

# CSE-016 Programming Lab Assignment № 2

16/03/2024

Youssef Ahmed Samy Kassem

ID 9545 – Group 3 – Lab 1

SSP – Faculty of Engineering, Alexandria University

*Document structure is detailed in the second page.*

***Solutions begin from the third page.***

## 1 Problems

### 1.1 Problem (1)

John is responsible for planting the street with trees; he can give you the length of the street in meters, the distance between each two trees in a meter, and the cost of planting each tree in dollars. Write a program that should read this information and then print the number of trees needed and the total cost.

### 1.2 Problem (2)

Write a program that calculates the squares, cubes, square root, and exponent ( $e^x$ ) of the numbers from 0 to 5 and uses tabs to print the following table of values:

Number	Square	Cube	Root	Exponent
0	0	0	0.0	1.0
..	..	..	..	..
..	..	..	..	..

# Contents

<b>1</b>	<b>Problems</b>	<b>1</b>
1.1	Problem (1) . . . . .	1
1.2	Problem (2) . . . . .	1
<b>2</b>	<b>Solutions</b>	<b>3</b>
2.1	Solution to Problem (1) . . . . .	3
2.1.1	Source Code . . . . .	3
2.1.2	Outcome . . . . .	4
2.2	Solution to Problem (2) . . . . .	5
2.2.1	Source Code . . . . .	5
2.2.2	Outcome . . . . .	5
2.3	Evidence of Work (Screenshots) . . . . .	6
2.3.1	Problem (1) Screenshots . . . . .	6
2.3.2	Problem (2) Screenshots . . . . .	8
2.4	Specifications . . . . .	10
<b>3</b>	<b>Licenses &amp; Warranty</b>	<b>10</b>

## 2 Solutions

### 2.1 Solution to Problem (1)

#### 2.1.1 Source Code

---

Program's main.c File – console input/output-oriented application to solve the problem

---

```
1  #include <math.h>
2  #include <stdio.h>
3
4  int main() {
5      // Runtime variable declarations
6      float streetlen, treedist, treecost, totalcost;
7      int treenum;
8
9      // Input prompts
10     printf("Hello, John! Let's help you fill the street with trees.\n\n");
11     printf("Please enter the length of the street (m): ");
12     scanf("%f", &streetlen);
13     printf("Now enter the distance you want between each tree (m): ");
14     scanf("%f", &treedist);
15     printf("Great! Finally, enter the cost of planting one tree ($): ");
16     scanf("%f", &treecost);
17     printf("\n"); // To separate input from results
18
19     // Calculations
20     treenum = 1 + floor(streetlen / treedist);
21     totalcost = (float) treecost * treenum;
22
23     // Outputting the results
24     printf("You will have to plant %d tree(s)\n", treenum);
25     printf("The total price you'll pay will be $%.2f\n", totalcost);
26
27     return 0;
28 }
```

---

## 2.1.2 Outcome

### Test Input Samples

#	Street Length	Distance Btn. Trees	Cost/ Tree
(1)	100	10	12
(2)	987	10	13.98

	#	Number of Trees	Total Cost
<b>Obtained Results</b>	(1)	11	132.00
	(2)	99	1384.02

The obtained results match the expected results.

### Console Output

---

Program's output to console in plaintext – using inputs from test sample (1)

---

Hello, John! Let's help you fill the street with trees.

Please enter the length of the street (m): 100

Now enter the distance you want between each tree (m): 10

Great! Finally, enter the cost of planting one tree (\$): 12

You will have to plant 11 tree(s)

The total price you'll pay will be \$132.00

---

*Turn over the page for the solution to problem (2)*

## 2.2 Solution to Problem (2)

### 2.2.1 Source Code

Program's main.c File – console application to output the full table as described in the problem

```
1 #include <math.h>
2 #include <stdio.h>
3
4 int main() {
5     printf("Number\tSquare\tCube\tRoot\tExponent\n");
6     printf("0\t%d\t%d\t%.1f\t%.1f\n", ((int)pow(0,2)), ((int)pow(0,3)), sqrt(0), exp(0));
7     printf("1\t%d\t%d\t%.1f\t%.1f\n", ((int)pow(1,2)), ((int)pow(1,3)), sqrt(1), exp(1));
8     printf("2\t%d\t%d\t%.1f\t%.1f\n", ((int)pow(2,2)), ((int)pow(2,3)), sqrt(2), exp(2));
9     printf("3\t%d\t%d\t%.1f\t%.1f\n", ((int)pow(3,2)), ((int)pow(3,3)), sqrt(3), exp(3));
10    printf("4\t%d\t%d\t%.1f\t%.1f\n", ((int)pow(4,2)), ((int)pow(4,3)), sqrt(4), exp(4));
11    printf("5\t%d\t%d\t%.1f\t%.1f\n", ((int)pow(5,2)), ((int)pow(5,3)), sqrt(5), exp(5));
12    return 0;
13 }
```

Please note that the line numbers are purely for readability purposes and are not part of the code.

Please also note that on lines where there is no new line number, there is **NO** actual line break in the code and that this is meant to be read as one single line.

### 2.2.2 Outcome

#### Console Output

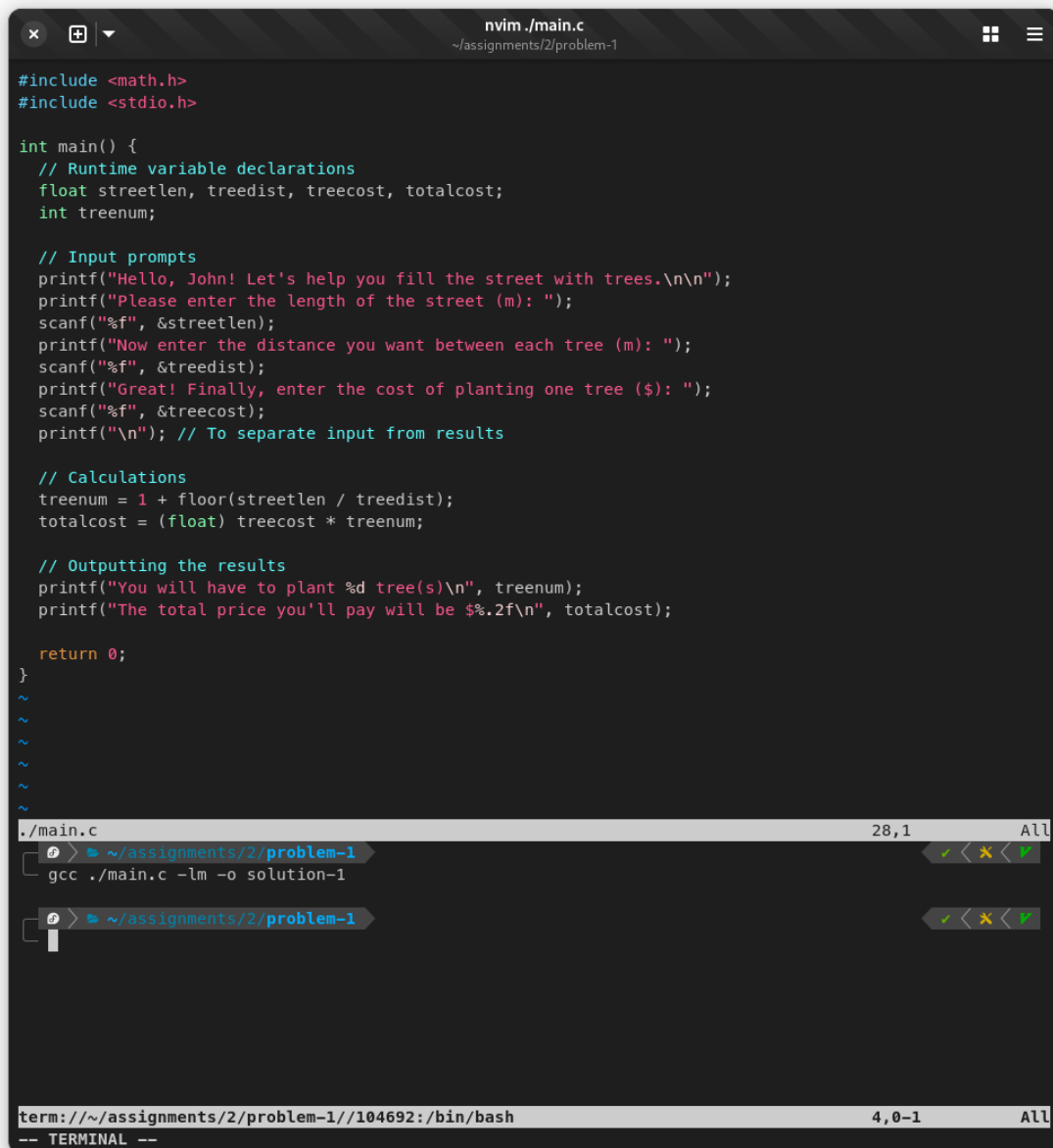
Program's output to console in plaintext

Number	Square	Cube	Root	Exponent
0	0	0	0.0	1.0
1	1	1	1.0	2.7
2	4	8	1.4	7.4
3	9	27	1.7	20.1
4	16	64	2.0	54.6
5	25	125	2.2	148.4

## 2.3 Evidence of Work (Screenshots)

### 2.3.1 Problem (1) Screenshots

#### Close Up Source-code Screenshot



The screenshot shows a code editor window titled "nvim ./main.c" with the file path "~/assignments/2/problem-1". The code is a C program that calculates the number of trees to plant and the total cost based on user input. The code is as follows:

```
#include <math.h>
#include <stdio.h>

int main() {
    // Runtime variable declarations
    float streetlen, treedist, treecost, totalcost;
    int treenum;

    // Input prompts
    printf("Hello, John! Let's help you fill the street with trees.\n\n");
    printf("Please enter the length of the street (m): ");
    scanf("%f", &streetlen);
    printf("Now enter the distance you want between each tree (m): ");
    scanf("%f", &treedist);
    printf("Great! Finally, enter the cost of planting one tree ($): ");
    scanf("%f", &treecost);
    printf("\n"); // To separate input from results

    // Calculations
    treenum = 1 + floor(streetlen / treedist);
    totalcost = (float) treecost * treenum;

    // Outputting the results
    printf("You will have to plant %d tree(s)\n", treenum);
    printf("The total price you'll pay will be $%.2f\n", totalcost);

    return 0;
}
~
~
~
~
~
```

Below the code editor, there is a terminal window showing the compilation command:

```
~/main.c 28,1 All
> ~/assignments/2/problem-1
gcc ./main.c -lm -o solution-1

> ~/assignments/2/problem-1
```

The terminal window title is "term://~/assignments/2/problem-1/104692:/bin/bash" with a cursor at "4,0-1 All".

## Close Up Output Console Screenshot

```
youssef@silverblue:~/assignments/2/problem-1
~/assignments/2/problem-1

youssef@silverblue:~/assignments/2/problem-1$ ./solution-1
Hello, John! Let's help you fill the street with trees.

Please enter the length of the street (m): 100
Now enter the distance you want between each tree (m): 10
Great! Finally, enter the cost of planting one tree ($): 12

You will have to plant 11 tree(s)
The total price you'll pay will be $132.00
youssef@silverblue:~/assignments/2/problem-1$
```

## Linux Desktop Screenshot

The screenshot shows a Linux desktop environment with three windows open. The top-left window is a code editor showing the C program `main.c`. The top-right window is a terminal showing the execution of the program. The bottom window is a PDF viewer displaying a document from Alexandria University.

**Code Editor (main.c):**

```
#include <math.h>
#include <stdio.h>

int main() {
    // Runtime variable declarations
    float streetlen, treedist, treecost, totalcost;
    int treenum;

    // Input prompts
    printf("Hello, John! Let's help you fill the street with trees.\n\n");
    printf("Please enter the length of the street (m): ");
    scanf("%f", &streetlen);
    printf("Now enter the distance you want between each tree (m): ");
    scanf("%f", &treedist);
    printf("Great! Finally, enter the cost of planting one tree ($): ");
    scanf("%f", &treecost);
    printf("\n"); // To separate input from results

    // Calculations
    treenum = 1 + floor(streetlen / treedist);
    totalcost = (float) treecost * treenum;

    // Outputting the results
    printf("You will have to plant %d tree(s)\n", treenum);
    printf("The total price you'll pay will be %.2f\n", totalcost);

    return 0;
}
```

**Terminal:**

```
youssef@silverblue:~/assignments/2/problem-1$ ./solution-1
Hello, John! Let's help you fill the street with trees.

Please enter the length of the street (m): 100
Now enter the distance you want between each tree (m): 10
Great! Finally, enter the cost of planting one tree ($): 12

You will have to plant 11 tree(s)
The total price you'll pay will be $132.00
youssef@silverblue:~/assignments/2/problem-1$
```

**PDF Viewer (Lab 2 - Spring 2023.pdf):**

Alexandria University  
Faculty of Engineering  
Specialized Scientific Programs

Computer and Communication  
CSE016 Computers & Programming

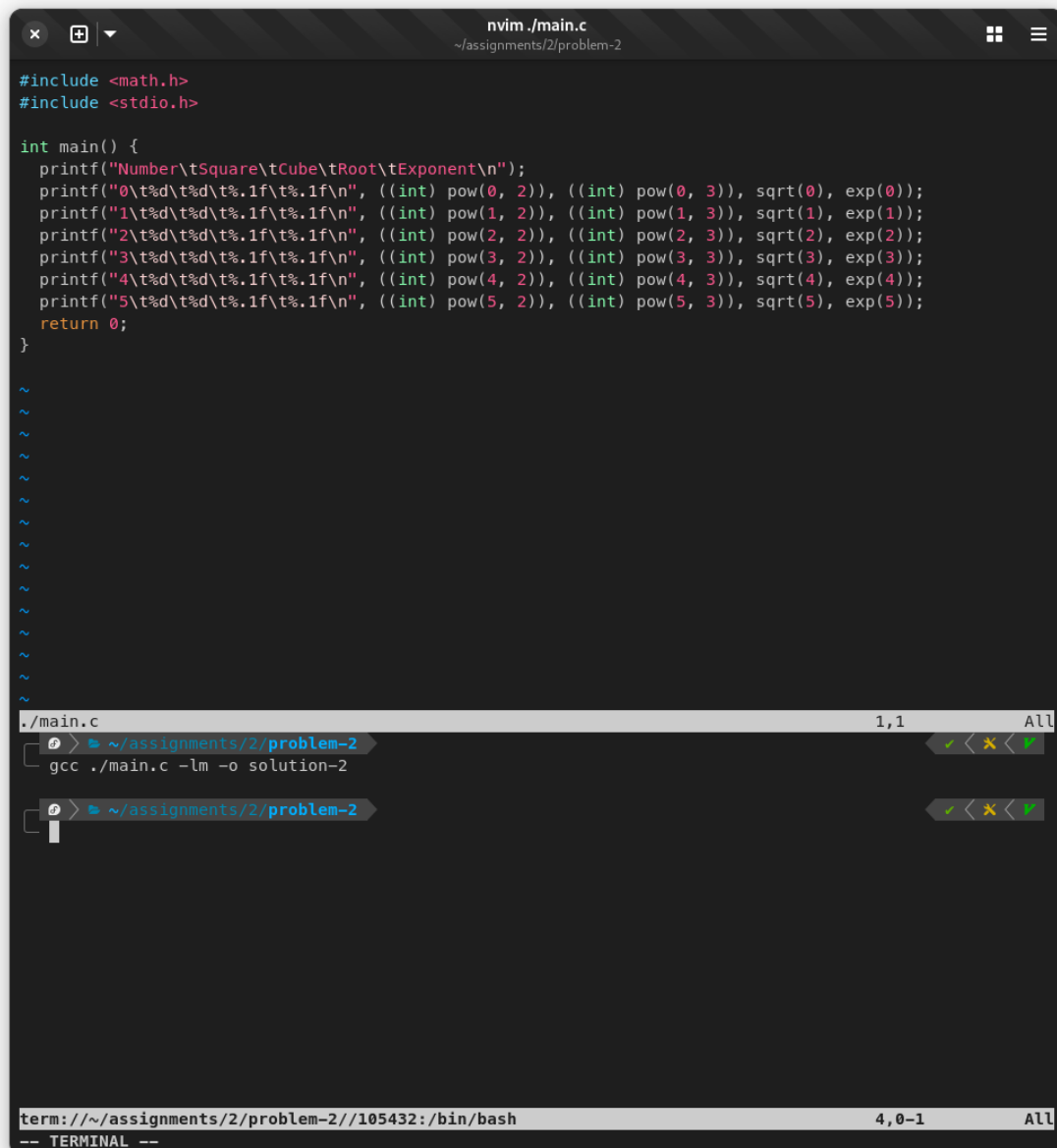
**Take home Assignments:**

- John is responsible for planting the street with trees; he can give you the **length** of the street in meters, the **distance** between each two trees in a meter, and the **cost** of planting each tree in dollars. Write a program that should read this information and then print the **number** of trees needed and the **total cost**.
- Write a program that calculates the **squares**, **cubes**, **square root**, and **exponent** ( $e^x$ ) of the numbers from 0 to 5 and uses tabs to print the following table of values:

Number	Square	Cube	Root	Exponent
0	0	0	0.0	1.0
..	..	..	..	..
..	..	..	..	..

## 2.3.2 Problem (2) Screenshots

### Close Up Source-code Screenshot



The screenshot shows the nvim editor interface. The main window displays the source code for `./main.c`, which includes `math.h` and `stdio.h`. The `main` function prints a table of mathematical values for exponents 0 through 5. Below the code, there are two terminal windows. The top terminal window shows the command `gcc ./main.c -lm -o solution-2` being executed. The bottom terminal window shows the prompt `term://~/assignments/2/problem-2/105432:/bin/bash`.

```
#include <math.h>
#include <stdio.h>

int main() {
    printf("Number\tSquare\tCube\tRoot\tExponent\n");
    printf("0\t%d\t%d\t%.1f\t%.1f\n", ((int) pow(0, 2)), ((int) pow(0, 3)), sqrt(0), exp(0));
    printf("1\t%d\t%d\t%.1f\t%.1f\n", ((int) pow(1, 2)), ((int) pow(1, 3)), sqrt(1), exp(1));
    printf("2\t%d\t%d\t%.1f\t%.1f\n", ((int) pow(2, 2)), ((int) pow(2, 3)), sqrt(2), exp(2));
    printf("3\t%d\t%d\t%.1f\t%.1f\n", ((int) pow(3, 2)), ((int) pow(3, 3)), sqrt(3), exp(3));
    printf("4\t%d\t%d\t%.1f\t%.1f\n", ((int) pow(4, 2)), ((int) pow(4, 3)), sqrt(4), exp(4));
    printf("5\t%d\t%d\t%.1f\t%.1f\n", ((int) pow(5, 2)), ((int) pow(5, 3)), sqrt(5), exp(5));
    return 0;
}
```

```
~/main.c 1,1 All
> ~/assignments/2/problem-2
gcc ./main.c -lm -o solution-2

> ~/assignments/2/problem-2

term://~/assignments/2/problem-2/105432:/bin/bash 4,0-1 All
-- TERMINAL --
```



## Close Up Output Console Screenshot

```
youssef@silverblue:~/assignments/2/problem-2
~/assignments/2/problem-2

youssef@silverblue:~/assignments/2/problem-2$ ./solution-2
Number Square Cube Root Exponent
0 0 0 0.0 1.0
1 1 1 1.0 2.7
2 4 8 1.4 7.4
3 9 27 1.7 20.1
4 16 64 2.0 54.6
5 25 125 2.2 148.4
youssef@silverblue:~/assignments/2/problem-2$
```

## Linux Desktop Screenshot

The screenshot shows a Linux desktop environment with three windows open:

- Code Editor (nvim):** Displays the C program `main.c` which calculates and prints the square, cube, square root, and exponent for numbers 0 to 5. The code uses `pow` and `sqrt` functions from `math.h`.
- Terminal:** Shows the command `gcc ./main.c -lm -o solution-2` being executed successfully, and the command `./solution-2` being run, which produces the same output table as the close-up screenshot.
- PDF Viewer:** Displays a PDF document titled "Lab 2 - Spring 2023.pdf" from Alexandria University. It contains "Take home Assignments" with two tasks: one about planting trees and another about calculating mathematical values for numbers 0 to 5.

The PDF document also includes a table of values for numbers 0 to 5, matching the output of the program:

Number	Square	Cube	Root	Exponent
0	0	0	0.0	1.0
1	1	1	1.0	2.7
2	4	8	1.4	7.4
3	9	27	1.7	20.1
4	16	64	2.0	54.6
5	25	125	2.2	148.4

## 2.4 Specifications

- **Libraries:**
  - `stdio.h`
  - `math.h`
- **Compiler:** GNU C Compiler (gcc) version 13.2.1 20231205 (Red Hat 13.2.1-6)
- **Supported Platforms:** OS: (any), architecture: (any)
- **Tested On:**
  - Compiled on Fedora 39 Workstation Linux
  - Ran on Fedora 39 Silverblue Workstation Linux

## 3 Licenses & Warranty

The source code will be released past the assignment's deadline under my public git repository at <https://github.com/y-samy/9545-assignments/tree/main/CSE-016/Lab-2/assignment-2>.

This document, my additions to its L<sup>A</sup>T<sub>E</sub>X source code, the software included and its C source code are all subject to the BSD 3-Clause Open Source License:

<https://opensource.org/license/bsd-3-clause>.

COPYRIGHT © 2024, Youssef Ahmed Samy

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.