SISTEME DE OPERARE

- Laborator 8 -

PROGRAMARE ÎN LIMBAJUL C

1. TIPURI DE DATE DE BAZĂ

|  |  |  |
| --- | --- | --- |
| Tip | Dimensiune de stocare | Domeniu de valori |
| char | 1 octet | [-128, 127] sau [0, 255] |
| int | 2 sau 4 octeți | [-32.768, 32.767] sau [-2.147.483.648, 2.147.483.647] |
| float | 4 octeți | [1,2E-38, 3,4E+38] |
| double | 8 octeți | [2,3E-308, 1,7E+308] |

* tipurilor de date char și int li se poate aplica calificatorul: unsigned
* tipului de date int i se mai pot aplica și calificatorii: short și long

|  |  |  |
| --- | --- | --- |
| Tip | Dimensiune de stocare | Domeniu de valori |
| unsigned char | 1 octet | [0, 255] |
| unsigned int | 2 sau 4 octeți | [0, 65.535] sau [0, 4.294.967.295] |
| short (int) | 2 octeți | [-32.768, 32.767] |
| unsigned short | 2 octeți | [0, 65.535] |
| long (int) | 4 octeți | [-2.147.483.648, 2.147.483.647] |
| unsigned long | 4 octeți | [0, 4.294.967.295] |

2. CUVINTE REZERVATE (CUVINTE-CHEIE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| auto | break | case | char | const | continue |
| default | do | double | else | enum | extern |
| float | for | goto | if | int | long |
| register | return | short | signed | sizeof | static |
| struct | switch | typedef | unsigned | union | void |
| volatile | while |  |  |  |  |

3. CONSTANTE

* pot fi definite în două moduri:
  + folosind directiva de preprocesare #define:

#define TEN 10

#define NEWLINE '\n'

* + utilizând prefixul const:

const int TEN = 10;

const char NEWLINE = '\n';

4. VARIABILE

* definirea unei variabile:

*tip\_variabila* ***nume\_variabila****;*

* + *tip\_variabila* poate fi: char, int, short, long etc. *(vezi pct. 1)*
  + ***nume\_variabila*** poate fi alcătuit din litere, cifre și caracterul „ \_” *(underscore)*
  + primul caracter trebuie să fie o literă
  + cuvintele rezervate *(vezi pct. 2)* nu pot fi utilizate ca nume de variabile
  + limbajul C face deosebire între litere mici și litere mari *(case-sensitive)*
* exemple:

int n; int n = 10;

char c; char c = 'a';

5. OPERATORI

* specifică operațiile care se efectuează cu variabile și constante
* categorii de operatori:
  + *operatori aritmetici:* + - \* / % ++ --
  + *operatori relaționali:* == != > < >= <=
  + *operatori logici:* && || !
  + *operatori logici pe biți:* & | ^ ~ << >>
  + *operatori de atribuire:* = += -= \*= /= %= <<= >>= &= ^= |=
  + *alți operatori:* sizeof() & \* ?:
* prioritatea operatorilor:

https://en.cppreference.com/w/c/language/operator\_precedence

6. TIPURI DERIVATE DE DATE

6.a. Tablouri unidimensionale (vectori)

*tip\_tablou* ***nume\_tablou****[dimensiune\_tablou];*

* exemple:

int list[5];

int list[5] = {10, 20, 30, 40, 50};

double values[] = {100.0, 2.0, 300.0, 40.0, 50.0};

6.b. Șiruri de caractere

char msg[] = "Hello";

char msg[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

6.c. Pointeri

* *pointer = o* *variabilă care conține adresa unei alte variabile*
* exemple:

int \*p; // pointer către o variabilă de tip întreg

char \*c; // pointer către o variabilă de tip float

float \*f; // pointer către o variabilă de tip float

double \*d; // pointer către o variabilă de tip double

* obținerea adresei la care este stocată variabila x: &x
* obținerea valorii variabilei x: \*p *(dacă p este un pointer către variabila x)*

6.d. Structuri de date

* definirea unei structuri de date:

struct Books

{

int id;

char author[50];

char title[100];

}

* declararea și utilizarea unei structuri de date:

int main(int argc, char\*\* argv)

{

struct Books book1;

...

book1.id = 1000;

strcpy(book1.author, "B.W. Kernighan, D.M. Ritchie");

strcpy(book1.title, "The C Programming Language");

...

return 0;

}

7. FUNCȚII

* declararea unei funcții:

*tip\_returnat* ***nume\_functie****(tip\_param* ***param1****, tip\_param* ***param2****, ...);*

unde:

* + *tip\_returnat* poate fi orice tip de date de bază/tip de date derivat sau void
  + ***nume\_functie*** poate fi alcătuit din litere, cifre și caracterul „ \_” *(underscore)*
  + primul caracter trebuie să fie o literă
  + cuvintele rezervate *(vezi pct. 2)* nu pot fi utilizate ca nume de funcții
  + *tip\_param* ***param1****, tip\_param* ***param2****, ...* este lista parametrilor formali sau void (dacă funcția nu are parametrii)
* exemple:

void afiseaza\_matrice(int\*\* matrice)

float media\_aritmetica(int a, int b)

int\*\* citeste\_matrice(FILE\* file)

* funcția main():
  + este punctul de intrare principal în program *(the program main entry point)*
  + prototipul recomandat este:

int main(int argc, char\*\* argv)

deoarece permite accesul la argumentele furnizate de utilizator în linia de comandă.

8. FUNCȚII DE INTRARE/IEȘIRE

int getchar(void)

int putchar(void)

char \*gets(char \*s)

int puts(const char \*s)

int scanf(const char \*format, ...)

int printf(const char \*format, ...)

9. FUNCȚII PENTRU LUCRUL CU FIȘIERE

9.a. Pentru fișiere de tip text:

FILE \*fopen(const char \*filename, const char \*mode)

int fgetc(FILE \*fp)

int fgets(char \*buf, int n, FILE \*fp)

int fputc(int c, FILE \*fp)

int fputs(const char \*s, FILE \*fp)

int fclose(FILE \*fp)

9.b. Pentru fișiere binare:

size\_t fread(void \*buf, size\_t bsize, size\_t nbyte, FILE \*fp)

size\_t fwrite(const void \*buf, size\_t bsize, size\_t nbyte, FILE \*fp)

sau:

int open(const char \*path, int oflag, ... )

ssize\_t read(int fd, void \*buf, size\_t nbyte)

ssize\_t write(int fd, const void \*buf, size\_t nbyte)

int close(int fd)

10. PRIMUL PROGRAM C ÎN UNIX

* editoare de texte în UNIX: vi, nano, joe
* exemplu: hello.c

|  |
| --- |
| #include <stdio.h>  // the program main entry point  int main(int argc, char\*\* argv)  {  printf("Hello world !\n");  return 0;  } |

* compilarea: gcc -Wall -g -o hello hello.c
* lansarea în execuție: ./hello

11. EXEMPLE

* obținerea numărului și a listei de argumente furnizate în linia de comandă:

wget http://www.cs.ubbcluj.ro/~dbota/SO/lab2/exemple/lab2\_2.c

* obținerea și afișarea variabilelor de mediu:

wget http://www.cs.ubbcluj.ro/~dbota/SO/lab2/exemple/lab2\_3.c

* utilizarea tablourilor unidimensionale:

wget http://www.cs.ubbcluj.ro/~dbota/SO/lab2/exemple/lab2\_4.c

* citirea unui întreg de la tastatură:

wget http://www.cs.ubbcluj.ro/~dbota/SO/lab2/exemple/lab2\_5.c

* citirea conținutului unui fișier de tip text:

wget http://www.cs.ubbcluj.ro/~dbota/SO/lab2/exemple/lab2\_6.c

* citirea conținutului unei matrice dintr-un fișier de tip text:

wget http://www.cs.ubbcluj.ro/~dbota/SO/lab2/exemple/lab2\_7.c

12. ERORI/ATENȚIONĂRI LA COMPILARE

* erori de sintaxă
* omiterea unui fișier de tip antet
* folosirea unei variabile care nu a fost definită
* folosirea a două variabile definite cu același nume
* folosirea unei funcții care nu a fost declarată
* apelul unei funcții fără respectarea prototipului acesteia (număr incorect de argumente, inversarea ordinii argumentelor etc.)

13. DETECȚIA ERORILOR DE GESTIUNE A MEMORIEI

* detecția erorilor de gestiune a memoriei folosind utilitarul valgrind:

valgrind ./myprog

REFERINȚE:

* Programare C: https://www.tutorialspoint.com/cprogramming/index.htm
* Manual Valgrind: http://valgrind.org/docs/manual/quick-start.html