**Tipuri fundamentale de date. Constante. Variabile. Operatori. Instrucțiuni. Intrare/ieșire**

**I. Repere teoretice**

**1. Unități lexicale ale limbajului C**

* cuvinte cheie
* constante
* identificatori
* operatori
* separatori: ( ) { } [ ] “ “ ‘ ‘ ; /\* \*/

**2. Tipuri fundamentale de date**

* tipuri intregi - specificatori char, int (calificatori unsigned, long, short)
* tipuri flotante/în virgulă mobilă – specificatori float (simplă precizie), double (dublă precizie)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Storage size** | **Value range** | **Specifier** |
| **char** | 1 byte | -128 to 127 or 0 to 255 | %c or %hhd |
| **unsigned char** | 1 byte | 0 to 255 | %c or %hhu |
| **signed char** | 1 byte | -128 to 127 | %c or %hhd |
| **int** | 2 or 4 bytes | -32.768 to 32.767 or  -2.147.483.648 to 2.147.483.647 | %d or %i |
| **unsigned int** | 2 or 4 bytes | 0 to 65.535 or  0 to 4.294.967.295 | %u |
| **short int** | 2 bytes | -32.768 to 32.767 | %hd or %hi |
| **unsigned short int** | 2 bytes | 0 to 65.535 | %hu |
| **long int** | 4 bytes | -2.147.483.648 to 2.147.483.647 | %d or %i  %ld or %li |
| **unsigned long int** | 4 bytes | 0 to 4.294.967.295 | %u or %lu |
| **long long int** | 8 bytes | –9.223.372.036.854.775.808 to 9.223.372.036.854.775.807 | %lld or %lli |
| **unsigned long long int** | 8 bytes | 0 to 18.446.744.073.709.551.615 | %llu or %I64 |
| **float** | 4 bytes | 1.2E-38 to 3.4E38  (6 decimal places) | %f |
| **double** | 8 bytes | 2.3E-308 to 1.7E308  (15 decimal places) | %f – printf  %lf - scanf |
| **long double** | 10 bytes | 3.4E-4932 to 1.1E4932  (19 decimal places) | %Lf |

**3. Constante**

* constante întregi: zecimal, octal, hexazecimal

exemple: 31, 037 (=31 în baza 8), 0x1f (=31 în baza 16), 0X1F (=31 în baza 16)

* constante reale/în virgulă mobilă

exemple: 123.45, 12345e-2, 0.12345E3, 0.12, .12, 12e2

* constante caracter

exemple: ‘x’, ‘9’ (are valoarea 57), ‘\n’, ‘\t’, ‘\b’, ‘\\’, ‘\’’, ‘\040’ (secvența octală 040 – caracterul spațiu)

* constante șir de caractere

exemple: “aceasta este o constanta sir”, “primul rand\nal doilea”

* constante simbolice - definite prin directive preprocesor

exemple: #define PI 3.14

* enumerări
* mecanisme implicite și explicite de memorare: sufixele u/U, l/L, f/F și combinate

**4. Variabile**

* nume
* atribute: tip (structura, gama de valori, dimensiunea spațiului ocupat), clasa de memorie (auto, extern, static, register)

**5. Operatori**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Operators' precedence Category** | **Operator** | **Associativity** |
| **1.** | Postfix | () [] -> . ++ -- | 🡪 |
| **2.** | Unary | + - ! ~ ++ -- (type) \* & sizeof | 🡨 |
| **3.** | Multiplicative | \* / % | 🡪 |
| **4.** | Additive | + - | 🡪 |
| **5.** | Shift | << >> | 🡪 |
| **6.** | Relational | < <= > >= | 🡪 |
| **7.** | Equality | == != | 🡪 |
| **8.** | Bitwise AND | & | 🡪 |
| **9.** | Bitwise XOR | ^ | 🡪 |
| **10.** | Bitwise OR | | | 🡪 |
| **11.** | Logical AND | && | 🡪 |
| **12.** | Logical OR | || | 🡪 |
| **13.** | Conditional | ?: | 🡨 |
| **14.** | Assignment | = += -= \*= /= %= >>= <<= &= ^= |= | 🡨 |
| **15.** | Comma | , | 🡪 |

**6. Instrucțiuni**

* instrucțiunea expresie
* instrcțiunea compusă
* instrucțiunea condițională – decizie simplă (if) și multiplă (switch)
* instrucțiuni repetitive (while, do-while, for)
* alte instrucțiuni: break, continue, return, goto

**7. Intrare/ieșire**

* fluxuri (stream-uri) de intrare-ieșire standard: de intrare, de ieșire, de editare a erorilor
* stdio.h – conține definiții ale fluxurilor standard stdin, stdout, stderr
* funcții

int printf(“sir”,expresie1, expresie2,…);

sir: <secventa de car><specificator format><secventa de car><specificator format>…

specificator de format: %[aliniere semn][latime][.precizie][h/l/L]tip\_conversie

int scanf(“sir”, adresaVar1, adresaVar2,…);

sir: <secventa de car><specificator format><secventa de car><specificator format>…

specificator de format: %[\*][latime][h/l/L]tip\_conversie

**8. Structura unui program**

* directive preprocesor – exemple #include<stdio.h>, #define PI 3.14>
* declarații globale
* funcții – un program conține o singură funcție main

**II. a) Exerciții**

1. Scrieți un program care conține declarația

int a=-12;

și apoi afișează liniile de mai jos:

a=-12

a= 0000012

a=-0000012

a=fffffff4

a=FFFFFFF4

a=37777777764

a=4294967284

2. Scrieți un program care conține declarațiile

char a=-12, b=’a’;

unsigned char c=12, d=-12, e=’x’;

și apoi afișează liniile de mai jos:

-12 a 97 61 *// valoarea lui a, valoarea lui b, codul caracterului memorat in b in baza 10 si in baza 16*

12 244 x 120 *// valoarea lui c, numarul corespunzator lui d, valoarea lui e, codul caracterului memorat in e in baza 10*

3. Scrieți un program care conține declarațiile

float a=67.78; double b=-98.9;

long double c=32.7; long int d=100000;

și apoi afișează liniile de mai jos:

a=+67.8

b=-98.900

c=32.70

d=100000

4. Scrieți un program care cuprinde instrucțiunile de mai jos. Afișați valorile lui a și b după fiecare citire, dacă introduceți de la tastatură, pentru fiecare citire, numărul 123. Explicați efectul.

int a,b,c,d;

scanf(“%d”,&a);

scanf(“%1d”,&a);

scanf(“%1d %\*1d %1d”,&a,&b);

5. Scrieți un program care cuprinde instrucțiunile de mai jos. Afișați valorile lui a și b după fiecare citire. Explicați efectul.

char a,b;

scanf(“%c”,&a); //introduceți de la tastatură litera x

scanf(“%c %c”,&a,&b); //introduceți de la tastatură: x y

scanf(“%c%c”,&a,&b); //introduceți de la tastatură: x y

scanf(“1%c”,&a); //introduceți de la tastatură: 12, apoi 21

6. Scrieți un program care cuprinde instrucțiunile de mai jos. Afișați valorile lui a, b și c după fiecare citire. Explicați efectul.

float a; double b; long double c;

scanf(“%f %lf %Lf”,&a,&b,&c); //introduceți de la tastatură: 2.5 2.7 4.3

scanf(“%f%lf”,&a,&b); //introduceți de la tastatură: 2 3

scanf(“%3f”,&a); //introduceți de la tastatură: 1.25

scanf(“%f 1 %lf”,&a,&b); //introduceți de la tastatură: 1.2 1 -5.7

**II. b) Aplicații**

**<tipuri de date, constante, citiri si afisari>**

1. Scrieți un program care să definească și să afișeze pe ecran constante din fiecare tip cunoscut.

2. Scrieți un program care să citească și apoi să afișeze pe ecran valorile limită pentru fiecare dintre tipurile de date întregi precizate în tabelul de mai sus, precum și câte o valoare pentru fiecare tip real.

3.Scrieți un program care să afișeze codurile următoarelor caractere albe: spațiu, tab, newline.

4. Scrieți un program care tipărește un pătrat în care fiecare latură este alcătuită din câte 10 caractere \*.

***<operatori aritmetici, relationali, logici>***

5. Se citește ora de plecare a unui autobuz, sub forma , precum și durata cursei, în aceeași formă. Afișați ora de sosire a autobuzului, tot sub forma de mai sus.

6. Se citesc două numere naturale şi . Să se afiseze câte numere pare sunt în intervalul .

7. Se citesc numerele n și k. Să se afișeze valoarea sumei

k+(k+1)+(k+2)+...+n

8. Se citește înălțimea și raza bazei unui cilindru. Să se calculeze suprafața totală a acestuia.

9. Considerăm două cutii în formă de cub având lungimile laturilor de L1, respectiv L2 centimetri. Scrieți un program care calculează volumul de aer care va rămâne între pereţii celor două cutii după ce se va introduce una dintre cutii în cealaltă.

10. Se citesc două numere naturale a și b. Să se afișeze media aritmetică, media geometrică și media armonică a acestora.

11. Se citesc trei numere a,b,c. Să se verifice dacă pot fi laturile unui triunghi.

12. Se citesc coeficientii a,b ai unei ecuatii de gradul intai. Afisati solutiile acesteia fara a utiliza instructiuni de decizie, ci doar operatorul ?:

13. Se citesc patru numere întregi, reprezentând capetele a două intervale. Să se afișeze mesajul DA dacă cele două se intersectează sau mesajul NU în caz contrar.

***<operatori pe biti>***

14. Se citește un număr natural. Să se afișeze valoarea reținută de

a) bitul de rang 2

b) bitul de rang k

15. Se citește un număr natural k (k<32). Să se afișeze 2k.

16. Se citeşte un număr natural n. Să se afişeze câtul şi restul împărţirii sale la 8.

17.Se citeşte un număr natural n. Să se verifice dacă este o putere a lui 2.

18. Se citesc doua numere naturale. Sa se afiseze numarul de pozitii binare prin care difera.

19. Se citesc n numere naturale, in care toate apar de exact doua ori, cu exceptia unui numar care apar doar o data. Sa se afiseze numarul care apare o singura data.

20. Se considera un sir de n-1 numere naturale distincte cuprinse intre 1 si n. Sa se determine numarul din intervalul [1,n] care lipseste.

21. Fiind date două numere naturale a și b a căror reprezentare binară ocupă 32 de biți, spunem că a este șeful lui b dacă pentru orice poziție binară 0 a lui a, poziția corespunzătoare a lui b este 0. Se citește un număr natural n și se cere să se afișeze numărul șefilor săi.

**<instrucțiuni>**

22. Se citește un număr natural nenul . Să se afișeze cel mai mic și cel mare număr ce pot fi formate din cifrele lui . De exemplu, pentru trebuie afișate numerele și .