**LAB 9: ITERATIONS IN C LANGAUGE**

**MCT-242L: Computer Programming-I Lab (Fall-2022)**

Registration No. **2022mc58**

**OBJECTIVE:**

This lab will introduce loops available in C Language Program. At the end of this lab, you should be able to:

* Understand ***while***, ***for*** and ***do***…***while*** loops in a C language program
* Implement ***while***, ***for*** and ***do***…***while*** to comprehend the concepts

**APPARATUS:**

* Laptop\PC with following tools installed
  + Visual Studio Code with **C/C++** and **Code Runner Extensions**
  + C/C++ mingw-w64 tools for Windows 10

**ITRATION STATEMENTS**

You may encounter situations, when a block of code needs to be executed several numbers of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.

Programming languages provide various control structures that allow for more complicated execution paths. A *loop* statement allows us to execute a statement or group of statements multiple times. C programming language provides the following types of loops to handle looping requirements.

C programming language provides the following types of loops to handle looping requirements:

|  |  |
| --- | --- |
| **Loop Type** | **Description** |
| ***while*** loop | Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body. |
| ***for*** loop | Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. |
| ***do*…*while*** loop | It is more like a ***while*** statement, except that it tests the condition at the end of the loop body. |
| ***nested*** loops | You can use one or more loops inside any other ***while***, ***for***, or ***do*...*while*** loop. |

***while* Loop**

A ***while*** loop in C programming repeatedly executes a target statement if a given condition is true. The syntax of a while loop in C programming language is given below:

|  |
| --- |
| while(condition)  {      statements(s);  } |
| **Syntax of *while* Loop** |

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any nonzero value. The loop iterates while the condition is true. When the condition becomes false, the program control passes to the line immediately following the loop.

Here, the key point to note is that a ***while*** loop might not execute at all. When the condition is tested and the result is false, the loop body will be skipped and the first statement after the ***while*** loop will be executed.

|  |  |
| --- | --- |
| **Example 9.1: *while* loop in C** | |
| /\* Example\_9\_1.c: while loop in C  ---------------------------------------------------------------------------  This program demonstrates the use of while loop in C. It takes a number  as input form user and displays values on screen from 0 to given number -1.  ---------------------------------------------------------------------------  Written by Shujat Ali (engrshujatali@gmail.com) on 4-Oct-2021.  IDE: Visual Studio Code 1.60.0  C Compiler: GCC (Rev. 5, Built by MSYS2 Project) 10.3.0 \*/  #include <stdio.h>  void while\_loop(int num); int main()  {  int num = 0;  printf("Enter a number >> ");   scanf("%d", &num);   // Call the while\_loop() function  while\_loop(num);   return 0; }  void while\_loop(int num)  { //Prints the values of the loop control variable from 0 to num - 1.  int i = 0;  while (i < num)   {printf("The value of loop control variable is %d\n", i++);} } // End of program | |
| **Program Output 1** | Enter a number >> 5  The value of loop control variable is 0  The value of loop control variable is 1  The value of loop control variable is 2  The value of loop control variable is 3  The value of loop control variable is 4 |
| **Program Output 2** | Enter a number >> 2  The value of loop control variable is 0  The value of loop control variable is 1 |

This function takes an integer as input and prints the values of the loop control variable from 0 to num - 1. The loop control variable is initialized to 0 and incremented by 1 after each iteration of the loop. The loop terminates when the loop control variable is equal to num. If we want to print the values from 0 to 99, we can simply call the while\_loop() function with the value 100 as the argument. We don't need to change the implementation of the function at all.

**LAB SUBMISSION:**

*Create a Visual Studio Code Workspace, Lab\_9 and c files (Task\_9\_1.c to Task\_9\_17.c) for individual tasks and add them to Lab\_9 workspace.*

|  |  |  |
| --- | --- | --- |
| **TASK 9.1: Print Stars** | | **[1 point]** |
| Write a program that will take a number from user and displays the number of \* on screen in a column. Make a function **Print\_Stars**(). Use ***while*** loop. | | |
| **Sample Output 1** | Enter the number >> 5  \*  \*  \*  \*  \* | |
| **Sample Output 2** | Enter the number >> 2  \*  \* | |

**Answer:** /\*--------------------------------------------------------------------------------   Author        :  Haseeb-ul-hassan   Roll No       :  2022MC58   Dated         :  16/10/23   Purpose       :  T6ask 9\_1 to get know about the user defined function                     and loops and print no of stars   Disclaimer    :  open source Code has been writen for educational purpose only   -------------------------------------------------------------------------------\*/ #include<stdio.h> void abc(int); void main() {     int a=0;     printf("enter a number >>");     scanf("%d",&a);     abc(a); } void abc(int d) {     int c=0;     while (c<d){printf("\*\n",c++);} }

|  |  |  |
| --- | --- | --- |
| **TASK 9.2: Multiplication Tables** | | **[1 point]** |
| Write a program that ask user to enter a number and prints its multiplication table of it from 1 to 10 using ***while*** loop. Make a function **Tables**(). | | |
| **Sample Output 1** | Enter the number >> 12  12 x 1 = 12  12 x 2 = 24  12 x 3 = 36  12 x 4 = 48  12 x 5 = 60  12 x 6 = 72  12 x 7 = 84  12 x 8 = 96  12 x 9 = 108  12 x 10 = 120 | |

**Answer:** /\*--------------------------------------------------------------------------------   Author        :  Haseeb-ul-hassan   Roll No       :  2022MC58   Dated         :  16/10/23   Purpose       :  Task 9\_2 to get know about the loops                     and print the table of given number   Disclaimer    :  open source Code has been writen for educational purpose only   -------------------------------------------------------------------------------\*/ #include<stdio.h> void table(int); void main() {     int a=0;     printf("enter a number >>");     scanf("%d",&a);     table(a); } void table(int d) {     int c=1,f=0;     while (c<11)     {         f=d\*c;         printf("%d\*%d=%d\n",d,c-1,f,++c);     } }

|  |  |  |
| --- | --- | --- |
| **TASK 9.3: Squares and Cubes** | | **[1 point]** |
| Write a program that will take a number as user input and print number 1 to that number, their squares, and cubes, separated by a tab using ***while*** loop. Make a function **Squares\_Cubes**(). | | |
| **Sample Output 1** | Enter the number >> 3  Number Square Cube  1 1 1  2 4 8  3 9 27 | |
| **Sample Output 2** | Enter the number >> 5  Number Square Cube  1 1 1  2 4 8  3 9 27  4 16 64  5 25 125 | |

**Answer:** /\*--------------------------------------------------------------------------------   Author        :  Haseeb-ul-hassan   Roll No       :  2022MC58   Dated         :  16/10/23   Purpose       :  Task 9\_2 to get know about the loops                     and print the square cube  of given number   Disclaimer    :  open source Code has been writen for educational purpose only   -------------------------------------------------------------------------------\*/ #include<stdio.h> void square\_qube(int); void main() {     int a=0;     printf("enter a number >>");     scanf("%d",&a);     square\_qube(a); } void square\_qube(int d) {     int c=1,f=1,g=1;             printf("Number\tSquare\t Qube \n");     while (c<d+1)     {         f=c\*c;         g=c\*f;         printf("%d\t%d\t%d\n",c,f,g);         c++;     } }

|  |  |  |
| --- | --- | --- |
| **TASK 9.4: Advanced Multiplication Tables** | | **[1 point]** |
| Write a program that asks user to enter a number and limit and prints its multiplication table of it from 1 to limit using ***while*** loop. Make a function **Adv\_Mul\_Tables**(). | | |
| **Sample Output 1** | Enter the number >> 12  Enter the limit >> 6  12 x 1 = 12  12 x 2 = 24  12 x 3 = 36  12 x 4 = 48  12 x 5 = 60  12 x 6 = 72 | |

**Answer:** /\*--------------------------------------------------------------------------------   Author        :  Haseeb-ul-hassan   Roll No       :  2022MC58   Dated         :  16/10/23   Purpose       :  Task 9\_4 to get know about the loops                     and print the table of given number at the given line   Disclaimer    :  open source Code has been writen for educational purpose only   -------------------------------------------------------------------------------\*/ #include<stdio.h> void table(int,int); void main() {     int a=0,b=0;     printf("enter a number >>");     scanf("%d",&a);     printf("enter a limit  >>");     scanf("%d",&b);     table(a,b); } void table(int d,int q) {     int c=1,f=0;     while (c<q+1)     {         f=d\*c;         printf("%d\*%d=%d\n",d,c-1,f,++c);     } }

|  |  |  |
| --- | --- | --- |
| **TASK 9.5: Find Factorial** | | **[1 point]** |
| Write a program that will take a number from user and will print its factorial at output. You must take care of negative number and zero as well. Make a function **Fact**(). | | |
| **Sample Output 1** | Enter the number >> 5  5! = 120 | |
| **Sample Output 2** | Enter the number >> -2  Factorial of -ve number doesn’t exist. | |
| **Sample Output 3** | Enter the number >> 0  0! = 1 | |

**Answer:** Paste your code here.

*Diagram

Description automatically generated****for* Loop**

A ***for*** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

|  |
| --- |
| for(init; condition; increment)  {     statements(s);  } |
| **Syntax of *for* Loop** |

Here is the flow of control in a ***'for***' loop −

* The **init** step is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here if a semicolon appears.
* Next, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement just after the '***for***' loop.
* After the body of the '***for***' loop executes, the flow of control jumps back up to the **increment** statement. This statement allows you to update any loop control variables. This statement can be left blank if a semicolon appears after the condition.
* The condition is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then increment step, and then again condition). After the condition becomes false, the '***for***' loop terminates.

|  |  |
| --- | --- |
| **Example 9.2: *for* loop in C** | |
| /\* Example\_9\_2.c: for loop in C  ---------------------------------------------------------------------------  This program demonstrates the use of for loop in C. It takes a number as  input form user and displays values on screen from 0 to given number -1.  ---------------------------------------------------------------------------  Written by Shujat Ali (engrshujatali@gmail.com) on 5-Oct-2021.  IDE: Visual Studio Code 1.60.0  C Compiler: GCC (Rev. 5, Built by MSYS2 Project) 10.3.0 \*/  #include <stdio.h>  void for\_loop(int num);  int main()  {    int num = 0;    printf("Enter a number >> ");    scanf("%d", &num);  // Call the for\_loop() function    for\_loop(num);    return 0;  }  void for\_loop(int num)  {    //Prints the values of the loop control variable from 0 to num - 1.    for (int i = 0; i < num; i++)   {      printf("The value of loop control variable is %d\n", i);    }  }  // End of program | |
| **Program Output 1** | Enter a number >> 3  The value of loop control variable is 0  The value of loop control variable is 1  The value of loop control variable is 2 |
| **Program Output 2** | Enter a number >> 5  The value of loop control variable is 0  The value of loop control variable is 1  The value of loop control variable is 2  The value of loop control variable is 3  The value of loop control variable is 4 |

|  |  |
| --- | --- |
| **TASK 9.6: Print Stars (again)** | **[1 point]** |
| Do TASK 9.1 again by using ***for*** loop. | |

**Answer:** Paste your code here.

|  |  |
| --- | --- |
| **TASK 9.7: Multiplication Tables (again)** | **[1 point]** |
| Do TASK 9.2 again by using ***for*** loop. | |

**Answer:** Paste your code here.

|  |  |
| --- | --- |
| **TASK 9.8: Squares and Cubes (again)** | **[1 point]** |
| Do TASK 9.3 again by using ***for*** loop. | |

**Answer:** Paste your code here.

|  |  |
| --- | --- |
| **TASK 9.9: Advanced Multiplication Tables (again)** | **[1 point]** |
| Do TASK 9.4 again by using ***for*** loop. | |

**Answer:** Paste your code here.

|  |  |
| --- | --- |
| **TASK 9.10: Find Factorial (again)** | **[1 point]** |
| Do TASK 9.5 again by using ***for*** loop. | |

**Answer:** Paste your code here.

|  |  |  |
| --- | --- | --- |
| **TASK 9.11: Print Line of Stars** | | **[1 point]** |
| Write a program to display the date sheet of your midterm exam.  Define a function **PrintStarLine**() in order to print the three stars lines in the below table. Note there are 50 stars in each line. | | |
| **Sample Output** | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Date Subject  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  November, 1 Vector and Complex Analysis  November, 2 Engineering Dynamics  November, 3 Digital Logic Design  November, 4 Computer Programming-I  November, 5 Translation of Holy Quran-II  November, 8 Electric Machinery  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

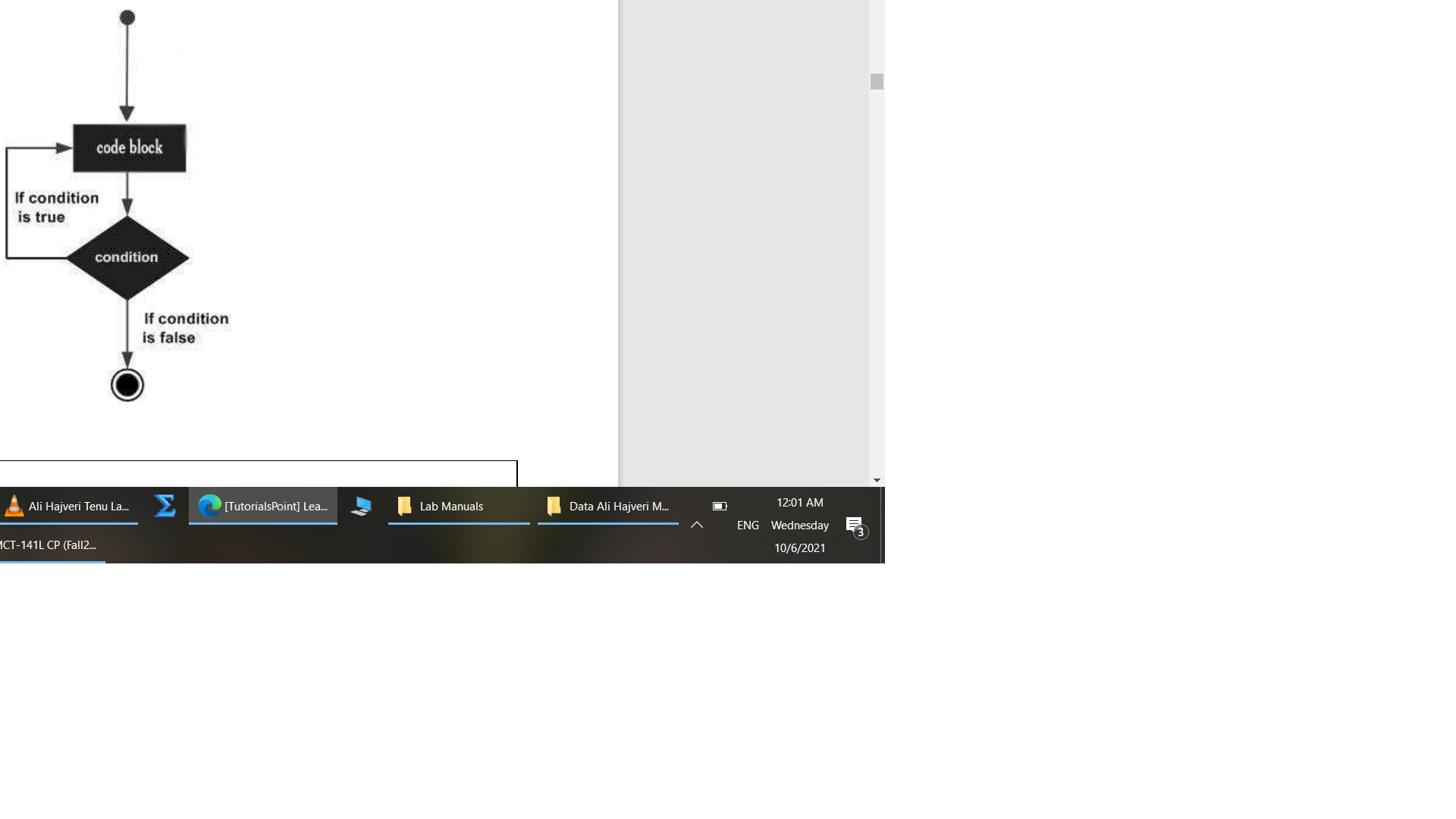
**Answer:** Paste your code here.

|  |  |  |
| --- | --- | --- |
| **TASK 9.12: Print Line of Counted Stars** | | **[1 point]** |
| Modify the above task such that instead of printing 50 stars in each line, function **PrintStarLine**() must take the number of stars to be printed as an input. Print 55 stars on 1st and 2nd line and 60 stars on last line. | | |
| **Sample Output** | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Date Subject  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  November, 1 Vector and Complex Analysis  November, 2 Engineering Dynamics  November, 3 Digital Logic Design  November, 4 Computer Programming-I  November, 5 Translation of Holy Quran-II  November, 8 Electric Machinery  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

**Answer:** Paste your code here.

|  |  |  |
| --- | --- | --- |
| **TASK 9.13: Print Line of Characters** | | **[1 point]** |
| Modify the above task such that instead of printing any number of stars in each line, function **PrintCharLine**() must take an input of the character (instead of star) to be printed on each line. Print 50 $ on 1st and 2nd line and 55 % on last line. | | |
| **Sample Output** | $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  Date Subject  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  November, 1 Vector and Complex Analysis  November, 2 Engineering Dynamics  November, 3 Digital Logic Design  November, 4 Computer Programming-I  November, 5 Translation of Holy Quran-II  November, 8 Electric Machinery  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% | |

**Answer:** Paste your code here.

***do*…*while* Loop**

Unlike ***for*** and ***while*** loops, which test the loop condition at the top of the loop, the ***do*...*while*** loop in C programming checks its condition at the bottom of the loop. A ***do*...*while*** loop is like a ***while*** loop, except the fact that it is guaranteed to execute at least one time.

|  |
| --- |
| Do  {     statement(s);  }while( condition ); |
| **Syntax of *do*…*while* Loop** |

Notice that the conditional expression appears at the end of the loop, so the statement(s) in the loop executes once before the condition is tested. If the condition is true, the flow of control jumps back up to ***do***, and the statement(s) in the loop executes again. This process repeats until the given condition becomes false.

|  |  |
| --- | --- |
| **Example 9.3: *do*…*while* loop in C** | |
| /\* Example\_9\_3.c: do...while loop in C  ---------------------------------------------------------------------------  This program demonstrates the use of do...while loop in C. It takes a number  as input form user and displays values on screen from 0 to given number -1.  ---------------------------------------------------------------------------  Written by Shujat Ali (engrshujatali@gmail.com) on 6-Oct-2021.  IDE: Visual Studio Code 1.60.0  C Compiler: GCC (Rev. 5, Built by MSYS2 Project) 10.3.0 \*/  #include <stdio.h>  void do\_while\_loop(int num)  {    //Prints the values of the loop control variable from 0 to num - 1.    int i = 0;    do    {      printf("The value of loop control variable is %d\n", i);      i++;    }    while (i < num);  }  int main()  {    int num = 0;    printf("Enter a number >> ");    scanf("%d", &num);    // Call the do\_while\_loop() function    do\_while\_loop(num);    return 0;  }// End of program | |
| **Program Output 1** | Enter a number >> 3  The value of loop control variable is 0  The value of loop control variable is 1  The value of loop control variable is 2 |
| **Program Output 2** | Enter a number >> 0  The value of loop control variable is 0 |

|  |  |  |
| --- | --- | --- |
| **TASK 9.14: Password Protected** | | **[1 point]** |
| Write a program that asks user to enter the 6-digit password and if the password is correct, it displays your name otherwise asks user to enter the password again. For defining password, use **const unsigned int** variable. Use ***do while*** loop. Make a function **Pass\_protect**(). | | |
| **Sample Output 1** | Enter the password >> 645289  Congratulations Mr. Shujat, your password is correct. | |
| **Sample Output 2** | Enter the password >> 347581  You entered the wrong password! Try again.  Enter the password >> 645289  Congratulations Mr. Shujat, your password is correct. | |

**Answer:** Paste your code here.

|  |  |  |
| --- | --- | --- |
| **TASK 9.15: Sum of 5-digit Number** | | **[1 point]** |
| Write a program that asks user to enter a 5-digit number and prints the sum of its digit. Use ***do while*** loop to force the user to enter only 5-digit number. Make a function **Sum\_Num**(). | | |
| **Sample Output 1** | Enter a 5-digit number >> 43298  Sum of the digits is 26 | |
| **Sample Output 2** | Enter a 5-digit number >> 1987  You entered an invalid number. Try again.  Enter a 5-digit number >> 12365  Sum of the digits is 17 | |

**Answer:** Paste your code here.

|  |  |  |
| --- | --- | --- |
| **TASK 9.16: Collatz Conjecture** | | **[1 point]** |
| The conjecture is concerned with [sequences](https://en.wikipedia.org/wiki/Sequence) defined as follows: start with any [positive integer](https://en.wikipedia.org/wiki/Positive_integer) n. Then each term is obtained from the previous term as follows: if the previous term is [even](https://en.wikipedia.org/wiki/Parity_(mathematics)), the next term is one half of the previous term. If the previous term is odd, the next term is 3 times the previous term plus 1. The conjecture is that no matter what value of n, the sequence will always reach 1.  Write a program that asks user to enter the number n and prints all the values of this sequence till 1, along-with step number. If the entered number is less than 2, print an error message. Make a function **Conjecture**(). | | |
| **Sample Output 1** | Enter the value of n >> 9  Value at step 1 is 28  Value at step 2 is 14  Value at step 3 is 7  Value at step 4 is 22  Value at step 5 is 11  Value at step 6 is 34  Value at step 7 is 17  Value at step 8 is 52  Value at step 9 is 26  Value at step 10 is 13  Value at step 11 is 40  Value at step 12 is 20  Value at step 13 is 10  Value at step 14 is 5  Value at step 15 is 16  Value at step 16 is 8  Value at step 17 is 4  Value at step 18 is 2  Value at step 19 is 1 | |
| **Sample Output 2** | Enter the value of n >> -3  You entered an invalid number. Try again. | |

**Answer:** Paste your code here.

|  |  |  |
| --- | --- | --- |
| **TASK 9.17: Find Average** | | **[1 point]** |
| Write a program that takes numbers from the user and when the user enters -1, it returns the average of the numbers entered (except the last entered -1). Allow for floating point values. Make a function **Avg**(). | | |
| **Sample Output 1** | Enter the number >> 9  Enter the number >> 6  Enter the number >> 12  Enter the number >> 3  Enter the number >> 8  Enter the number >> 7  Enter the number >> -1  The average of the given 6 numbers is 7.5 | |
| **Sample Output 2** | Enter the number >> 15  Enter the number >> 5  Enter the number >> 11  Enter the number >> 9  Enter the number >> 0  Enter the number >> -1  The average of the given 5 numbers is 8 | |

**Answer:** Paste your code here.

Students are advised to fill the manual and submit it before the upcoming lab. Kindly rename the file as   
‘MCT-242L\_CP1\_2022\_LM9\_XX’, where XX is your roll number. After completing the manual, turn it in Google Classroom.