

I confirm that this assignment is my own work.

Where I have referred to academic sources, I have provided in-text citations and included the sources in the final reference list.

Programming assignment

Part 2

Bora Sozer  
P438457

İçindekiler

[Section 1: Analysis 3](#_Toc164005559)

[Key Challanges 3](#_Toc164005560)

[Success Criteria 7](#_Toc164005561)

[Consideration of user experience 7](#_Toc164005562)

[Scope and limitations of the program 8](#_Toc164005563)

[Section 2: Design 9](#_Toc164005564)

[Main 9](#_Toc164005565)

[Functions 10](#_Toc164005566)

[Section 3: Technical Overview 19](#_Toc164005567)

[Section 4: Coded solution 24](#_Toc164005568)

[Example users CSV File (a file named “users.csv” needs to be created with this information) 24](#_Toc164005569)

[Words CSV file (a file named words.csv needs to be created with this information) 24](#_Toc164005570)

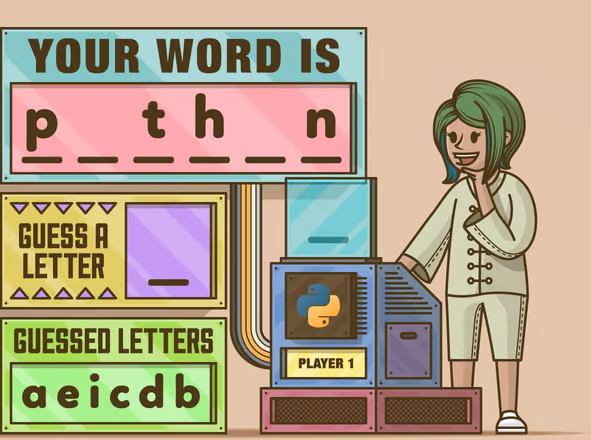
[Code 29](#_Toc164005571)

[Section 5: Testing 40](#_Toc164005572)

[Section 6: Evaluation 45](#_Toc164005573)

[Section 7: References 47](#_Toc164005574)

# Section 1: Analysis

The aim of this python programme is to create an application to help students learn vocabulary. In the application, there are 3 different games (Word Guess, Word Match and Letter Hunt), 5 different categories in each game (Maths, Science, History, Art, and Computer Science), 3 different difficulty levels in each category that the user can choose according to his/her level, and 3 levels to be played sequentially within each difficulty level.

Each level has 5 words and definitions. The user needs to guess these 5 words correctly and level up in order to progress in the game. When the user fails to complete 5 words or exits without completing them, progress does not record.

The application has a user login and stores the user's progress. In order to ensure friendly competition among students, lists ranked according to user score can be seen both within the whole application and on the basis of game, subject and difficulty level.

Apart from Python code, there are two different CSV files that store user data and words.

## Key Challanges

**Challenge 1:**

I used dictionary as a data structure before and I thought I had solved it completely. However, when I was doing this programme, it was difficult to handle the data with the dictionary structure at first. Yes, I understood the dictionary structure, but I had never done this with such complex data. Both writing this data to csv and reading it back from csv and using it in the code was quite challenging and confusing.

**Solution 1:**

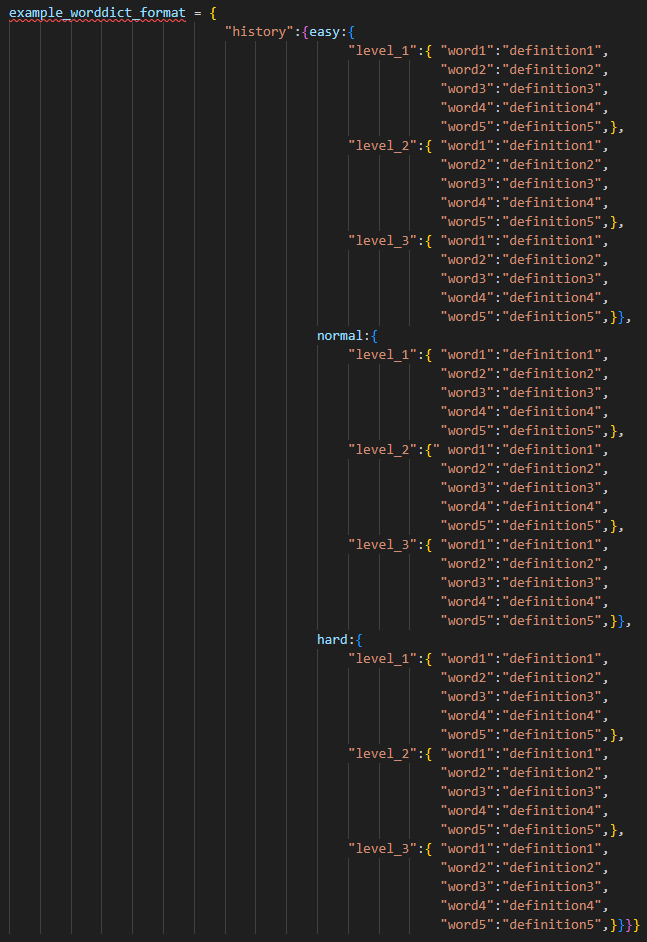
Before the read and write functions, I wanted to determine how to store the data in the dictionary by making a sample dictionary.



As can be seen, it contains a deep and complex data structure with many dictionaries. Storing the data in this format allowed the data to be accessed easily without using a for loop or another variable each time the program needs to access the user's specific data while user playing the game.

Using only the game, subject and difficulty preference previously selected by the user in the categories, it was enabled to access specific level info with the following line.

users[username]["progress"][game\_choice][subject\_choice][difficulty\_choice]

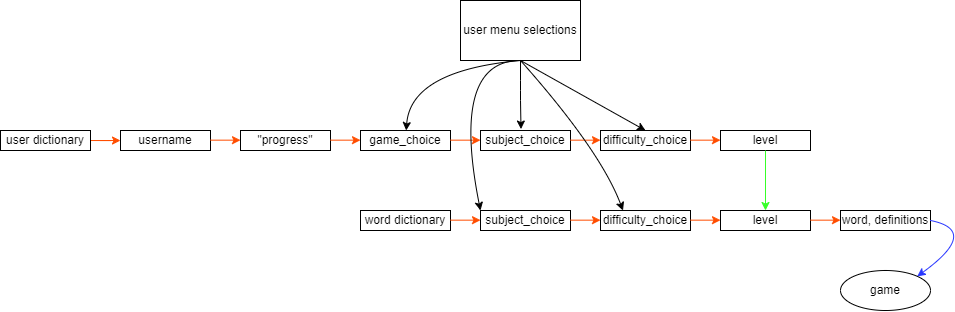
In the same way, I created the sample form of the word dictionary. 

Using the line below, I accessed the words and definitions of the level with for loop and used them in the game.

for word,definition in worddict[subject\_choice][difficulty\_choice][level].items():

I then practised many times in different ways and got used to using these deep and complex lexical structures.

I finally managed to handle the data comfortably. The associated data in the word dictionary was accessed according to the user data as follow.



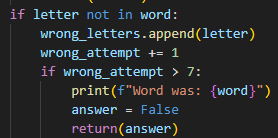
**Challenge 2:**

I had to determine how to give score to the user and how the game progressed.

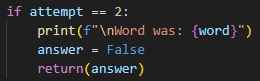
**Solution 2:**

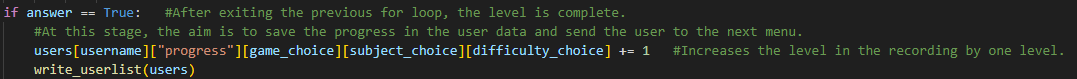
For this, I assigned a total of 3 levels to each difficulty, 1,2 and 3. I decided that there should be 5 words in each level. If the user completes the level below a certain number of mistakes (different in each game), I decided to complete the level and move to the next level and get points.

For example, in the letter hunt game, I defined a wrong\_attempt variable for letter guessing, increased this variable for each wrong letter guess, and when it reached a certain number, I warned the user and made the game exit the level.



Similarly, in the word match game, the user is warned after the second wrong guess and asked if they want to try the same level again.



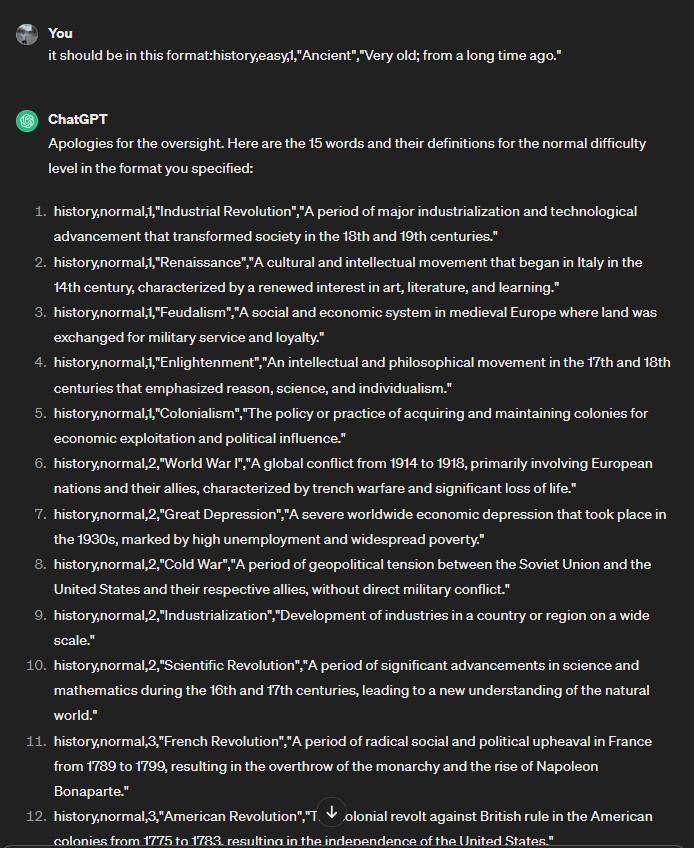
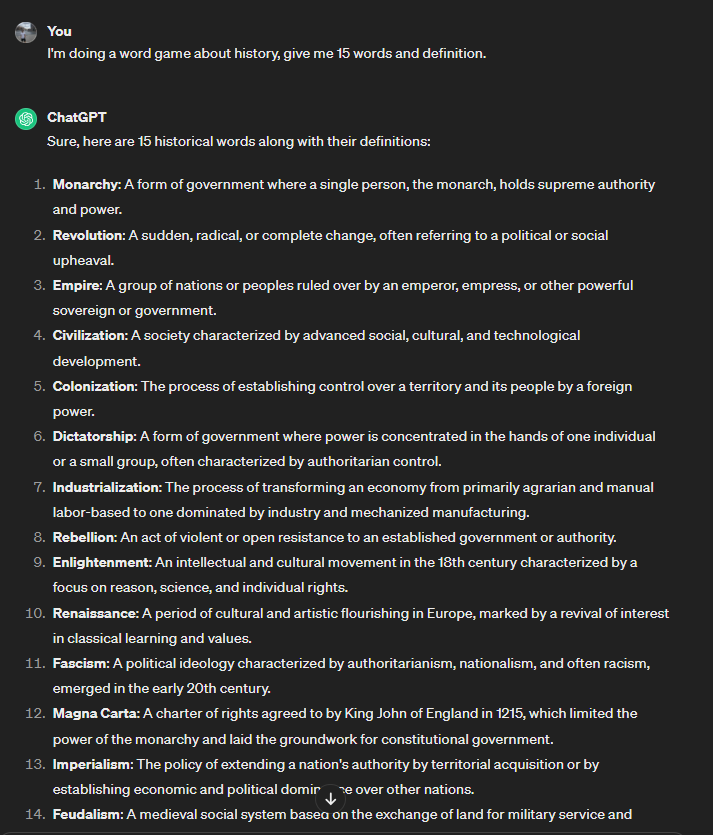
If all 5 words are correct, the user's level is increased. 

If the user leaves without completing 5 words, level progress is not provided. Therefore, it is important to complete 5 words in each level for progression.

**Challenge 3:**

It was quite difficult to find so many words and definitions on so many different subjects. It would take a lot of time to find the words and their meanings one by one and then write them in the format I wanted.

**Solution 3:** However, as Simon said in the lesson, getting help from ChatGPT for this job was very helpful. With proper explanations and the right prompts, I got the relevant words and then printed these words in the format I wanted. As can be seen in the pictures and references below. (ChatGPT, 2024)



## Success Criteria

Each game has a different success criteria.

In Game 1 - Word guess the user is given 6 wrong guesses. If the 6 guesses expire and the user does not guess correctly, he/she is considered to have failed the level.

In Game 2 - Word match, 5 words are given in advance, so the user is only given 2 guesses. If the user does not guess, the level fails.

In Game 3 - Letter hunt, the user is allowed to guess 7 wrong letters. As a result, if the user cannot guess all the letters of the word, the level fails.

## Consideration of user experience

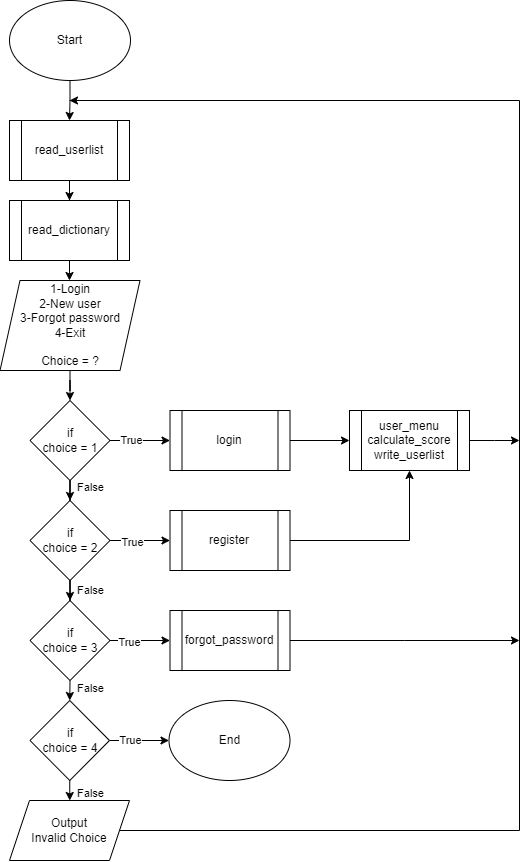
This programme is console based and has a very simple and easy to use operation. The information to be given to the user is given clearly. Where the user needs to make a choice, each option is numbered. It is enough for the user to enter a number. This both prevents mistakes and speeds up the use of the application. This is one of the points that improve user experience. In addition, considering possible wrong entries, the programme is coded in such a way that it does not give any detected error.

## Scope and limitations of the program

There are 3 different games and 5 different subjects in the programme. More subjects can be added in the future. In addition, I decided the words of 3 different difficulty levels with the help of ChatGPT. Considering that it is for students, words and word difficulties can be decided with the help of an instructor. In addition, the definitions of the words can be made more accurate and clearer with the help of an editor.

# Section 2: Design

## Main

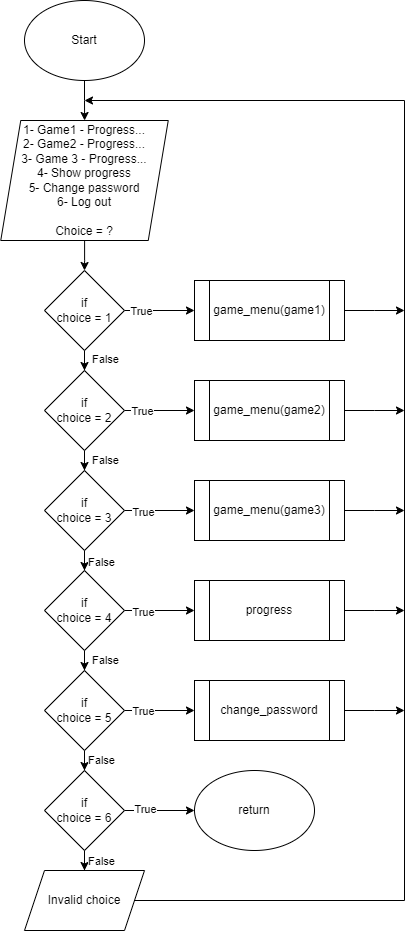
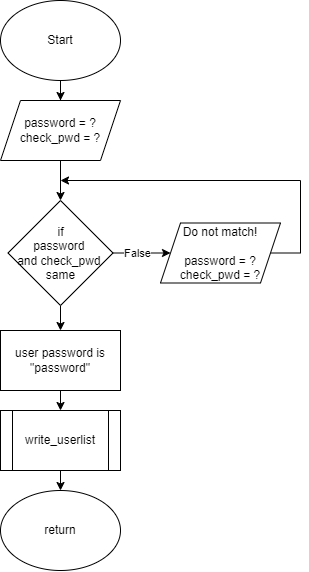


## Functions

The flowchart provides a broad overview of the logic and doesn't include every single detail. I designed it to be a helpful guide during the application preparation phase. I then improved the application by including additional functionality not explicitly mentioned in the flowchart.

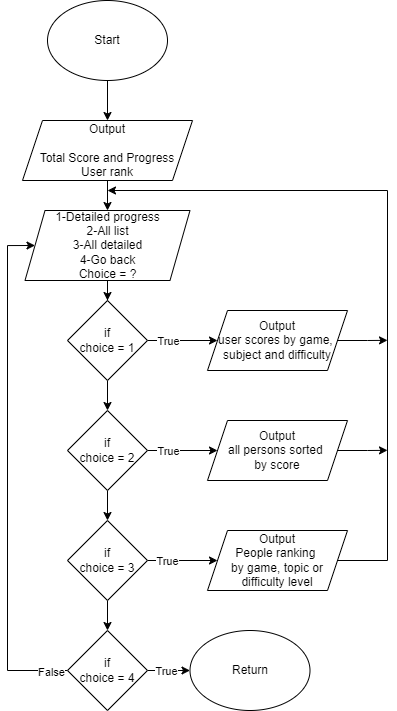
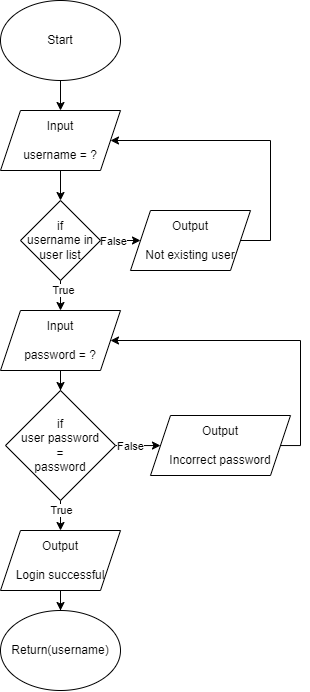
user\_menu

change\_password



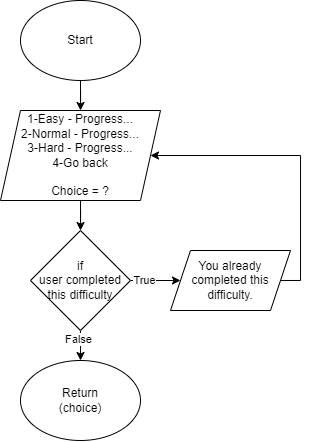
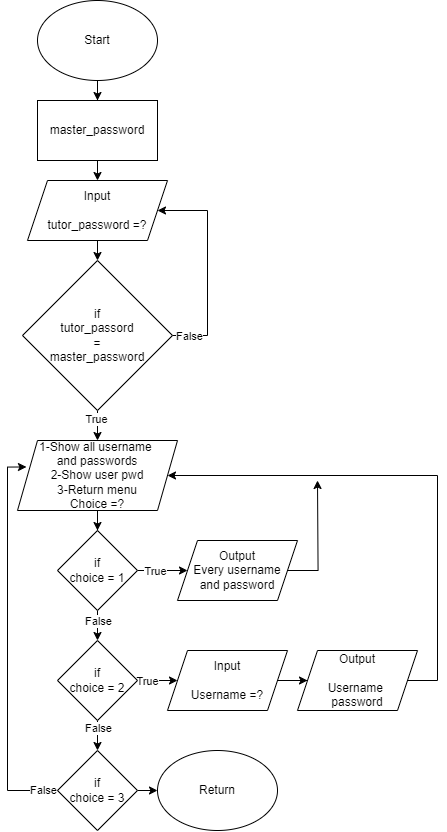
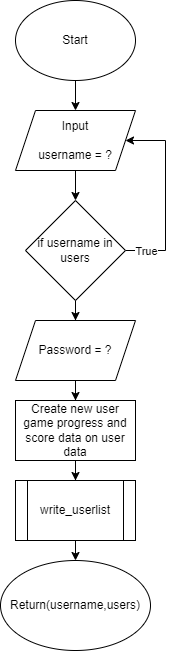
progress

login



forgot\_password

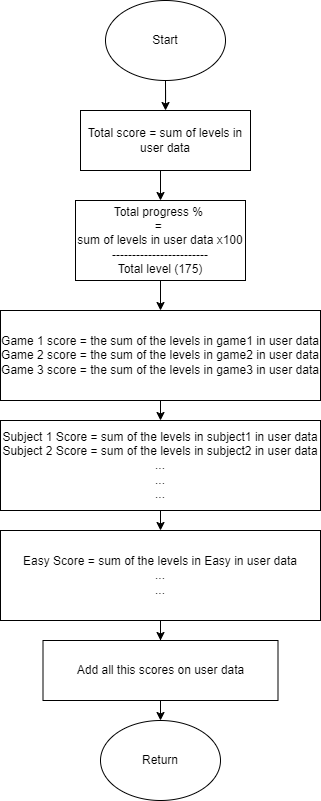
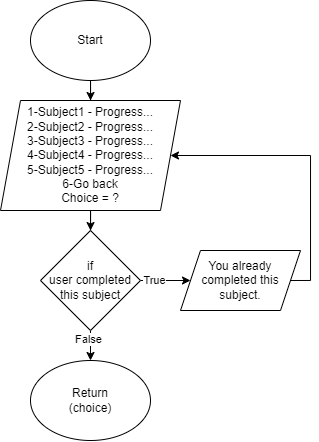
register



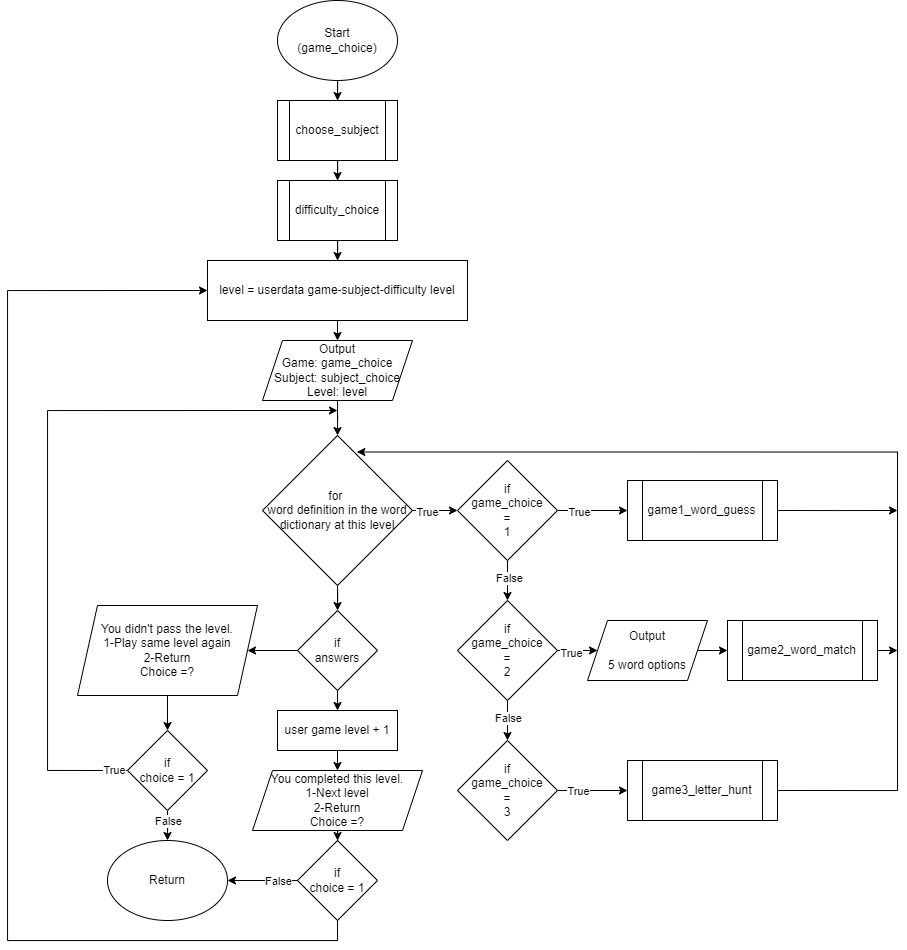
choose\_difficulty

forgot\_password

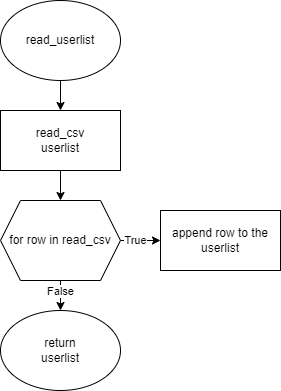
choose\_subject



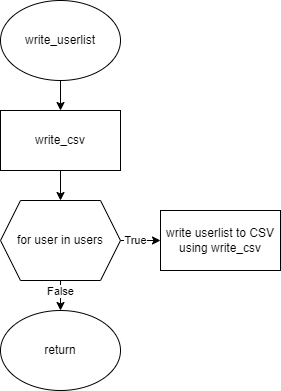
game\_menu



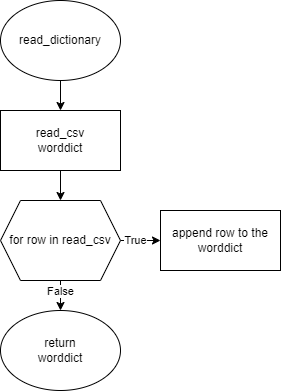
write\_userlist



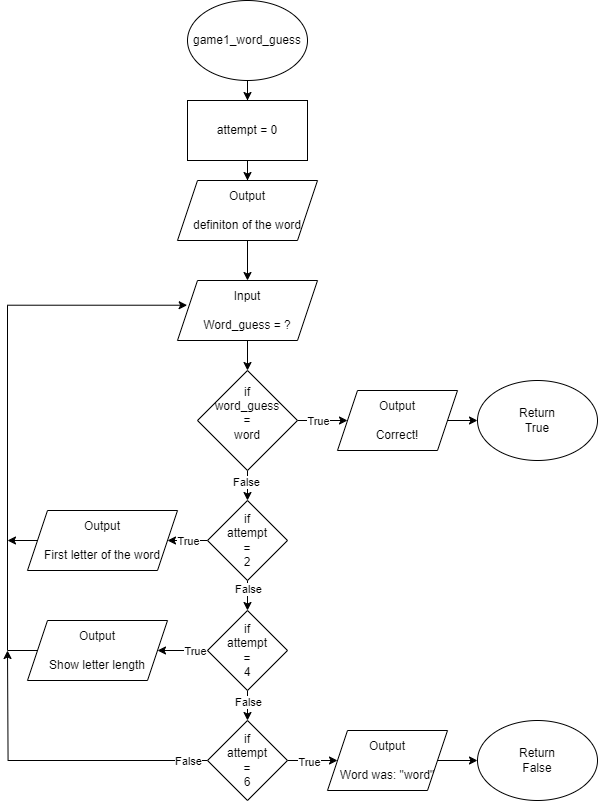
read\_userlist



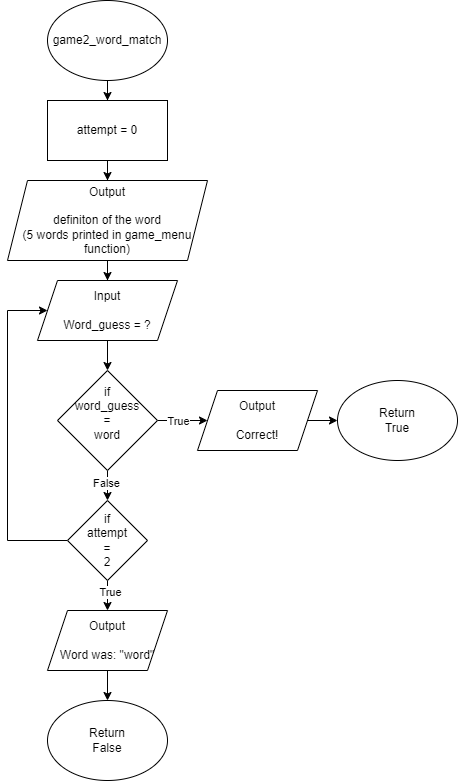
read\_dictionary



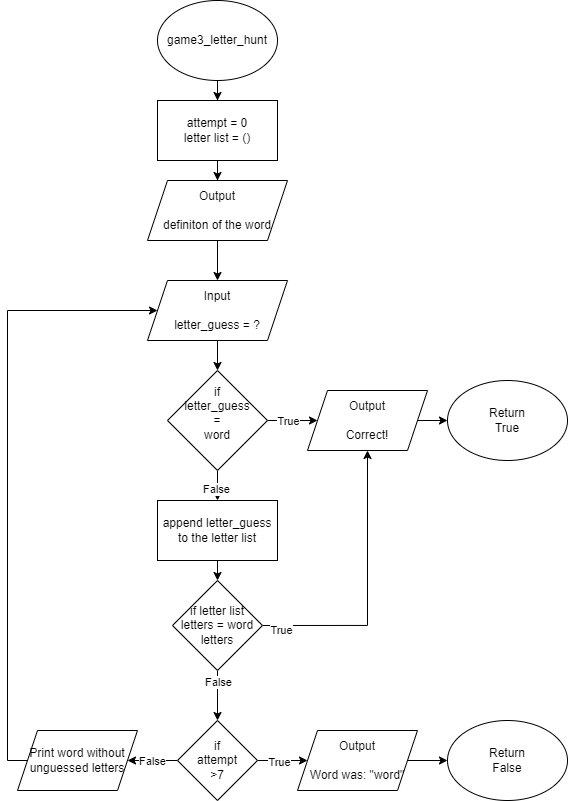
game1\_word\_guess



game2\_word\_match



game3\_letter\_hunt



# Section 3: Technical Overview

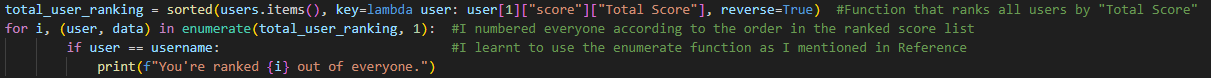
Many variables were used due to the length of the code, the layered structure of the application, having 3 games and the large number of functions. In addition, the code is divided into functions as much as possible to make it more understandable.

You can see all variables, data structures and functions in detail below.

|  |  |  |
| --- | --- | --- |
| **Variables** | **Type** | **Description** |
| wordfile | string | it contains CSV word file name  wordfile = "words.csv"  I assigned the file name to the variable because when I used different word csvs for testing and experimenting while working with the code, it was more convenient to do this with 1 change at the top of the code rather than looking for the location of the file in the code. |
| userfile | string | it contains CSV user file name  userfile = "users\_test.csv" |
| master\_pwd | string | master password for the teacher to log in and help in case the student forgets the password |
| game1 = "Word Guess"  game2 = "Word Match"  game3 = "Letter Hunt"  subject1 = "math"  subject2 = "science"  subject3 = "history"  subject4 = "art"  subject5 = "computer science"  easy = "easy"  normal = "normal"  hard = "hard" | string | Variables used for ease of writing code, easier use of functions and possible changes |
| max\_level = 3  number\_of\_difficulty = 3  max\_total\_level\_for\_each\_subject = max\_level \* number\_of\_difficulty  number\_of\_subjects = 5  max\_total\_level\_for\_each\_game = max\_total\_level\_for\_each\_subject \* number\_of\_subjects  number\_of\_games = 3  max\_total\_level = max\_total\_level\_for\_each\_game \* number\_of\_games | integer | Variables created to be used in progress display and score calculation. They are defined separately because they can be used separately when needed. Also, since I was still designing the general outline of the programme while writing the code, it was much easier to make changes here than in the function. |
| write\_csv | \_csv.writer | variable which contains “.writer()” function |
| user | string | a variable representing each key in the users dictionary. |

|  |  |
| --- | --- |
| **Defined Functions** | **Description** |
| write\_userlist(users) | Saves all user data and progress in the userlist to a CSV file with one user in each row. |
| read\_userlist() | Opens a new users dictionary. Using csv.reader, appends the user information and progress in each line in the CSV into the users dictionary so that each user is a dictionary.  **It returns users dictionary.** |
| login(users) | Provides user login. First it checks the username. Then it checks username's password. If it matches, **it returns the username.** |
| register(users) | It is used to create a new user. First it asks the user for username and checks that it is non existent. Then it asks for the password. Then it adds a user dictionary with progress and score from scratch to the users dictionary.  **It returns the username and users dictionary.** |
| forgot\_password | This function allows the tutor to use the master password to find the user's password when the user forgets it. The tutor can access the password by entering the name of a specific student or all students. |
| calculate\_score(users,username) | Based on the user's progress, it calcualtes both an overall score and individual game, subject, and difficulty-based scores. It also saves them in user data. |
| choose\_difficulty  (users,username,game\_choice,subject\_choice) | It shows the user their progress on the respective difficulty levels and asks them to choose a difficulty. It also checks whether the respective difficulty level has been completed. It prevents the selection of the finished difficulty.  **It returns difficulty\_choice.** |
| choose\_subject  (users,username,game\_choice) | Shows the user the subject list and progress. It then makes the subject selection. If the relevant subject is completed, it warns the user. **It returns subject\_choice.** |
| game1\_word\_guess(word,definition) | It prints the definition of the word and asks the user to guess the word.  On a correct guess, answer = True.  The 2nd wrong guess shows the first letter of the word,  the 4th wrong guess shows the length of the word.  On the 6th wrong guess, it says the word and answer = False.  **It returns answer.** |
| game2\_word\_match(word,definition) | It prints the definition of the word and asks the user to guess the word.  On a correct guess, answer = True.  Since 5 possible words are printed in the Game\_menu function,  answer = False on the 2nd wrong guess  **It returns answer.** |
| game3\_letter\_hunt(word,definition) | It prints the definition of the word and asks the user to guess letters of the word.  Shows the letter length of the word with each letter hidden.  If a letter in the word is predicted, it shows that letter.  On a correct all letter guesses, answer = True.  For 7 incorrect letter guesses, answer = False.  **It returns answer.** |
| game\_menu  (users,user,game\_choice) | This is the menu common to all 3 games. The topic selection and difficulty selection functions are called here.  Then, for every 5 words in the related level, the related game function is executed.  When all 5 words are guessed correctly according to the answer from the game function, the user's progress is changed.  In case of a wrong guess, the user is asked what he/she wants to do. |
| progress(users,username) | You can see the user's detailed progress, the list of all students by overall game score and the list by detailed score.  It sorts all users according to their score with the lambda function (GeeksforGeeks, 2023) and then enumerate to give everyone a number in order. **(Figure 1.)** |
| change\_password(users,username) | Allows the user to change their password after logging in. Checks the new password twice to avoid confusion.  It changes users data and **returns users.** |
| user\_menu(users,username) | After the user logs in, this function is the main menu they will see unless they log out.  It shows the 3 game options and their progress in the games.  It also offers a show progress option where the user can see their own and other users' progress.  There is also an option for the user to change their current password and log out. |
| main() | It is the main function that runs when the application is opened.  It has login, register and forgot password sections.  There is also an option to exit the application.  The read\_userlist and read\_dictionary functions run at the beginning of the function.  At the end, calculate\_score and write\_userlist functions are executed. |

|  |  |
| --- | --- |
| **Functions** | **Description** |
| lambda user: user[1]["score"]["Total Score"] | The lambda function is used to temporarily retrieve the "Total Score" value from user data |
| total\_user\_ranking =  sorted(users.items(), key=lambda user: user[1]["score"]["Total Score"], reverse=True) | The Sorted function uses lambda to sort all users according to the "Total score" value and assigns it to the "total\_user\_ranking" variable. |
| enumerate(total\_user\_ranking, 1) | The Enumerate function assigns a number to each item in the list in sequence. |
| **csv.reader()** | Function used to read data from CSV. It reads the CSV file line by line and returns each line as a list. |
| **csv.writer()** | It is a python function used to write to a CSV file. |
| **input()** | It is a function used to get input from the user. |
| **print()** | Used to print data to the console. |
| **exit()** | It is a function used to terminate the programme. |



**Figure 1.**

# Section 4: Coded solution

This section is divided into 3 parts. I don't embed the data into the code, I just store it as a CSV file. For this reason, the application needs two pre-created CSV files named user and words to run. The first part contains users CSV, the second part contains words CSV and the third part contains Python code.

Since I cannot upload more than 1 file via VLE in case of any problems due to copy-paste, I uploaded the users and words CSV files to Google Drive just in case, and shared the link here.

Here is a copy of the same data in CSV file:

<https://drive.google.com/drive/folders/1kIxpN822UAB4MwovZp_vMPRlUhzz_7Ub?usp=sharing>

## Example users CSV File (a file named “users.csv” needs to be created with this information)

username,password,game 1 subject 1 difficulty 1,g1s2d1,g1s3d1,g1s4d1,g1s5d1,g1s1d2,g1s2d2,g1s3d2,g1s4d2,g1s5d2,g1s1d3,g1s2d3,g1s3d3,g1s4d3,g1s5d3,g2s1d1,g2s2d1,g2s3d1,g2s4d1,g2s5d1,g2s1d2,g2s2d2,g2s3d2,g2s4d2,g2s5d2,g2s1d3,g2s2d3,g2s3d3,g2s4d3,g2s5d3,g3s1d1,g3s2d1,g3s3d1,g3s4d1,g3s5d1,g3s1d2,g3s2d2,g3s3d2,g3s4d2,g3s5d2,g3s1d3,g3s2d3,g3s3d3,g3s4d3,g3s5d3,Total Score,Total progress,Word Guess,Word Match,Letter Hunt,math,science,history,art,computer science,easy,normal,hard

bora,1234,2,4,4,1,4,4,4,4,4,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,24,13.7,22,1,1,8,6,9,0,1,5,9,10

mertcan,1991,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1.1,0,2,0,2,0,0,0,0,2,0,0

nadeen,2005,1,1,1,1,2,1,2,1,1,1,1,1,1,1,2,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0.6,1,0,0,0,0,0,0,1,0,0,1

## Words CSV file (a file named “words.csv” needs to be created with this information)

subject,difficulty,level,word,definition

math,easy,1,"Addition","A mathematical operation that combines two or more numbers to find their total."

math,easy,1,"Subtraction","A mathematical operation that finds the difference between two numbers."

math,easy,1,"Multiplication","A mathematical operation that combines repeated addition."

math,easy,1,"Division","A mathematical operation that splits a number into equal parts or groups."

math,easy,1,"Fraction","A number that represents a part of a whole, expressed as a ratio of two integers."

math,easy,2,"Geometry","A branch of mathematics that deals with shapes, sizes, and properties of space."

math,easy,2,"Algebra","A branch of mathematics that deals with symbols and the rules for manipulating those symbols to solve equations."

math,easy,2,"Trigonometry","A branch of mathematics that deals with the relationships between the sides and angles of triangles, especially right triangles."

math,easy,2,"Calculus","A branch of mathematics that deals with the study of rates of change and accumulation of quantities."

math,easy,2,"Statistics","A branch of mathematics that deals with the collection, analysis, interpretation, and presentation of data."

math,easy,3,"Equation","A mathematical statement that asserts the equality of two expressions."

math,easy,3,"Function","A relation between a set of inputs and a set of possible outputs where each input is related to exactly one output."

math,easy,3,"Variable","A symbol or letter that represents a quantity that can change or vary in value."

math,easy,3,"Graph","A visual representation of data showing the relationship between variables."

math,easy,3,"Matrix","A rectangular array of numbers, symbols, or expressions arranged in rows and columns."

math,normal,1,"Pythagorean Theorem","In a right-angled triangle, the square of the length of the hypotenuse is equal to the sum of the squares of the lengths of the other two sides."

math,normal,1,"Exponential Function","A mathematical function in which the independent variable appears as an exponent."

math,normal,1,"Derivative","A measure of how a function changes as its input changes, represented by the rate of change."

math,normal,1,"Integral","A mathematical concept that represents the area under a curve or the accumulation of quantities."

math,normal,1,"Probability","The likelihood of an event occurring, expressed as a fraction between 0 and 1."

math,normal,2,"Fractal","A complex geometric shape that can be split into parts, each of which is a reduced-scale copy of the whole."

math,normal,2,"Vector","A quantity that has both magnitude and direction, represented by an arrow in space."

math,normal,2,"Algorithm","A step-by-step procedure or set of rules for solving a problem, often performed by a computer."

math,normal,2,"Complex Number","A number that can be expressed in the form a + bi, where a and b are real numbers, and i is the imaginary unit (√-1)."

math,normal,2,"Topology","A branch of mathematics that deals with the properties of space that are preserved under continuous deformations, such as stretching or bending."

math,normal,3,"Equation","A mathematical statement that asserts the equality of two expressions."

math,normal,3,"Function","A relation between a set of inputs and a set of possible outputs where each input is related to exactly one output."

math,normal,3,"Variable","A symbol or letter that represents a quantity that can change or vary in value."

math,normal,3,"Graph","A visual representation of data showing the relationship between variables."

math,normal,3,"Matrix","A rectangular array of numbers, symbols, or expressions arranged in rows and columns."

math,hard,1,"Differential Equation","An equation that relates one or more functions and their derivatives."

math,hard,1,"Limit","A fundamental concept in calculus representing the value that a function or sequence approaches as the input approaches some value."

math,hard,1,"Vector Calculus","A branch of mathematics that extends classical calculus to higher dimensions, typically dealing with vector fields and vector-valued functions."

math,hard,1,"Discrete Mathematics","A branch of mathematics dealing with countable, distinct objects, often characterized by the study of integers, graphs, and formal logic."

math,hard,1,"Number Theory","A branch of mathematics that deals with the properties and relationships of numbers, particularly integers."

math,hard,2,"Abstract Algebra","A branch of mathematics dealing with algebraic structures such as groups, rings, and fields, where arithmetic operations are generalized and abstracted."

math,hard,2,"Partial Differential Equation","An equation involving partial derivatives of a function of several variables."

math,hard,2,"Multivariable Calculus","A branch of calculus dealing with functions of multiple variables, including differentiation and integration."

math,hard,2,"Real Analysis","A branch of mathematics that rigorously studies the properties of real numbers, sequences, and limits."

math,hard,2,"Combinatorics","A branch of mathematics concerning the study of finite or countable discrete structures, particularly the combinations of objects."

math,hard,3,"Abstract Geometry","A branch of geometry dealing with abstract, axiomatic systems and their properties."

math,hard,3,"Topological Space","A set with a collection of open subsets satisfying certain properties, studied in topology."

math,hard,3,"Functional Analysis","A branch of mathematics dealing with vector spaces endowed with some kind of limit-related structure and the linear operators acting upon these spaces."

math,hard,3,"Game Theory","A branch of mathematics and economics dealing with the analysis of strategies for dealing with competitive situations where the outcome of a participant's choice depends on the choices of others."

math,hard,3,"Cryptanalysis","The study of analyzing information systems to study hidden aspects of the systems, particularly for deciphering encrypted messages or securing communication protocols."

science,easy,1,"Observation","The action or process of observing something or someone carefully or in order to gain information."

science,easy,1,"Experiment","A scientific procedure undertaken to make a discovery, test a hypothesis, or demonstrate a known fact."

science,easy,1,"Hypothesis","A supposition or proposed explanation made on the basis of limited evidence as a starting point for further investigation."

science,easy,1,"Theory","A well-substantiated explanation of some aspect of the natural world that is acquired through the scientific method and repeatedly tested and confirmed through observation and experimentation."

science,easy,1,"Data","Facts and statistics collected together for reference or analysis."

science,easy,2,"Variable","Any factor, trait, or condition that can exist in differing amounts or types."

science,easy,2,"Control Group","In an experiment, the group that does not receive the treatment being tested; used as a benchmark to compare the results of the experimental group."

science,easy,2,"Independent Variable","The variable that is changed or controlled in a scientific experiment to test the effects on the dependent variable."

science,easy,2,"Dependent Variable","The variable being tested and measured in a scientific experiment; its value depends on the independent variable."

science,easy,2,"Peer Review","The evaluation of scientific, academic, or professional work by others working in the same field to ensure its validity and quality."

science,easy,3,"Quantum Mechanics","A fundamental theory in physics that describes the behavior of matter and energy on the atomic and subatomic levels."

science,easy,3,"Genetics","The branch of biology that deals with the study of genes, heredity, and variation in living organisms."

science,easy,3,"Relativity","The theory developed by Albert Einstein that describes the relationships between space and time, and between energy and matter."

science,easy,3,"Biochemistry","The branch of science that explores the chemical processes within and related to living organisms."

science,easy,3,"Climate Change","A long-term change in the average weather patterns that have come to define Earth's local, regional, and global climates."

science,normal,1,"Cell Theory","The scientific theory stating that all living organisms are composed of cells, the basic structural and functional units of life."

science,normal,1,"Natural Selection","The process by which organisms with favorable traits are more likely to survive and reproduce, leading to the evolution of species over time."

science,normal,1,"Plate Tectonics","The theory that Earth's outer shell is divided into several plates that glide over the mantle, causing continental drift, earthquakes, volcanic activity, and the formation of mountains and oceanic trenches."

science,normal,1,"Photosynthesis","The process by which green plants and some other organisms use sunlight to synthesize nutrients from carbon dioxide and water, producing oxygen as a byproduct."

science,normal,1,"Atomic Theory","The theory that all matter is composed of small, indivisible particles called atoms, which combine in specific ratios to form compounds."

science,normal,2,"Ecosystem","A biological community of interacting organisms and their physical environment."

science,normal,2,"Natural Resources","Materials or substances such as minerals, forests, water, and fertile land that occur in nature and can be used for economic gain."

science,normal,2,"Biological Evolution","The change in the heritable characteristics of populations of organisms over successive generations, driven by natural selection, mutation, genetic drift, and gene flow."

science,normal,2,"Chemical Reaction","A process that involves the rearrangement of the molecular or ionic structure of a substance, typically accompanied by the release or absorption of energy."

science,normal,2,"Cognitive Psychology","The branch of psychology that focuses on the study of mental processes such as perception, memory, attention, language, problem-solving, and decision-making."

science,normal,3,"Newtonian Mechanics","The branch of physics that deals with the motion of macroscopic objects and the forces acting on them, according to Newton's laws of motion."

science,normal,3,"Neurotransmitter","A chemical substance that is released at the end of a nerve fiber by the arrival of a nerve impulse and causes the transfer of the impulse to another nerve fiber, muscle fiber, or some other structure."

science,normal,3,"Quantum Entanglement","A phenomenon in quantum mechanics where the quantum states of two or more particles become correlated in such a way that the state of one particle cannot be described independently of the state of the others."

science,normal,3,"Biodiversity","The variety and variability of life on Earth, including genetic diversity, species diversity, and ecosystem diversity."

science,normal,3,"Greenhouse Effect","The process by which radiation from Earth's atmosphere warms the planet's surface to a temperature above what it would be without its atmosphere."

science,hard,1,"Astrophysics","The branch of astronomy that deals with the physics of the universe, including the physical properties of celestial objects and the interactions between them."

science,hard,1,"Quantum Field Theory","A theoretical framework in which the classical fields of physics are described using quantum mechanics."

science,hard,1,"Biotechnology","The use of living organisms, cells, or biological molecules to develop products and technologies that improve the quality of life."

science,hard,1,"Particle Physics","The branch of physics that studies the nature and behavior of particles that constitute matter and radiation."

science,hard,1,"String Theory","A theoretical framework in which the point-like particles of particle physics are replaced by one-dimensional objects called strings."

science,hard,2,"Artificial Intelligence","The simulation of human intelligence processes by machines, especially computer systems."

science,hard,2,"Stem Cells","Undifferentiated cells that have the potential to develop into various types of cells in the body during early life and growth."

science,hard,2,"Dark Matter","A hypothetical form of matter that is thought to make up approximately 85% of the matter in the universe and does not emit or interact with electromagnetic radiation, making it invisible or 'dark'."

science,hard,2,"Nanotechnology","The manipulation of matter on an atomic, molecular, and supramolecular scale to create materials, devices, and systems with new properties and functions."

science,hard,2,"Neuroscience","The scientific study of the nervous system, including its structure, function, and disorders."

science,hard,3,"Quantum Mechanics","A fundamental theory in physics that describes the behavior of matter and energy on the atomic and subatomic levels."

science,hard,3,"Genetics","The branch of biology that deals with the study of genes, heredity, and variation in living organisms."

science,hard,3,"Relativity","The theory developed by Albert Einstein that describes the relationships between space and time, and between energy and matter."

science,hard,3,"Biochemistry","The branch of science that explores the chemical processes within and related to living organisms."

science,hard,3,"Climate Change","A long-term change in the average weather patterns that have come to define Earth's local, regional, and global climates."

history,easy,1,"Ancient","Very old; from a long time ago."

history,easy,1,"Artifact","An object made or used by people in the past."

history,easy,1,"Archaeologist","A scientist who studies ancient people and their cultures by examining artifacts."

history,easy,1,"Civilization","A society with cities, a central government, and social classes."

history,easy,1,"Culture","The beliefs, customs, arts, etc., of a particular society, group, place, or time."

history,easy,2,"Timeline","A line that shows the order of events in history."

history,easy,2,"Dynasty","A series of rulers from the same family."

history,easy,2,"Empire","A group of countries or regions that are controlled by one ruler or government."

history,easy,2,"Mesopotamia","An ancient region located in the eastern Mediterranean bounded in the northeast by the Zagros Mountains and in the southeast by the Arabian Plateau."

history,easy,2,"Pharaoh","A ruler of ancient Egypt."

history,easy,3,"Revolution","Fundamental change in the way of thinking or organizing society."

history,easy,3,"Industrialization","Development of industries in a country or region on a wide scale."

history,easy,3,"Imperialism","Policy of extending a country's power and influence through diplomacy or military force."

history,easy,3,"Decolonization","Process of dismantling colonial empires and granting independence to colonized nations."

history,easy,3,"Communism","Political and economic ideology advocating for collective ownership of property and resources."

history,normal,1,"Timeline","A line that shows the order of events in history."

history,normal,1,"Dynasty","A series of rulers from the same family."

history,normal,1,"Empire","A group of countries or regions that are controlled by one ruler or government."

history,normal,1,"Mesopotamia","An ancient region located in the eastern Mediterranean bounded in the northeast by the Zagros Mountains and in the southeast by the Arabian Plateau."

history,normal,1,"Pharaoh","A ruler of ancient Egypt."

history,normal,2,"Revolution","Fundamental change in the way of thinking or organizing society."

history,normal,2,"Industrialization","Development of industries in a country or region on a wide scale."

history,normal,2,"Imperialism","Policy of extending a country's power and influence through diplomacy or military force."

history,normal,2,"Decolonization","Process of dismantling colonial empires and granting independence to colonized nations."

history,normal,2,"Communism","Political and economic ideology advocating for collective ownership of property and resources."

history,normal,3,"Feudalism","A political and economic system prevalent in medieval Europe, characterized by the granting of land in exchange for loyalty and service."

history,normal,3,"Renaissance","A period of cultural, artistic, and intellectual revival in Europe, marking the transition from the Middle Ages to modernity."

history,normal,3,"Enlightenment","An intellectual and philosophical movement in 18th century Europe that emphasized reason, science, and individualism."

history,normal,3,"Reformation","A 16th century religious movement that aimed to reform the Roman Catholic Church and resulted in the establishment of Protestantism."

history,normal,3,"Colonialism","The policy or practice of acquiring and controlling foreign territories for economic exploitation and political dominance."

history,hard,1,"Feudalism","A political and economic system prevalent in medieval Europe, characterized by the granting of land in exchange for loyalty and service."

history,hard,1,"Renaissance","A period of cultural, artistic, and intellectual revival in Europe, marking the transition from the Middle Ages to modernity."

history,hard,1,"Enlightenment","An intellectual and philosophical movement in 18th century Europe that emphasized reason, science, and individualism."

history,hard,1,"Reformation","A 16th century religious movement that aimed to reform the Roman Catholic Church and resulted in the establishment of Protestantism."

history,hard,1,"Colonialism","The policy or practice of acquiring and controlling foreign territories for economic exploitation and political dominance."

history,hard,2,"Neolithic Revolution","The transition from hunting and gathering to agriculture and settlement, marking the beginning of settled societies."

history,hard,2,"Globalization","The process of increased interconnectedness and interdependence among countries, economies, and cultures."

history,hard,2,"Industrial Revolution","A period of rapid industrialization, technological advancement, and urbanization in the 18th and 19th centuries."

history,hard,2,"Cold War","A state of geopolitical tension between the Soviet Union and the United States and their respective allies, without direct military conflict, from the mid-20th century to the early 1990s."

history,hard,2,"French Revolution","A period of radical social and political upheaval in France from 1789 to 1799, resulting in the overthrow of the monarchy and the rise of democracy."

history,hard,3,"Hammurabi's Code","One of the earliest and most complete written legal codes, proclaimed by the Babylonian king Hammurabi, consisting of 282 laws covering various aspects of daily life."

history,hard,3,"Great Depression","A severe worldwide economic depression that took place mostly during the 1930s, originating in the United States following the stock market crash of 1929."

history,hard,3,"World War I","A global war primarily centered in Europe that lasted from 1914 to 1918, involving many of the world's great powers and resulting in millions of casualties."

history,hard,3,"World War II","A global war that lasted from 1939 to 1945, involving many of the world's nations and characterized by significant events including the Holocaust and the use of atomic weapons."

history,hard,3,"Civil Rights Movement","A social movement in the United States during the 1950s and 1960s aimed at ending racial segregation and discrimination against African Americans and achieving legal recognition and federal protection of their civil rights."

art,easy,1,"Expression","The process of making known one's thoughts or feelings, especially in a creative way."

art,easy,1,"Creativity","The use of imagination or original ideas to create something; inventiveness."

art,easy,1,"Aesthetic","Concerned with beauty or the appreciation of beauty."

art,easy,1,"Medium","The materials or methods used by an artist to create a work of art."

art,easy,1,"Composition","The arrangement of elements within a work of art."

art,easy,2,"Palette","A range of colors or qualities available to an artist."

art,easy,2,"Perspective","The technique used to represent three-dimensional objects and depth on a two-dimensional surface."

art,easy,2,"Impressionism","A style of painting characterized by capturing the overall impression of a scene, rather than details."

art,easy,2,"Sculpture","The art of making two- or three-dimensional representative or abstract forms, especially by carving stone or wood or by casting metal or plaster."

art,easy,2,"Abstract","Art that does not attempt to represent an accurate depiction of visual reality but instead uses shapes, colors, forms, and gestural marks to achieve its effect."

art,easy,3,"Renaissance","A period in European history, covering the span between the 14th and 17th centuries, marking the transition from the Middle Ages to modernity and the revival of interest in classical art, literature, and culture."

art,easy,3,"Baroque","A highly ornate and often extravagant style of architecture, art, and music that flourished in Europe from the early 17th to late 18th century."

art,easy,3,"Cubism","An early-20th-century art movement that revolutionized European painting and sculpture and inspired related movements in music, literature, and architecture."

art,easy,3,"Surrealism","An artistic movement and style that emerged in the early 20th century, characterized by dreamlike imagery, unexpected juxtapositions, and non sequitur."

art,easy,3,"Pop Art","An art movement that emerged in the mid-20th century, characterized by imagery from popular culture and mass media, such as advertising, comic books, and consumer products."

art,normal,1,"Palette","A range of colors or qualities available to an artist."

art,normal,1,"Perspective","The technique used to represent three-dimensional objects and depth on a two-dimensional surface."

art,normal,1,"Impressionism","A style of painting characterized by capturing the overall impression of a scene, rather than details."

art,normal,1,"Sculpture","The art of making two- or three-dimensional representative or abstract forms, especially by carving stone or wood or by casting metal or plaster."

art,normal,1,"Abstract","Art that does not attempt to represent an accurate depiction of visual reality but instead uses shapes, colors, forms, and gestural marks to achieve its effect."

art,normal,2,"Renaissance","A period in European history, covering the span between the 14th and 17th centuries, marking the transition from the Middle Ages to modernity and the revival of interest in classical art, literature, and culture."

art,normal,2,"Baroque","A highly ornate and often extravagant style of architecture, art, and music that flourished in Europe from the early 17th to late 18th century."

art,normal,2,"Cubism","An early-20th-century art movement that revolutionized European painting and sculpture and inspired related movements in music, literature, and architecture."

art,normal,2,"Surrealism","An artistic movement and style that emerged in the early 20th century, characterized by dreamlike imagery, unexpected juxtapositions, and non sequitur."

art,normal,2,"Pop Art","An art movement that emerged in the mid-20th century, characterized by imagery from popular culture and mass media, such as advertising, comic books, and consumer products."

art,normal,3,"Abstract Expressionism","An art movement developed in the mid-20th century, characterized by large-scale, nonrepresentational paintings that express the artist's emotions and subconscious thoughts."

art,normal,3,"Minimalism","An art movement that emerged in the 1960s, characterized by simplicity, geometric forms, and an emphasis on the materiality of the artwork."

art,normal,3,"Fauvism","An early 20th-century art movement led by Henri Matisse, characterized by bold colors, simplified forms, and gestural brushwork."

art,normal,3,"Realism","An artistic movement that emerged in the mid-19th century, emphasizing accurate depiction of ordinary people and everyday life."

art,normal,3,"Dadaism","An art movement of the European avant-garde in the early 20th century, characterized by a rejection of logic, reason, and aestheticism in favor of nonsense, irrationality, and anti-bourgeois protest."

art,hard,1,"Chiaroscuro","The use of strong contrasts between light and dark to create a sense of volume in modeling three-dimensional objects and figures."

art,hard,1,"Pointillism","A technique of painting in which small, distinct dots of pure color are applied in patterns to form an image."

art,hard,1,"Avant-garde","New and experimental ideas and methods in art, music, or literature, often characterized by challenging traditional forms or conventions."

art,hard,1,"Encaustic","A painting technique in which pigments are mixed with hot wax and fixed as an emulsion on the surface with heat, especially popular in ancient Greece and Rome."

art,hard,1,"Trompe-l'oeil","An art technique that uses realistic imagery to create the optical illusion that the depicted objects exist in three dimensions."

art,hard,2,"Dadaism","An art movement of the European avant-garde in the early 20th century, characterized by a rejection of logic, reason, and aestheticism in favor of nonsense, irrationality, and anti-bourgeois protest."

art,hard,2,"Futurism","An early 20th-century art movement centered in Italy, emphasizing speed, technology, youth, and violence, and the objectification of the dynamic character of modern life."

art,hard,2,"Installation","An artistic genre of three-dimensional works that are often site-specific and designed to transform the perception of a space."

art,hard,2,"Monochromatic","Using only one color or shades of one color in an artwork."

art,hard,2,"Found Object","An object that is found and considered by an artist to have aesthetic value, which is later used in an artwork."

art,hard,3,"Postmodernism","A late 20th-century movement in art, architecture, and criticism that represents a departure from modernism and is characterized by a mixing of styles, borrowing from historical sources, and often ironic or humorous treatments of traditional ideas and forms."

art,hard,3,"Neo-Expressionism","An art movement of the late 20th century characterized by a return to emotionalism and the depiction of recognizable objects and figures."

art,hard,3,"Conceptual Art","Art in which the concept or idea behind the work is more important than the finished art object."

art,hard,3,"Gothic","An art and architectural style that originated in medieval Europe and is characterized by its pointed arches, ribbed vaults, and flying buttresses."

art,hard,3,"Kinetic Art","Art that depends on motion for its effects and can include sculpture, mobiles, and installations."

computer science,easy,1,"Algorithm","A set of instructions designed to perform a specific task or solve a particular problem, especially when applied to a computer program."

computer science,easy,1,"Programming","The process of writing code to instruct a computer to perform specific tasks or operations."

computer science,easy,1,"Data Structure","A way of organizing and storing data in a computer so that it can be accessed and modified efficiently."

computer science,easy,1,"Computer","An electronic device capable of processing data according to programmed instructions."

computer science,easy,1,"Software","The programs and other operating information used by a computer."

computer science,easy,2,"Binary","A number system that uses only two digits, 0 and 1, to represent numbers and perform arithmetic operations."

computer science,easy,2,"Algorithm Analysis","The study of the efficiency and performance of algorithms, typically measured in terms of time and space complexity."

computer science,easy,2,"Database","An organized collection of data, typically stored and accessed electronically from a computer system."

computer science,easy,2,"Object-Oriented Programming","A programming paradigm based on the concept of objects, which can contain data and code to manipulate that data."

computer science,easy,2,"Compiler","A program that translates source code written in a high-level programming language into machine code or intermediate code."

computer science,easy,3,"Artificial Intelligence","The simulation of human intelligence processes by machines, especially computer systems."

computer science,easy,3,"Machine Learning","A subset of artificial intelligence that enables computers to learn from data and improve performance on specific tasks without being explicitly programmed."

computer science,easy,3,"Operating System","The software that manages computer hardware and provides common services for computer programs."

computer science,easy,3,"Encryption","The process of converting information or data into a code to prevent unauthorized access."

computer science,easy,3,"Network","A collection of computers and other devices connected together to share resources and information."

computer science,normal,1,"Recursion","A programming technique where a function calls itself in order to solve smaller instances of the same problem."

computer science,normal,1,"Variable","A named storage location in a computer's memory that contains data that can be manipulated and changed during program execution."

computer science,normal,1,"Conditional Statement","A programming construct that performs different actions depending on whether a condition is true or false."

computer science,normal,1,"Loop","A programming construct that repeats a block of code until a certain condition is met."

computer science,normal,1,"Function","A self-contained block of code that performs a specific task and can be reused throughout a program."

computer science,normal,2,"Recursion","A programming technique where a function calls itself in order to solve smaller instances of the same problem."

computer science,normal,2,"Variable","A named storage location in a computer's memory that contains data that can be manipulated and changed during program execution."

computer science,normal,2,"Conditional Statement","A programming construct that performs different actions depending on whether a condition is true or false."

computer science,normal,2,"Loop","A programming construct that repeats a block of code until a certain condition is met."

computer science,normal,2,"Function","A self-contained block of code that performs a specific task and can be reused throughout a program."

computer science,normal,3,"Recursion","A programming technique where a function calls itself in order to solve smaller instances of the same problem."

computer science,normal,3,"Variable","A named storage location in a computer's memory that contains data that can be manipulated and changed during program execution."

computer science,normal,3,"Conditional Statement","A programming construct that performs different actions depending on whether a condition is true or false."

computer science,normal,3,"Loop","A programming construct that repeats a block of code until a certain condition is met."

computer science,normal,3,"Function","A self-contained block of code that performs a specific task and can be reused throughout a program."

computer science,hard,1,"Multithreading","The ability of a CPU to execute multiple threads or processes concurrently, allowing for improved performance and responsiveness in multitasking environments."

computer science,hard,1,"Abstraction","The process of simplifying complex systems or ideas by focusing only on the essential aspects, often used in programming to manage complexity and improve maintainability."

computer science,hard,1,"Asynchronous Programming","A programming paradigm that allows tasks to be executed independently of the main program flow, enabling non-blocking execution and improved concurrency."

computer science,hard,1,"Parallel Computing","A type of computation in which many calculations or processes are carried out simultaneously, typically using multiple processors or cores."

computer science,hard,1,"Cybersecurity","The practice of protecting computer systems, networks, and data from unauthorized access or attacks."

computer science,hard,2,"Big Data","Extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions."

computer science,hard,2,"Cloud Computing","The delivery of computing services, including servers, storage, databases, networking, software, and analytics, over the internet."

computer science,hard,2,"Computer Graphics","The field of visual computing, including the creation, manipulation, and rendering of images and animations using computers."

computer science,hard,2,"Quantum Computing","A type of computing that takes advantage of the laws of quantum mechanics, allowing exponentially faster processing of certain types of problems."

computer science,hard,2,"Blockchain","A decentralized, distributed ledger technology used to record transactions across multiple computers in a way that is secure, transparent, and tamper-resistant."

computer science,hard,3,"Virtual Reality","An artificial environment created with software and presented to the user in such a way that they suspend belief and accept it as a real environment."

computer science,hard,3,"Internet of Things (IoT)","The network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data."

computer science,hard,3,"Artificial Neural Network","A computational model inspired by the structure and function of the human brain, consisting of interconnected nodes (neurons) that process information and learn from experience."

computer science,hard,3,"Natural Language Processing","A field of artificial intelligence and linguistics concerned with the interaction between computers and humans through natural language, enabling computers to understand, interpret, and generate human language."

computer science,hard,3,"Machine Vision","The ability of a computer to interpret and understand visual information from the real world, often used in applications such as object recognition and autonomous vehicles."

## Code

import csv

import random

wordfile = "words.csv"

userfile = "users.csv"

master\_pwd = "1234"

game1 = "Word Guess"

game2 = "Word Match"

game3 = "Letter Hunt"

subject1 = "math"

subject2 = "science"

subject3 = "history"

subject4 = "art"

subject5 = "computer science"

easy = "easy"

normal = "normal"

hard = "hard"

max\_level = 3

number\_of\_difficulty = 3

max\_total\_level\_for\_each\_subject = max\_level \* number\_of\_difficulty

number\_of\_subjects = 5

max\_total\_level\_for\_each\_game = max\_total\_level\_for\_each\_subject \* number\_of\_subjects

number\_of\_games = 3

max\_total\_level = max\_total\_level\_for\_each\_game \* number\_of\_games

def write\_userlist(users):

with open(userfile, "w", newline="") as users\_csv:

write\_csv = csv.writer(users\_csv) #Writing header to csv

write\_csv.writerow(["username","password","game 1 subject 1 difficulty 1","g1s2d1","g1s3d1","g1s4d1","g1s5d1","g1s1d2","g1s2d2","g1s3d2","g1s4d2","g1s5d2","g1s1d3","g1s2d3","g1s3d3","g1s4d3","g1s5d3",

"g2s1d1","g2s2d1","g2s3d1","g2s4d1","g2s5d1","g2s1d2","g2s2d2","g2s3d2","g2s4d2","g2s5d2","g2s1d3","g2s2d3","g2s3d3","g2s4d3","g2s5d3",

"g3s1d1","g3s2d1","g3s3d1","g3s4d1","g3s5d1","g3s1d2","g3s2d2","g3s3d2","g3s4d2","g3s5d2","g3s1d3","g3s2d3","g3s3d3","g3s4d3","g3s5d3",

"Total Score","Total progress",

game1,game2,game3,

subject1,subject2,subject3,subject4,subject5,

easy,normal,hard

])

for user in users: #It writes each data of each user in the user dictionarry to a CSV file

write\_csv.writerow([user,users[user]["pwd"],users[user]["progress"][game1][subject1][easy],users[user]["progress"][game1][subject1][normal],users[user]["progress"][game1][subject1][hard],

users[user]["progress"][game1][subject2][easy],users[user]["progress"][game1][subject2][normal],users[user]["progress"][game1][subject2][hard],

users[user]["progress"][game1][subject3][easy],users[user]["progress"][game1][subject3][normal],users[user]["progress"][game1][subject3][hard],

users[user]["progress"][game1][subject4][easy],users[user]["progress"][game1][subject4][normal],users[user]["progress"][game1][subject4][hard],

users[user]["progress"][game1][subject5][easy],users[user]["progress"][game1][subject5][normal],users[user]["progress"][game1][subject5][hard],

users[user]["progress"][game2][subject1][easy],users[user]["progress"][game2][subject1][normal],users[user]["progress"][game2][subject1][hard],

users[user]["progress"][game2][subject2][easy],users[user]["progress"][game2][subject2][normal],users[user]["progress"][game2][subject2][hard],

users[user]["progress"][game2][subject3][easy],users[user]["progress"][game2][subject3][normal],users[user]["progress"][game2][subject3][hard],

users[user]["progress"][game2][subject4][easy],users[user]["progress"][game2][subject4][normal],users[user]["progress"][game2][subject4][hard],

users[user]["progress"][game2][subject5][easy],users[user]["progress"][game2][subject5][normal],users[user]["progress"][game2][subject5][hard],

users[user]["progress"][game3][subject1][easy],users[user]["progress"][game3][subject1][normal],users[user]["progress"][game3][subject1][hard],

users[user]["progress"][game3][subject2][easy],users[user]["progress"][game3][subject2][normal],users[user]["progress"][game3][subject2][hard],

users[user]["progress"][game3][subject3][easy],users[user]["progress"][game3][subject3][normal],users[user]["progress"][game3][subject3][hard],

users[user]["progress"][game3][subject4][easy],users[user]["progress"][game3][subject4][normal],users[user]["progress"][game3][subject4][hard],

users[user]["progress"][game3][subject5][easy],users[user]["progress"][game3][subject5][normal],users[user]["progress"][game3][subject5][hard],

users[user]["score"]["Total Score"],

users[user]["score"]["Total progress %"],

users[user]["score"][game1],users[user]["score"][game2],users[user]["score"][game3],

users[user]["score"][subject1],users[user]["score"][subject2],users[user]["score"][subject3],users[user]["score"][subject4],users[user]["score"][subject5],

users[user]["score"][easy],users[user]["score"][normal],users[user]["score"][hard],

])

def read\_userlist():

users = {}

with open(userfile, "r", newline="") as users\_csv:

read\_csv = csv.reader(users\_csv)

next(read\_csv)

for row in read\_csv: #Each line in the CSV represents a user. Imports the data from each row into the user's dictionarry

users[row[0]] = {"pwd":row[1],

"progress":{game1:{subject1:{easy:int(row[2]),normal:int(row[3]),hard:int(row[4])},

subject2:{easy:int(row[5]),normal:int(row[6]),hard:int(row[7])},

subject3:{easy:int(row[8]),normal:int(row[9]),hard:int(row[10])},

subject4:{easy:int(row[11]),normal:int(row[12]),hard:int(row[13])},

subject5:{easy:int(row[14]),normal:int(row[15]),hard:int(row[16])}},

game2:{subject1:{easy:int(row[17]),normal:int(row[18]),hard:int(row[19])},

subject2:{easy:int(row[20]),normal:int(row[21]),hard:int(row[22])},

subject3:{easy:int(row[23]),normal:int(row[24]),hard:int(row[25])},

subject4:{easy:int(row[26]),normal:int(row[27]),hard:int(row[28])},

subject5:{easy:int(row[29]),normal:int(row[30]),hard:int(row[31])}},

game3:{subject1:{easy:int(row[32]),normal:int(row[33]),hard:int(row[34])},

subject2:{easy:int(row[35]),normal:int(row[36]),hard:int(row[37])},

subject3:{easy:int(row[38]),normal:int(row[39]),hard:int(row[40])},

subject4:{easy:int(row[41]),normal:int(row[42]),hard:int(row[43])},

subject5:{easy:int(row[44]),normal:int(row[45]),hard:int(row[46])}}

},

"score":{"Total Score":int(row[47]),

"Total progress %":float(row[48]),

game1:int(row[49]),game2:int(row[50]),game3:int(row[51]),

subject1:int(row[52]),subject2:int(row[53]),subject3:int(row[54]),subject4:int(row[55]),subject5:int(row[56]),

easy:int(row[57]),normal:int(row[58]),hard:int(row[59])

}

}

return(users)

def read\_dictionary():

global worddict

worddict = {}

with open(wordfile, "r", newline="") as words\_csv:

read\_csv = csv.reader(words\_csv)

next(read\_csv)

for row in read\_csv:

subject = row[0] #I have made temporary assignments for ease of understanding

difficulty = row[1]

level = row[2]

word = row[3]

definition = row[4]

if subject not in worddict: #If there is no dictionary on that subject, create a new dictionary

worddict[subject] = {}

if difficulty not in worddict[subject]: #If there is no dictionary on this difficulty in the subject, create a new dictionary

worddict[subject][difficulty] = {}

if level not in worddict[subject][difficulty]: #If there is no dictionary on this level in this difficulty in the subject, create a new dictionary

worddict[subject][difficulty][level] = {}

worddict[subject][difficulty][level][word] = definition #Place the word and definition in correct subject-difficulty-level dictionary

def login(users):

username = input("Username:")

while username not in users: #Confirms the existence of the User

print("Not existing username. Try another username or press enter to go menu.")

username = input("Username:")

if username == "": #If the user cannot remember their username, they can press enter and go back.

enter = False

return(username,enter)

password = input("Password:")

while password != users[username]["pwd"]:

print("Incorrect password. Try another password or press enter to go menu.")

password = input("Password:")

if password == "": #If the user cannot remember their password, they can press enter and go back.

enter = False

return(username,enter)

print(f"Login successful. \nWelcome {username}!")

enter = True

return(username,enter)

def register(users):

username = input("Username:")

while username in users: #Prevents username duplication

print("Username already exists. Try another username or press enter to go menu.")

username = input("Username:")

if username == "":

enter = False

return(username,users,enter)

password = input("Password:")

while password == "":

print("You need to choose a password!")

password = input("Password:")

check\_password = input("Re-enter password:")

while password != check\_password: #Double checking for password

print("Passwords do not match. Type again.")

password = input("Password:")

check\_password = input("Re-enter password:")

#Creating new user's username, password and other dictionaries used for progress in Users dictionarry

users[username] = {"pwd":password,"progress":{

game1:{

subject1:{easy:1,normal:1,hard:1},

subject2:{easy:1,normal:1,hard:1},

subject3:{easy:1,normal:1,hard:1},

subject4:{easy:1,normal:1,hard:1},

subject5:{easy:1,normal:1,hard:1}},

game2:{

subject1:{easy:1,normal:1,hard:1},

subject2:{easy:1,normal:1,hard:1},

subject3:{easy:1,normal:1,hard:1},

subject4:{easy:1,normal:1,hard:1},

subject5:{easy:1,normal:1,hard:1}},

game3:{

subject1:{easy:1,normal:1,hard:1},

subject2:{easy:1,normal:1,hard:1},

subject3:{easy:1,normal:1,hard:1},

subject4:{easy:1,normal:1,hard:1},

subject5:{easy:1,normal:1,hard:1}}

},

"score":{

"Total Score":0,

"Total progress %":0,

game1:0,game2:0,game3:0,

subject1:0,subject2:0,subject3:0,subject4:0,subject5:0,

easy:0,normal:0,hard:0

}

}

print(f"User succesfully created. \nWelcome {username}!")

write\_userlist(users)

enter = True

return(username,users,enter)

def forgot\_password(users):

print("Talk with your tutor.")

while True:

tutor\_login = input("Press enter to go menu or if you are a tutor, type master password(\*\*\*\*Master password:1234\*\*\*\*):")

#I wrote the master passcode here so that you can check the function. Otherwise it will not be written in the original version.

if tutor\_login == "":

return

else:

if tutor\_login == master\_pwd:

print("")

while True:

print("1-Show all username and passwords") #Password recovery option with teacher login

print("2-Enter username")

print("3-Return menu")

choice = input("Choice:")

print("")

if choice == "1":

for user,details in users.items():

print(f"{user} : {details["pwd"]}")

elif choice == "2":

username = input("Username:")

if username in users:

print(f"{username}'s password: {users[username]["pwd"]}")

else:

print("Username not exists.")

elif choice == "3":

return

else:

print("Invalid choice.")

print("")

else:

print("Wrong password.")

def calculate\_score(users,username): #User score calculation with for loop in user data

users[username]["score"]["Total Score"] = sum(values-1 for game in users[username]["progress"].values() for subject in game.values() for values in subject.values())

users[username]["score"]["Total progress %"] = round(((sum(values-1 for game in users[username]["progress"].values() for subject in game.values() for values in subject.values()))/175\*100),1)

users[username]["score"][game1] = sum(values-1 for subject in users[username]["progress"][game1].values() for values in subject.values()) #Collects all of game1 scores.

users[username]["score"][game2] = sum(values-1 for subject in users[username]["progress"][game2].values() for values in subject.values())

users[username]["score"][game3] = sum(values-1 for subject in users[username]["progress"][game3].values() for values in subject.values())

users[username]["score"][subject1] = sum(values-1 for game in users[username]["progress"].values() for values in game[subject1].values()) #Collects subject 1 scores in each game

users[username]["score"][subject2] = sum(values-1 for game in users[username]["progress"].values() for values in game[subject2].values())

users[username]["score"][subject3] = sum(values-1 for game in users[username]["progress"].values() for values in game[subject3].values())

users[username]["score"][subject4] = sum(values-1 for game in users[username]["progress"].values() for values in game[subject4].values())

users[username]["score"][subject5] = sum(values-1 for game in users[username]["progress"].values() for values in game[subject5].values())

users[username]["score"][easy] = sum(subject[easy]-1 for game in users[username]["progress"].values() for subject in game.values()) #Collects easy difficulty scores for each game and subject

users[username]["score"][normal] = sum(subject[normal]-1 for game in users[username]["progress"].values() for subject in game.values())

users[username]["score"][hard] = sum(subject[hard]-1 for game in users[username]["progress"].values() for subject in game.values())

#The reason for writing "values-1" in all of them is to store the level in user data.

#However, what we need here are not the level user on, the levels made before. That's why they all have "-1".

#All calculations are saved in the users dictionary

def choose\_difficulty(users,username,game\_choice,subject\_choice):

print("")

print(f"1-Easy - Progress: {users[username]["progress"][game\_choice][subject\_choice][easy]-1}/{max\_level}")

print(f"2-Normal - Progress: {users[username]["progress"][game\_choice][subject\_choice][normal]-1}/{max\_level}")

print(f"3-Hard - Progress: {users[username]["progress"][game\_choice][subject\_choice][hard]-1}/{max\_level}")

print(f"4-Go back")

while True:

difficulty\_choice = input("Choice:")

if difficulty\_choice == "1":

difficulty\_choice = easy

elif difficulty\_choice == "2":

difficulty\_choice = normal

elif difficulty\_choice == "3":

difficulty\_choice = hard

elif difficulty\_choice == "4":

return(difficulty\_choice)

else:

print("Invalid option.")

continue

if users[username]["progress"][game\_choice][subject\_choice][difficulty\_choice]-1 == max\_level:

print("You already completed this difficulty. Please choose another difficulty or go back.")

continue

break

return(difficulty\_choice)

def choose\_subject(users,username,game\_choice):

if sum(values-1 for subject in users[username]["progress"][game\_choice].values() for values in subject.values()) == max\_total\_level\_for\_each\_game:

print("You already completed this game. Please choose another game or exit.")

subject\_choice = "6"

return(subject\_choice)

print("")

print(f"1-{subject1} - Progress: {sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject1].values())}/{max\_total\_level\_for\_each\_subject}")

print(f"2-{subject2} - Progress: {sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject2].values())}/{max\_total\_level\_for\_each\_subject}")

print(f"3-{subject3} - Progress: {sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject3].values())}/{max\_total\_level\_for\_each\_subject}")

print(f"4-{subject4} - Progress: {sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject4].values())}/{max\_total\_level\_for\_each\_subject}")

print(f"5-{subject5} - Progress: {sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject5].values())}/{max\_total\_level\_for\_each\_subject}")

print(f"6-Go back")

subject\_choice = input("Choice:")

while True:

if subject\_choice == "1":

subject\_choice = subject1

elif subject\_choice == "2":

subject\_choice = subject2

elif subject\_choice == "3":

subject\_choice = subject3

elif subject\_choice == "4":

subject\_choice = subject4

elif subject\_choice == "5":

subject\_choice = subject5

elif subject\_choice == "6":

return(subject\_choice)

else:

subject\_choice = input("Invalid choice. Type again:")

continue

if sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject\_choice].values()) == max\_total\_level\_for\_each\_subject:

print("You completed this subject already. Choose another one or press 6 to go back.")

subject\_choice = input("Choice:")

else:

return(subject\_choice)

def game1\_word\_guess(word,definition): #The game functions are short because I have done common operations with the help of functions.

print(f"\nDefinition: {definition}") #I gave only "word", "definition" variables to these 3 game functions and made them return True or False with "answer" variable.

print("Guess the word")

attempt = 0

while True:

guess = input("Word:")

if guess.lower() == word.lower():

print("\nCorrect!")

answer = True

return(answer)

else:

print("\nWrong. Guess again.")

attempt += 1

if attempt == 2: #On 2 incorrect attempts I gave the first letter of the word as a clue

print(f"First letter: {word[0]}")

elif attempt == 4:

print(word[0], end=" ") #In 4 incorrect attempts I gave the first letter and length of the word as a clue

for letter in range(len(word)-1):

print('\_', end=' ')

print(" ")

elif attempt == 6: #6 incorrect attempts, the right of error expires and has to start all over again

print(f"\nWord was: {word}") #All these 3 numbers can be changed to make the game easier or harder.

answer = False

return(answer)

def game2\_word\_match(word,definition): #Gives the user 5 words and 1 definition and asks for the word belonging to the given definition

print(f"Definition: {definition}")

print("Guess the word")

attempt = 0

while True:

guess = input("Word:")

if guess.lower() == word.lower():

print("\nCorrect!")

answer = True

return(answer)

else:

attempt += 1

if attempt == 2: #2 incorrect attempts, the right of error expires and has to start all over again

print(f"\nWord was: {word}")

answer = False

return(answer)

print("\nWrong. Guess again.")

def game3\_letter\_hunt(word,definition): #It gives the user the length of the word and ask a letter guess

word = word.lower()

wrong\_attempt = 0

letters = []

wrong\_letters = []

print(f"\nDefinition: {definition}\n")

print("Word:", end=" ")

for i in word:

if i == " ":

print(" ", end=" ")

else:

print("\_", end=" ")

while True:

answer = True

letter = input("\nGuess letter:").lower()

while letter == "":

letter = input("\nInvalid letter.\nGuess letter:")

if letter in letters:

print(f"You already type {letter}")

continue

else:

letters.append(letter)

if letter == word:

return(answer)

if letter not in word:

wrong\_letters.append(letter)

wrong\_attempt += 1

answer = False

if wrong\_attempt > 7: #7 incorrect attempts, the right of error expires and has to start all over again

print(f"Word was: {word}")

return(answer)

print(f"\nRemaining attempt: {8-wrong\_attempt}")

print("Non-existent letters: ", end=' ')

print(\*wrong\_letters)

for i in word:

if i in letters or i == " ":

print(i, end=' ')

else:

print('\_', end=' ')

answer = False

if answer == True:

print("\nCorrect!")

print(f"Word is {word}\n")

break

return(answer)

def game\_menu(users,username,game\_choice): #I made a common game\_menu function for 3 games because I needed to select the subject and difficulty before each game.

#Also, according to this data, I learnt the progress of the related level from user data and accessed the word-definition pair related to that level from word dictionary.

return\_selection = "Choose subject"

while return\_selection == "Choose subject":

subject\_choice = choose\_subject(users,username,game\_choice)

if subject\_choice == "6":

return()

return\_selection = "Choose difficulty"

while return\_selection == "Choose difficulty":

difficulty\_choice = choose\_difficulty(users,username,game\_choice,subject\_choice)

if difficulty\_choice == "4":

return\_selection = "Choose subject"

continue

return\_selection = "1"

while return\_selection == "1":

level = users[username]["progress"][game\_choice][subject\_choice][difficulty\_choice]

#It takes the level of the relevant difficulty level of the relevant subject of the relevant game of the relevant user stored in the user data and assigns it to the "level" variable.

#Because we will use it to get the words of the related level from the "worddict" dictionary.

print("")

print(f"Game: {game\_choice}")

print(f"Subject: {subject\_choice}")

print(f"Difficulty: {difficulty\_choice}")

print(f"Level: {level}")

question = 1

for word,definition in worddict[subject\_choice][difficulty\_choice][str(level)].items():

if game\_choice == game1:

answer = game1\_word\_guess(word,definition)

elif game\_choice == game2:

word\_list = list(worddict[subject\_choice][difficulty\_choice][str(level)].keys())

random.shuffle(word\_list)

print("\nWords: " + " - ".join(word\_list))

answer = game2\_word\_match(word,definition)

elif game\_choice == game3:

answer = game3\_letter\_hunt(word,definition)

if answer == False:

print("")

print("You didn't pass the level.")

print("1-Start over on the same level.") #Although it is difficult to control nested functions and while loops,

print("2-Return to the difficulty selection.") #I offered the user the opportunity to return to whichever menu he wants.

print("3-Return to the subject selection.")

print("4-Return to the main menu.")

return\_selection = input("Choice:")

while return\_selection not in ["1","2","3","4"]:

print("Invalid choice.Try again.")

return\_selection = input("Choice:")

if return\_selection == "1":

break

elif return\_selection == "2":

return\_selection = "Choose difficulty"

break

elif return\_selection == "3":

return\_selection = "Choose subject"

break

elif return\_selection == "4":

return\_selection = "Main menu"

return

if question < 5:

if answer == True:

answer = False

question += 1

print("\n1-New word")

print("2-Go back (Your progress will not be recorded because you have not completed 5 words at this level.)")

new\_question = input("Choice:")

while new\_question not in ["1","2"]:

new\_question = input("Invalid choice.Type again.Choice:")

if new\_question == "2":

return\_selection = "Choose difficulty"

break

if answer == True: #After exiting the previous for loop, the level is complete.

#At this stage, the aim is to save the progress in the user data and send the user to the next menu.

users[username]["progress"][game\_choice][subject\_choice][difficulty\_choice] += 1 #Increases the level in the recording by one level.

calculate\_score(users,username)

write\_userlist(users)

if sum(values-1 for game in users[username]["progress"].values() for subject in game.values() for values in subject.values()) == max\_total\_level:

#If the sum of all levels in the user data is equal to the max level, it means that the user has completed all games.

print("")

print("!!!!!!YOU FINISHED THE GAME!!!!!!")

print("")

break

elif sum(values-1 for subject in users[username]["progress"][game1].values() for values in subject.values()) == max\_total\_level\_for\_each\_game:

#If the sum of this game levels in the user data equals the maximum level, the user has completed this game.

print(f"Congratulations! You have completed {game\_choice}.")

break

elif sum(difficulty-1 for difficulty in users[username]["progress"][game\_choice][subject\_choice].values()) == max\_total\_level\_for\_each\_subject:

#If the sum of this subjects all difficulty levels in the user data equals the maximum level, the user has completed this subject.

print(f"Congratulations! You have completed {subject\_choice}")

return\_selection = "Choose subject" #In the previous function, this variable will be used to return to the subject selection.

break

elif users[username]["progress"][game\_choice][subject\_choice][difficulty\_choice] == 4:

#When the level kept in the user data passes level 3 and becomes 4, that difficulty is completed.

print(f"You have completed {subject\_choice} - {difficulty\_choice} difficulty.")

return\_selection = "Choose difficulty" #In the previous function, this variable will be used to return to the difficulty selection.

break

else: #If the above statements are not fulfilled, it means that there is still more level left in this difficulty level.

print("")

print(f"You have completed {level}. level.")

print("1-Start next level.")

print("2-Return to the difficulty selection.") #Thanks to the layered use of both function and while loops, it is possible to return to the desired part at any stage.

print("3-Return to the subject selection.")

print("4-Return to the main menu.")

return\_selection = input("Choice:")

while return\_selection not in ["1","2","3","4"]:

print("Invalid choice.Try again.")

return\_selection = input("Choice:")

if return\_selection == "1":

continue

elif return\_selection == "2":

return\_selection = "Choose difficulty"

break

elif return\_selection == "3":

return\_selection = "Choose subject"

break

elif return\_selection == "4":

return\_selection = "Main menu"

return

def progress(users,username): #Show progress function

print("\nTotal correctly recognised words: ", sum(values-1 for game in users[username]["progress"].values() for subject in game.values() for values in subject.values())\*5,"/875",sep="")

print(f"Your Total Score: {users[username]["score"]["Total Score"]}")

print(f"Your Total Progress: %{users[username]["score"]["Total progress %"]}")

total\_user\_ranking = sorted(users.items(), key=lambda user: user[1]["score"]["Total Score"], reverse=True) #Function that ranks all users by "Total Score"

for i, (user, data) in enumerate(total\_user\_ranking, 1): #I numbered everyone according to the order in the ranked score list

if user == username: #I learnt to use the enumerate function as I mentioned in Reference

print(f"You're ranked {i} out of everyone.")

while True:

print("")

print("Progress:")

print("1-Your detailed progress")

print("2-All list")

print("3-All detailed")

print("4-Go back")

choice = input("Choice:")

while choice not in ["1","2","3","4"]:

choice = input("Invalid. Try again. Choice:")

if choice == "1":

print("-----")

print(f"Your {game1} score: {users[username]["score"][game1]}")

print(f"Your {game2} score: {users[username]["score"][game2]}")

print(f"Your {game3} score: {users[username]["score"][game3]}")

print("---")

print(f"Your {subject1} score: {users[username]["score"][subject1]}")

print(f"Your {subject2} score: {users[username]["score"][subject2]}")

print(f"Your {subject3} score: {users[username]["score"][subject3]}")

print(f"Your {subject4} score: {users[username]["score"][subject4]}")

print(f"Your {subject5} score: {users[username]["score"][subject5]}")

print("---")

print(f"Your {easy} score: {users[username]["score"][easy]}")

print(f"Your {normal} score: {users[username]["score"][normal]}")

print(f"Your {hard} score: {users[username]["score"][hard]}")

print("-----")

elif choice == "2":

print("\nAll users scores:")

for i, (user, data) in enumerate(total\_user\_ranking, 1):

print(f"{i}. {user}: {data["score"]["Total Score"]}")

elif choice == "3": #Detailed game, subject and difficulty level based score lists

print("")

print("1-Ranking by games")

print("2-Ranking by subjects")

print("3-Ranking by difficulties")

print("4-Go back")

detailed\_choice = input("Choice:")

while detailed\_choice not in ["1","2","3","4"]:

detailed\_choice = input("Invalid. Try again. Choice:")

if detailed\_choice == "1": #It shows users according to the scores of the selected game.

print("")

print(f"1-{game1}")

print(f"2-{game2}")

print(f"3-{game3}")

print("4-Go back")

choice = input("Choice:")

while choice not in ["1","2","3","4"]:

choice = input("Invalid. Try again. Choice:")

if choice == "1":

game = game1

elif choice == "2":

game = game2

elif choice == "3":

game = game3

game\_ranking = sorted(users.items(), key=lambda user: user[1]["score"][game], reverse=True)

print(f"\n{game.capitalize()} ranking of all players:")

for i, (user, data) in enumerate(game\_ranking, 1):

print(f"{i}. {user}: {data["score"][game]}")

else:

continue

elif detailed\_choice == "2": #It shows users according to the scores of the selected subject.

print("")

print(f"1-{subject1}")

print(f"2-{subject2}")

print(f"3-{subject3}")

print(f"4-{subject4}")

print(f"5-{subject5}")

print("6-Go back")

choice = input("Choice:")

while choice not in ["1","2","3","4","5","6"]:

choice = input("Invalid. Try again. Choice:")

if choice == "1":

subject = subject1

elif choice == "2":

subject = subject2

elif choice == "3":

subject = subject3

elif choice == "4":

subject = subject4

elif choice == "5":

subject = subject5

else:

continue

subject\_ranking = sorted(users.items(), key=lambda user: user[1]["score"][subject], reverse=True)

print(f"\n{subject.capitalize()} ranking of all players:")

for i, (user, data) in enumerate(subject\_ranking, 1):

print(f"{i}. {user}: {data["score"][subject]}")

elif detailed\_choice == "3": #It shows users according to the scores of the selected difficulty

print("")

print(f"1-{easy}")

print(f"2-{normal}")

print(f"3-{hard}")

print("4-Go back")

choice = input("Choice:")

while choice not in ["1","2","3","4"]:

choice = input("Invalid. Try again. Choice:")

if choice == "1":

difficulty = easy

elif choice == "2":

difficulty = normal

elif choice == "3":

difficulty = hard

difficulty\_ranking = sorted(users.items(), key=lambda user: user[1]["score"][difficulty], reverse=True)

print(f"\n{difficulty.capitalize()} ranking of all players:")

for i, (user, data) in enumerate(difficulty\_ranking, 1):

print(f"{i}. {user}: {data["score"][difficulty]}")

else:

continue

else:

continue

else:

break

def change\_password(users,username):

password = input("New password:")

check\_password = input("Re-enter new password:")

while password != check\_password:

print("Passwords do not match. Type again.")

password = input("Password:")

check\_password = input("Re-enter password:")

users[username]["pwd"] = password

write\_userlist(users)

print("Password succesfully changed.")

return(users)

def user\_menu(users,username): #Main menu after the user logs in

while True:

print("")

print(f"1-{game1} - Progress: {sum(values-1 for subject in users[username]["progress"][game1].values() for values in subject.values())}/{max\_total\_level\_for\_each\_game}")

print(f"2-{game2} - Progress: {sum(values-1 for subject in users[username]["progress"][game2].values() for values in subject.values())}/{max\_total\_level\_for\_each\_game}")

print(f"3-{game3} - Progress: {sum(values-1 for subject in users[username]["progress"][game3].values() for values in subject.values())}/{max\_total\_level\_for\_each\_game}")

print(f"4-Show progress")

print(f"5-Change password")

print(f"6-Log out")

game\_choice = input("Choice:")

if game\_choice == "1":

game\_choice = game1 #I linked the game\_choice variable to game to simplify the code

game\_menu(users,username,game\_choice)

elif game\_choice == "2":

game\_choice = game2

game\_menu(users,username,game\_choice)

elif game\_choice == "3":

game\_choice = game3

game\_menu(users,username,game\_choice)

elif game\_choice == "4":

progress(users,username)

elif game\_choice == "5":

change\_password(users,username)

elif game\_choice == "6":

return(users)

elif game\_choice not in ["1","2","3","4","5","6"]:

print("Invalid choice.")

def main():

while True:

users = read\_userlist()

read\_dictionary()

print("")

print("1-Login")

print("2-New user")

print("3-Forgot password")

print("4-Exit")

choice = input("Choice:")

enter = False

if choice == "1":

username,enter = login(users)

elif choice == "2":

username,users,enter = register(users)

elif choice == "3":

forgot\_password(users)

elif choice == "4":

exit()

else:

print("Invalid choice.\n")

if enter == True:

user\_menu(users,username)

calculate\_score(users,username)

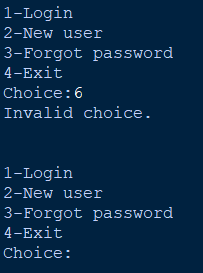
write\_userlist(users)

main()

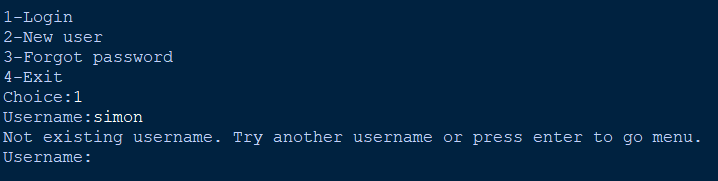
# Section 5: Testing

|  |  |  |
| --- | --- | --- |
| **Purpose of test** | **How the test was carried out** | **Outcome from that test** |
| **Validate user input for main menu choice**  Ensure the program does not accept invalid option. | Invalid option tried as input from user | Program warned the user that they had entered invalid and asked them to type again.  **(Figure 1.)** |
| **Validate username input for login**  Ensure the program does not accept non-existent username. | Non-existent username tries as input from user | Program warned the user that they had entered non-existent and asked them to write again or go back to create a new user.  **(Figure 2.)** |
| **Check user password input for login**  Ensure that the entered input and the user's registered password are the same | Incorrect password attempted | The programme warned the user. It offered the option to try again or go back.  **(Figure 3.)** |
| **Validate user input for game choice**  Ensure the program does not accept invalid option. | Invalid option tried as input from user | Program warned the user that they had entered invalid and asked them to type again.  **(Figure 4.)** |
| **Validate user input for subject choice**  Ensure the program does not accept invalid option. | Invalid option tried as input from user | Program warned the user that they had entered invalid and asked them to type again.  **(Figure 5.)** |
| **Validate user input for difficulty choice**  Ensure the program does not accept invalid option. | Invalid option tried as input from user | Program warned the user that they had entered invalid and asked them to type again.  **(Figure 6.)** |
| **Validate that the programme works properly when you guess the wrong word in Game1**  Ensure that the programme does not accept a wrong guess as correct. | Wrong guess tried as input from user. | The game proceeded as expected and ended when the predetermined number of wrong attempts had been completed.  **(Figure 7.)** |
| **Validate the options that come up when the attempt is over, when user guess the wrong word in Game1**  Ensure the program does not accept invalid option. | Invalid option tried as input from user | Program warned the user that they had entered invalid and asked them to type again.  **(Figure 8.)** |
| **Validate that the programme works properly when you guess the wrong word in Game2**  Ensure that the programme does not accept a wrong guess as correct. | Wrong guess tried as input from user. | The game proceeded as expected and ended when the predetermined number of wrong attempts had been completed.  **(Figure 9.)** |
| **Validate that the programme does not re-accept as input the letter that the user has already tried in Game3**  Ensure that the programme does not accept a same letter as new attempt. | A previously tried letter was tried again. | The programme tells the user that the letter has been tried before.  **(Figure 10.)** |
| **Validate that the programme does not give any error when user tried wrong letters.**  Ensure that the programme does not accept a wrong letter as correct letter. | Wrong guess tried as input from user. | The game proceeded as expected and ended when the predetermined number of wrong attempts had been completed.  **(Figure 11.)** |
| **Validate that the programme save user progress on CSV after completing each level.**  Ensure that the app retains progress even if the user closes the app after completing the level without logging out of their account. | The application was closed after the level was completed. | It was verified that the progress was recorded on the CSV. |

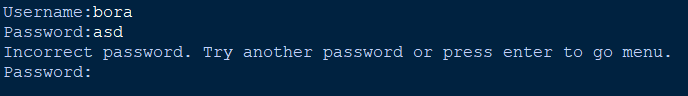
**Figure 1.**



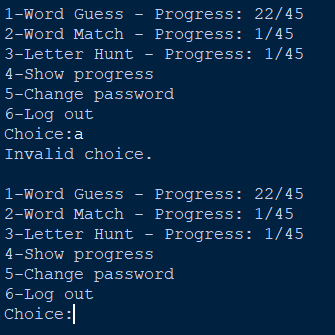
**Figure 2.**



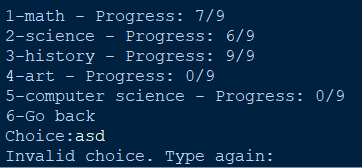
**Figure 3.**



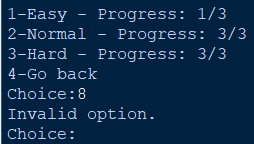
**Figure 4.**



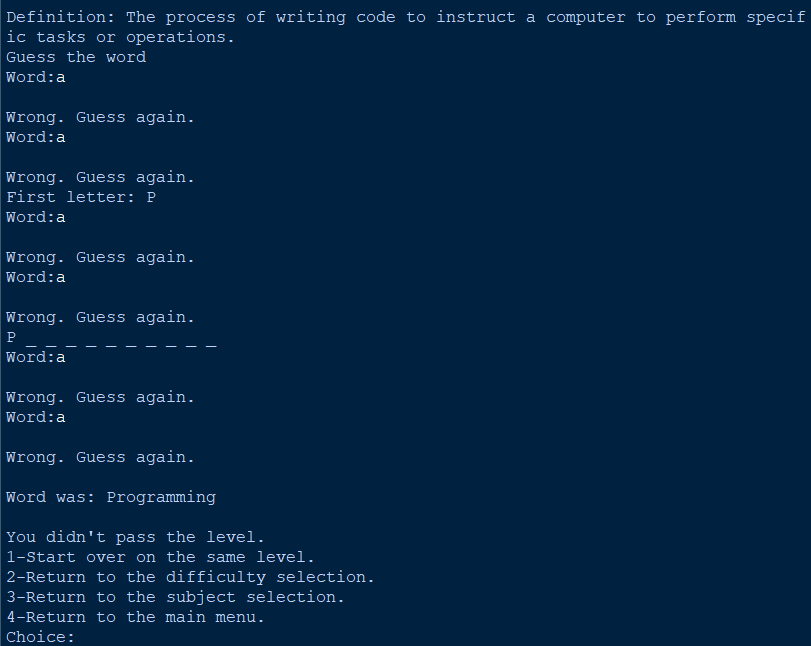
**Figure 5.**



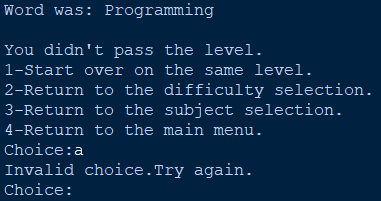
**Figure 6.**



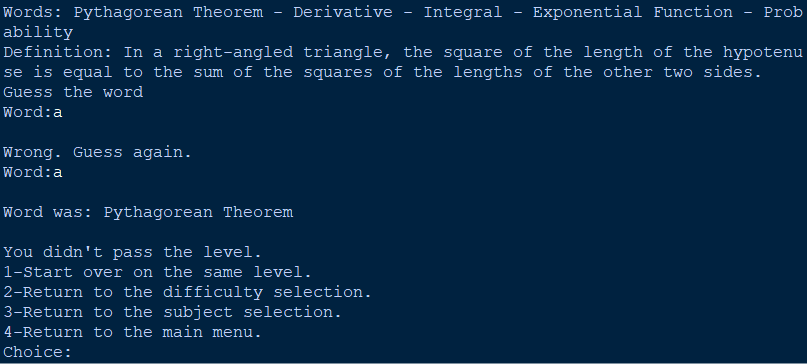
**Figure 7.**



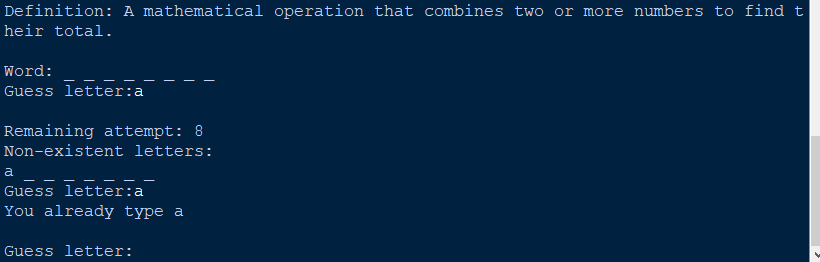
**Figure 8.**



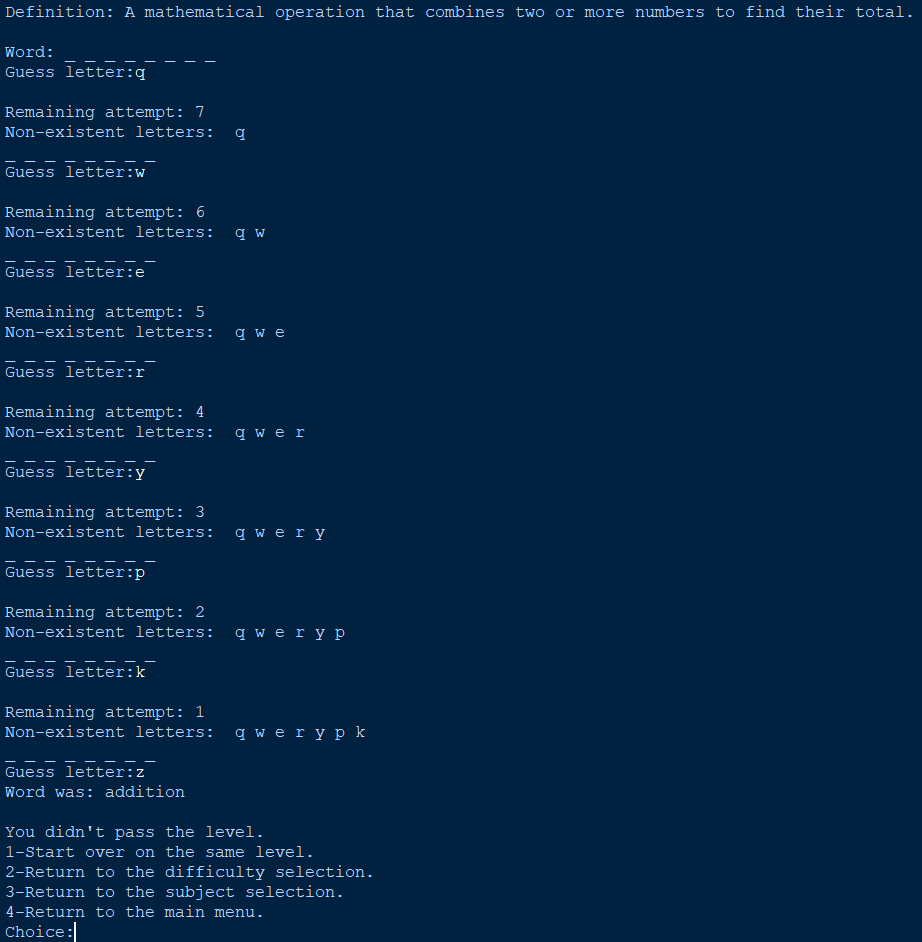
**Figure 9.**



**Figure 10.**



**Figure 11.**

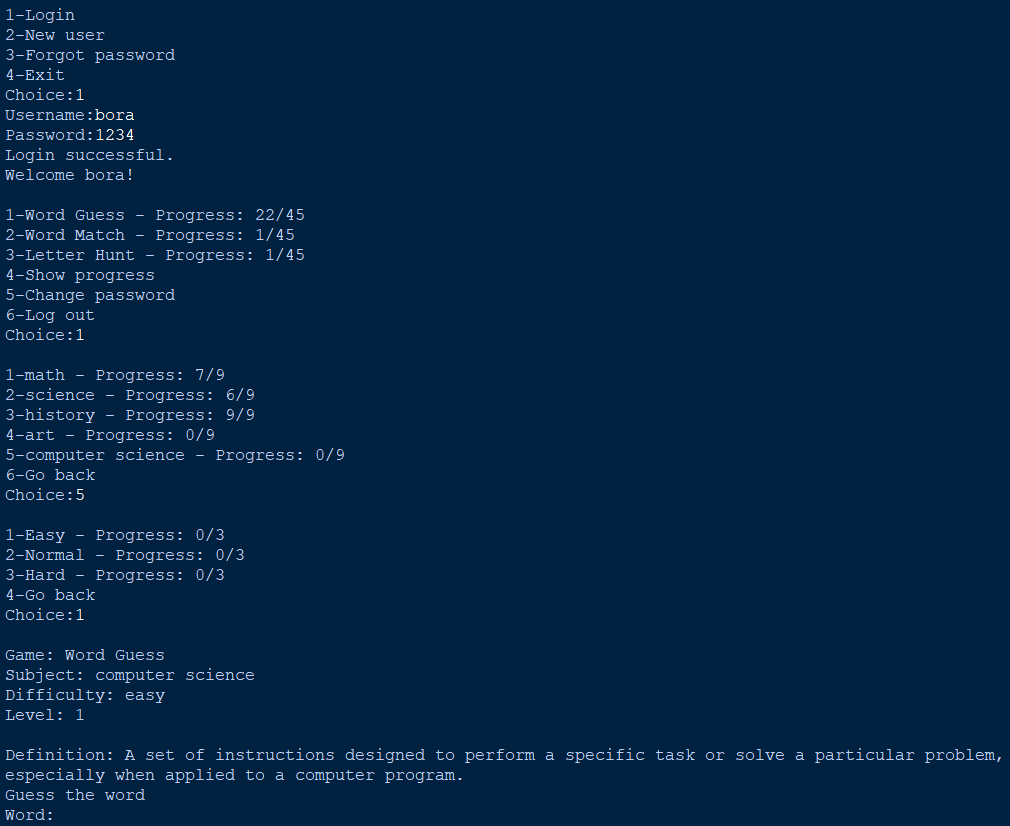


# Section 6: Evaluation

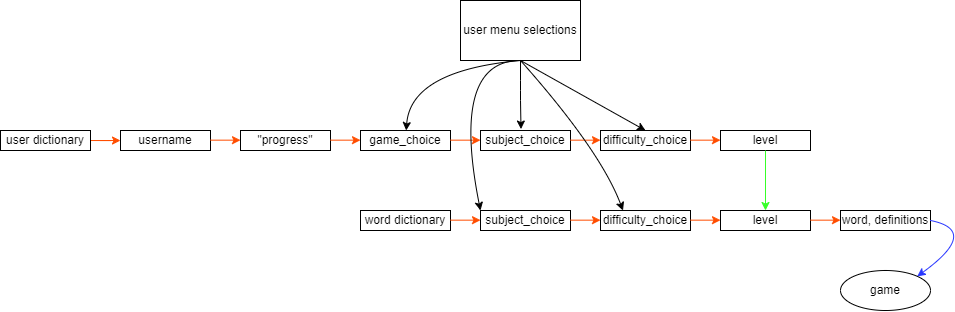
The program meets all specified criteria and effectively achieves its goal without any detected error.

The application retrieves user data and progress from the CSV file. It also retrieves word-definitions from the CSV file. Login, register, forgot password section welcomes the user. After registering or logging in, the user can choose one of the **3 games** or **view the progress** or **change the password**. According to the user's game selection, the subject selection appears. You can also see your progress in each subject on this screen. After the subject selection, the **difficulty selection** section comes, in the same way, after selecting the difficulty level, the word game starts to be played according to the level of the user.

The figure shows the process from running the programme to starting the game.

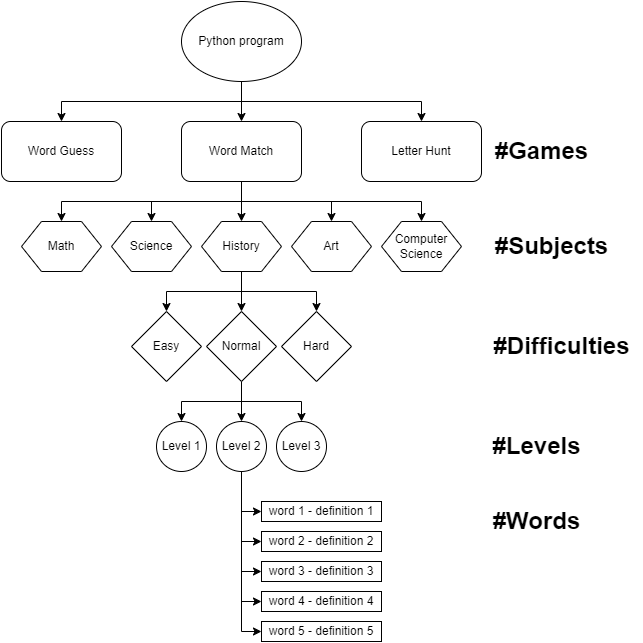


The application successfully matches the level of the relevant difficulty level of the relevant subject of the user's game in the user data with the relevant word-definition pairs in the word dictionary. In the following image, the way of working can be understood more clearly.



Each level has 5 word-definition pairs. After successfully guessing these 5 words, the user's level is increased. Care has been taken to use a user-friendly interface to ensure easy use by younger students.The application checks for invalid entries at every possible stage, in every question.

A summary of the overall structure of the programme can be seen below.



To summarise, this application makes the learning process easier, interesting and fun by providing different games for a student to learn new vocabulary in different subjects. It increases the learning experience. It also motivates the student by keeping the user's progress. In addition, it also provides friendly competition by comparing its own progress with other students in the show progress section. Since it has a user account system, it allows different people to play from the same device and keep their personal progress. It offers 3 different difficulty options as easy, normal, and hard in every subject of each game according to the class and age of the student.

You can find my video presentation explaining the general structure of the code at this link.

Link: <https://drive.google.com/drive/folders/1kIxpN822UAB4MwovZp_vMPRlUhzz_7Ub?usp=sharing>

# Section 7: References

1-Used to understand working mechanism of sorted() and lambda function:

*Python program to sort the list according to the column using Lambda* (2023) *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/python-program-to-sort-the-list-according-to-the-column-using-lambda/>

2-Used to understand working mechanism of Python built-in function enumerate():

Enumerate() in Python (2023) *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/enumerate-in-python/>

3-Used ChatGPT to get help in finding and organising a total of 225 words, 45 words in each subject, and their meanings in the format I wanted to use in the game.

<https://chat.openai.com/share/89809c73-b407-433c-8875-2048e38cb19e>

(The language of communication is Turkish, not English.)