

# Laborator 1. Programe simple; variabile și constante, operații de intrare/ieșire formatare

1.1 Scrieți un program care declară câte o variabilă pentru fiecare tip fundamental disponibil în limbajul C (exluzând tipul `void`).

1.2 Scrieți un program care declară câte o variabilă pentru fiecare tip fundamental disponibil în limbajul C (exluzând tipul `void`), și apoi le inițializează, folosind operatorul de atribuire, cu valori arbitrare (orice valoare de tipul respectiv de dată).

1.3 Scrieți un program care declară și inițializează o variabilă întreagă fără semn, o variabilă de tip caracter fără semn, o variabilă întreagă fără semn cu reprezentare lungă, și o variabilă reală cu dublă precizie cu reprezentare lungă, cu valori corespunzătoare, arbitrare.

1.4 Scrieți un program care să exploreze fișierele antent `limits.h` și `float.h`, și să afișeze\* domeniul de cuprindere pentru tipurile fundamentale de bază, precum și variațiile date de modificatorii de tip.

\*studiați funcția de I/O `printf()`.

## Exemplu ieșire:

```
char -> val_minima : val_maxima
```

```
int -> val_minima : val_maxima
```

etc.

1.5 Scrieți un program care declară și inițializează cu valori arbitrare câte o variabilă de tip `double`, `int` și `char`. Apoi, declarați și inițializați un pointer către fiecare dintre cele trei variabile. Programul dvs. va afișa atât adresa, cât și valoarea stocată pentru fiecare dintre cele șase variabile (cele trei variabile inițiale și cei trei pointeri corespunzători). Utilizați specificatorul de formatare potrivit pentru a afișa adresele în format hexazecimal.

Exemplu de adresă: „0xbfe55918”. Caracterele inițiale „0x” indică faptul că se utilizează notația hexazecimală; restul cifrelor reprezintă adresa în sine. Utilizați „%f” pentru a afișa o valoare de tip real (`double`).

## Exemplu de ieșire:

Adresa variabilei `char` este 0x\_\_\_\_\_

Adresa variabilei `int` este 0x\_\_\_\_\_

Adresa variabilei `double` este 0x\_\_\_\_\_

Adresa pointerului către `char` este 0x\_\_\_\_\_

Adresa pointerului către `int` este 0x\_\_\_\_\_

Adresa pointerului către `double` este 0x\_\_\_\_\_

Valoarea variabilei `char` este \_\_\_\_\_

Valoarea variabilei `int` este \_\_\_\_\_

Valoarea variabilei `double` este \_\_\_\_\_

Valoarea pointerului către `char` este 0x\_\_\_\_\_

Valoarea pointerului către `int` este 0x\_\_\_\_\_

Valoarea pointerului către `double` este 0x\_\_\_\_\_\*

1.6 Scrieți un program care să interschimbe valorile a două variabile de tip întreg folosind strict pointeri. Se vor afișa pe ecran valorile inițiale, urmate de valorile interschimbate.

#### Intrare

var1: 10  
var2: 20

#### Ieșire

var1: 20  
var2: 10

1.7 Scrieți un program care să citească de la tastatură un caracter folosind funcția `getchar()` și apoi să îl afișeze în consolă folosind funcția `putchar()`.

1.8 Scrieți un program care să citească de la tastatură un șir de caractere de dimensiune maximă 255 caractere folosind funcția `fgets()` și apoi să îl afișeze în consolă folosind funcția `puts()`.

1.9 crieți un program care să citească de la tastatură un șir de caractere de dimensiune maximă 255 caractere, o valoare întreagă, și o valoare reală. Se va afișa în consolă fiecare dată, în ordinea citirii, pe cate o linie nouă.

1.10 Scrieți un program care să citească de la tastatură o valoare reală, și o afișează în consolă cu doar 4 zecimale.

1.11 Scrieți un program care să citească de la tastatură o valoare întreagă, și o afișează în consolă formatând ieșirea cu specificatorul de precizie de valoare 10.

1.12 Scrieți un program care să citească de la tastatură o valoare întreagă, și o afișează în consolă formatând ieșirea cu specificatorul pentru mărimea minimă a câmpului de valoare 10, împreună cu o aliniere la stâng.

1.13 Scrieți un program care să citească de la tastatură două valori întregi, și o valoare reală, printr-o singură apelare a funcției `scanf()` și afișează pe ecran numărul de date citite de la tastatură.

**Intrare:**

10 20 1.0

**Ieșire:**

3