***PROBLEM OF THE WEEK – 01***

***Assignment Given: 16/09/2020***

***Assignment Completed: 11/10/2020***

**PROJECT DISCRIPTION: Write or create a C++ application for the unzipping or copying extracted folder from source to desired destination location and perform operations. Display a menu asking for 3 options ATP, OTE, DMI and Exit.**

|  |
| --- |
| **Option 1): if selected takes the choice of version number and searches the file with suffix of given input and select it and copy that zip file from the source, extract it to destination by removing or renaming the file given Name (OTE\_Tester).**  **Note: the Extracted file should contain a text file with version number.**  **Option 2): if selected take the version of the file to be chosen, search it in OTE source directory, open the file copy or extract the zip file in the directory to a User defined location and copy the OTStart.exe file to OTE\_Tester folder.**  **Note:** if the OTEstart file is already exiting Replace it.  **Option 3):** if selected takes the file name, searches it, and extracts it to destination folder or Directory where OTE\_Tester is present and again extracts it and keeps the extracted files under different name.  **Note:** Only one Zip is used to extract it twice. |

**Note:**

1. Try to use the in-Built C++ functions and methods for unzipping if possible.
2. Use the command prompt to run the program as .exe and to perform operations.
3. Keep all files in one main directory and code.
4. Make all necessary comments and display message corresponding to invalid inputs.
5. If possible try to keep an Error logs for the program.

**Program Structure:**

------------------------------------**Schematic-**----------------------------------------

<Pre processor directives>

<type def>

<pragma>

<Global Variables>

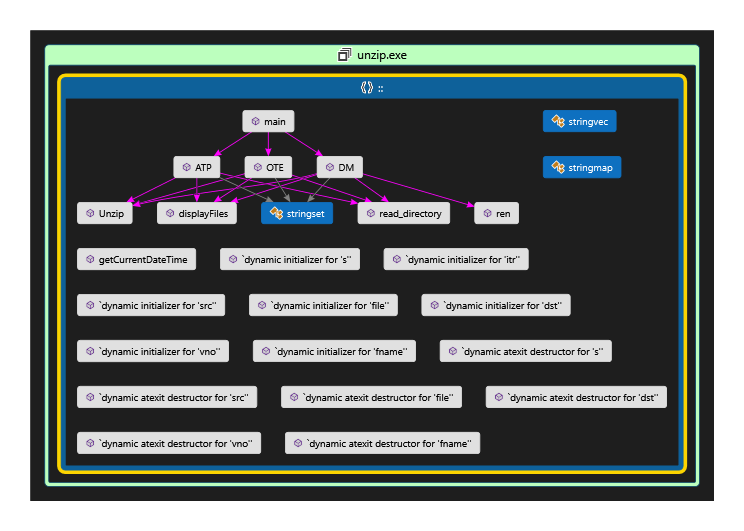
Function Declaration;

Function Definitions ()

Main()

------------------------------------**Schematic**-------------------------------------------

------------------------------------**CODE TREE**-------------------------------------------



------------------------------------**CODE TREE**-------------------------------------------

**Function Definition:**

1. **string getCurrentDateTime() :**

Returns the Current time in a String format to log file name generation.

1. **void read\_directory(const string& name, stringset& s) :**

Reads the Directory path and an unordered set and reads the files in the directory to the set.

1. **void displayFiles(stringset s) :**

Displays the filenames stored the unordered set by read\_directory function.

1. **void ATP() :** This performs the first use case of the assignment statement.
2. Take source file path from user.
3. Read and display files in the source.
4. Wait for few seconds and press any key to continue.
5. Enter the file version number without any white space.
6. Add .zip to file name and append it to source file path.
7. Take the destination path.
8. Convert the source and destination file path to const char \* and send it as parameters to unzip function.
9. Perform file rename for the unzip folder.
10. **void OTE() :** This performs the Second use case of the assignment statement.
    1. Take source file path from user.
    2. Read and display files in the source.
    3. Wait for few seconds and press any key to continue.
    4. Enter the file version number without any white space.
    5. Add .zip to file name and append it to source file path.
    6. Take the destination path.
    7. Convert the source and destination file path to const char \* and send it as parameters to unzip function.
    8. Perform file moving operation by copying .exe file to OTE\_Tester folder.
    9. Performing remove directory for the empty folder once the .exe is copied
11. **void DMI() :** This performs the Third use case of the assignment statement.
    1. Take source file path from user.
    2. Read and display files in the source.
    3. Wait for few seconds and press any key to continue.
    4. Enter the file version white space.
    5. Add .zip to file name and append it to source file path.
    6. Take the destination path.
    7. Convert the source and destination file path to const char \* and send it as parameters to unzip function.
    8. Renaming the unzip folder to New name and performing same when next file is extracted as subsequent DMIAppA, DMIAppB.
12. **void Unzip(const char\* a, const char\* b):**

The function definition takes the const char \* string to perform extraction of the zip file and move it to the destination folder defined by the user. It is implementation try catch so any exception during execution will be recorded in log file for Debugging.

1. **int main () :**

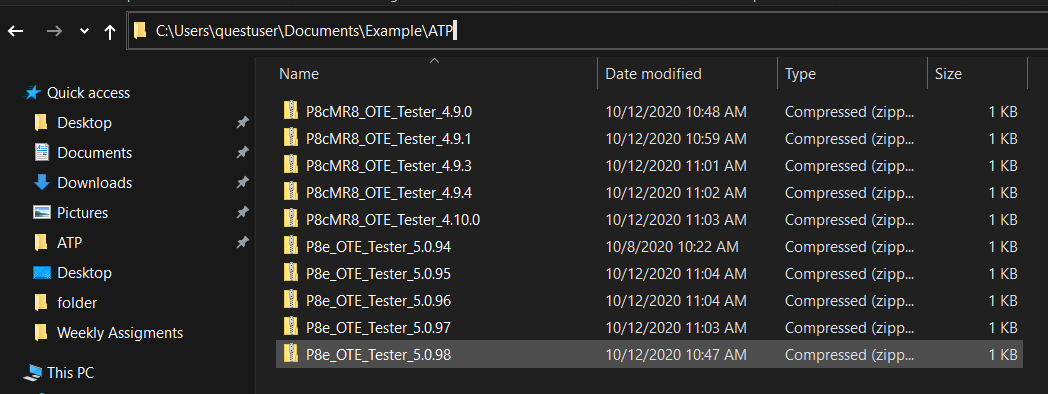
Main function handles the Choices or options the user selects and calls the predefined functions. It also gives the log filename for the current program in execution once the user choice operation is performed it returns the control to the menu and loops whether to continue or exit the program.

1. **string checkdmi(string vno),string checkatp(string vno),string checkote(string vno) :** Checks version number of ATP, DMI, OTE and appends it to particular filename.
2. **ren (string dst, string fname), rem(string dst, string fname), void rep (string dst, string fname):** rep takes the old filename of ATP unzipped file to OTE\_TESTER,

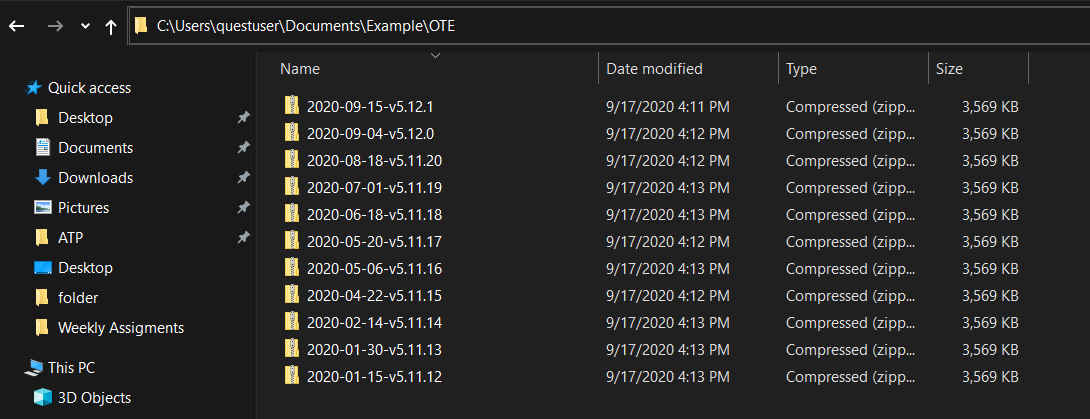
ren takes the old filename of unzipped DMI file and renames it con currently as DMIAppA, DMIAppB, rem takes the unzipped OTE folder file and copies to OTE\_Tester folderand removes the Empty directory of DMI unzipped folder.

**INPUT SCREENSHOTS**

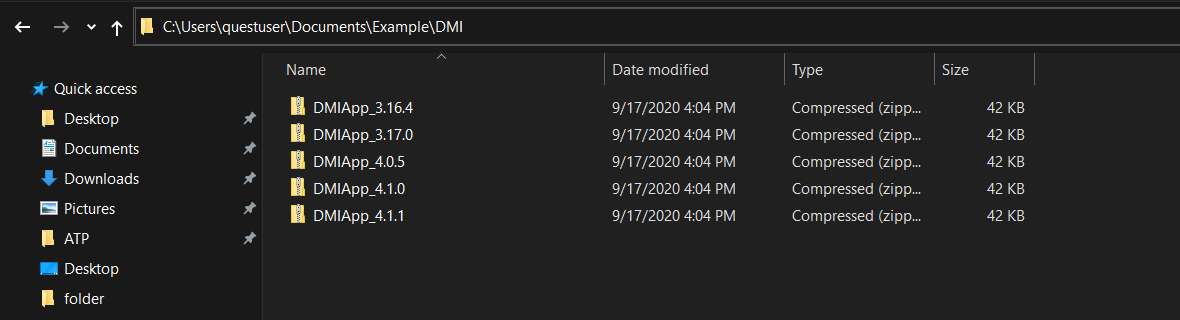
**ATP .zip FILE.**

****

**OTE .zip FILE.**

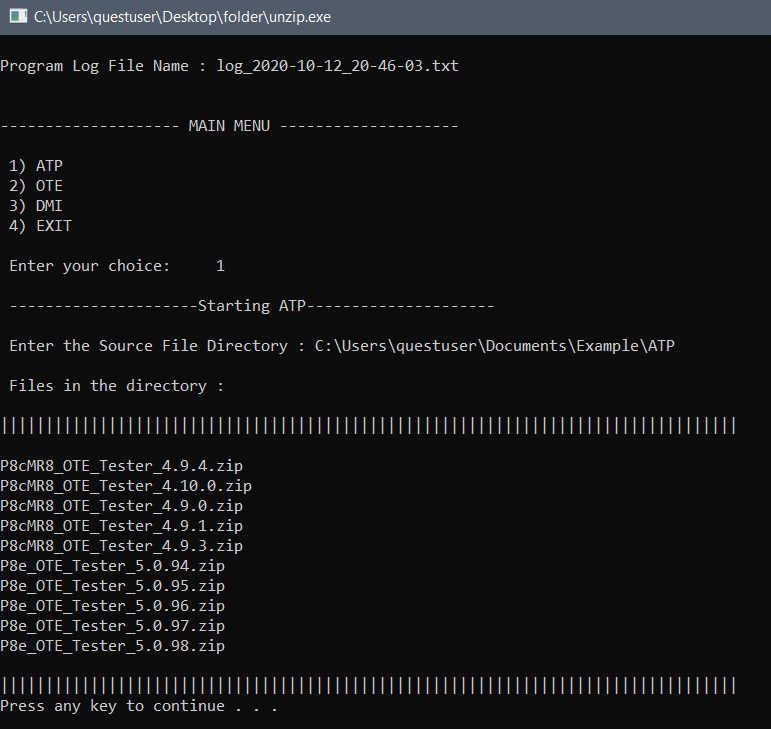
****

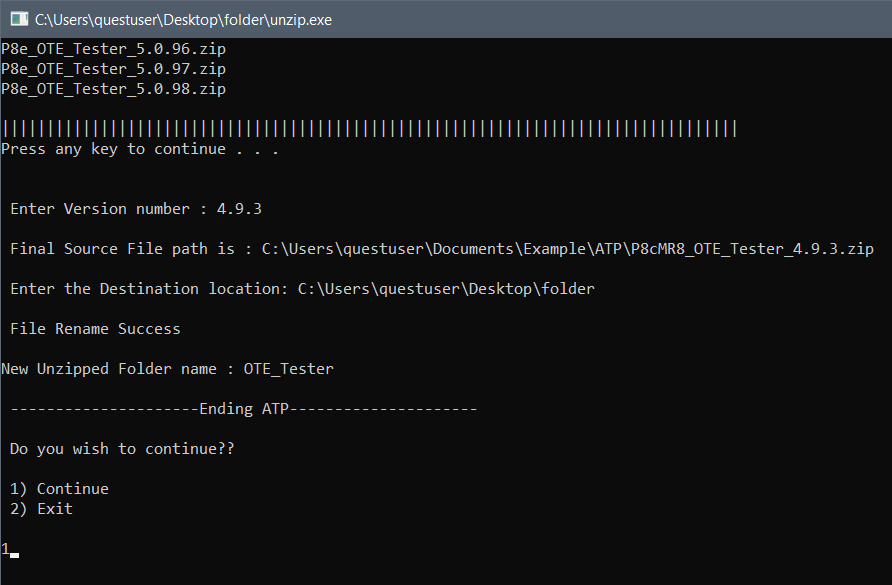
**DMI .zip FILE.**

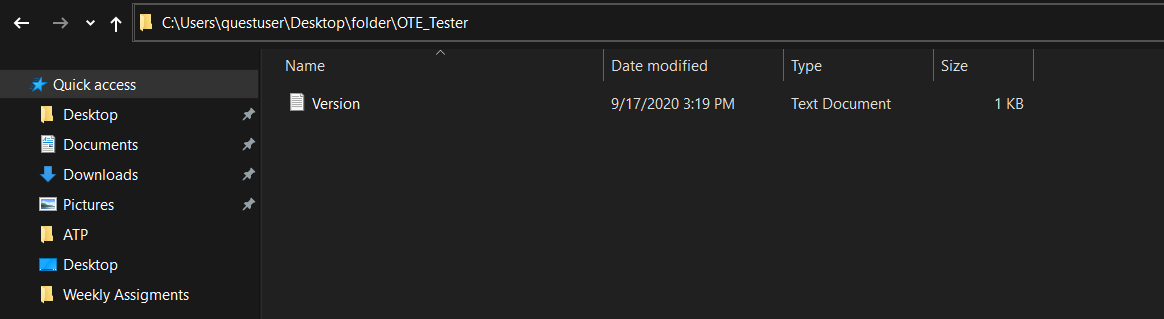
****

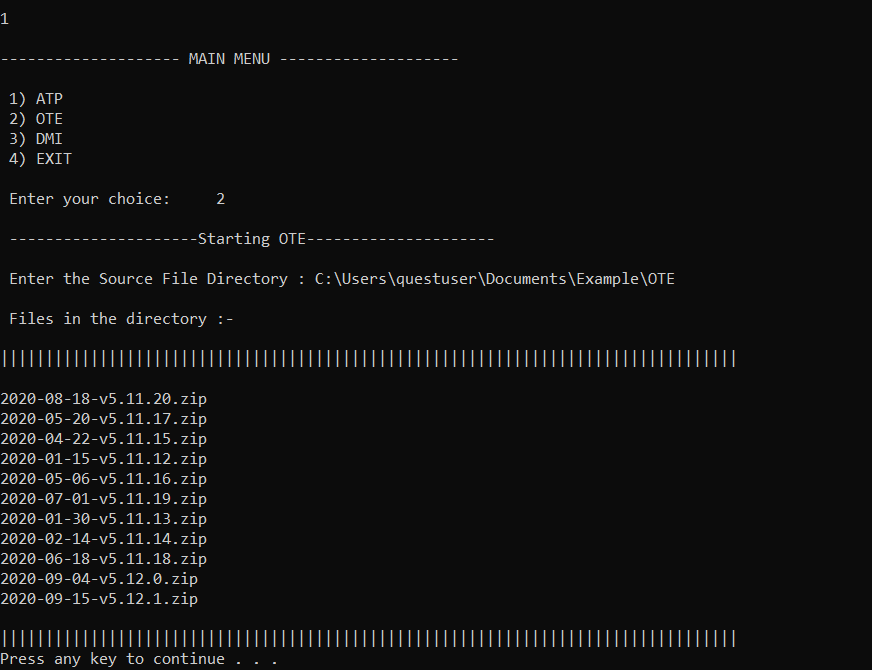
**OUTPUT SCREENSHOTS.**

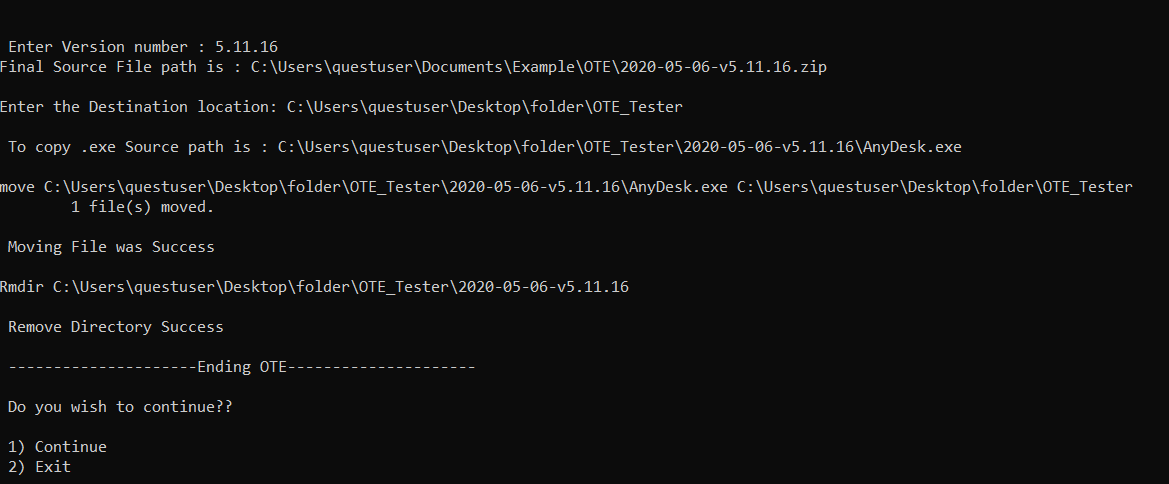
1. **ATP**

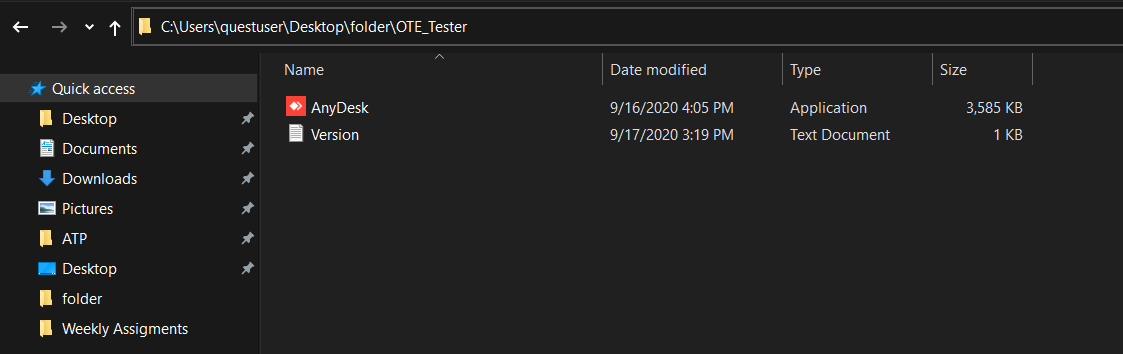
****

****

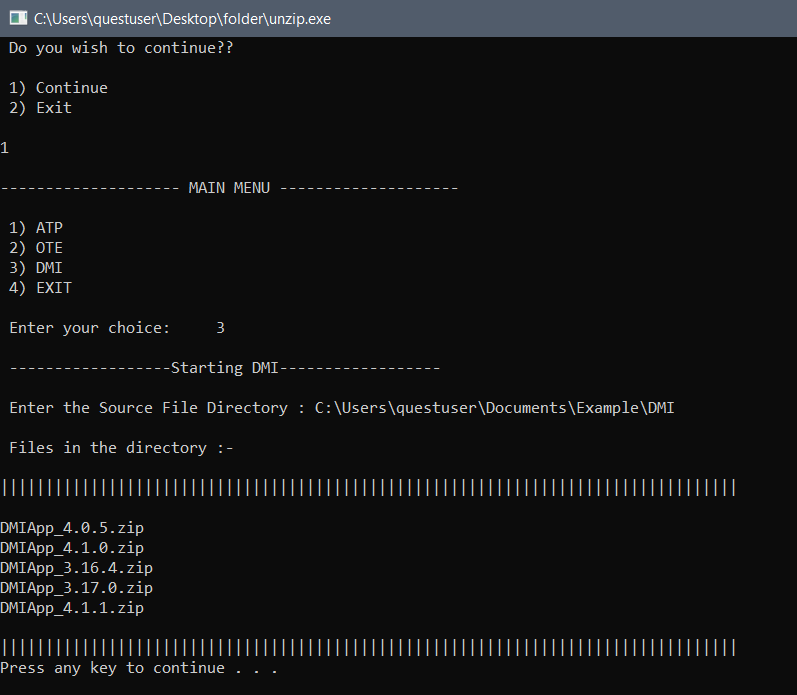
****

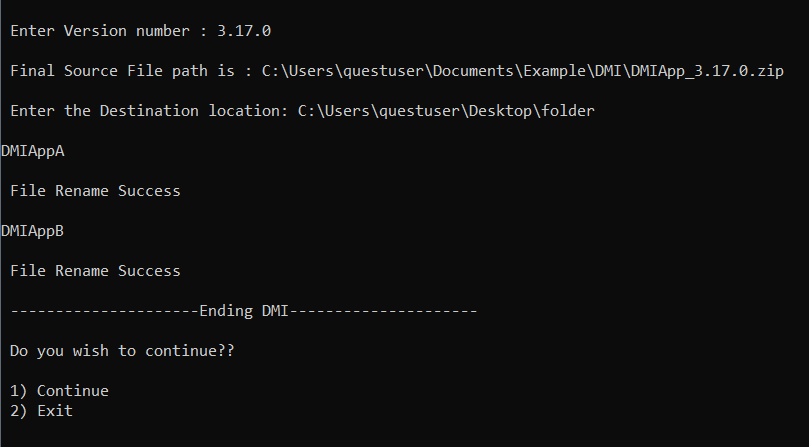
1. **OTE**

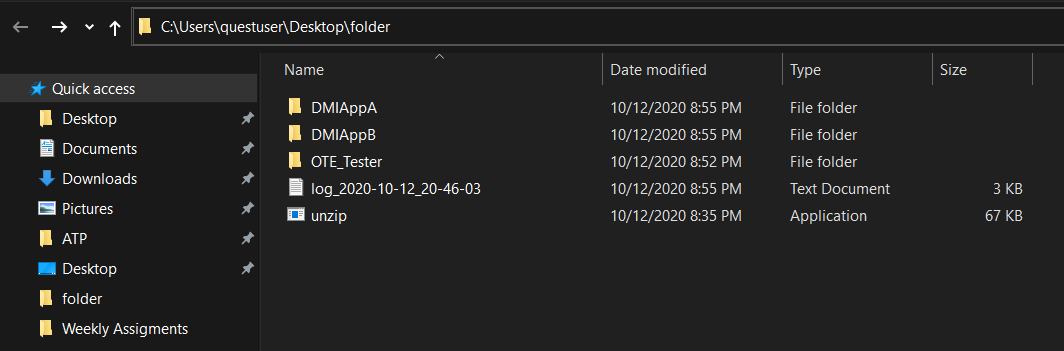
****

****

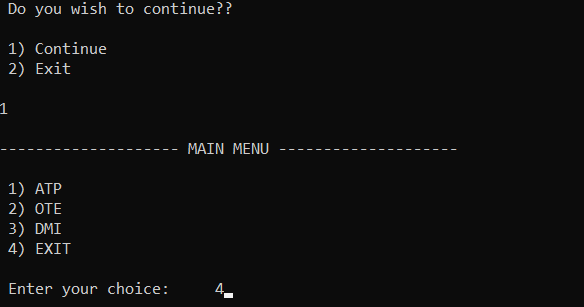
1. **DMI**

****

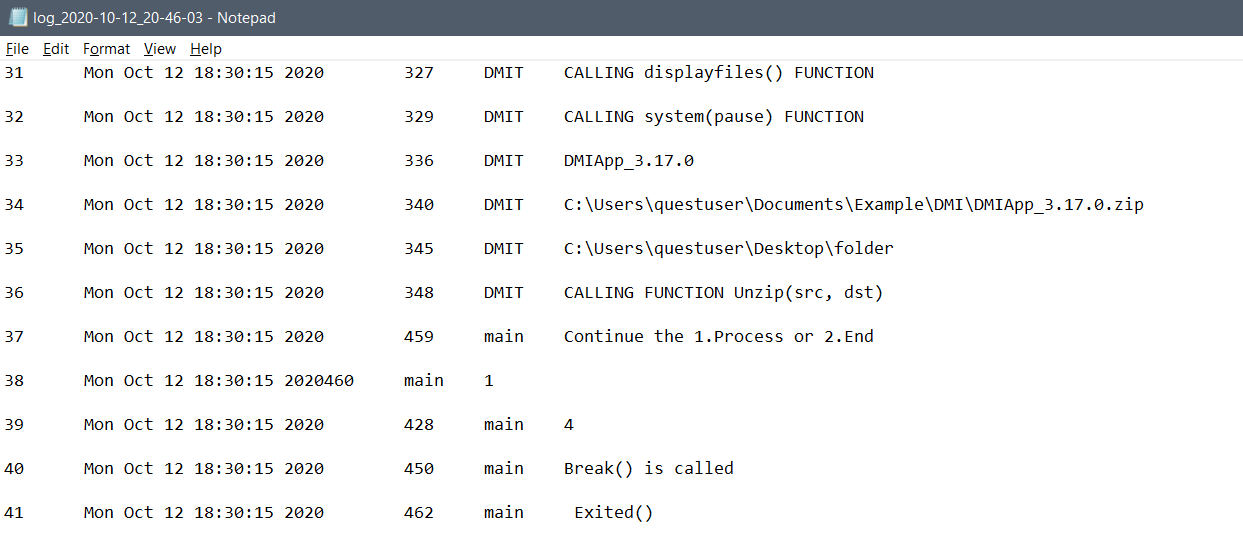
****

****

1. **EXIT**

****

**LOG FILE :**

****