ЛАБОРАТОРНА РОБОТА № 2_2
ПРИНЦИПИ РОБОТИ В СИСТЕМІ ПРОГРАМУВАННЯ TURBO C ++

Мета роботи: отримання базових практичних навиків використання середовищ програмування (*programming environment*), необхідних для подальшого виконання циклу лабораторних робіт.

1 Порядок виконання роботи

1) Ознайомтесь з теоретичними відомостями про роботу в середовищі програмування Turbo C ++.
2) Виконайте налаштування параметрів середовища програмування, а саме: задайте шляхи до основних каталогів середовища, задайте параметри компіляції (*compilation*), програми, компонувальника (*linker*) та процесу відлагодження (*debugging*). Збережіть встановлені параметри.
3) Наведіть приклади довідкової інформації з контекстної довідкової системи (*context help*) середовища програмування.
4) Згідно з п. 2.4 теоретичних відомостей напишіть просту програму мовою програмування С та збережіть на диску Y , у власній папці. Виконайте компіляцію програми. Виправте необхідні помилки та запустіть програму на виконання.
5) Запустіть програму в покроковому режимі та прослідкуйте за зміною значень змінних в процесі відлагодження програми. Встановіть декілька контрольних точок та виконайте відлагодження програми.
6) Підготуйте звіт та зробіть висновки.

В ході цієї лабораторної роботи студенту пропонується виконати ряд дій, які повинні познайомити його з використанням середовища програмування для розробки, компіляції, виконання і відлагодження програм. Використання засобів, що автоматизують зазначені етапи є основою успішної розробки професійних програмних систем в будь-якому іншому середовищі.

2 Теоретичні відомості

2.1 Основи роботи в середовищі *Turbo C ++*

Підготовка каталогу і запуск середовища програмування

Середовище програмування ***Turbo C ++*** складається з великого числа файлів, зберігання яких структуроване по каталогах. Кореневий каталог середовища програмування потрібно переписати до власного ди ску Y:\: ***Y:\TC***. Файл запуску середовища ***D:\TC\BIN\TC.EXE***.

Решта каталогів є підкаталогами в цьому каталозі. Основні з них:

Y:\TC\BIN - у цьому підкаталозі розміщені програмні модулі середовища програмування, підказка і т. д.

Y:\TC\INCLUDE - у цьому підкаталозі розміщені файли-заголовки з описами стандартних функцій середовища програмування.

Y:\TC\LIB - у цьому підкаталозі розміщені бібліотеки стандартних функцій середовища програмування.

2.2 Настроювання робочого середовища

Нижче наведені параметри середовища програмування, які потрібно встановити для його роботи.

Виберіть в ***Головному Меню*** пункт ***Options****(Параметри)*, і в підменю, що відкрилось, виберіть пункт ***Directories****...(Каталоги…)* (далі послідовність виборів з меню ми подаємо у вигляді: ***Головне Меню -> Options -> Directories****...)*.

В полях діалогу, що з'явився на екрані, встановіть значення шляхів, які відповідають місце розташуванню програми, наприклад для рядка Include Directories - Y:\TC\INCLUDE і т.д.

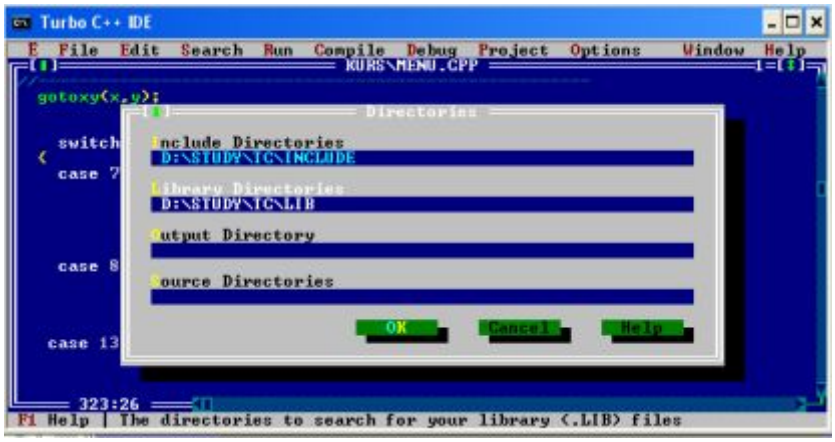

Рисунок 1 – Встановлення шляхів середовища

Після встановлення значень натисніть кнопку ***OK***.

Встановлені значення задаватимуть пошук файлів-заголовків і бібліотек в стандартних каталогах середовища програмування. Ті ж програмні модулі, які створюватимете Ви – текстові, об'єктні і завантажувальні, – розміщуватимуться у Вашому поточному каталозі.

Для налаштування параметрів компіляції виконайте такі дії:

Виберіть команду головного меню: ***Options -> Compiler -> Code generation...***

В полях діалогу, що з'явився на екрані, встановіть такі значення:

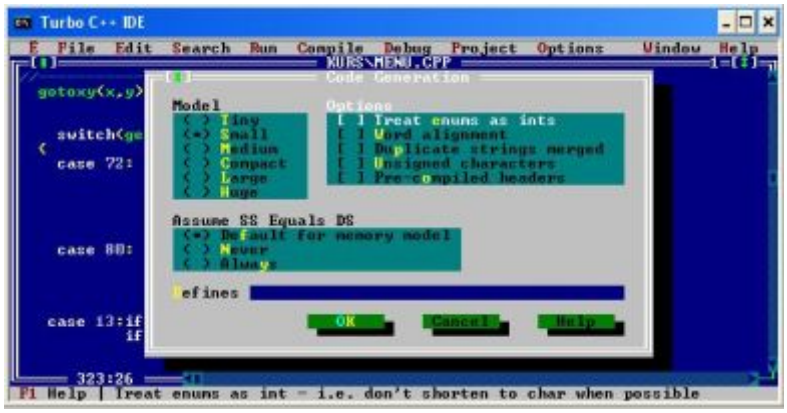

Рисунок 2 – Налаштування параметрів компіляції

Після встановлення значень натисніть кнопку ***OK***.

Виберіть команду головного меню: ***Options - > Compiler -> Advanced code generation...***

В полях діалогу, що з'явився на екрані, встановіть такі значення:

Рисунок 3 – Розширені налаштування компіляції

Після встановлення значень натисніть кнопку ***OK***.

Виберіть команду головного меню: ***Options -> Compiler -> Entry/Exit Code...***

В полях діалогу, що з'явився на екрані, встановіть такі значення:

Рисунок 4 – Параметри генерування коду

Після встановлення значень натисніть кнопку ***OK***.

Виберіть команду головного меню: ***Options -> Compiler -> Source...***

В полях діалогу, що з’явився на екрані, встановіть такі значення:

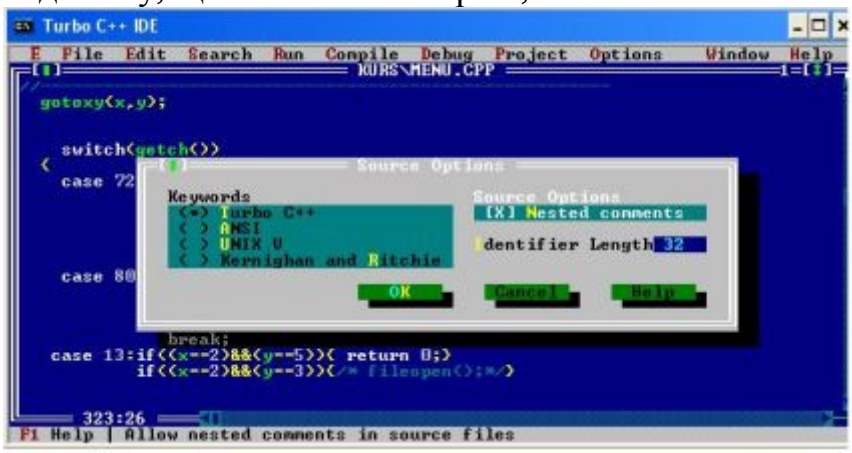

Рисунок 5 – Вибір стандарту

Після встановлення значень натисніть кнопку ***OK***.
Виберіть команду головного меню ***Options -> Make ...***
В полях діалогу, що з’явився на екрані, встановіть такі значення:

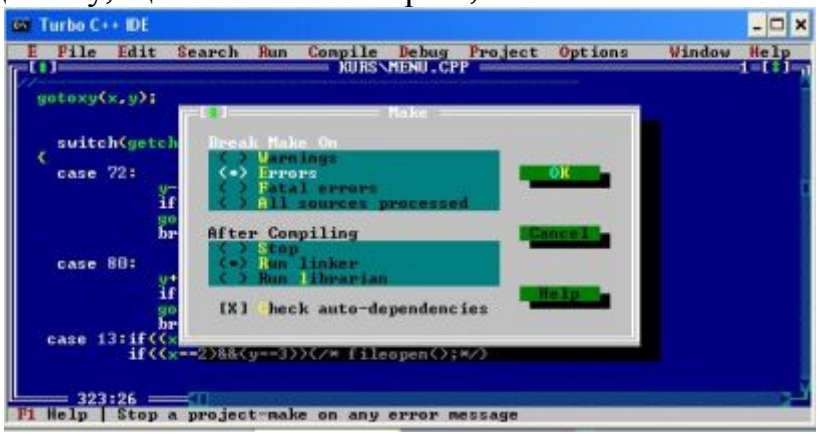

Рисунок 6 – Вибір налаштувань побудови

Після встановлення значень натисніть кнопку ***OK***.
Виберіть команду головного меню: ***Options -> Linker -> Settings ...***
В полях діалогу, що з’явився на екрані, встановіть такі значення:

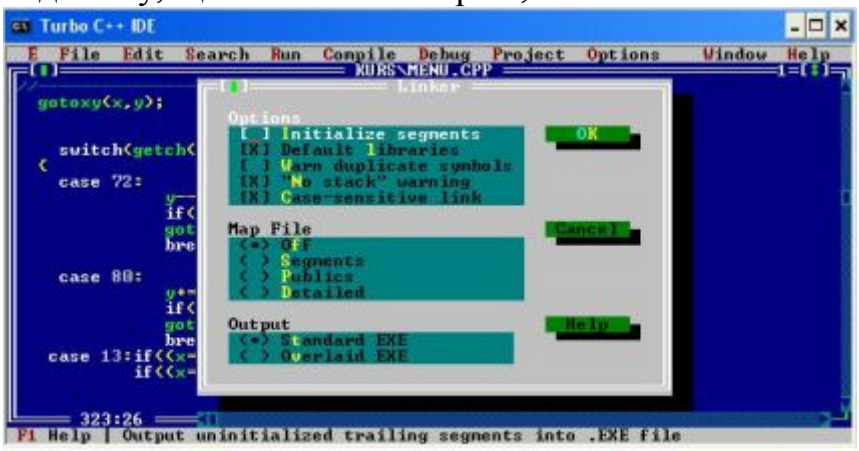

Рисунок 7 – Налаштування лінкування

Після встановлення значень натисніть кнопку ***OK.***

Виберіть в головному меню команду ***Options -> Linker -> Libraries ...***
В полях діалогу, що з'явився на екрані, встановіть такі значення:

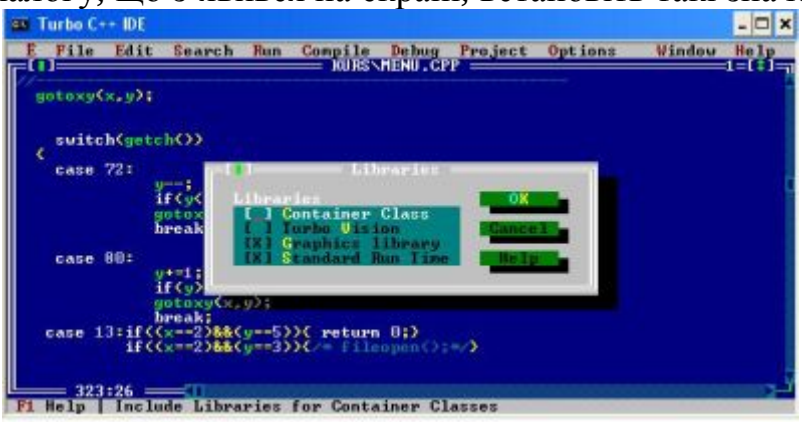

Рисунок 8 – Налаштування лінкування для бібліотек

Після встановлення значень натисніть кнопку ***OK.***
Виберіть в головному меню команду ***Options - > Debugger...***
В полях діалогу, що з'явився на екрані, встановіть такі значення:

Рисунок 9 – Налаштування відлагодження

Після встановлення значень натисніть кнопку ***OK***.
Виберіть в головному меню команду ***Головне Меню -> Options -> Save…***. У вікні, що з'явилось на екрані, натисніть кнопку ***OK***.

2.3 Користування підказкою

Середовище програмування ***Turbo C++*** має систему онлайн-підказки, яка дуже корисна, особливо для програміста-початківця.

Підказка є гіпертекстовою, тобто, в усіх текстах, які з'являються на екрані при виведенні підказки, є деякі виділені слова. Вибравши одне з виділених слів, можна одержати підказку, пов'язану з цим словом.

Перша можливість звернення до підказки – через ***Головне Меню -> Help***. В меню ***Help*** найкориснішими є пункти ***Contents*** і ***Index***.

Пункт ***Contents*** виводить на екран тематичний список розділів, з яких можна одержати підказку. Зі всіх цих розділів перш за все ми рекомендуємо:

How to Use Help – як користуватися підказкою;
Menus and Hot Keys – меню і гарячі клавіші;
Editor Commands – команди редактора;

Turbo C++ Language – мова ***Turbo C++***;
Error Messages – повідомлення про помилки;
Functions – функції;
Header Files – файли заголовків.

Вибір будь-якого з цих розділів приводить до появи списку підрозділів і так далі.

Пункт ***Index*** виводить на екран алфавітний список всіх ключових слів і імен функцій мови і середовища програмування ***Turbo C++.*** Вибираючи пункт цього списку, можна одержати докладну підказку щодо нього.

Інша можливість звернення до підказки – через клавіші ***F1*** і комбінацію клавіш ***Ctrl+F1***. Ця підказка – контекстно-залежна, тобто, підказка, яка з'являється на екрані, стосується саме того стану середовища програмування, в якому вона зараз знаходиться.

За клавішею ***F1*** Ви одержуєте підказку стосовно активного на даний момент меню або вікна.

Комбінацією клавіш ***Ctrl+F1*** можна користуватися тільки в активному вікні редактора. Якщо при натисканні цієї комбінації курсор знаходиться на якомусь ключовому слові або на функції, то на екран виводиться докладна підказка щодо цього ключового слова або функції. Інакше дія цієї комбінації аналогічна ***Головне Меню -> Help -> Index.***

2.4 Створення, редагування і збереження програми

Виберіть в головному меню команду ***File -> New.*** На екрані відкриється порожнє вікно редактора із заголовком ***00.CPP***. У цьому вікні наберіть текст програми, наведений нижче (нумерацію рядків, яка дана зліва, набирати не потрібно).

```
1  #include <stdio.h>
2  int main(void){
3   int a, b, c, d, x, y;
4     a=1;
5     b=2;
6     c=a+b;
7     d=a*b;
8     if (c==d) {
9 x=100;
10 y=200;
11 }
12     else {
13 x=200;
14 y=100;
15 }
16     printf("%d %d\n",x,y);
17     return 0;
18  }
```

При наборі і подальшому редагуванні тексту намагайтесь якнайчастіше використовувати такі спеціальні клавіші і комбінації клавіш, як: ***Home, End, Ctrl<-, Ctrl->, Ctrl+Y*** і ін. Також корисними можуть бути блокові команди: ***Ctrl+K B*** (виділити початок блоку), ***Ctrl+K K*** (виділити кінець блоку), ***Ctrl+K B*** (копіювати блок), ***Ctrl+K V*** (перемістити блок), ***Ctrl+K H*** (відмінити виділення). Докладну підказку щодо спеціальних клавіш і комбінацій можна одержати, обравши в головному меню команду ***Help -> Contents -> Editor Commands***.

Текст, який Ви набираєте, зберігається в оперативній пам'яті, тому, якщо під час набору виникне якась аварія (наприклад, вимкнення живлення), все, що Ви набрали, буде втрачено. Щоб уникнути такої небезпеки, потрібно зберегти текст у файлі на зовнішній пам'яті. Набравши декілька перших рядків, збережіть текст. Для цього оберіть команду головного меню ***File -> Save as...*** . При цьому екрані з'являється діалог ***Save File As.*** У верхньому полі цього діалогу введіть ім'я, під яким Ви хочете зберегти текст програми. Якщо Ви введете тільки ім'я (без розширення), система автоматично додасть до імені Вашої програми розширення ***CPP*** – стандартне розширення для програм, написаних на мові **C++**. Ми рекомендуємо явно задавати розширення, наприклад, ***myfile.c***, оскільки програми, які ми пишемо в цій частині лабораторного практикуму, обмежуються можливостями базової мови С. Набравши ім'я, натисніть на клавішу ***Enter*** або натисніть на кнопку ***OK***.

При першому збереженні програми Ви дали їй ім'я (зверніть увагу – це ім'я тепер є заголовком вікна редактора). Далі після кожного додавання декількох рядків або при внесенні деякої кількості змін в текст зберігайте програму під тим же ім'ям. Для цього достатньо викликати команду головного меню ***File -> Save*** або натиснути клавішу ***F2***.

Згодом, якщо Вам потрібно буде знову завантажити в редактор текст тієї ж програми, оберіть команду головного меню ***File -> Open*** (або натисніть клавішу ***F3***). Відкриється діалог ***Open a File***, який дуже схожий на діалог ***Save File As***. У верхньому полі цього діалогу виводиться маска для файлів, з яких можна вибирати файл для відкриття, в нижньому – список файлів, імена яких відповідають цій масці. Змінюючи маску у верхньому полі, Ви змінюєте склад списку в нижньому. Досягнувши того, що в нижньому полі буде саме той список, який Вам потрібен, перейдіть в нижнє поле. Виберіть потрібний файл і натисніть на клавішу ***Enter*** або натисніть на кнопку ***OK***. Файл, який Ви обрали, відкриється у вікні редактора.

2.5 Компіляція і виконання програми

Закінчивши набір тексту і зберігши файл програми на диску, виконайте компіляцію програми. Для цього оберіть в головному меню команду ***Compile -> Compile*** або натисніть комбінацію клавіш ***Alt+F9***. На екрані з'являється вікно ***Compiling***, в якому відображається хід компіляції. При нормальному завершенні компіляції в нижньому рядку цього вікна повинно бути виведено повідомлення ***Success: Press any key***

Якщо там виводиться повідомлення

Errors: Press any key

або

Warnings: Press any key,

то Ваша програма вимагає корекції.

Роботу з діагностикою помилок ми докладніше розглянемо в наступному розділі. Після того, як Ви відкомпілювали програму без помилок, запустіть її на виконання. Для цього оберіть в головному меню команду ***Run -> Run*** або натисніть комбінацію клавіш ***Ctrl+F9***. На екрані з'являється вікно ***Linking***, в якому відображається хід компонування. При нормальному завершенні компонування це вікно зникає само собою і виконується програма.

Якщо ж в цьому вікні виводиться повідомлення ***Errors: Press any key***, то Ваша програма вимагає корекції.

Якщо Ваша програма не вимагає введення даних (а саме така та програма, з якою ми зараз працюємо), Ви навіть не встигнете помітити, як ця програма виконується. Щоб подивитися результати, які програма видала на екран, оберіть в головному меню команду ***Window -> User screen*** або натисніть комбінацію клавіш ***Alt+F5***. Ви побачите чорний екран з тим, що вивела Ваша програма (у нашому випадку це повинне бути: 200 100). Щоб вийти з режиму перегляду результатів, натисніть будь-яку клавішу.

Перевірте зміст Вашого робочого каталогу. Якщо текст Вашої програми був збережений у файлі ***myfile.c***, то після компіляції в каталозі повинен з'явиться файл ***myfile.obj***, а після виконання – ще і ***myfile.exe.***

2.6. Діагностика помилок і попереджень компілятора і компонувальника

На цьому етапі виконання лабораторної роботи ми пропонуємо Вам поекспериментувати з повідомленнями компілятора і компонувальника. Заздалегідь рекомендуємо зробити копію файлу програми. Для цього оберіть в головному меню команду ***File -> Save as...*** і введіть якесь нове ім'я програми, наприклад: ***myfilex.c***. Тепер у Вас є дві копії програми в двох файлах (***myfile.c*** і ***myfilex.c***). Одна копія (хай це буде ***myfile.c***) зберігатиме правильну версію програми, а в іншу (це буде ***myfilex.c***) ми навмисне вводитимо помилки.

Відкрийте в текстовому редакторі файл ***myfilex.c***. Внесіть такі зміни в текст програми (тут і далі ми вказуємо номери рядків, в які потрібно внести зміни):

```
 6 c=a+b1;
 9 x=100; a123
18  /*}*/
```

Запустіть програму на компіляцію (***Ctrl+F9***). Ви одержите повідомлення ***Errors: Press any key*** у вікні ***Compiling***. Коли Ви натиснете будь-яку клавішу, в нижній частині екрана відкриється вікно ***Message*** з таким вмістом:

```
Compiling MYFILEX.C:
Error       MYFILEX.C          6: Undefined symbol 'b1'
```

Error MYFILEX.C 10: Undefined symbol 'a123'
Error MYFILEX.C 10: Statement missing ;
Error MYFILEX.C 17: Compound statement missing }

Це повідомлення компілятора про помилки. Перший рядок – заголовок. У наступних рядках: ознака помилки, ім'я файлу, в якому знайдена помилка, номер рядка тексту програми, в якому була знайдена помилка, діагностика помилки.

Повідомлення до рядка 6 – «***Невизначений символ 'b1'»***. У цьому операторі використовується змінна з таким ім'ям, якої немає серед оголошених змінних.

Перше повідомлення до рядка 10 – «***Невизначений символ 'a123'»***. Текст ***a123*** схожий на ім'я змінної, але така змінна не оголошена. Інше повідомлення до рядка 10 – «***В операторі відсутній ;***» . Текст ***a123*** може бути окремим оператором, але в ньому немає ознаки кінця оператора. Зверніть увагу на те, що хоча помилку ми внесли в рядок 9, повідомлення видається до рядка 10, оскільки помилка була знайдена тільки при обробці цього рядка.

Повідомлення до рядка 17 – «***В складеному операторі відсутній }***» – компілятор виявив непарність операторних дужок { }. Де б не була пропущена закривальна операторна дужка, її відсутність може бути знайдена тільки на останньому операторі програми.

Коли Ви переміщаєтеся списком повідомлень про помилки у вікні ***Message***, у вікні редактора те місце тексту програми, якого стосується поточне повідомлення, виділяється кольором. Коли Ви перемикаєтеся у вікно редактора (клавіша ***F6***), курсор встановлюється на це саме місце.

Відновіть правильний вміст файлу ***myfilex.c***. (Це можна зробити, відкривши в редакторі файл ***myfile.c*** і знов зберігши його з ім'ям ***myfilex.c.*** При цьому Ви одержите попередження про те, що файл ***myfilex.c*** буде змінений, на яке Вам потрібно відповісти ***Yes***.) Внесіть такі зміни в текст програми:

```
3   int a, b, з, d, x, y, z=2;
4     a=b;
8     if (c=d) {
```

Запустіть програму на компіляцію (***Ctrl+F9***). Ви одержите повідомлення: ***Warnings: Press any key*** у вікні ***Compiling.*** Коли Ви натиснете будь-яку клавішу, в нижній частині екрана відкриється вікно ***Message*** з таким текстом:

Compiling MYFILEX.C:
Warning MYFILEX.C 4: Possible use of 'b' before definition
Warning MYFILEX.C 8: Possible incorrect assignment
Warning MYFILEX.C 18: 'z' is assigned value that is never used
Warning MYFILEX.C 18: 'c' is assigned value that is never used

Це попередження компілятора.

Попередження до рядка 4 – «***Можливе використовування 'b' до визначення***». У цьому операторі значення змінної ***b*** привласнюється змінній ***a***, але яке значення має ***b*** на цей момент виконання програми – невідомо.

Попередження до рядка 8 – «***Можливе некоректне привласнення».*** Вираз ***c=d*** має сенс: «***привласнити змінній c значення змінної d***». Вираз ***c==d,*** який застосований в правильній програмі, має сенс: «***порівняти змінні c і d***». Оскільки вираз внесений в умовний оператор, компілятор має підстави припускати, що тут повинно бути порівняння, а не привласнення.

Попередження до рядка 18 – ***«Змінній 'z' привласнюється значення, яке ніде не використовується».*** У операторі 3 ми дали змінній ***z*** початкове значення 2. Але далі в програмі значення ***z*** ніде нічому не привласнюється і ні з чим не порівнюється.

Ще одне таке ж попередження до рядка 18 стосується змінної *c*. Привласнення значення цій змінній відбувається в рядку 8. Ситуація невикористання значення може бути виявлена тільки в кінці програми.

Ви можете одержати не всі попередження з тих, які тут перераховані. При налаштуванні середовища можна відмінити або відновити видачу тих або інших попереджень. Щоб зробити відповідне налаштування, оберіть в головному меню команду ***Options -> Compiler -> Messages.***

Відновіть правильний вміст файлу ***myfilex.c.*** Внесіть такі зміни в текст програми:

```
4     a=1; abc();
```

Запустіть програму на виконання (***F9***).

Ви одержите повідомлення ***Errors: Press any key*** у вікні ***Linking***. У вікні ***Message*** буде:

```
Linking MYFILEX.EXE:
Linker Error:    Undefined symbol _abc in module MYFILEX.C
```

Це повідомлення компонувальника. «***Невизначений символ _abc в модулі MYFILEX.C***». Синтаксично оператор ***abc();*** є звертанням до функції, саме так його трактує компілятор. Але коли компонувальник намагається знайти функцію з таким ім'ям в доступних йому модулях і бібліотеках, він її не знаходить, про що і повідомляє нас.

Якщо Ви запустите окремим кроком компіляцію тієї ж програми, Ви одержите:

```
Compiling MYFILEX.C:
Warning      MYFILEX.C      4: Call to function 'abc' with
                            no prototype
```

Компілятор попереджає про те, що в програмі є «***Виклик функції 'abc' без прототипу***».

2.7 Відлагодження програми

В середовищі програмування є відладник, який працює на рівні текстового коду. Використовуючи його, Ви можете виконувати програму в покроковому режимі, встановлювати брейкпойнти виконання і стежити за поточними значеннями змінних програми.

Відкрийте в текстовому редакторі файл ***myfile.c*.**

Оберіть в головному меню команду ***Debug -> Watches -> Add watch*** (або натисніть комбінацію клавіш ***Ctrl+F7***). У вікні ***Add Watch***, яке з'явиться на екрані, введіть в полі ***Watch Expression*** ім'я змінної ***a***, натисніть кнопку ***OK***. У нижній частині екрана з'явиться вікно ***Watch***, а в ньому – «***a: Undefined symbol 'a'***».

Повторіть ці дії кілька разів, вводячи в полі ***Watch Expression*** імена ***b, з, d***. У вікні ***Watch*** додаватимуться аналогічні повідомлення. Цими діями ми даємо середовищу програмування інструкцію відстежувати і відображати у вікні ***Watch*** поточні значення вибраних змінних програми. Оскільки програма ще не виконується, ці змінні поки що «невідомі» системі програмування, про що і свідчать повідомлення.

Зробіть активним вікно редактора і натисніть клавішу ***F8***. У вікні редактора кольором (швидше за все – блакитним) буде виділений рядок 2 тексту програми. Це ми почали відлагодження нашої програми в покроковому режимі. Ще раз натисніть клавішу ***F8***. Виділення зміститься на рядок 4, а у вікні відображатимуться якісь значення змінних. Виконання програми почалося і ці змінні вже «відомі», але їх значення ще не встановлені, тому вони – якісь випадкові числа. З наступним натисканням клавіші ***F8*** значення змінної ***a*** зміниться на 1. Кожне наступне натискання клавіші ***F8*** просуватиме виконання програми на оператор вперед і, відповідно до виконання операторів програми, мінятимуться значення змінних.

Зверніть увагу на те, що після рядка 8 виконання зразу ж «перестрибне» на рядок 13. Оскільки умова в умовному операторі 8 не виконується, виконання обходить рядки 9–12. Якщо Ви змініте: ***4 a=2;*** і знову виконаєте програму в покроковому режимі, умова в рядку 8 виконуватиметься, отже, і виконання програми пройде через рядки 9, 10, 11 і обійде рядки 12–15.

Закінчивши виконання програми, встановіть курсор на рядок 8 і оберіть в головному меню команду ***Debug -> Toggle breakpoint*** (або натисніть комбінацію клавіш ***Ctrl+F8***). Рядок 8 виділиться кольором (швидше за все – червоним). Цим ми задали брейкпойнт програми. Тепер запустіть програму на виконання (***F9***). Виконання зупиниться на рядку 8. При цьому у вікні ***Watch*** відображатимуться поточні значення змінних. Ви можете продовжити виконання в покроковому (***F8***) або в автоматичному режимі (***F9*)**.

Команди ***Debug*** дають можливість повністю керувати стеженням за значеннями і точками зупинки.

3 Контрольні питання

1. Яке основне призначення середовища Turbo C++?
2. Опишіть структуру головного вікна середовища. Дайте характеристику складових елементів.
3. Опишіть головне меню, його основні пункти та їх призначення.
4. Як створити новий файл?
5. Які основні операції роботи з текстом в вікні текстового редактора? Які комбінації клавіш використовуються у редакторі?
6. Опишіть алгоритм запуску програми.
7. Опишіть основні етапи відлагодження програми.
8. Як переглянути значення програмних об’єктів під час роботи програми?
9. Розробіть найпростішу програму на мові C++, виведення привітання на екран та покажіть основні етапи відлагодження програми.