

## Laborator 2 – Mediul integrat de programare C

### 1. Etapele de realizare ale unui program C

În vederea obținerii unui cod executabil în C se parcurg următoarele etape:

1. Crearea codului sursă

2. Compilarea programului

3. Legarea programului cu funcțiile necesare din bibliotecă (linkeditarea)

1. Majoritatea compilatoarelor de C utilizate în prezent conțin un editor propriu pentru scrierea **codului sursă** al programului în limbajul C și salvarea lui într-un fișier numit **fișier sursă**. Dacă se lucrează cu un compilator de sine stătător, acesta va accepta și fișiere create cu alte editoare cu condiția ca acestea să fie fișiere text standard. Codul sursă constituie datele de intrare ale compilatorului C.

2. În urma **compilării**, codul sursă se traduce în cod mașină, pe care calculatorul îl poate citi și executa nemijlocit. Acest cod rezultat se numește **cod obiect** și constituie datele de intrare pentru programul de legare (cu extensia .OBJ).

Modul de compilare al programului depinde de mediul de integrare folosit, detaliile de utilizare fiind prezentate în manualul utilizatorului.

Cele mai multe din programele C sunt conținute complet într-un singur fișier sursă. Pentru a reduce durata de compilare (care crește odată cu creșterea lungimii unui program), C permite unui program să fie conținut în mai multe fișiere și permite compilarea separată a fiecăruia. Această compilare separată prezintă avantajul că dacă se modifică codul sursă a unui fișier nu este nevoie să se recompileze întregul program.

3. Operația de **linkeditare** este realizată cu ajutorul unui program care leagă programul obținut la etapa de compilare cu bibliotecile de sistem, inserează diferite secvențe standard și rezolvă referințele externe. Datele de ieșire constituie un program executabil (cu extensia .EXE)

Toate compilatoarele de C conțin o bibliotecă standard de funcții C. Standardul ANSI C furnizează setul minimal de funcții inclus în bibliotecă. La ora actuală însă, majoritatea compilatoarelor folosite conțin biblioteci cu mult mai multe funcții decât cele definite de ANSI.

### 2. Structura unui program C

Programul C poate începe cu o **linie de comentariu**, în care se specifică pe scurt numele și scopul programului. După aceasta, programul poate conține **directive de preprocesor** care conțin diverse instrucțiuni destinate compilatorului C. Programul C

poate conține apoi o listă de **declarații de variabile** deoarece în acest limbaj toate variabilele utilizate **trebuie declarate** înainte de a fi utilizate.

```
/*Linie de comentariu-Structura unui program în C */
Directive de preprocesor
Declarații de variabile
tip returnat main(lista parametri)
{
  secvența de declarații și instrucțiuni
}

tip returnat fu1(lista1 parametri)
{
  secvența1 de declarații și instrucțiuni
}

tip returnat fu2(lista2 parametri)
{
  secvența2 de declarații și instrucțiuni
}

.
.
.
tip returnat fun(lista3 parametri)
{
  secvența de declarații și instrucțiuni
}
```

**Fig. 1** Structura unui program în C

Orice program scris în limbajul C este format din una sau mai multe funcții care specifică operațiile ce trebuie efectuate. Funcția **main( )** este obligatoriu prezentă și se apelează la începutul programului. Într-un program bine conceput, în această funcție se realizează un rezumat cu ceea ce face programul. Acest rezumat constă din apelul altor funcții, unele aflându-se în același program iar altele în biblioteci ce conțin funcții scrise anterior. Deși **main( )** nu reprezintă un cuvânt-cheie, programatorul trebuie să-l considere ca atare și să nu-l folosească niciodată ca nume de variabilă.

Funcțiile **fu1( )**, **fu2( )**, ... **fun( )** reprezintă funcțiile definite de utilizator.

O metodă de a comunica date între funcții este cea care utilizează parametrii acestora. Listele **lista1 parametri**, **lista2 parametri**, ..., **listan parametri** conțin argumentele funcțiilor **fu1( )**, **fu2( )**, ... **fun( )**. Prezența parantezelor este obligatorie chiar dacă funcția nu conține nici un parametru.

Acoladele { } existente în fiecare funcție includ **secvența de declarații și instrucțiuni** care alcătuiesc funcția.

### 3. Mediul de dezvoltare integrat Borland C++

Mediul de dezvoltare integrat Borland C++ 3.1 este un instrument foarte puternic, la îndemâna oricărui utilizator. Acesta conține: un editor, un compilator rapid, un editor de legături, încărcător și numeroase utilitare (programe de manipulare a fișierelor, programe pentru asistarea programatorului la depanarea programelor în faza de execuție). În momentul lansării în execuție al programului apare ecranul principal al mediului (figura 1.3). Ecranul conține următoarele componente:

- meniul principal (pull down) situat în partea superioară a ecranului,
- fereastra de editare;
- fereastra de mesaje;
- linia de opțiuni rapide situată în partea de jos a ecranului.

Meniul principal conține 11 submeniuri (elemente de meniuri) care la rândul lor se compun din opțiuni specifice. Activarea unui element de meniu se face cu ajutorul tastei <F10>. Navigația prin elementele de meniu se face folosind tastele săgeți →, ←, ↑, ↓ urmată de activarea comenzii dorite cu tasta <Enter>. Ieșirea din meniul principal se realizează cu ajutorul tastei <Esc>. Activarea unei element de meniu sau a unei opțiuni se poate realiza și prin tastarea literei scrise îngroșat sau color (la monitoarele color) din nume.

### Submeniul System

Conține comanda sistemului Repaint Destop urmată de o listă de programe care au fost instalate cu comanda **Options|Transfer**: GREP, Turbo Assembler, Turbo Debugger, Turbo Profiler, Resource Compiler, Import Librarian. Rularea unui astfel de program se face prin selectarea lui urmată de apăsarea tastei <Enter>.

### Submeniul File

Acest meniu pune la dispoziție opțiuni de deschidere și salvare a fișierelor, de schimbare a directoarelor, de tipărire la imprimantă a textului, de părăsire temporară sau definitivă a mediului (tabelul 1).

**Tab. 1**

Comanda	Combinatia de taste	Funcție îndeplinită
New		Deschide o nouă fereastră de editare a unui nou fișier, având numele implicit NONAMEnr.CPP (nr=00,...,31).
Open	F3	Încarcă un cod sursă al unui program existent într-o nouă fereastră de editare.
Save	F2	Salvează conținutul ferestrei de editare active.
Save as...		Salvează conținutul ferestrei de editare active într-un fișier cu nume nou.
Save all		Salvează conținutul tuturor ferestrelor deschise în fișierele corespunzătoare respective.
Change dir		Alege o nouă unitate de disc și/sau un nou director curent.
Print		Tipărește conținutul ferestrei active de editare.
DOS shell		Părăsește temporar mediul și trece în sistemul de operare DOS pentru a da comenzi. Revenirea se face cu comanda <b>Exit</b> .

Comanda	Combinăția de taste	Funcție îndeplinită
Quit	Alt+X	Părăsește definitiv mediul Borland C++.

### Submeniul Edit

Conține opțiuni specifice editării textelor, în general a programelor C/C++ (tab. 2). Este necesar să definim mai întâi noțiunea de clipboard.

**Clipboard** reprezintă un tampon unde poate fi copiat un bloc dintr-un fișier sursă care va fi ulterior prelucrat.

**Tab. 2**

Comanda	Combinăția de taste	Funcție îndeplinită
Undo	Alt+BKSp	Anulează efectul ultimei acțiuni de editare.
Redo	Shift+Alt+BKSp	Revine asupra efectului celei mai noi comenzi Undo.
Cut	Shift+Del	Șterge de pe ecran textul selectat și-l copiază în Clipboard.
Copy	CTRL+Ins	Copiază textul selectat în Clipboard.
Paste	Shift+Ins	Introduce conținutul din Clipboard în textul ferestrei de editare active începând de la poziția curentă a cursorului.
Clear	Ctrl+Del	Șterge textul selectat fără a-l scrie în Clipboard.
Copy example		Copiază un exemplu de program din fereastra Help în Clipboard.
Show Clipboard		Deschide și afișează fereastra Clipboard care conține toate textele șterse sau copiate din diferite ferestre de editare.

### Submeniul Search

Conține o serie de opțiuni pentru localizarea diferitelor informații, cum ar fi: porțiuni de text, declarații de funcții sau erori ale programului.

**Tab. 3**

Comanda	Combinăția de taste	Funcție îndeplinită
Find		Afișează fereastra de dialog "Find Text" unde se caută o porțiune de text din fereastra activă de editare.
Replace		Afișează o fereastră dialog "Replace text" unde se înlocuiește o porțiune de text, existentă în fereastra activă a codului sursă, cu o alta, specificată de utilizator.
Search Again		Repetă ultima operație de tip Find sau Replace.
Go to Line Number		Deplasează cursorul pe o nouă linie cu numărul specificat de utilizator (fereastra de dialog "Go to Line Number").
Previous Error	Alt+F7	Mută cursorul la locația erorii sau mesajului de avertisment anterioare.

Comanda	Combinatia de taste	Funcție îndeplinită
Next Error	Alt+F8	Mută cursorul la locația erorii sau mesajului de avertisment următoare.
Locate Function		Afișează fereastra de dialog “Locate Function” unde se localizează declarația unei funcții.

### Submeniul Run

Conține opțiuni pentru lansarea în execuție a programului și controlul execuției acestuia.

**Tab. 4**

Comanda	Combinatia de taste	Funcție îndeplinită
Run	Ctrl+F9	Rulează (execută) programul din fereastra de editare curentă. Dacă s-au făcut modificări în fișierul sursă, acesta se recompilază.
Program Reset	Ctrl+F2	Închide sesiunea de depanare curentă, eliberând memoria alocată programului. Fiecare fișier deschis și utilizat de program este închis.
Go to Cursor	F4	Rulează programul de la bara de execuție (bara supraluminată care indică instrucțiunea ce va fi executată) până la linia ce conține cursorul în fereastra de editare activă.
Trace into	F7	Rulează programul pas cu pas intrând în funcțiile apelate.
Step over	F8	Rulează programul pas cu pas fără a intra în funcțiile apelate.
Arguments...		Permite introducerea unor argumente ale liniei de comandă pentru rularea programului (fereastra de dialog “Program Arguments”).

### Submeniul Compile

Conține o serie de opțiuni pentru compilarea programului din fereastra activă și crearea codului executabil precum și opțiuni pentru prelucrarea proiectelor (programelor cu surse multiple) (tabelul 5).

**Tab. 5**

Comanda	Combinatia de taste	Funcție îndeplinită
Compile	Alt+F9	Compilează programul sursă NUME.C sau NUME.CPP din fereastra de editare activă, generând fișierul obiect NUME.OBJ.
Make	F9	Creează fișierul executabil compilând și link-editând fișierele componente ale unui program (care nu sunt actuale) cu surse multiple (proiect). Numele fișierului executabil generat NUME.EXE derivă din numele fișierului proiect sau din numele fișierului din fereastra activă de editare.

Comanda	Combinatia de taste	Funcție îndeplinită
Link		Realizează linkeditarea (editarea de legături) și se creează fișierul executabil.
Build All		Compilează și link-editează toate fișierele componente ale fișierului proiect indiferent de data la care au fost create.
Information...		Afișează o fereastră cu informații despre directorul curent, fișierul sursă curent, memoria folosită, număr de linii compilate, număr de erori/avertismente, timpul de rulare și codul de ieșire al programului.
Remove messages		Șterge toate mesajele din fereastra de mesaje "Message".

### Submeniul Debug

Meniul Debug oferă opțiuni de depanare a programelor scrise în C (tabelul 6).

**Tab.6**

Comanda	Combinatia de taste	Funcție îndeplinită
Inspect...	Alt+F4	Examinează valoarea unei variabile (simplă, șir, pointer, structură, tip, uniune și funcție)
Evaluate/Modify	Ctrl+F4	Evaluează și vizualizează valoarea unei variabile sau a unei expresii; modifică valoarea unei variabile simple (fereastra de dialog "Evaluate and Modify")
Call stack...	Ctrl+F3	Afișează o fereastră cu lista în ordine inversă a funcțiilor apelate (împreună cu parametrii transmiși fiecărei funcții) până la punctul curent de execuție al programului.
Watches	Ctrl+F7	Selectează sau deschide fereastra "Watch" de gestionare a variabilelor pe parcursul rulării programului.
	Add watch...	Introduce o variabilă sau o expresie în fereastra "Add Watch" cu scopul de a putea fi urmărită.
	Delete Watch	Șterge variabila sau expresia curentă din fereastra "Watch".
	Edit Watch...	Editează, într-o fereastră separată "Edit Watch", expresia curentă din fereastra "Watch".
	Remove all watches	Șterge toate elementele din fereastra "Watch".
Toggle Breakpoint	Ctrl+F8	Stabilește sau anulează puncte de întrerupere în

Comanda	Combinatia de taste	Funcție îndeplinită
		linia pe care se află cursorul; acestea vor întrerupe rulara programului când se ajunge la acele linii.
Breakpoints...		Afișează fereastra "Breakpoints" cu lista punctelor de întrerupere și caracteristici ale acestora.

### Submeniul Project

Acest submeniu conține comenzi pentru crearea și întreținerea unui proiect ce conține mai multe fișiere (tabelul 7).

**Tab. 7**

Comanda	Funcție îndeplinită
Open project...	Deschide un proiect existent sau creează un nou proiect. Fișierele proiect au extensia .PRJ (fereastra "Open Project File").
Close Project	Inchide proiectul curent.
Add item...	Adaugă fișiere la lista de fișiere componente ale proiectului (fereastra "Add to Project List").
Delete item	Sterge fișierul selectat din fereastra "Project" (deschisă cu comanda Open Project).
Local options...	Afișează o fereastră de dialog "Local Options" în cadrul căreia se pot specifica opțiuni ale liniei de comandă, se atribuie nume sau o cale pentru fișierul obiect etc.
Include files...	Deschide o fereastră de dialog "Include Files", unde se pot vedea și deschide toate fișierele pe care le include fișierul selectat din fereastra "Project".

### Submeniul Options

Meniul Options permite comenzi privind vizualizarea și modificarea setărilor implicite ce configurează mediul Borland C++ (tabelul 8).

**Tab. 8**

Comanda	Funcție îndeplinită
Application...	Afișează fereastra de dialog "Set Application Options" unde pot fi selectate global opțiuni de compilare și linkeditare pentru aplicații DOS și Windows.
Compiler	Stabilește opțiunile compilatorului.
Code generation...	Deschide o fereastră de dialog unde pot fi selectate modelul de memorie utilizat, opțiuni privind codul generat, modul de interpretare al segmentului de stivă de către compilator, macrodefiniții pentru preprocesor.
Advanced code generation...	Specifică alte opțiuni cu privire la codul specificat.

<b>Comanda</b>		<b>Funcție îndeplinită</b>
	Entry/Exit Code...	Deschide fereastra de dialog “Entry/Exit Code Generation” unde se poate stabili tipul codului prolog/epilog, convenția de apelare, opțiuni privind stivele.
	C++Options	Setează opțiuni specifice cu privire la fișierul obiect pentru limbajul C++.
	Advanced C++ Options	Setează opțiuni avansate cu privire la fișierul obiect pentru limbajul C++.
	Optimizations...	Deschide o fereastră de dialog “Optimization Options” unde se stabilește modul de optimizare al codului generat.
	Source...	Stabilește modul în care compilatorul prelucrează codul sursă și tipul acestuia.
	Messages	Afișează o listă de subcomenzi cu ajutorul cărora pot fi setate opțiuni legate de modul în care sunt raportate diferite tipuri de erori care apar la compilare.
	Names	Modifică numele implicite de segmente, grup și clasă pentru secțiunile de cod, date și BBS.
	Transfer...	Adaugă sau șterge programe de transfer în cadrul submeniuului System (fereastra “Transfer”).
	Make...	Selectează condițiile pentru gestionarea și prelucrarea proiectelor.
Linker		Stabilește opțiunile pentru editarea de legături.
	Settings...	Deschide fereastra de dialog “Linker” unde se setează diferite opțiuni pentru editarea de legături.
	Libraries	Alege bibliotecile care se pot utiliza la linkeditare.
	Librarian...	Deschide fereastra de dialog “Librarian” cu diferite opțiuni de lucru cu biblioteci.
	Debugger...	Selectează o serie de opțiuni care afectează depanatorul integrat.
	Directories...	Specifică directoarele care conțin fișierele necesare pentru compilare, linkeditare precum și unde se vor salva fișierele executabile
Environment		Stabilește mediul de lucru Borland C++.
	Preferences...	Deschide fereastra de dialog “Preferences” cu opțiuni privind comportamentul și aspectul fizic al mediului integrat.
	Editor...	Deschide o fereastră dialog “Editor Options” unde sunt setate diferite opțiuni ale editorului de texte.
	Mouse	Deschide o fereastră de dialog “Mouse Options” unde sunt setate opțiuni cu privire la utilizarea mouse-ului.
	Desktop...	Deschide fereastra de dialog “Desktop Preferences” unde se specifică dacă conținutul desktop-ului se salvează între sesiuni.

Comanda		Funcție îndeplinită
	Startup...	Deschide fereastra de dialog "Startup Options" unde se specifică diferite opțiuni de lansare a mediului integrat.
	Colors...	Afișează fereastra de dialog "Colors" unde se selectează culorile pentru fiecare componentă a mediului.
Save...		Deschide fereastra de dialog "Save Options" unde se specifică fișierul care va conține toate setările făcute în cadrul meniurilor Options și Search.

### Submeniul Window

Meniul Window oferă opțiuni de gestionare a ferestrelor. Aceste opțiuni permit închiderea, deschiderea, afișarea ecranului utilizator, trecerea dintr-o funcție în alta.

**Tab. 9**

Comanda	Combinatia de taste	Funcție îndeplinită
Size/Move	Ctrl+F5	Schimbă dimensiunile sau poziția ferestrei active.
Zoom	F5	Redimensionează fereastra activă la dimensiunea maximă. Dacă fereastra este deja maximă, în urma comenzii fereastra revine la dimensiunea anterioară.
Cascade		Afișează în cascadă (una peste alta) ferestrele de editare deschise, fereastra activă fiind cea superioară.
Tile		Alătură ferestrele pe orizontală astfel încât pot fi văzute simultan.
Next	F6	Activează fereastra următoare.
Close	Alt+F3	Închide fereastra activă.
Close all		Închide toate ferestrele.
Message		Deschide fereastra de mesaje "Message" unde sunt afișate mesaje de eroare și avertisment.
Output		Deschide fereastra "Output" unde se afișează textul din linia de comandă DOS și textul generat de programul care se execută.
Watch		Deschide fereastra de supraveghere „Watch” unde se afișează valorile unor expresii în timpul execuției programului (în mod depanare).
User screen	Alt+F5	Vizualizează în mod text și grafic rezultatele obținute în urma execuției unui program, revenirea la ecranul mediului făcându-se prin apăsarea oricărei taste.
Register		Deschide fereastra „Register” unde se afișează regiștrii și indicatorii CPU.
Project		Deschide fereastra „Project” unde sunt listate fișierele folosite la crearea unui proiect.
Project notes		Deschide un fișier Project note unde se scriu note și informații legate de fișierele unui proiect.
List all	Alt+O	Deschide fereastra „Window List” care conține lista cu toate ferestrele deschise curent și toate fișierele sursă deschise după momentul lansării în execuție a

Comanda	Combinatia de taste	Funcție îndeplinită
		programului.

## Submeniul Help

Submeniul Help furnizează o serie de opțiuni de asistență soft permanentă a programatorului.

**Tab. 10**

Comanda	Combinatia de taste	Funcție îndeplinită
Contents		Afișează cuprinsul comenzilor conținute în documentația de ajutor cu privire la modul de utilizare al Help-ului, meniuri și taste funcționale, comenzile editorului, linia de comandă, limbaj, mesaje de eroare, funcții, fișiere antet etc.
Index	Shift+F1	Afișează un index alfabetic al noțiunilor din documentația de ajutor. Selectarea conținutului unei noțiuni se face prin deplasarea cu tastele săgeți a barei de selecție sau prin tastarea primelor litere din numele acesteia, ambele moduri fiind urmate de apăsarea tastei <Enter>.
Topic search	Ctrl+F1	Oferă informații referitoare la sintaxa și semantica limbajului C (funcții, instrucțiuni, cuvinte-cheie, fișiere antet, constante, variabile etc).
Previous Topic	Alt+F1	Reafișează fereastra Help anterioară. Dacă se repetă comanda, mediul integrat permite reafișarea ultimelor 20 ferestre de ajutor vizualizate.
Help on help		Afișează o fereastră Help în care este descris modul de apel și folosire al help-ului inclus.
Active file...		Afișează o fereastră de dialog unde se selectează fișierul Help utilizat de mediul integrat.
About		Furnizează informații despre versiunea produsului Borland C++ și despre drepturile de autor.

## Fereastra de mesaje

**Fereastra de mesaje** care apare în partea de jos a ecranului afișează mesajele de eroare și avertisment care sunt rezultatul compilării și linkeditării. Pentru a închide o fereastră de mesaje se apasă cu mouse-ul pe cutia de închidere [◇].

Prima eroare va fi marcată de o bară supraluminată, locul acesteia va fi indicat în fereastra de editare printr-o linie și ea supraluminată. Dacă mesajul de eroare este prea lung, se pot utiliza tastele săgeți pentru deplasarea sa pe orizontală. Pentru vizionarea altor mesaje se folosesc tastele ↑ și ↓ pentru deplasarea barei supraluminate din fereastra de mesaje. Odată cu aceasta se va deplasa și bara supraluminată din fereastra editorului, poziționându-se pe linia noii erori.

Pentru corectarea unei erori se plasează bara supraluminată din fereastra de mesaje pe mesajul de eroare corespunzător și se tastează <Enter>. Cursorul se va poziționa în fereastra editorului pe locul care a generat mesajul de eroare, iar linia de stare afișează mesajul ales.

Acum eroarea poate fi corectată. Dacă există mai multe erori se poate reveni în fereastra de mesaje tastând F6 și se selectează următorul mesaj de eroare.

### Exemplu:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
void func(double, double*, double*, double*);
void main(void)
{
int i, n_eq, n_iter;
double a,b,d,eps,x1,x2,y,yd,ys;
clrscr();
printf("\n\n\t\t Metoda newton-raphson \n");
printf("\n\t a="); scanf("%le",&a);
printf("\n\t b="); scanf("%le",&b);
printf("\n\t nb [numarul maxim de iteratii] = "); scanf("%d",&n_iter);
printf("\n\t eps [distanța maximă între două rădăcini] = "); scanf("%le",&eps);
func(a, &y, &yd, &ys);
x1=y*ys<0.0 ? b : a;
for(i=0;i<n_iter;i++)
{
func(x1, &y, &yd, &ys);
x2=x1-y/yd; d=fabs(x1-x2);
printf("\n\t i=%d x=%25.16le", i, x2);
if (d<eps)
{
printf("\n\t numărul de iteratii = %4d"
"\n\t\t [x2-x1] = %le < %le\n\t\t r = %16.9le",
i,d,eps,x2);
getch(); exit(0);
}
x1=x2;
}
func(x2, &y, &yd, &ys);
printf("\n\n\t numărul de iteratii = %4d |x1-x2|=%le"
"\n\t r=%25.16le f(r)=%25.16le",
n_iter, d, x2, y);
getch();
}

void func(double x, double *y, double *yd, double *ys)
{
*y=pow(x,5)-5.0;
*yd=5.0*pow(x,4);
*ys=20.0*pow(x,3);
}
```

**Cerințe:**

1. Salvați programul în directorul dvs.
2. Realizați modificări ale acestuia utilizând submeniul Edit (nealterând însă programul de bază).
3. Realizați căutări în program (submeniul Search).
4. Realizați compilarea programului
5. Rulați programul
6. În cazul în care apar erori, folosiți submeniul Debug pentru corectarea acestora.
7. Căutați în submeniul Help funcțiile utilizate în program.