

Міністрество освіти і науки України  
Національний технічний університет  
«Харківський політехнічний інститут»  
Кафедра «Обчислювальна техніка та програмування»

## **ЗВІТ**

Про виконання лабораторної роботи №4  
« Розробка лінійних програм »

Виконавець: Мокрослоєв А.В. (КІТ-120є)  
Перевірив викладач: Бульба С. С.

Харків 2020

# Лабораторна робота № 4. Розробка програм, що розгалужуються

## 1. Вимоги

### 1.1 Розробник

Студент групи КІТ-120є Мокрослоєв Антон Вікторович

### 1.2 Загальне завдання

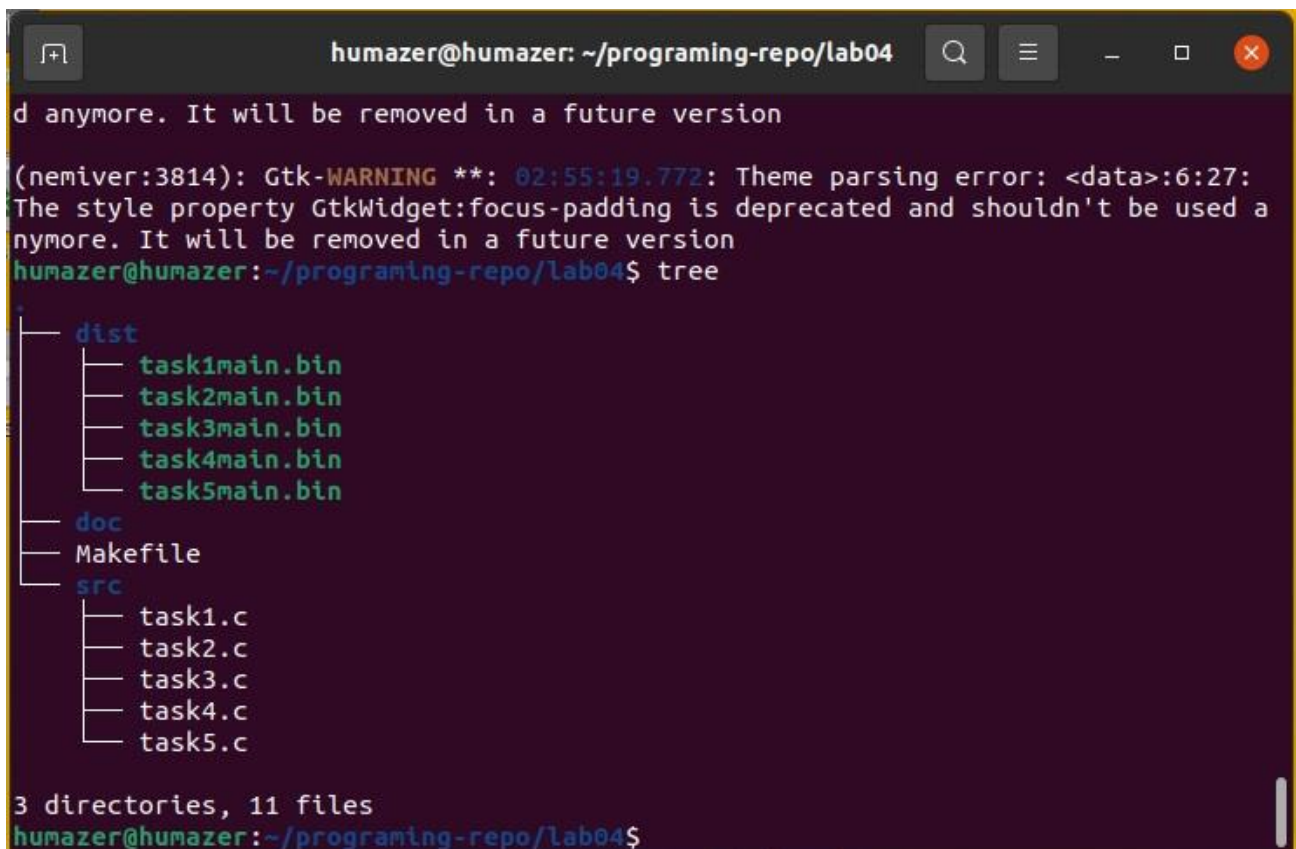
Виконати всі завдання з категорії “відмінно”

#### Індивідуальне завдання:

№1.

## 2. Опис програми

2.1 Було створено структуру проекту (Рис. 1), створено папку *lab04*, файл *Makefile* для компіляції та папку *src* для вихідного коду.



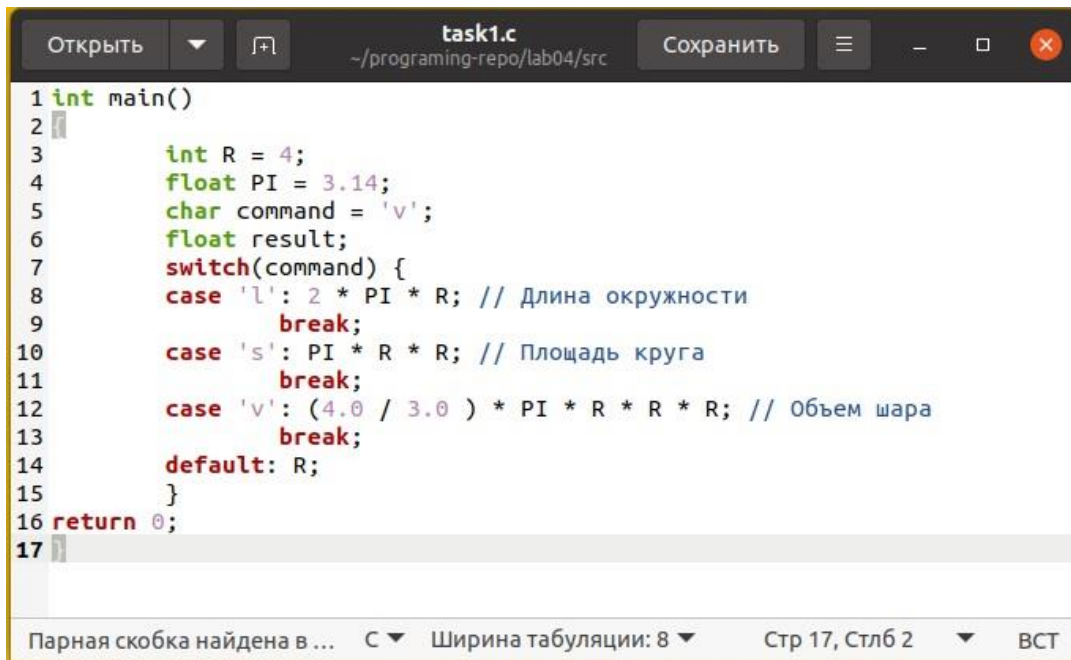
```
humazer@humazer: ~/programming-repo/lab04
d anymore. It will be removed in a future version
(nemiver:3814): Gtk-WARNING **: 02:55:19.772: Theme parsing error: <data>:6:27:
The style property GtkWidget:focus-padding is deprecated and shouldn't be used a
nymore. It will be removed in a future version
humazer@humazer:~/programming-repo/lab04$ tree
.
├── dist
│   ├── task1main.bin
│   ├── task2main.bin
│   ├── task3main.bin
│   ├── task4main.bin
│   └── task5main.bin
├── doc
├── Makefile
└── src
    ├── task1.c
    ├── task2.c
    ├── task3.c
    ├── task4.c
    └── task5.c

3 directories, 11 files
humazer@humazer:~/programming-repo/lab04$
```

Рисунок 1 – Структура проекту

2.2 Виконано задачі 1-5, код збережено у файлах *task1.c* – *task5.c*.

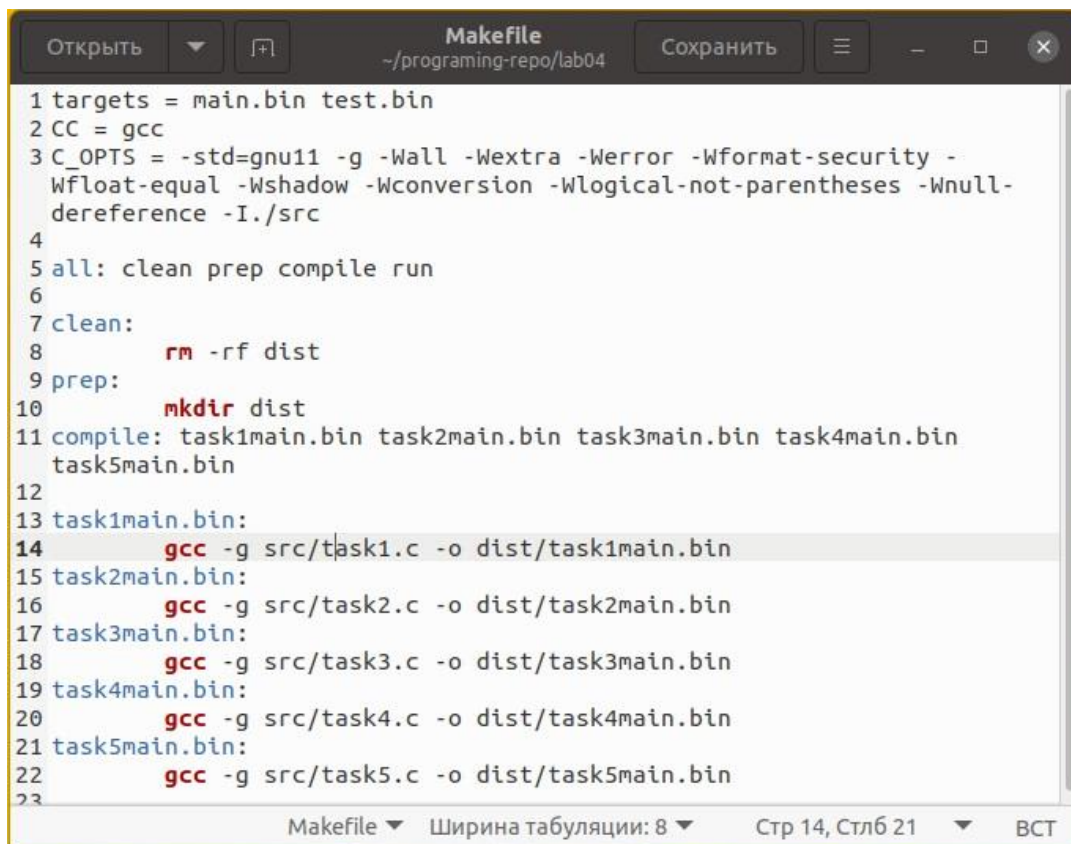
### 2.3 Оголошено змінні: $R$ – радіус, $PI$ = число $\Pi$ .



```
1 int main()
2 {
3     int R = 4;
4     float PI = 3.14;
5     char command = 'v';
6     float result;
7     switch(command) {
8     case 'l': 2 * PI * R; // Длина окружности
9         break;
10    case 's': PI * R * R; // Площадь круга
11        break;
12    case 'v': (4.0 / 3.0) * PI * R * R * R; // Объем шара
13        break;
14    default: R;
15    }
16    return 0;
17 }
```

Рисунок 2 – Реалізація програми

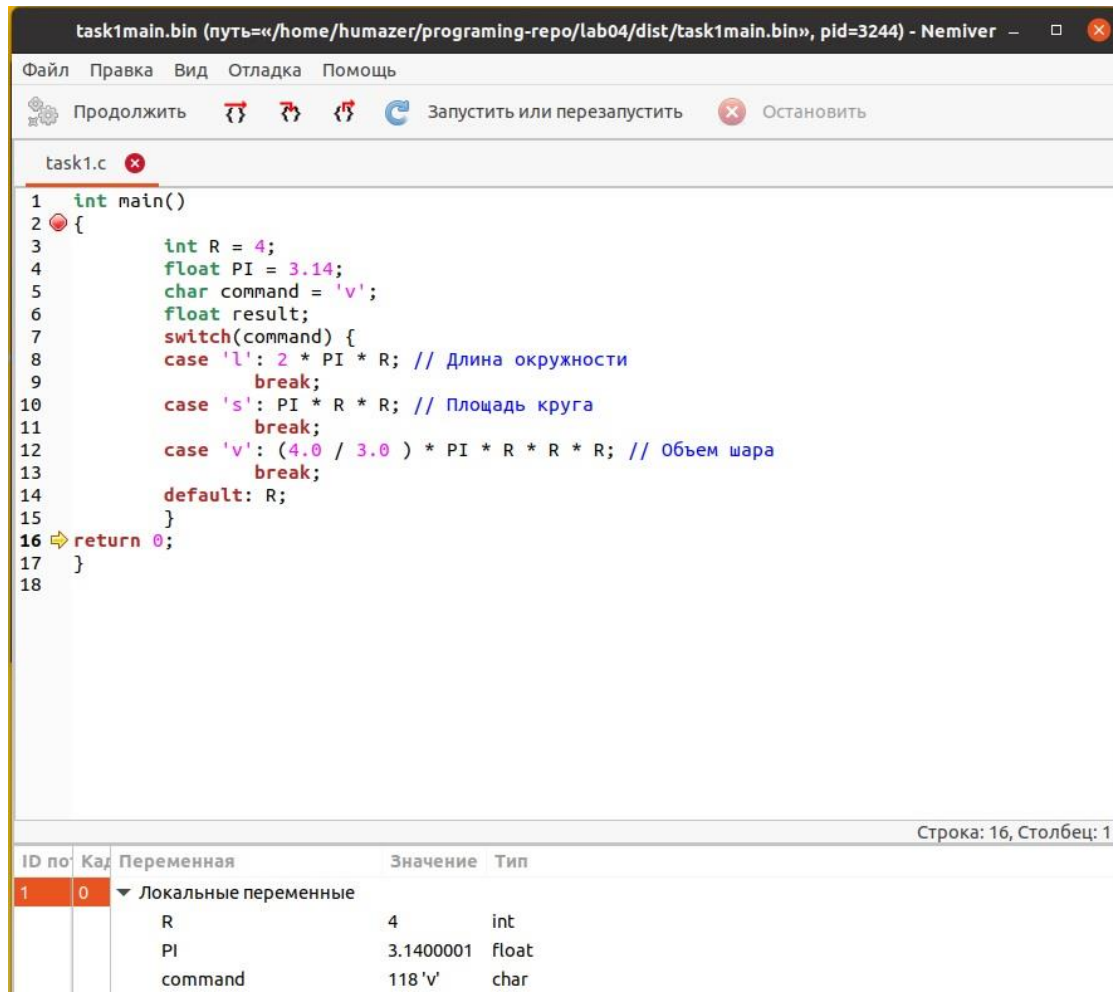
### 2.4 Внесено зміни до *Makefile* для компіляції проекту (Рис. 3).



```
1 targets = main.bin test.bin
2 CC = gcc
3 C_OPTS = -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -
4         Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-
5         dereference -I./src
6
7 all: clean prep compile run
8
9 clean:
10     rm -rf dist
11
12 prep:
13     mkdir dist
14
15 compile: task1main.bin task2main.bin task3main.bin task4main.bin
16         task5main.bin
17
18 task1main.bin:
19     gcc -g src/task1.c -o dist/task1main.bin
20
21 task2main.bin:
22     gcc -g src/task2.c -o dist/task2main.bin
23
24 task3main.bin:
25     gcc -g src/task3.c -o dist/task3main.bin
26
27 task4main.bin:
28     gcc -g src/task4.c -o dist/task4main.bin
29
30 task5main.bin:
31     gcc -g src/task5.c -o dist/task5main.bin
```

Рисунок 3 – Структура Makefile

**2.5** Проект було компільовано, використовуючи *Makefile* та запущено за допомогою *nemiver*. Програма виконалася успішно, результати завдання вказані на скриншоті (Рис.4).



```
task1main.bin (путь=«/home/humazer/programing-repo/lab04/dist/task1main.bin», pid=3244) - Nemiver
Файл  Правка  Вид  Отладка  Помощь
Продолжить  Запустить или перезапустить  Остановить

task1.c
1  int main()
2  {
3      int R = 4;
4      float PI = 3.14;
5      char command = 'v';
6      float result;
7      switch(command) {
8          case 'l': 2 * PI * R; // Длина окружности
9              break;
10         case 's': PI * R * R; // Площадь круга
11             break;
12         case 'v': (4.0 / 3.0) * PI * R * R * R; // Объем шара
13             break;
14         default: R;
15     }
16     return 0;
17 }
18
```

Строка: 16, Столбец: 1

ID по	Ка	Переменная	Значение	Тип
1	0	Локальные переменные		
		R	4	int
		PI	3.1400001	float
		command	118 'v'	char

Рисунок 4 – Відлагодження програми з *nemiver* та результати завдання.

**2.6** Зміни було зафіксовано та завантажено у *git*-репозиторії.

### 3. Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду розробки лінійних програм, роботи з операторами *if*, *switch*. Набув практичних навичок у програмуванні програм, які розгалужуються.

## Контрольні питання:

1. Як працює умовний оператор if?
2. Який вираз називається складеним логічним? Наведіть приклади.
3. Який оператор називають оператором множинного вибору? Наведіть приклад.
4. Як працює оператор switch?
5. Як працює тернарний оператор? Наведіть приклад.
6. Коли умовний оператор називається вкладеним?
7. Навіщо в операторі switch використовується оператор break?
8. Чи можуть бути вкладеними оператори switch?
9. Чи можна замість оператора if використовувати тернарний оператор і навпаки, - замість тернарного - оператор if?

## Відповіді:

1. Умовний оператор if реалізує виконання команд, якщо умова приймає значення "істина" (true).
2. Складене висловлювання А И В, утворене в результаті об'єднання двох простих висловлювань А і В логічною операцією И, істинно тоді, коли А і В одночасно істинні

Наприклад: Число 456 трехзначное и четное.

А: Число трехзначное, В: Число четное. Высказывания А и В соединены вместе логической операцией И, в результате получено составное высказывание А И В. Высказывание А і В истинно, поэтому А И В = 1.

3. Оператором множинного вибору називають switch.

Пример:

```
switch {  
case c1:  
    /*Тут находится код, который необходимо выполнить*/  
    break;
```

4. Оператор switch порівнює значення однієї змінної з декількома константами. Основний формат для використання оператора множинного вибору switch case.
5. Тернарний оператор (x == t) ? y : x, є альтернативним варіантом конструкції if...else, що дозволяє скоротити код, фактично записуя умову в одну строчку.
6. Коли після ключових слів then або else знову використовуються умовні оператори, вони називаються вкладеними. Число вкладень може бути довільно, при цьому діє правило: else завжди відноситься до найближчого оператору if.
7. Для закінчення роботи оператора switch. Ще можна примусове закінчити цикл, беручи до уваги стандартну перевірку умови.
8. Так, оператор switch может бути серед послідовності іншого оператора switch.
9. В деяких ситуаціях це можливо.