

CL1002 – Programming Fundamentals Lab

Assignment # 02

Note:

- First think about problem statement and then write/draw your logic on paper.
- After designing the logic on paper, code the problem statement on any editor (VS Code, Gedit, etc).
- Copied tasks will be awarded **zero** marks without any investigation.
- Comments you code properly.
- Note that these assignment marks could be graded through a viva (quiz) in the lab.
- Submit a pdf file containing all of your C code with all possible screenshots of every task output on Google Classroom.
- Please submit your file with this naming convention (ROLLNO-NAME) i.e (22P-8743-Zain.pdf)

Problem: 1 | Cricket Match

Suppose there is a cricket match between Pakistan and India. India played first and gave a target of 300 runs to Pakistan. Now write a program that obtains a cricket match score and remaining wickets from user. If the score is greater than 300 outputs a message indicating “Pakistan win the match by X wickets” while X are remaining wickets. Otherwise shows a message how many remaining runs required to win with X wickets in hand. Use single if-else statement. If score is 300 with 0 wickets in hand then display match is draw.

Note: Score and wickets are not less than zero

Problem: 2 | (if with logical operators)

A program uses a char variable named membership and an int variable named age. The membership variable contains one of the following letters (entered in either uppercase or lowercase): M or N. The letter M stands for member, and the letter N stands for non- member. The program should display the appropriate seminar fee, which is based on a person’s membership status and age. The fee schedule is shown in Figure. Write the C code to display the fee.

\$10	Club member less than 65 years old
\$5	Club member atleast 65 years old

\$20	Non Member
------	------------

Problem: 3 | Quadratic Equation

Write a program to find out the roots of quadratic equation ($ax^2 + bx + c = 0$).

The user enter values of a, b and c only. The values entered must be greater than 0, if any of the value is 0 then print “The value can’t be zero”. If any value entered is less than zero then print “Values must be greater than Zero”.

The roots are calculated as:

i) If disc. is greater than zero then roots are real and unequal.

$$\text{root1} = \frac{-b + (\text{disc})^{1/2}}{2*a}$$

$$\text{root2} = \frac{-b - (\text{disc})^{1/2}}{2*a}$$

ii) If disc. is less than zero then roots are imaginary.

$$\text{root1} = \frac{-b + i(\text{disc})^{1/2}}{2*a}$$

$$\text{root2} = \frac{-b - i(\text{disc})^{1/2}}{2*a}$$

iii) If disc. is equal to zero then roots are real.

$$\text{root1} = \text{root2} = -b/(2*a)$$

Note:

- “disc” mean Discriminator having value $\text{disc} = b^2 - 4*a*c$.
- Use `<math.h>` library in your code and use ‘sqrt’ for square root.
- The value of iota is constant i.e. $i = -1$.

Problem: 4 | Largest palindrome product

A palindrome number is a number that remains the same when digits are reversed. For example, the number 12321 is a palindrome number, but 1451 is not a palindrome number.

The largest palindrome made from the product of two 2-digit numbers is $9009 = 91 \times 99$.

Write an efficient program that Find the largest palindrome made from the product of two 3-digit numbers.

Problem: 5 | Highly divisible triangular number

The sequence of triangle numbers is generated by adding the natural numbers. So the 7th triangle number

would be $1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$.

Similarly,

The 3rd triangular number would be $1 + 2 + 3 = 6$

The 4th triangular number would be $1 + 2 + 3 + 4 = 10$

So, the first ten terms would be:

1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...

Let us list the factors of the first seven triangle numbers:

1: 1

3: 1,3

6: 1,2,3,6

10: 1,2,5,10

15: 1,3,5,15

21: 1,3,7,21

28: 1,2,4,7,14,28

We can see that 28 is the first triangle number to have over five divisors.

Write a program that will display the first five triangular numbers to have over 10 divisors.

Note:

Display the triangular number with their all divisors as shown above.

Problem: 6 | Random Walk

In this problem, the program starts up assuming that it is following an object on a 2- dimensional plane as it takes many steps to perform a sort-of “random walk”. The object starts off at the “origin” ($x = 0$ and $y = 0$). You will input two real numbers that will represent the next coordinates of the object. The computer will calculate the straight-line distance that it has to travel to get there and position the object at that point. Then you will input two more real numbers representing a move from that location, then another position (another pair of points) and another position (pair of points) and so on. To stop the program, you have to input the identical new coordinates as the previous step—that is, there is no distance to be traveled for that final step. This is a “sentinel” indicating that the loop should

terminate. This last zero-step should not be counted in the number of steps. When the program exits the loop at this point make the program print out the following points.

1. Report the total distance travelled.
2. Report the total number of steps.
3. Report the average distance traveled per step. (Make sure that the case of zero steps and zero distance traveled does not cause problems.)

Use the following formula to calculate the distance of the step.

$$s = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}.$$

Here,

x_i, y_i represent (New coordinates),

x_0, y_0 represent (Current coordinates) and

s is the Distance of a single step

Problem: 7 | Pattern

Write programs to display each of the following patterns.

<p>(i)</p> <pre> 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 </pre>	<p>(ii)</p> <pre> 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 </pre>	<p>(iii)</p> <pre> A A B A B C A B C D A B C D E </pre>	<p>(iv)</p> <pre> 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 1 2 3 4 5 4 3 2 1 </pre>
<p>(v)</p> <pre> A B C D E F A B C D E A B C D A B C A B A </pre>	<p>(vi)</p> <pre> * * * * * * * * * * * * * * * * </pre>	<p>(vii)</p> <pre> * * * * * * * * * * * * * * * * </pre>	<p>(viii)</p> <pre> * * * * * * * * * * * * * * * * * * </pre>

Good Luck