Honors Computer Science C++
Mr. Clausen
Program 17A, 17B, 17C

Program 17A: “Four For Time” (30 points)

Write a program that practices “for loops”, functions, and a menu. For each of the “for loops” described, write a separate function with a corresponding menu choice.

1. Type comments at the beginning of the program to display your name and other information just like those used for program 4A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**

2. Include <iostream> so we can use the **cout** and **cin** commands. Include <iomanip> so we can use the **setw** command.

3. Type: using namespace std;

4. There are no constants to declare.

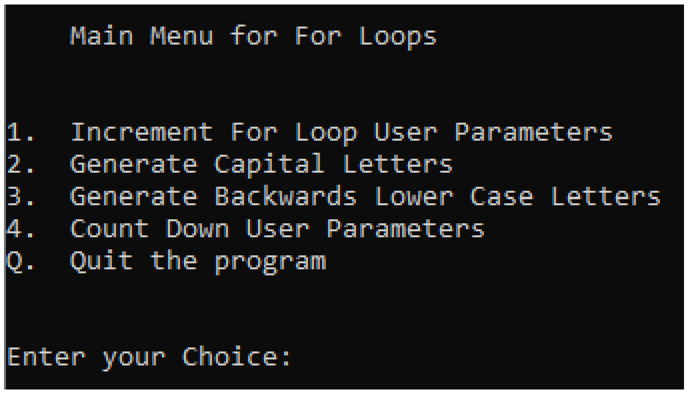
5. Don’t forget to include comments for your functions, right **before** each of your function declarations.

6. Declare the function Display_My_Info ( ) in the function declaration section of the program. This should be a void function that doesn’t take any parameters and doesn’t return any values. In addition, declare functions for Menu(), Increment_For_Loop_User_Params(), Generate_Capital_Letters(), Generate_Backwards_Lower_Case_Letters(), Count_Down_User_Params(), and Control_Menu_Execution() to call all of the functions.

7. Use a comment line with **equal signs** to separate all of the above from the int main function. For example:
   //===

8. Inside the **int** main() function, declare all of your variables for the menu and then call the menu function inside a loop like we have done in previous programs. Don’t forget to call the function Display_My_Info() before calling the menu function, before the loop.
   Don’t forget that value-returning functions are “called” in assignment statements, while void functions can be called just by writing the function name including any of the “actual” parameters.
   Don’t forget your **return 0** command at the end of int main.

The menu should look like the following.



```
    Main Menu for For Loops


1.  Increment For Loop User Parameters
2.  Generate Capital Letters
3.  Generate Backwards Lower Case Letters
4.  Count Down User Parameters
Q.  Quit the program


Enter your Choice:
```

9. After the **int** main() function, have another comment line of **subtraction signs** to separate the above from your function implementation lines below. For example:
   //---

10. Implement all of the functions, separating each one from the other ones using comment lines of **subtraction signs**.

11. When you implement the function Increment_User_Params(), ask the user for an increment value and an upper limit. Make these variables local to this function. You will also need a variable for the "display numbers counter" to ensure that you are displaying 10 numbers across each row (you will need to increment this variable inside the loop body), since the user will choose to increment by a value other than one. Start the for loop with an initial value of 1. Declare the loop counter variable in the for loop header.
   To format the numbers nicely use the setw command to set a width of 5 for each of the columns. You will need an "if" statement in the loop to display 10 numbers across like we did in program 8 (see Ch. 6 lecture notes).

12. When you implement the function Generate_Capital_Letters(), start the loop at 65 (A) and stop the loop at 90 (Z) incrementing by one. You will also need a variable for the "display numbers counter" to ensure that you are displaying 10 letters across each row (you will need to increment this variable inside the loop body). Declare the loop counter variable in the for loop header. If you do this, you can reuse the variable name for the loop counter.
   To format the letters nicely use the setw command to set a width of 5 for each of the columns. You will need an "if" statement in the loop to display 10 letters across like we did in program 8 (see Ch. 6 lecture notes). **Review the Chapter 3 lecture notes to see that you need to explicitly type cast the integer values in the loop counter to char to make letters display in the output.**

13. When you implement the function Generate_Backwards_Lower_Case_Letters(), start the loop at 122 (z) and stop the loop at 97 (a) decrementing by one (subtracting). You will also need a variable for the “display numbers counter” to ensure that you are displaying 10 letters across each row (you will need to increment this variable inside the loop body). Declare the loop counter variable in the for loop header. If you do this, you can reuse the variable name for the loop counter.
    To format the letters nicely use the setw command to set a width of 5 for each of the columns. You will need an “if” statement in the loop to display 10 letters across like we did in program 8 (see Ch. 6 lecture notes). Review the Chapter 3 lecture notes to see that you need to explicitly type cast the integer values in the loop counter to char to make letters display in the output. **The purpose of this function is to practice using a “for loop” to correctly decrement. If the loop header is wrong, you will not get credit for this portion of the program.**

14. When you implement the function Count_Down_User_Params(), ask the user for a decrement value, a starting value, and an ending value. Make these variables local to this function. You will also need a variable for the “display numbers counter” to ensure that you are displaying 10 numbers across each row (you will need to increment this variable inside the loop body). Start the for loop with an initial value of “starting_value”, terminate the loop at “ending_value”, and use the “decrement_value” to count backwards and to display the numbers from higher to lower. Declare the loop counter variable in the for loop header.
    To format the numbers nicely use the setw command to set a width of 5 for each of the columns. You will need an “if” statement in the loop to display 10 numbers across like we did in program 8 (see Ch. 6 lecture notes). **The purpose of this function is to practice using a “for loop” to correctly decrement. If the loop header is wrong, you will not get credit for this portion of the program.**

15. After the last function implementation of every program, end your program with two comment lines of equal signs. This signifies the end of your source code. For Example:
    ```
    //=====================================================
    //=====================================================
    ```

When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don’t need to make any changes, then click the Submit button.

PROGRAM 17B Give Me A Penny (20 points)

If you give someone one penny today, two pennies tomorrow, four pennies the next day, eight pennies the day after that, and continue to double the amount of pennies you give the person each day, how long would it take for that person to accumulate one million **dollars** (not pennies) or more? (It will not calculate to exactly $1,000,000.)

1. **For this program we don't need to use functions.**
   Type comments at the beginning of the program to display your name and other information just like those used for program 4A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**

2. Include <iostream> so we can use the **cout** and **cin** commands. Include <iomanip> so we can use the **setw** command and the setprecision command.

3. **Type**: using namespace std;

4. There are no constants to declare.

5. Don't forget to include comments for your function, Display_My_Info(), right **before** your function declaration.

6. Declare the function Display_My_Info ( ) in the function declaration section of the program. This should be a void function that doesn't take any parameters and doesn't return any values.

7. Use a comment line with **equal signs** to separate all of the above from the int main function. For example:
   //===

8. Inside the **int** main() function, declare all of your variables and then call the function, Display_My_Info(). Don't forget your **return 0** command should be the last line in int main.

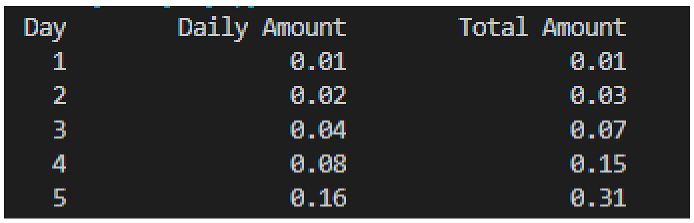
9. Initialize all of the variables that are to be used in this program. Initialize each variable on a separate line. (Initialize integers to 0 (zero), decimal numbers to 0.0. You will need variables for all of the following: dayNumber, dailyAmount, and sum. If you think that you might need more variables. Make sure that you use descriptive identifiers for all of your variables to model "self-documenting code".

10. Leave a blank line after the variable initialization statements.

11. Print a "header" for each of the three columns in your output. The day number, daily amount, total amount to date.

12. For the Calculations AND Output section, type the following comment:
    #--------------------------------Calculations & Output-------------------------------
    You will need to use a while loop to do the calculations and print the output for each

day. The loop should stop **after** the total/sum is equal to or has exceeded $1,000,000. Your output should look like the following except that it will continue until the running total equals or exceeds one million **dollars**. You need to use set precision and setw in your cout statements to make the output look nice.



```
Day        Daily Amount        Total Amount
  1                0.01                0.01
  2                0.02                0.03
  3                0.04                0.07
  4                0.08                0.15
  5                0.16                0.31
```

13. When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don't need to make any changes, then click the SUBMIT button in repl.it to turn it in.

PROBLEM 17C: Probability Probably (30 points)

Write a program to give the user a choice to flip a coin or roll a standard and fair six sided die. Ask the user how many times they want to flip a coin or roll the die. Display each outcome and display how many of each were randomly determined. Display 10 coin outcomes or 10 die rolls across each line.

1. Type comments at the beginning of the program to display your name and other information just like those used for program 4A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**

2. Include <iostream> so we can use the **cout** and **cin** commands. Include <iomanip> so we can use the **setw** command. Also include, #include <ctime> and #include <cstdlib>.

3. **Type**: using namespace std;

4. There are no constants to declare.

5. Don't forget to include comments for your functions, right **before** each of your function declarations.

6. Declare the function Display_My_Info ( ) in the function declaration section of the program. This should be a void function that doesn't take any parameters and doesn't return any values. In addition, declare functions for all of the other parts of this program, which include a menu, control menu execution, flipping a coin, and rolling a die. There should be separate functions for each of the above.

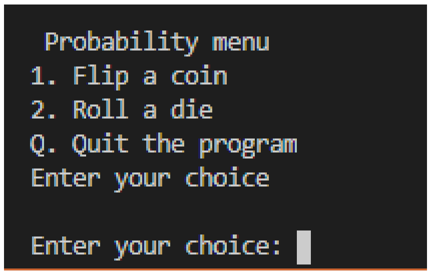
7. Use a comment line with **equal signs** to separate all of the above from the int main function. For example:
   //===

8. Inside the **int** main() function, declare all of your variables and then call the functions. **Don't forget that value-returning functions are "called" in assignment statements, while void functions can be called just by writing the function name including any of the "actual" parameters.**
   Don't forget your **return 0** command.

9. Also in the main() function, create a "while loop" that will quit when the user enters either a lowercase 'q' or upper case 'Q' and keep running otherwise.
   Now call the "menu" function and control menu execution function from the main function **inside the "while loop"** and run your program to see if it works without errors, and will quit when the user enters either an uppercase or lowercase letter 'Q'.

10. After the **int** main() function, have another comment line of **subtraction signs** to separate the above from your function implementation lines below. For example:
    //---

11. Implement the function "menu" that displays the menu (as pictured below) and asks the user to enter their choice and returns this choice to the main function. Upper and lower case values should work for the menu choice to quit. Pass the variable representing the user's choice to this function when calling it from the main function.



```
 Probability menu
1. Flip a coin
2. Roll a die
Q. Quit the program
Enter your choice

Enter your choice: 
```

12. You can add the "if, else if" statements in a control menu execution function to call the functions to flip a coin or roll the die.

13. Implement all of the functions, separating each one from the other ones using comment lines of **subtraction signs**.

14. Define the function to flip a coin that asks the user how many times to flip the coin (no error trapping), uses a loop to generate a random number representing heads or tails, displays these outcomes, and counts the occurrences of each. Include your cout statements in this function as well so that you are repeatedly generating a random number, printing out heads or tails, keeping totals for each

and then displaying how many heads and tails were chosen after the loop is over. Display each outcome and display how many of each were randomly determined. Display 10 coin outcomes across each line. **Make any variables used in this function local to this function.** Call this function from **inside the control menu execution function** and run your program to see if it works without errors.

15. Define the function to roll the die that asks the user how many times to roll the die (no error trapping), uses a loop to generate random number representing a 1 through 6, displays these outcomes, and counts the occurrences of each. Include your cout statements in this function as well so that you are repeatedly generating a random number, printing out the outcome, keeping totals for each and then displaying how many ones, twos, threes, etc. were chosen after the loop is over. Display each outcome and display how many of each were randomly determined. Display 10 die rolls across each line. **Make any variables used in this function local to this function.** Call this function from **inside the control menu execution function** and run your program to see if it works without errors.

16. After the last function implementation of every program, end your program with two comment lines of equal signs. This signifies the end of your source code. For Example:
```
//=========================================================
//=========================================================
```

17. When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don’t need to make any changes, then click the SUBMIT button in repl.it to turn it in.