

Laborator 2

1. Structura unui program C

Un program în limbajul C se compune din una sau mai multe funcții. Dintre acestea una este funcția principală și are numele *main*. Un program în C se lansează cu prima instrucțiune a funcției *main*.

Exemplu:

```
#include <stdio.h>
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

2. Funcția standard printf

Funcția `printf` este utilizată pentru a scrie date la dispozitivul standard de ieșire, sub un anumit format. Prototipul funcției se găsește în fișierul `stdio.h`. Funcția convertește, formatează și afișează argumentele conform formatului specificat și returnează numărul de caractere afișate.

Se apelează astfel:

```
printf(control, param1, param2, . . . , paramn)
```

unde:

`control` – este un șir de caractere care conține textele de scris și specificatorii de format pentru datele care se scriu; textele din acest parametru sunt copiate la ieșirea standard

`param1, . . . , paramn` – sunt expresii. Valorile lor se convertesc și se scriu conform specificatorilor de format prezenți în parametrul de control.

Parametrul de control este inclus între ghilimele, iar numărul specificatorilor de format trebuie să coincidă cu numărul parametrilor `param1, . . . , paramn`.

În cazul în care funcția va afișa doar text, parametrul de control nu conține niciun specificator de format, iar `param1, . . . , paramn` sunt absenți.

Un specificator de format începe cu un caracter procent și se termină cu una sau două litere. Între procent și prima literă, un specificator de format mai poate conține opțional:

- un caracter “-“

Implicit, datele sunt aliniate în dreapta câmpului în care ele se scriu. Dacă este prezent caracterul “-“ atunci data corespunzătoare lui se aliniază la stânga.

- un șir de cifre care definește lungimea minimă a câmpului afectat datei respective

Literele care pot fi utilizate într-un specificator de format sunt:

d	valori de tip int
o	valori în format octal
x	valori în format hexazecimal
u	valori de tip unsigned
ld	valori de tip longint
c	char
s	șiruri de caractere
f	valori de tip float
e	float în format exponențial
g	alege %f sau %e – cel care se reprezintă pe un număr minim de caractere

Pentru formatarea ieșirii se mai utilizează *secvențe escape* care sunt precedate de caracterul “\” (backslash) ce conțin caractere neafișabile.

revenire cu un pas (backspace)	‘\b’
retur de car	‘\r’
transformare de format	‘\f’
tabulator	‘\t’
newline	‘\n’
NULL	‘\0’

Exemple:

```
printf("Invat limbajul C");
```

Funcția are un singur argument - parametrul de control . Acesta conține text care se va afișa ca atare pe ecran.

```
printf("Valoarea este %d", a);
```

Funcția are un două argumente - parametrul de control și expresia a. Parametrul de control conține textul de afișat și un specificator de format (%d) ce reprezintă formatul cu care se va afișa valoarea variabilei a. Dacă variabila a are valoarea 4, la execuție se va afișa pe ecran

Valoarea este 4

```
printf("Valoarea este %4d", a);
```

Dacă variabila a are valoarea 4, se va afișa pe ecran:

```
Valoarea este    4
```

```
printf("%04d", a);
```

Dacă variabila a are valoarea 4, se va afișa pe ecran:

```
0004
```

3.Probleme

1. Să se scrie un program în C care afișează numele studentului, grupa și specializarea la care sunteți. Afișarea se va face pe 3 rânduri.

2. Să se scrie un program în C care afișează următoarele informații:

NrMatricol	Student	Media
111	Popescu Ion	10
2	Ionescu Ana	9.75
35	Popa Dan	8.45
4	Anghel Mihai	7.33
145	Croitoru Andrei	6.00

3. Afișați valoarea 123 in octal și hexazecimal.

4.Declararea variabilelor

Sintaxa generală:

```
<tip> <lista_declaratori>;
```

Exemple:

```
int a, oVariabila = 0;
float element;
```

5.Funcția standard scanf

Funcția `scanf` este utilizată pentru citirea datelor de la tastatura și conversia lor conform formatului specificat. Prototipul funcției se găsește în fișierul `stdio.h`.

Se apelează astfel:

```
scanf(control, param1, param2. . . ., paramn);
```

unde:

`control` – este un șir de caractere ce conține specificatorii de format pentru datele care se citesc

`param1, . . . , paramn` definesc zonele receptoare ale datelor citite prin intermediul funcției `scanf`. Fiecare dintre acești parametri trebuie să fie de forma:

```
&nume
```

și reprezintă adresa zonei de memorie rezervată variabilei `nume`.

Ca și la funcția `printf`, parametrul de control este inclus între ghilimele, iar numărul specificatorilor de format trebuie să coincidă cu numărul parametrilor `param1, . . . , paramn`.

Exemplu:

Fie declarațiile:

```
int i;  
long n;  
float x;
```

Prin funcția `scanf` apelată mai jos se citesc date care se atribuie acestor variabile:

```
scanf("%d%ld%f", &i, &n, &x);
```

Datele sunt atribuite în ordinea citirii. Astfel, prima valoare introdusă va fi atribuită lui `i`, a doua lui `n`, iar a treia lui `x`.

6.Probleme

4. Să se scrie un program care citește de la tastatură un număr întreg și îl afișează pe ecran.
5. Să se scrie un program care citește de la tastatură două numere reale și le afișează pe ecran.

7.Operatori aritmetici

+	adunare
-	scădere
*	înmulțire
/	împărțire
%	modulo

8.Probleme

6. Să se scrie un program care citește de la tastatură două numere întregi și afișează suma și media lor.
7. Să se scrie un program care citește de la tastatură latura unui pătrat și afișează perimetrul și aria pătratului.
8. Să se scrie un program care citește de la tastatură laturile unui dreptunghi și afișează perimetrul și aria dreptunghiului.
9. Să se scrie un program care citește de la tastatură laturile unui triunghi și afișează perimetrul și aria triunghiului
10. Să se scrie un program în C care afișează următoarele informații:

Student: *Nume Prenume*

NrCrt	Proba	Nota
-------	-------	------

1	Scrisa	9
---	--------	---

2	Practica	8
---	----------	---

Media: 9.50

Numele studentului și notele vor fi preluate de la tastatură. Media se calculează.