

Laborator 1 – Limbaje de Programare

1 Compilare și rulare

Programele C nu pot fi executate ca atare de către procesor calculatorului. Ele trebuie traduse în instrucțiuni pe care procesorul le înțelege și le poate executa direct: în cod mașină. Programul care face această traducere, din cod sursă în cod mașină, este un compilator. Pornind de la unul sau mai multe fișiere sursă care conțin programul scris în limbajul C, el creează un fișier executabil care poate fi rulat pe calculator.

Compilatorul **gcc** (GNU Compiler Collection) este unul din cele mai performante compilatoare pentru limbajul C. În cadrul acestui laborator, după posibilități, vom lucra ori direct cu compilatorul **gcc**, din linia de comandă, ori indirect, și anume cu un mediu integrat de dezvoltare (IDE – Integrated Development Environment) care folosește **gcc** pentru compilare.

Fișierele sursă C au în mod convențional extensia **.c**.

Dacă lucrăm direct din linia de comandă, cea mai simplu mod de a compila programul **nume.c** este

gcc nume.c

În acest caz, compilatorul creează un fișier executabil numit **a.out** (prin convenție). Orice fișier executabil poate fi rulat dând ca și comandă numele lui, în acest caz:

./a.out

unde numele fișierului a fost precizat în raport cu directorul curent **.** (punct).

Putem specifica mai multe opțiuni de compilare

gcc opțiuni nume.c

De exemplu, adesea dorim să precizăm un nume (altul decât **a.out**) pentru fișierul executabil creat. Putem face aceasta cu opțiunea **-o numeexecutabil**. De regulă, alegem același nume ca și al fișierului sursă, fără extensia **.c** [În UNIX/Linux, fișierele executabile nu sunt identificate prin extensie (cum ar fi **.exe** în Windows), ci printr-un atribut (permisiune de execuție) care în acest caz e setat automat de compilator. În schimb, când compilăm din linia de comandă în Windows, numele executabilului va avea extensia **.exe**, de exemplu **nume.exe**]

gcc -o nume nume.c

și rulăm atunci programul cu comanda:

./nume

Opțiunea **-Wall** (Warnings: all) generează avertismente pentru multe situații de folosire a limbajului care constituie potențiale erori în program. E recomandabil să compilăm deci cu comanda

gcc -Wall -o nume nume.c

Opțiunea **-std=c99** specifică folosirea standardului ISO (1999), care oferă facilități suplimentare față de versiunile anterioare. Bibliotecile suplimentare necesare la compilare se specifică prin opțiunea **-l** urmată, fără spațiu, de numele bibliotecii (fără prefixul **lib** folosit convențional pentru biblioteci). De exemplu, pentru a compila programe cu funcții matematice folosiți opțiunea **-lm**.

Opțiunile de compilare pot fi date în orice ordine. Nu despărțim în două opțiunea **-o nume** intercalând alte opțiuni, ea este o singură opțiune formată din două cuvinte.

Pentru programe compuse, se indică toate fișierele sursă pe linia de comandă:

```
gcc opțiuni fișier1.c fișier2.c fișierN.c
```

2 Lucrarea 1

Să ne amintim exemplele de la curs:

$$f : \mathbb{Z} \rightarrow \mathbb{Z}, f(x) = 2x + 1$$

va fi în C:

```
int f( int x){
    return 2*x+1;
}
```

$$\text{medie} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}, \text{medie}(x, y) = \frac{x + y}{2}$$

va fi în C:

```
double medie ( double x, double y){
    return (x+y)/2;
}
```

Presupunând că am dori să și tipărim media a două numere reale, programul nostru ar arăta ca mai jos.

```
#include <stdio.h>

double medie ( double x, double y){
    return (x+y)/2;
}

int main(void){
    printf("%f", medie(1.5, 2));
    return 0;
}
```

Considerăm funcțiile matematice următoare:

- $f : \mathbb{Z} \rightarrow \mathbb{Z}, f(x) = x^2 - 7x + 4$
- $g : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}, g(x, y) = 2x - 5y + 1$
- $h : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}, h(x, y, z) = \frac{x}{3} + \frac{y}{2} - \frac{z^3}{5}$
- $d : \mathbb{R} \times \mathbb{Z} \rightarrow \mathbb{R}, d(x, y) = \frac{x}{y + 1.5}$
- $k : \mathbb{R} \rightarrow \mathbb{R}, d(x) = \frac{x^2}{9}$
- $m : \mathbb{N} \rightarrow \mathbb{R}, m(x) = \frac{x}{7}$

Se cere:

1. după modelul de la curs, implementați funcțiile de mai sus ca funcții C
2. apelați funcțiile în **main**, dându-le diverse constante numerice drept parametri efectivi și tipăriți rezultatele obținute
3. declarați cel puțin o variabilă de un tip întreg și cel puțin o variabilă de un tip real, citiți de la intrare valori de tipurile respective, apelați funcțiile pe care le-ați implementat dându-le ca parametri efectivi valorile citite de la intrare și tipăriți rezultatele obținute
4. încercați să creați funcții noi prin compunerea a 2 sau mai multe din cele pe care le-ați implementat deja (cum ar fi, de exemplu, $f \circ g$ sau $k \circ d$), apelați-le pentru valori constante și pentru valori citite de la tastatură, tipăriți rezultatele obținute