

PROGRAMARE PROCEDURALĂ

- lab. 2 -

1. Reamintim

Tipuri de date:

- char
- int
- float
- double
- bool

Modificatori de tip:

- short
- long
- signed
- unsigned

Declararea unei variabile:

```
tip_data nume_variabila;
```

Declararea unei functii:

```
tip_data1) nume_functie (lista_parametri2)) {  
    corpul_functiei  
    return3) rezultat;  
}
```

¹⁾ tipul de data pentru functii poate fi **void**, caz in care functia nu returneaza un rezultat;

²⁾ lista_parametri poate fi vida sau poate fi de forma:

```
tip_data1 nume_parametru1, tip_data2 nume_parametru2, ...
```

³⁾ **return** este cuvânt cheie.

Citiri/scrieri de la tastatura:

- Libraria <stdio.h> trebuie inclusa pentru a folosi functiile de citire, respectiv scriere de la tastatura **scanf**, **printf**.

- Declaratia functiei scanf:

```
int4) scanf ( const1) char * format2), lista_parametri3) );
```

- citește date din stdin (standard input) și le memorează conform parametrului format în locațiile de memorie indicate în lista de parametri.

¹⁾ cuvântul cheie **const** se folosește pentru a declara variabile constante, cele care nu își schimbă valoarea;

²⁾ format reprezintă un sir de caractere constant de forma : %[dimensiune][modificator]tip, unde

dimensiune	reprezintă numărul maxim de caractere ce pot fi citite în operația curentă de citire
modificator	h : short int (pentru d,i și n) sau unsigned short int (pentru o, u și x) l : long int (pentru d,i și n), unsigned long int (pentru o, u și x) sau double (pentru e, f și g) L : long double (pentru e, f și g)
tip	c : caracter d : întreg zecimal e, E, f, g, G : număr în virgulă mobilă o : întreg în baza 8 s : sir de caractere u : întreg zecimal fără semn (unsigned) x, X : întreg hexazecimal

³⁾ de forma **&a**, unde **a** este variabila în care se va stoca valoarea citită;

⁴⁾ în caz de succes, funcția returnează numărul de itemi citiți; în caz contrar returnează EOF (End Of File - constantă definită).

- Exemple:

```
scanf("%d",&n);
```

- citește un *întreg zecimal* și îl memorează în variabila **n** (declarată anterior prin **int n**);

```
scanf("%f%c%h",&a,&b,&c);
```

- citește un *număr în virgulă mobilă* în variabila **a** (deci a trebuie să fie

de tip float sau double și trebuie să fie declarată anterior), un *caracter* în variabila **c** (este declarată **char b**; anterior) și un *short int* în variabila **c** (deci avem declarat anterior **short int c**);

- Declaratia functiei printf:

```
int3) printf ( const char * format1), lista_parametri2) );
```

- scrie datele în stdout (standard output) conform formatului

¹⁾ format reprezintă un sir de caractere constant de forma : %[flag-uri][dimensiune][.precizie][lungime]tip, unde

flag-uri	- : left-justify conform dimensiunii + : numărul va fi precedat de semn(+ sau -) # : pentru o, x, X – valorile vor fi precedate de 0, 0x , respectiv 0X pentru e, E, f – virgulă va apărea în output, chiar dacă nu există zecimale după virgulă pentru g, G – ca pentru e, E, f , dar zero-urile de după virgulă vor apărea 0 : adaugă zero-uri la stânga în locul spațiilor libere
Dimensiune	numărul minim de caractere ce vor fi scrise; dacă numărul este mai mic decât dimensiunea, se adaugă spații; dacă este mai mare nu se trunchiază
.precizie	.numar : pentru întregi – numărul minim de scris pentru virgulă mobilă – numărul de cifre de scris după virgulă pentru siruri de caractere (c) – numărul maxim de caractere de scris

lungime	vezi modificador din scanf
Tip	vezi tip din scanf

²⁾ de forma **a, b**, unde a si b sunt variabilele ce vor fi scrise

³⁾ in caz de succes, returneaza numarul de itemi scrisi; in caz contrar, returneaza un numar negativ.

- Exemple:

printf("%d",a); - scrie valoarea variabilei intregi a

printf("%f%c",a,b) - scrie valoarea variabilei de tip float a si valoarea variabile de tip char b

2. Operatori

Clasificare	Operator	Exemplu
atribuire	=	int a = 5;
aritmetici unari	+, -	int a = 5, b; b = -a; b = +b;
aritmetici binari	+, -, *, /, %	int a = 5, b = 3; float c; c = (a + b) / (a - b) * 2;
preincrementali, predecrementali	++, --	int a = 5, b = 3, c; ++ a; c = a + b; (a va avea valoarea 6, b : 3, c : 9)
postincrementali, postdecrementali	++, --	int a = 5, b = 3, c; c = a++ + b; (a va avea valoarea 6, b: 3, c : 8)
logici	!, &&,	int a = 5; if (a>=3) a--;
relationali	>, >=, <, <=, ==, !=	
logici la nivel de bit	&, , ^, ~, >>, <<	int a = 3, b = 10, c; c = (a>>b) & a;
atribuire combinati	+=, -=, *=, /=, %=, &=, =, >>=, <<=	int a = 3; a+=5;
decizional	?:	int a; bool b = (a>5)?true:false;
de conversie	(tip)	int a,b,c; c = (int)a/b;
sizeof	sizeof(tip)	printf("%d", sizeof(int))

3. Probleme

1. Se da ecuatia de gradul al II-lea $3x^2 - 5x + 2 = 0$. Sa se calculeze radacinile ecuatiei folosind ori de cate ori se poate operatorii de atribuire combinati.
2. Se citesc trei numere intregi de la tastatura. Sa se afiseze maximul dintre cele 3 numere folosind operatorul decizional.
3. Se citeste un numar intreg n de la tastatura. Sa se calculeze $n*8$, $n/4$ si $n*10$ folosind operatorii logici de deplasare la nivel de bit. Sa se verifice apoi ca rezultatele sunt egale (folosind pe de o parte operatorii uzuali de inmultire si impartire si, pe de alta parte, pe cei de deplasare).
4. Se citeste un numar intreg de la tastatura. Sa se determine daca acesta este par sau impar folosind doar operatorii logici. Sa sa afiseze un mesaj corespunzator. (se vor folosi doar operatori, nu si instructiuni decizionale)