Firstly, I opened the original file of the word.cpp and then I used the debugging tool to find out the errors in the program. It was my very first time using a debugging tool. I asked help for the teacher and learnt about it. Debugger allowed me to check the program through line by line. Debugger helped me to identify the errors which made it possible to analyze and this helped me to remove errors in the program.

The program is created to replace the existing word list which consists of 10 numbers with the new word list which contains of 25 numbers. While going through the debugging process I figured out some errors related to the use of pointers, the passing of parameters, and dynamic memory management.

The first error was in the use of the function;

Passing of parameters error: void build\_wordlist (string\* word\_list, int capacity, string filename):the address of word\_list and for the int capacity was not allocated so I chaged the code to: (string\* &word\_list, int& capacity, string filename).

Another error was in the use of pointer during the process of creating another array as twice larger as the original word list.

string new\_word\_list = \*(new string[2 \* capacity]);

In the above statement the address of the new\_word list was not allocated instead the address of new string and capacity was allocated; but this was error since we needed to insert new strings in the new file. Therefore, for the dynamic memory allocation the pointer is being used as shown in the statement below:

string \*new\_word\_list = new string[2 \* capacity];

The next error was in the statement :

delete word\_list;

\*word\_list = new\_word\_list;

As seen above the word\_list is an array so, when we delete array from a dynamic memory special bracket “[]” is used. Here, the \* sign has been use but this is not supposed to be used in this condition since the word\_list and new\_word list both are pointers. As per to correct the error I corrected the statement as shown below:

delete [] word\_list;

word\_list = new\_word\_list;

And finally, all the errors were sorted out and the program run without an error and displayed all the strings of the new list. The result as shown in the following:

