Project Documentation

The main Functions:

1- **Perform the functions of the simple calculator, which are**:
   - Addition
   - Subtraction
   - Multiplication
   - Division
   - Modulus
2- **Perform the functions of the scientific calculator, which are**:
   - Get the square root of a real number.
   - Get the cubic root of a real number.
   - Get the absolute value of a number.
   - e to the power of a number x.
   - Prime factorization of real integer.
   - Factorial of a real integer.
   - Log of a real number.
   - Natural log of a real number.
   - Get the result of real number raised to real number.
   - Sin of a real angle.
   - Cos of a real angle.
   - Tan of a real angle.
   - Shift sin, cos, tan.
   - Solve first degree equation of the form $y = ax + c.$
   - Solve second degree equations of the form: $y = ax^2 + bx + c.$
   - Summation of many numbers.
   - Multiplication of many numbers.
3- **Perform the encoding and decoding of a given word**.
4- **Display the data stored in the file**.
5- **Clear the data from the file**.

Notes:

- All those functions will be done in the terminal.
- The libraries used are only: stdio.h, stdlib.h, math.h, string.h.
- For the file handling part, I applied it in the part of encoding and decoding words. In addition, the history of the mathematical operations done while using the program is been saved to the file.
- At first, there will be a menu from which the user must choose what function he wants to perform.
- After choosing from the three main functions which are simple calculator, scientific calculator, and encoding and decoding, there will be a new menu containing what function can be done in that part, which are mentioned above.
- In each menu, there will be an option that the user can end the program at any time he wants to do this. In addition, there will be an option to get him back to the main menu.
- The user will be asked to choose an option and write it in the terminal.
- I declared a structure called xy which contains two double numbers x and y to ease the use of the function that requires to parameters.
- **Inside the main.** there is only the line for opening or creating the file file.txt and the line for calling the function mainoptions which is responsible for performing the whole tasks for the program.

Functions:

1- choose_calc():

- It displays the options, which are 1: Simple Calculator 2: Scientific Calculator 3: Encoding and decoding of a word 4: Printing File Content (the history) 5: Clear all the file content(history) 0: End the program.
- The user will be asked to enter a value from 0 to 5, then the function returns that value.

2- get_option1():

- That function will display the options for the simple calculator which are the 5 main mathematical operations in addition to the option of ending the program.
- The user will be asked to choose one and that will be the value that the function returns.
- If the user chooses a value which is not one of these options, the user will be asked to enter a correct value.

3- get_option2():

- That function will display the options for the scientific calculator in addition to the option of ending the program.
- The user will be asked to choose one and that will be the value that the function returns.
- If the user chooses a value which is not one of these options, the user will be asked to enter a correct value.

4- getxy():

- It is called getxy which means get x and y. x and y are the elements of the defined structure before it. That function declares a struct of type xy and gets x and y from the user and return the structure xy.

5- getint():

- The function asks the user to enter an integer and returns it.

6- getdouble():

- The function asks the user to enter a double and returns it.

7- addition():

- It uses the function getxy to get the two numbers x and y. then it performs the addition of two numbers x and y and returning the sum of them called z.

8- subtraction():

- It uses the function getxy to get the two numbers x and y. then it performs the Subtraction of two numbers x and y and returning the result of them called z.

9- multiplication():

- It uses the function getxy to get the two numbers x and y. then it performs the multiplicatoion of two numbers and returns the product.

10- division():

- It uses the function getxy to get the two numbers x and y. then it performs the division of two numbers and returns the result.

11- modulus():

- It uses the function getxy to get the two numbers x and y. then it performs the modulus of two numbers and returns the result.
- In that function I used the built-in remainder function. There was a problem as that the remainder function returns negative value if the first number was less than the second one. To solve such bug as the function is called modulus not remainder and the modulus can not be negative value, I added an if condition to check if the resulted value was negative,

if it was the case, it will make the result = result + second number (y).

12- squareroot():

- it uses the function getdouble which return a double number. Then it performs the sqrt function and returns the result. If the entered number was negative it will by default return -nan.

13- cubicroot():

- it uses the function getdouble which return a double number. Then it performs the cubic root function and returns the result.

14- absolute():

- it uses the function getdouble which return a double number. Then it performs the absolute function and returns the result.

15- epowerx():

- it uses the function getdouble which return a double number. Then it raise e to the power of x and returns the result.

16- factorial():

- it uses the function getint which return an int number. It performs the factorial for that number.
- The number must be less than 21 that is because int cannot store more than 2147483647. That problem can be solved using arrays or linked list, I applied that logic in another task.
- That function returns the factorial of the number entered.

17- primeFactor():

- that function uses the function getint then perfoms prime factorization with the same way we learnt in the practice sessions. Then it prints every step to the terminal.

18- logg():

- it uses the function getdouble then perform the log to it and print the result in the console.
- Before performing the operation, the function makes sure first that the number is bigger than or equal to 0.

19- lnn():

- it uses the function getdouble then perform the natural log to it and print the result in the console.
- Before performing the operation, the function makes sure first that the number is bigger than or equal to 0.

20- power(xy xy1):

- it is clear that the paremeter of that function is a structure xy. That function gets a structure xy which contains x and y double, it raise x to the power of y and returns the result.

21- sinn(double x):

- it is *clear that the paremeter of that function is a double x*. At first it converts that angle from degrees to radians, as the built-in function sin deals with radians not degrees. Then it returns the sin of that angle.

22- coss(double x):

- It is clear that the paremeter of that function is a double x. At first it converts that angle from degrees to radians, as the built-in function cos deals with radians not degrees. Then it returns the cos of that angle.

23- tann(double x):

- it is clear that the paremeter of that function is a double x. At first it converts that angle from degrees to radians, as the built-in function tan deals with radians not degrees. Then it returns the cos of that angle.

24- angleS():

- It uses the function getdouble at first, then it make sure that the number entered is in the range of the inverse sin or not, which is between -1 and 1 included, then it performs the function of inverse sin and print the angle in radians and degrees in the console.

25- angleC():

- It uses the function getdouble at first, then it make sure that the number entered is in the range of the inverse cos or not, which is between -1 and 1 included, then it performs the function of inverse cos and print the angle in radians and degrees in the console.

26- angleT():

- It uses the function getdouble at first, then it performs the function of inverse tangent and print the angle in radians and degrees in the console.

27- feq():

- it solves the first-degree equation in the form Y = aX + c. the user is asked to enter both a and c. Then using mathematical manipulation, it gives the value of x where y = 0 and print it to the console.

28- seq():

- it solves second degree equations in the form Y = aX^2 + bx + c. the user is asked to enter both a and c. Then using mathematical manipulation, it gives the value of x where y = 0 and print it to the console.
- Here we have three main cases; all of them depends on the determinant which is b^2 – 4ac.

29- sum_of_many_numbers():

- the user is asked to enter a long line of numbers, at every successful entry of a *number the size of the pointer is increased by one until the user enters 0*,

using the same way introduced in the lecture- and the sum variable is increased by the entry number. At the end the sum is printed, and the memory allocated is released.

30- multiplication_of_manynumbers():

- the user is asked to enter a long line of numbers, at every successful entry of a number the size of the pointer is increased by one until the user enters 1, using the same way introduced in the lecture- and the product variable is multiplied by the entry number. At the end the product is printed, and the memory allocated is released.

31- printing_fileContent():

- it prints the content of the file. The history of the operations done.

32- simple_calc():

- it uses the function getoption1 and depends on the return value of the function getoption1 the function performs specific function using the switch loop and it ends when the user decide to end the program or go back to the main menu.

33- scientific_calc():

- it uses the function getoption2 and depends on the return value of the function getoption2 the function performs specific function using the switch loop and it ends when the user decide to end the program or go back to the main menu.

34- encode_decode():

- At first I save a memory location of 200 character for the stirring getting from the user and the same size for the encoded string and the decoded one (same as original).
- That function gets a string from the user and then prints the encoded string and the original one.

- The idea behind encoding is that for every character in the string before y, the function changes it to character + 2. For example, a or A will be c.
- For the characters y, z or Y and Z will be changed to character -24. So that y will be a and z will be 2.
- Then doing the reverse operation to get the original string back.
- At the end of the function, the memory location to be freed.

35- mainoptions():

- That is the main function of the program, which uses the function choosecalc return, depending on the return value of choosecalc function, the switch start working.