**Clock Program**

The beginning of the program was just copied from the sample given. It asks the user for a number between 0-9 inclusive and that is the time the clock begins from. Hours, minutes and seconds each have 2 variables that store their value to be printed at the end of the main loop.

In my program, the seconds are first incremented by one. The number of seconds is divided by 60 and if the number of seconds is 60 the quotient is 1, that value of one will be added to the number of minutes which represents an increment of 1 minute. If the number of seconds is less than 60, the quotient will be 0 hence 0 will be added to the number of minutes which represents no increment in minutes. The remainder will be divided by 10 where the digit in AH and AL register will be stored in 2 separate variables to represent the overall number of seconds which will be displayed at the end of the main loop.

Next, the minutes will be checked to see if they are equal to 60 and if the number of minutes is 60 the quotient is 1, that one will be added to the number of hours which represents an increment of 1 hour. If the number of minutes is less than 60, the quotient will be 0 hence 0 will be added to the number of hours which represents no increment in hours. The remainder will be divided by 10 where the digit in the AH and AL register will be stored in 2 separate variables to represent the overall number of minutes which will be displayed at the end of the main loop.

Finally, the hours will be divided by 12 to check to see if they are equal to 12 hours and if the number of hours is 12 the remainder will be 0 hence the hours would reset to 0.

The final part of the main loop is where the digits representing the number of hours will be displayed followed by the minutes and then the seconds. The time is displayed at the end after all calculations are done otherwise the time will be printed in reverse. A semi-colon will be printed between the hours and minutes and between the minutes and seconds. At the end of displaying these results, a new line will be displayed to print the next time below. The variable called ‘counter’ will be incremented at the end of the main loop and it is checked to see if it equals 43200 which is the number of seconds in 12 hours. If it is, the program will jump to a procedure that exits the program. If not, the main loop will execute again.

**Assumptions**

* The user will not enter a character other than 0-9 when prompted to do so
* Clock time does not need to replicate realistic counter time

Limitations I encountered were the inability to implement the use of a stack and procedures

During the project, I tried to find a way to implement the use of stack however I found it difficult considering data must be of type DW in order to be pushed into the stack. I wasn’t able to make use of procedures as error messages appeared saying the calls were for a procedure too far away. Not being able to use these 2 concepts prevented me from simplifying my code more than what is it.

**How to Use the program**

1. Enter a number between 0 and 9 when prompted and the beginning format will appear
2. Click any key when prompted and the clock will cycle through 12 hours and return to the beginning format