# Programming Fundamental

Programming Day - Week 12

## Introduction

Welcome to your favorite day of the week which is programming day🎉. In this lab manual, we shall work together to learn and implement new programming concepts.

## Let's do some coding.

By this week, you have learned how to write a program that contains functions, loops, arrays, conditional structures and stores the data on the file.

## Let's put your skills to the test ‼ Make us proud ❤️

**Task 01(CP):** Complete your business application that includes

1. Single responsibility functions for performing different operations
2. Proper naming convention for variable, array, and function names
3. Global arrays that store the data of multiple items/users/objects
4. Stores the data on the file in CSV format
5. Loads the data back into the arrays when you restart the program
6. CRUD (Create, Retrieve, Update, Delete) Operations and Other respective features of your project
7. Validation functions

Consider this sample code from the last class.
https://github.com/MuhammadIrzam447/Programming-Fundamentals-Lab-Manuals/tree/master/SignInSignUp%20v2

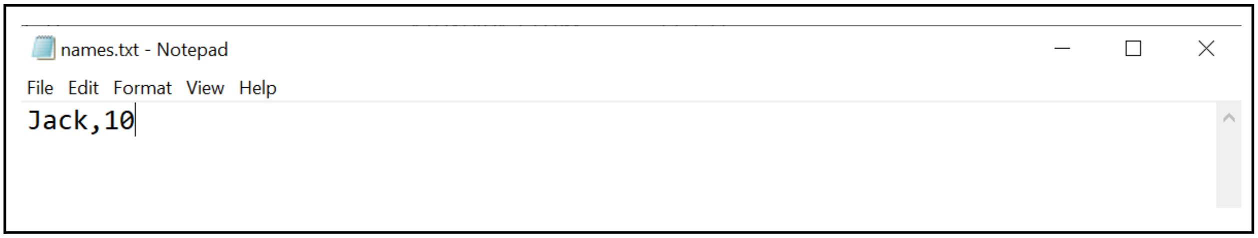
**Task 02(CP):** Complete your game application that includes

1. A Single Player and 3 different Enemies
2. Smart enemy that follows the main player (*check the github code attached as sample code* https://github.com/MaidaShahid/PacmanCpp)
3. Players Movement
4. Shooting/Firing from Players and Enemies
5. Collision Detection with maze/enemies/bullets
6. Shows user/enemy health and score

Students, you are in luck as we have **bonus tasks** for this programming day.
Attempt these to practice your programming and problem solving skills.

## Task 03(CP):

Create a function which constructs a rectangular birthday cake, based on someone's name and age! Name and age is given in a text file named **names.txt** in the following format.



```
Jack,10
```

Read the file to get the name and age of the person and make sure to surround the birthday message with the character that fits the rule:

1. If the age is an even number, surround the message with "#".
2. If the age is an odd number, surround the message with "*".

Store the output in another file.

**Other important rules:**

1. The output message should be in the format:

{age} HB {name}! {age}

2. Leave a space between the edge of the cake and the age numbers.
3. The number of characters (# or *) in the banner should be 15

**Test Cases:**

| Input | Output |
|---|---|
| getBirthdayCake("Jack", 10) | "###############"<br>"# 10 HB Jack! 10 #"<br>"###############" |
| getBirthdayCake("John", 19) | "***************"<br>"* 19 HB John! 19 *"<br>"***************" |
| getBirthdayCake("Mary", 20) | "###############"<br>"# 20 HB Mary! 20 #"<br>"###############" |

## Task 04(CP):

Create a function that takes a string str containing only letters from a to z from a file named **alphabets.tx**t in lowercase and appends the missing letter(s) in alphabetical order a-z in the next line of the file.

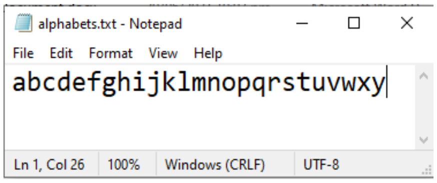
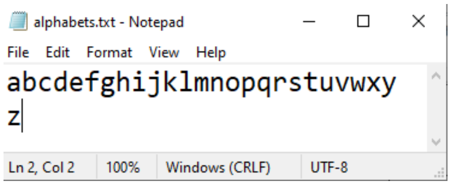
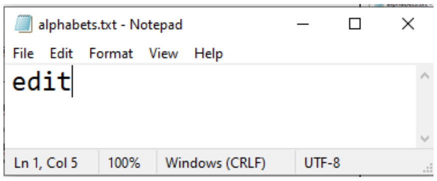
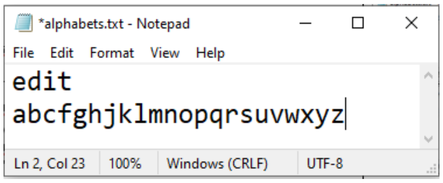
A set of letters is given by

abcdefghijklmnopqrstuvwxyz

**Notes:** If the string contains all letters from a-z, then an empty string will be appended.

**Hint:** You can convert int to char by writing: char(97) ➞ a

**Test Cases**

| Input | Output |
|---|---|
| <br> alphabets.txt - Notepad<br>File Edit Format View Help<br>Ln 1, Col 26 100% Windows (CRLF) UTF-8<br>abcdefghijklmnopqrstuvwxy | <br> alphabets.txt - Notepad<br>File Edit Format View Help<br>Ln 2, Col 2 100% Windows (CRLF) UTF-8<br>abcdefghijklmnopqrstuvwxy<br>z |
| <br> alphabets.txt - Notepad<br>File Edit Format View Help<br>Ln 1, Col 5 100% Windows (CRLF) UTF-8<br>edit | <br> *alphabets.txt - Notepad<br>File Edit Format View Help<br>Ln 2, Col 23 100% Windows (CRLF) UTF-8<br>edit<br>abcfghjklmnopqrsuvwxyz |

## Task 05(CP):

Suppose you are working for Secret Services. You have to write a function that takes a string and convert that into morseCode. The function outputs an encrypted letter string with Morse code.

**Note:**

- There will be a space in the output string after each letter’s morse code.

| Alphabets | MorseCode |
|---|---|
| A | .- |
| B | -... |
| C | -.-. |
| D | -.. |

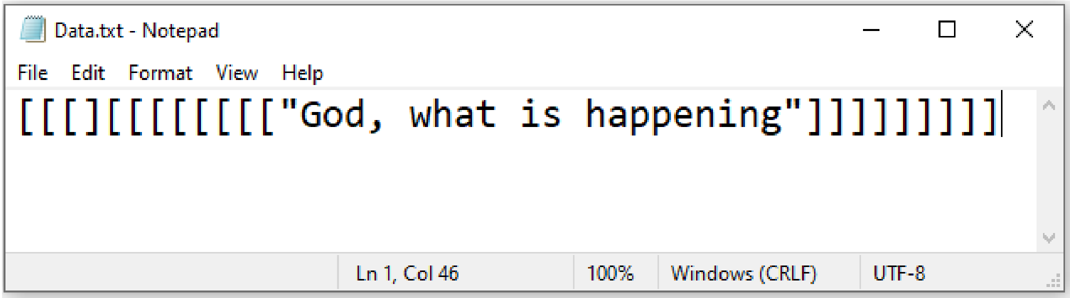
| | |
|---|---|
| E | . |
| F | ..-. |
| G | --. |
| H | .... |
| I | .. |
| J | .--- |
| K | -.- |
| L | .-.. |
| M | -- |
| N | -. |
| O | --- |
| P | .--. |
| Q | --.- |
| R | .-. |
| S | ... |
| T | - |
| U | ..- |
| V | ...- |
| W | .-- |
| X | -..- |
| Y | -.-- |
| Z | --.. |
| (Space) | -.-.-.- |

**Test Cases**

| Input | Output |
|---|---|
| morse("F Mueller") | "..-. ..... -- ..- . .-.. .-.. . .-." |
| morse("Barack Obama") | "-... .- .-. .- -.-. -.- ..... --- -... .- -- .-" |
| morse("JOHN F KENNEDY") | ".--- --- .... -. ..... ..-. ..... -.- . -. -. . -.. -.--" |

**Task 06(CP):**

John has made a mistake and overwrote a text file with some weird data, thankfully he had no back-up and he’s too lazy to fix it so now you guys can solve it for him!
Get the text strings out of those files.

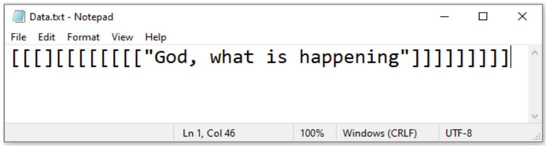


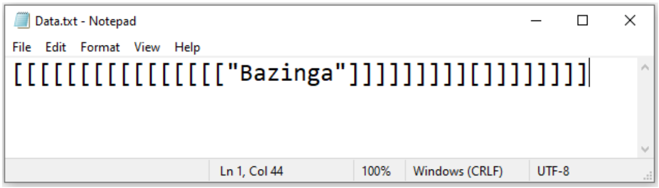
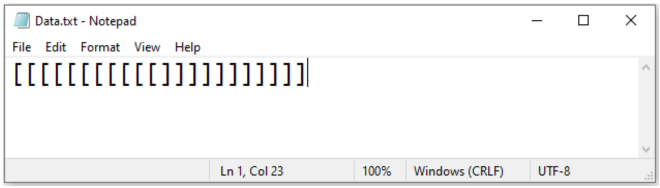
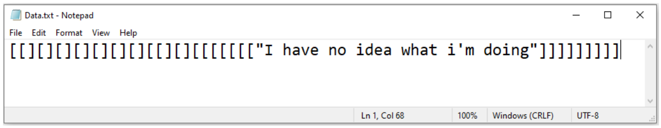
```
[[[][[[[[[[["God, what is happening"]]]]]]]]]
```

**Note:** if there is no string, you should return a pre-defined string (see Test Case 3).

Your task is to make a function that reads the data.txt file and then returns only the text string from the data.

**Test Cases**

| Input | Output |
|---|---|
| <br>[[[][[[[[[[["God, what is happening"]]]]]]]]] | "God, what is happening" |

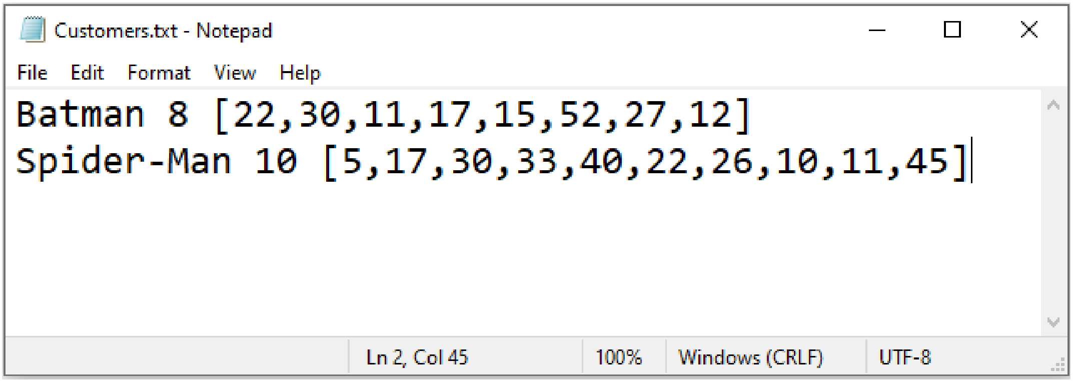
| | |
|---|---|
| <br>`[[[[[[[[[[[[[[[["Bazinga"]]]]]]]]][]]]]]]]` | "Bazinga" |
| <br>`[[[[[[[[[[[]]]]]]]]]]]` | "What... why did you make this?" |
| <br>`[[][][][][][][][[][][[[[[[[["I have no idea what i'm doing"]]]]]]]]]` | "I have no idea what i'm doing" |

**Task 07(CP):**

KamyabLife is launching a network of autonomous pizza delivery drones and wants you to create a flexible rewards system (Pizza Points) that can be tweaked in the future. The rules are simple:

if a customer has made at least N orders of at least Y price, they get a FREE pizza!

The information of the customers is stored in a file Customers.txt in the following format. First the name of the customer is given then after the space the number of orders are given then after the space within the brackets all the orders prices are given.



```
Batman 8 [22,30,11,17,15,52,27,12]
Spider-Man 10 [5,17,30,33,40,22,26,10,11,45]
```

Your task is to create a function that takes a minimum number of orders and a minimum order price then cout the names of the customers that are eligible for a free pizza.

**Test Cases**

| Input | Output |
|---|---|
| pizza_points(5, 20) | "Spider-Man" |
| pizza_points(3, 10) | "Batman"<br>"Spider-Man" |
| pizza_points(5, 100) | “” |

**Good Luck and Best Wishes ‼**

**Happy Coding ahead :)**