

CS1701 / CS1803

GROUP PROJECT

ASSIGNMENT 0: FORMATIVE TASK

Green 48

Tutor – Dr Nour Ali

Olivia Van Herteryck - 1840061

Jaskirat Singh - 1906151

Gabriel Karkafi - 2011904

Ahelal Iqbal - 2018227

Ahmed Alkhazraji - 2045016

Asmahaan Abubakar Ali – 1914189

Introduction

The purpose of this assignment is to build on teamwork and coding skills in order to successfully build and run a functioning calculator with eight specific operations. Not only does this task build on Assignment 1, but it will be the building block for the coming assignments.

Requirements Specification

Due to unforeseen circumstances, one member in our group left. This meant that the distribution of the tasks is not equal. However, the other group members made it up by offering help and support to complete the remaining tasks and report as well as giving recommendations to improve the work. Our group has worked harmoniously together to ensure the work produced was to the highest standard and have been excellent with meeting team deadlines.

In order to do this efficiently, each task was assigned to a group member:

* Task 1 – Addition, subtraction, multiplication and division of two numbers - **Gabriel**
* Task 2 – Power, modulo and square/cube roots - **Olivia**
* Task 3 – Conversions between temperature units (Celsius/Fahrenheit) - **Gabriel**
* Task 4 – Conversions between Metric and Imperial units for mass (g/oz) and length (cm/inches) - **Ahelal**
* Task 5 – Conversion of a decimal number to binary - **Jaskirat**
* Task 6 – Conversion of a binary number to decimal - **Ahmed**
* Task 7 – Conversion of a decimal number to hexadecimal - **Asmahaan**
* Task 8 – Conversion of a hexadecimal number to decimal - **Asmahaan**

Alongside this, we also had to:

* Display the list of operations and allow the user to choose which operation they would like to perform
* Ask the user to input the number(s)
* Perform the calculation and display the result
* Ask the user if they want to continue and if they select “Y”, it should continue, otherwise if its “N” then the program should exit.

Algorithm Design

**Menu**

**A picture containing text

Description automatically generated**

**Task 1**

**Diagram

Description automatically generatedFlowchart**

**Pseudocode**

Algorithm 1. Task\_1\_Operators (num1, num2, z, operator, answer)

Input: num1, num2, answer are double, z and operator are strings

1. Print “Enter your first number: ”
2. Print “Enter your operator (+, -, \*, /): ”
3. Switch (operator)
4. Case “+”
5. “Enter your second number: ”
6. Set answer = num1 + num2
7. Case “-”
8. Print “Enter your second number: ”
9. Set answer = num1 – num2
10. Case “\*”
11. Print “Enter your second number: ”
12. Set answer = num1 \* num2
13. Case “/”
14. Print “Enter your second number: ”
15. Set answer = num1 / num2
16. Print answer
17. Print “Type 'yes' to continue, ‘no’ to go to menu, ‘exit’ to exit”
18. Switch (final1c)
19. Case “yes”
20. Go to task 1
21. Case “no”
22. Go to menu
23. Case “exit”
24. Terminate program

Output: answer

**Task 2**

**Flowchart**

A picture containing diagram

Description automatically generated

**Pseudocode**

1. Print "choose a function"

2. Switch Case

3. Case 1:

4. Input int DIVISION

5. Input int DIVISOR

6. Int REM = DIVISION % DIVISOR

7. Print "DIVISION % DIVISOR = REM"

8. Case 2:

9. Input Double SQUARE

10. Double SQUAREROOT = square root of SQUARE

11. Print "the square root of number is ROOT"

12 Case 3:

13. Input Doule SQUARERT

14. Double SQR = SQUARERT \* SQUARERT

15. Print "the square of SQUARERT is SQR"

16. Case 4:

17. Return to Main

**Task 3**

**Diagram

Description automatically generatedFlowchart**

**Pseudocode**

Algorithm 3. Task\_3\_Temp\_Converter (celsius, fahrenheit, z, temp, answer)

Input: celsius, fahrenheit and answer are double, z is a string, temp is an integer

1. Print “Enter 1 to convert from Celsius to Fahrenheit”
2. Print “Enter 2 to convert from Fahrenheit to Celsius”
3. Switch (temp)
4. Case 1
5. Print “Enter Celsius value: ”
6. Set answer = (celsius \* 9/5) + 32
7. Case 2
8. Print “Enter Fahrenheit value: ”
9. Set answer = (fahrenheit - 32) \* 5/9
10. Print answer
11. Print “Type 'yes' to continue, ‘no’ to go to menu, ‘exit’ to exit”
12. Switch (final3c)
13. Case “yes”
14. Go to task 3
15. Case “no”
16. Go to menu
17. Case “exit”
18. Terminate program

Output: answer

**Task 4**

**Flowchart**

**Diagram

Description automatically generated**

**Pseudocode**

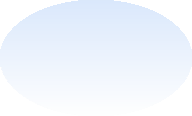
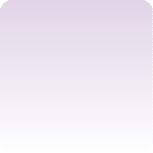
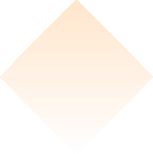
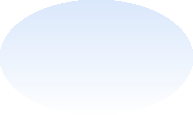
Algorithm 4: task4

Input: integer

1. Switch (sdkjfds)
2. Case 1
3. Goto cm to in
4. Output “Enter the number of inches”
5. Input number of inches
6. CM = inches \* 2.54
7. Output CM
8. Case 2
9. Goto in to cm
10. Output “Enter the number of CM”
11. Input number of CM
12. Inches = CM / 2.54
13. Output inches
14. Case 3
15. Goto g to oz
16. Output “Enter the number of Grams”
17. Input the number of grams
18. Ounce = Grams / 28.35
19. Output Ounce
20. Case 4
21. Goto oz to g
22. Output “Enter the number of Ounce”
23. Input the number of ounce
24. Grams = Ounce \* 28.35
25. Output Grams
26. Output “want to convert again?”
27. Switch (final4c)
28. Case “yes”
29. Goto start
30. Case “no”
31. Goto menu
32. Case “exit”
33. Terminate program
34. End case

**Task 5**

**Flowchart**



Start

Get an integer number from user

False

number !=0

Yes

True

Print binary

number

binary[i] = number%2

number = number/2

Would you like to convert another number?

No

i=i+1

Call the menu

Exit

Stop

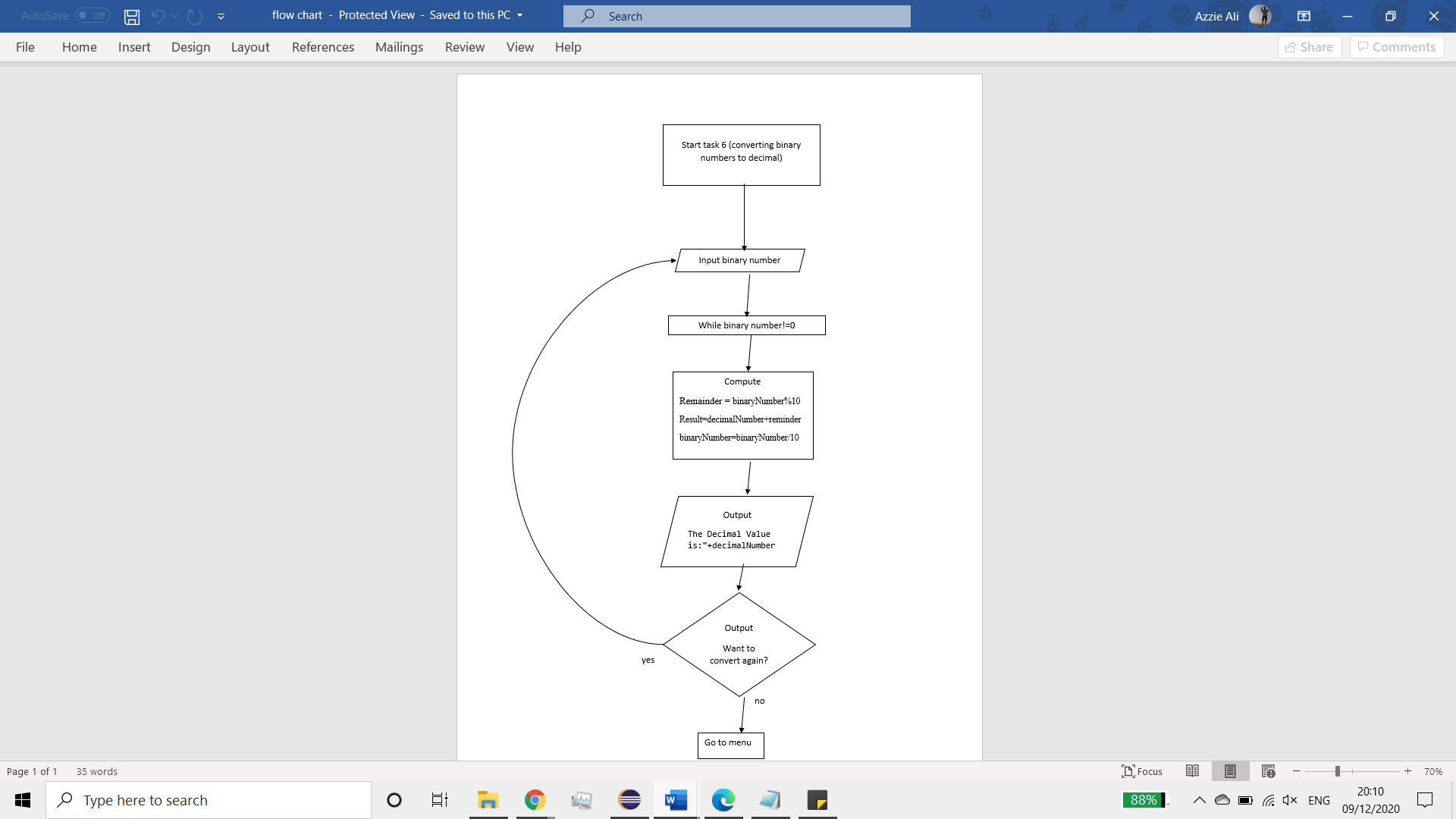
**Pseudocode**

Algorithm 5 – dectobinary (decimalnumber)

1. input decimal number
2. while decimalnumber <0 then
3. decimalnumber %2
4. store remainder in array (holds binary number)
5. decimalnunber=decimalnumber/2
6. repeat until value – equal to 0
7. Output result
8. Output “want to convert again”
9. Switch (final5c)
10. Case “yes”
11. Goto start
12. Case “no”
13. Goto menu
14. Case “exit”
15. Terminate program
16. End case

**Task 6**

**Flowchart**



**Pseudocode**

Algorithm 6 – binarytodec

Input – integer

1. While binaryvalue!=0
2. Repeat
3. Compute
4. Reminder=binaryNumber%10
5. Result=decimalNumber+reminder
6. binaryNumber=binaryNumber/10
7. decimalNumber=result+decimalNumber
8. Output result
9. Output “Do you want to convert again?”
10. Switch
11. Case “yes”
12. Goto start
13. Case “no”
14. Goto menu
15. Case “exit”
16. Terminate program
17. End case

**Task 7**

**A picture containing timeline

Description automatically generatedFlowchart**

**Pseudocode**

Algorithm 7 – task7 (decimalvalue)

Input: integer

1. while decimalvalue>0 then
2. repeat
3. compute
4. remainder = decimalvalue%16
5. result=hex[remainder]+result
6. decimalvalue=decimalvalue/16
7. Output result
8. Output “want to convert again?”
9. Switch (final7c)
10. Case “yes”
11. Goto start
12. Case “no”
13. Goto menu
14. Case “exit”
15. Terminate program
16. End case

**Task 8**

**Flowchart**

**A picture containing diagram

Description automatically generated**

**Pseudocode**

Algorithm 8. hextodec (hexvalue)

Input: a string hexvalue

1.       Get hex

2.       Compute

3.                       Decimal = 0

4.                       For i=0 to hexvalue.length()

5.                                       Char storevalue =hexvalue.charAt(i);

6.                                       Int I = hex.indexOf(storevalue)

7.                                       Decimal = 16\*decimal+in

8.       Output “The decimal value is: “ + decimal

9.       Output “want to convert again?”

10.                   Switch (final8c)

11.                                   Case “yes”

12.                                                   Goto start

13.                                   Case “no”

14.                                                   Goto menu

15.                                   Case “exit”

16.                                                   Terminate program

17.                                   End case