema nr. 2.2.

Un punct din plan este dat prin coordonatele sale (x, y) . Să se stabilească poziția punctului prin indicarea cadranelor (1, 2, 3, 4) în care este plasat. Pentru un punct situat pe una din semiaxe se vor preciza cadranele separate de semiaxa respectivă (de exemplu 2-3).

Problema nr. 2.3.

Să se calculeze durata unei conexiuni Internet cunoscându-se momentul conectării (dat prin oră, minut și secundă) și momentul deconectării (de asemenea dat prin oră, minut și secundă). Trebuie să se ia în considerație cazul în care o conexiune începe într-o zi și se încheie în ziua următoare. **Atenție: rezultatul trebuie afișat sub forma ore, minute, secunde**, deci pentru calcul NU se va transforma totul în secunde.

Problema nr. 2.4.

Să se calculeze momentul întreruperii unei conexiuni Internet dacă se cunoaște momentul de conectare (dat prin oră, minut și secundă) și durata totală a conexiunii

(dată prin ore, minute și secunde). Se ia în considerație și cazul în care o conexiune începută într-o anumită zi se poate încheia în ziua următoare. **Atenție: rezultatul trebuie afișat sub forma ore, minute, secunde**, deci pentru calcul NU se va transforma totul în secunde.

TEMA 3

Problema nr. 3.1.

O pereche de numere naturale a și b se numesc prietene dacă suma divizorilor unuia din numere este egală cu celălalt număr (și invers). De exemplu, 220 și 284 sunt numere prietene deoarece:

1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110 sunt divizori ai numărului 220 și suma lor este 284

1, 2, 4, 71, 142 sunt divizori ai numărului 284 și suma lor este 220.

Să se scrie un program care găsește toate numerele prietene dintr-un interval dat de numere naturale $[x, y]$. Cele două numere care reprezintă capetele intervalului sunt citite de la tastatură și programul face și validarea datelor (verifică corectitudinea datelor de intrare). Nu se vor folosi tablouri.

Indicații:

În afara funcției **main** se vor mai scrie două funcții ale căror prototipuri se vor indica într-un fișier header:

1. o funcție care primește ca parametru un număr natural și returnează suma divizorilor numărului.
2. o funcție având ca parametri două numere naturale și care returnează 1 sau 0 după cum cele numere sunt sau nu prietene.

Problema nr. 3.2.

Numerele naturale pot fi clasificate în: deficiente, perfecte sau abundente după cum suma divizorilor este mai mică, egală sau mai mare decât valoarea numărului. Astfel:

$n = 12$ este abundent deoarece are suma divizorilor $sd = 1+2+3+4+6 = 16 > 12$,

$n = 6$ este perfect pentru că $sd = 1+2+3 = 6$,

$n=14$ este deficient deoarece $sd = 1 + 2 + 7 = 10 < 14$.

Scrieți un program care citește două valori naturale x și y și, în funcție de opțiunea utilizatorului, afișează toate numerele perfecte, abundente sau deficiente din intervalul $[x, y]$. Programul trebuie să conțină o secvență de validare a datelor. Nu se vor folosi tablouri.

Indicații:

În afara funcției **main** se vor mai scrie două funcții ale căror prototipuri se vor indica într-un fișier header:

1. o funcție care primește ca parametru un număr natural și returnează suma divizorilor numărului.
2. o funcție având ca parametru un număr natural, funcție care returnează -1, 1 sau 0 după cum numărul transmis ca parametru este deficient, perfect sau abundent.

Problema nr. 3.3.

Se citesc de la tastatură un număr necunoscut de numere întregi nenule terminate cu caracterul CTRL/Z. Să se stabilească dacă acestea:

- formează un șir strict crescător (tip șir = 6)
- formează un șir crescător (tip șir = 5)

- formează un şir strict descrescător (tip şir = 2)
- formează un şir descrescător (tip şir = 3)
- sunt identice (şirul este constant) (tip şir = 4)
- nu sunt ordonate (tip şir = 1)

Programul are posibilitatea de a analiza mai multe seturi de date (va exista posibilitatea de reluare a programului).

Observații:

1. Numerele citite **nu** se vor memora într-un vector (tablou).
2. Caracterul CTRL/D se obține prin apăsarea simultană a tastelor **Ctrl** și **D** și are semnificație de sfârșit de fișier asociat tastaturii. Se va consulta **Help**-ul pentru a vedea cum se comportă funcția **scanf** în cazul întâlnirii sfârșitului de fișier. Pentru Windows sfârșitul de fișier este indicat prin CTRL/Z.

Indicații:

Între două elemente ale şirului se pot stabili următoarele relații de ordine:

- mai mic
- egal
- mai mare

Se determină numărul de relații de ordine de fiecare tip și în funcție de acest număr se stabilește tipul şirului astfel:

- a. numai relații mai mic - şir strict crescător
- b. relații mai și relații egal - şir crescător
- c. numai relații egal - şir constant
- d. relații egal și relații mai mare - şir descrescător
- e. numai relații mai mare - şir strict descrescător
- f. toate tipurile de relații - şir oarecare

În afara funcției **main** se vor mai scrie două funcții ale căror prototipuri se vor indica într-un fișier header:

1. o funcție care nu are nici un parametru și returnează o valoare în funcție de tipul şirului analizat. Această funcție citește și analizează şirul.
2. o funcție care primește ca parametru un număr natural reprezentând tipul şirului și afișează mesajul corespunzător.