

Лабораторна робота №4 Розробка програм, що розгалужуються

1 Вимоги

1.1 Розробник

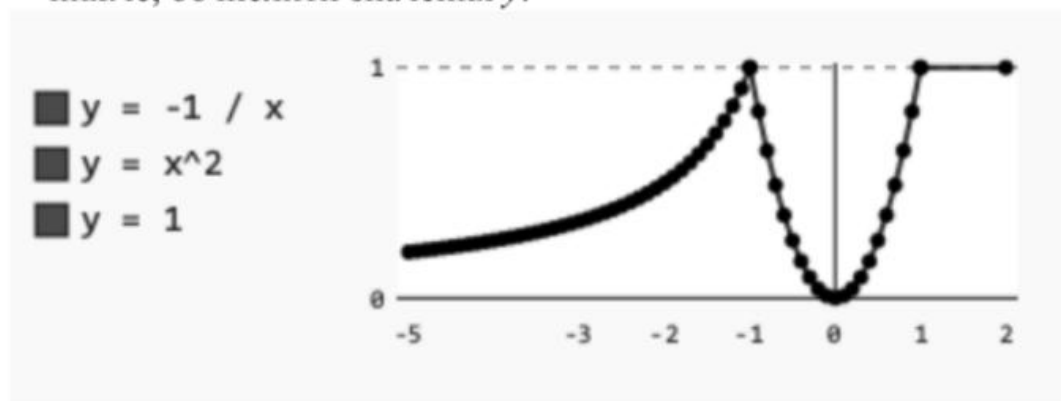
-Носов Микола
-студент групи кіт 1206
-26.10.2020

1.2 Загальне завдання:

Розробити програми.

1.3 Індивідуальне завдання:

1. За заданим радіусом r та командою ('l', 's' або 'v') користувача обчислити:
 - довжину окружності, якщо команда – 'l';
 - площу кола, якщо команда – 's';
 - об'єм кулі, якщо команда – 'v'.
2. Визначити, у скільки разів значення дробової частини числа більше за цілу. Організувати перевірку ділення на 0. Результат «обрізати» до другого знака після коми. Наприклад, $x = 123.656 \rightarrow y = 656/123 = 5.333333 = 5.330000$.
3. Дано три числа k, m, n . Змінити значення змінних таким чином, щоб виконувалась умова $k < m < n$.
4. Дано дійсне число x . Для функції $y(x)$, графік якої подано на рисунку нижче, обчислити значення y .



5. Дано дійсне число x . Для функції $y(x)$, графік якої поданий на рисунку

2 Опис програми

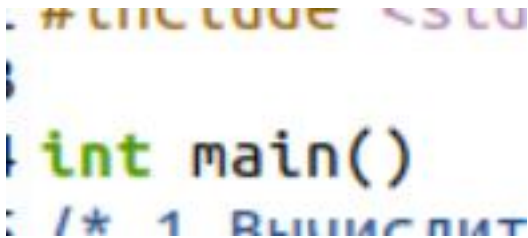
2.1 Функціональне призначення

1. Програма призначена для ;
 - пошуку площі кола, довжини кола, об'єму кулі за заданим радіусом.
 - того, щоб дізнатися у скільки разів дробова частина числа більше за його цілу частину.
 - для того, щоб змінити значення змінних таким чином, щоб виконувалась нерівність.
 - для того, щоб знайти значення у за рівнянням, маючи значення х.
 - того, щоб знайти значення у за рівнянням, маючи значення х.

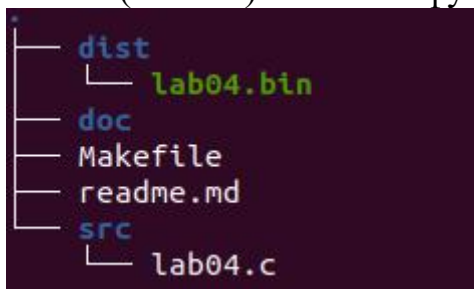
2.2 Опис логічної структури

Основна функція (див. Рисунок 2.1)

Структура проекту (див. Рисунок 2.2)



(Рис 2.1) - головна функція



(Рис 2.2) - tree

2.3 Важливі фрагменти коду

- початкові дані (константа π) та стандартна бібліотека (див. Рис. 2.3)
- функція `switch`. Використана для того, щоб програма розуміла, що їй потрібно обчислювати(об'єм, площу або довжину) (див. Рис. 2.4)
- округлення до 2 знаків після коми. Необхідно обрізати число за умовою задачі (див. Рис. 2.5)
- помилка при введенні одноразових значень у змінні. Адже при однакових значеннях не може виконуватись нерівність $k < m < n$ (див. Рис. 2.6)
- функція `if` (див. Рис. 2.7)

```
#define pi 3.14
#include <stdio.h>
```

(Рис 2.3) - початкові дані

```
switch (var)
{
case 's':
    result = pi*r*r;
    break;
case 'l':
    result = 2*pi*r;
    break;
case 'v':
    result = (4/3)*pi*r*r*r;
    break;
```

(Рис 2.4) - функція умови, 1 задача

```
d2 = d2 * 100;
d2 = (int) d2;
d2 = (float) d2 / 100;
```

(Рис 2.5) - округлення, 2 задача

```
if ( k == m || m == n || k == n )
{
char k [] = "Error: change variable value";
char m [] = "Error: change variable value";
char n [] = "Error: change variable value";
```

(Рис 2.6) - помилка вводу, 3 задача

```
if (x < -1)
{
y = -1/x;
} else if (x > -1 && x < 1)
{
y = x*x;
}
else
{
y = 1;
}
```

(Рис 2.7) - функція умови, 4 та 5 задачі

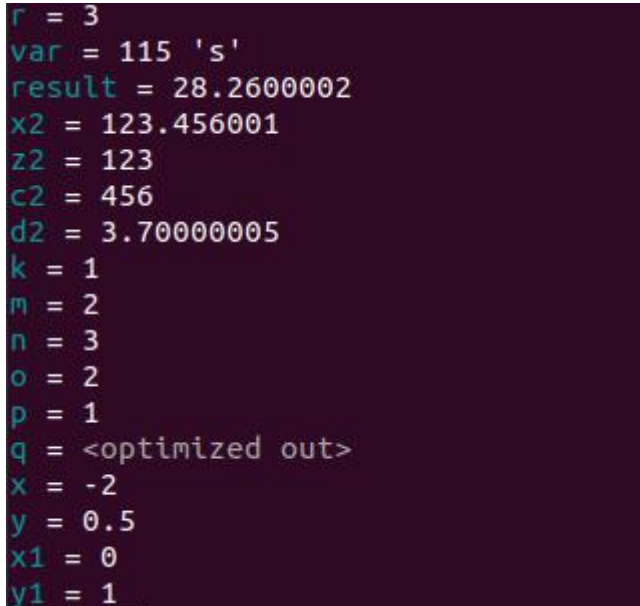
3.1 Обчислення результатів

Для обчислення результатів ми використаємо gdb - дебагер, влаштований в утиліті gcc.

Щоб подивитися на результат, ми повинні при компіляції вказати рівень інформації для відладки за допомогою -o, відкомпілювати код, запустити його бінарний файл у відлагоднику та вказати головну функцію, як місце, з якого починати відладку за допомогою команди “b main”, запустити програму та дивитись за результатом її роботи.

Щоб подивитись на значення змінних, використовуємо “info locals”
Значення початкових змінних у першому випадку (див. Рис. 3.1)

-r = 3 - радіус
-var = 's' - тип операції
-x2 = 123.456 - число для другої задачі
-k = 3
-m = 2
-n = 1 } числа для нерівності
-x = -2 - число для першого графіку
-x1 = 0 - - число для другого графіку



```
r = 3
var = 115 's'
result = 28.2600002
x2 = 123.456001
z2 = 123
c2 = 456
d2 = 3.70000005
k = 1
m = 2
n = 3
o = 2
p = 1
q = <optimized out>
x = -2
y = 0.5
x1 = 0
y1 = 1
```

(рис. 3.1) -значення змінних у gdb

Значення початкових змінних у другому випадку (див. Рис. 3.2)

-r = 3 - радіус
-var = 'l' - тип операції
-x2 = 123.456 - число для другої задачі

-k = 3
 -m = 2
 -n = 1

} числа для нерівності

-x = 2 - число для першого графіку

-x1 = -2 - - число для другого графіку

```

r = 3
var = 108 'l'
result = 18.8400002
x2 = 123.456001
z2 = 123
c2 = 456
d2 = 3.700000005
k = 1
m = 2
n = 3
o = 2
p = 1
q = <optimized out>
x = 2
y = 1
x1 = -2
y1 = 1
(gdb)

```

(Рис 3.2) - значення змінних у gdb

Значення початкових змінних у третьому випадку (див. Рис. 3.3)

-r = 3 - радіус
 -var = 'V' - тип операції
 -x2 = 123.456 - число для другої задачі
 -k = 3
 -m = 2
 -n = 1

} числа для нерівності

-x = 0.5 - число для першого графіку

-x1 = 0.5 - - число для другого графіку

```
gdb> print r
r = 3
gdb> print var
var = 118 'v'
gdb> print result
result = 84.7799988
gdb> print x2
x2 = 123.456001
gdb> print z2
z2 = 123
gdb> print c2
c2 = 456
gdb> print d2
d2 = 3.700000005
gdb> print k
k = 1
gdb> print m
m = 2
gdb> print n
n = 3
gdb> print o
o = 2
gdb> print p
p = 1
gdb> print q
q = <optimized out>
gdb> print x
x = 0.5
gdb> print y
y = 0.25
gdb> print x1
x1 = 0.5
gdb> print y1
y1 = 0.5
gdb>
```

(Рис 3.3) - значення змінних у gdb

Висновок :

У ході даної лабораторної роботи ми познайомились із функціями умови If та switch , навчилися їх використовувати, а також вирішили 5 лінійних задач.