Assignment 4

Name: Michael Skarlatov

Date: 11/29/2018

Part 1:

a)

0			
1	4371		
2			
3	1323	6173	
4	4344		
5			
6			
7			
8			
9	4199	9679	1989

b)

0	9679
1	4371
2	1989
3	1323
4	6173
5	4344
6	
7	
8	
9	4199

c)

0	9679
1	4371
2	
3	1323
4	6173
5	4344
6	
7	
8	1989
9	4199

d)

0	
1	4371
2	
3	1323
4	6173
5	9679
6	
7	4344
8	
9	4199

1989 does not fit on the table

Part 2:

This apart of the assignment wanted me to multiply two polynomials together and out put a simplified resulting polynomial. I did this by making two vectors that would be the first two polynomials being multiplied together by pushing back two numbers that would represent the exponent and constant. These two vectors where called P1 and P2 and where added to a function called pollymaker and then this function used temp vectors to store values to foil values for the final vector, and then compare exponents to add values together for the final polynomial vector. I then printed out the final vector.

Part 3:

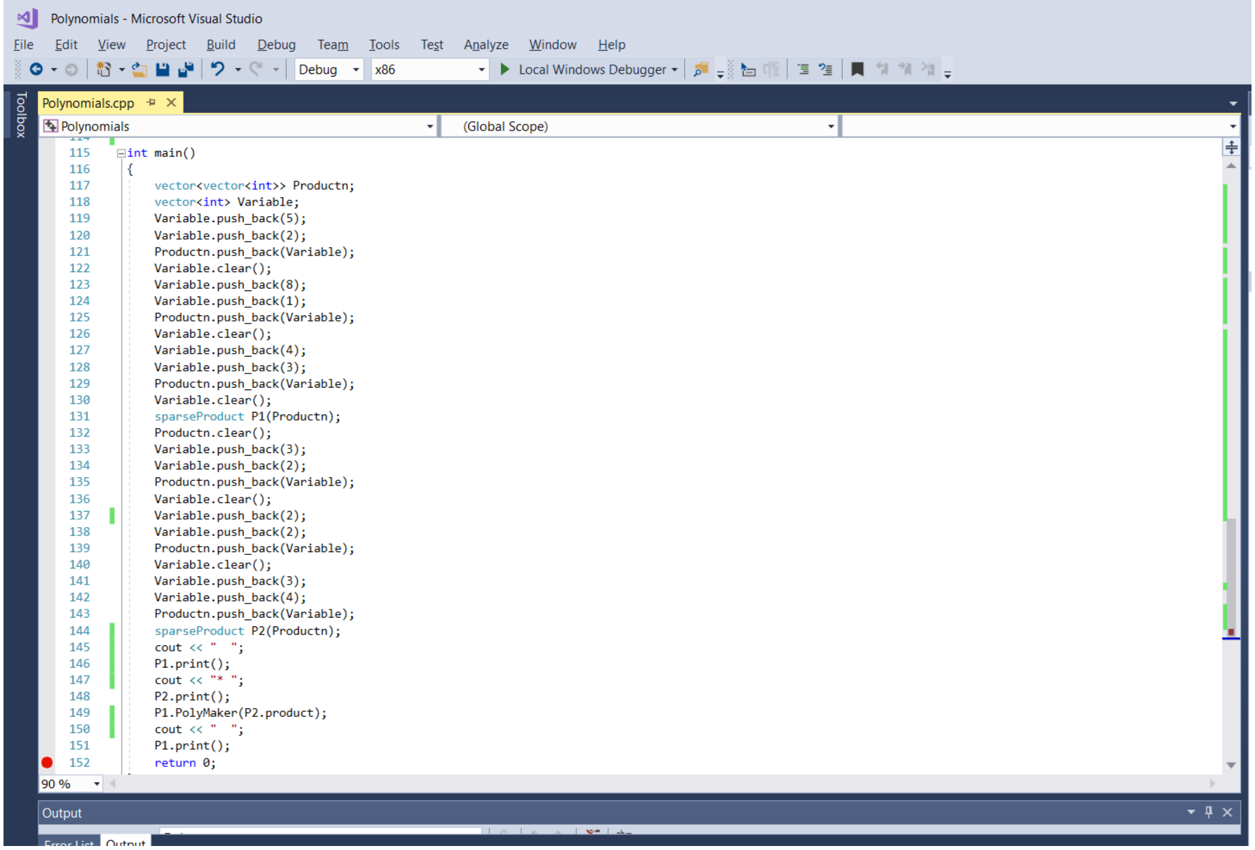
This part of the assignment required me to build a spell checker that would check words from a text file that I created with against a dictionary of words made from the text file provided by the professor. It would do this by first turning the words from the provided text file into a created type dictionary which I labeled as d. After this I created a function that would turn the text file into a vector of vectors that would hold the line position of all the words in each line on the text file I created. I then scrolled through each word in each line and used a comparison function that would check all the hashes in the unordered_map dictionary d, called search. This function would return a bool 1 or 0 depending on weather the word was in the dictionary or not. If it was not in the dictionary it would move into one of the functions that would try to either delete a letter from the word, add a letter to the word, or move all the letters around in the word and would then find all of the words in the dictionary that match this altered word and out put these words as alternative words for the misspelled words.

Part 4:

This part of the assignment wanted me to create a program that would be able to take a stream of integers and out put the median of the stream as more numbers where being added. To do this I created an array of integers that would represent the stream in the order it was being added. I then passed this array and the size of the array into a function that created two heaps called left and right, these heaps where a maxheap that was represented by left and a minheap that was represented by right. I then added the values from the array as well as the heaps into a function called getMedian that would then be assigned to an integer m which represented the effective median. In the function getMedian it would add the numbers from the array to the heaps depending on weather the number being added was more or less then the current median and would then move values from the other heap to make sure that the heaps where always the same size with an even amount of numbers added. If there where an even amount of numbers added the function would return the median as the average of the two middle values in both heaps. However, if there where an odd amount of numbers added the median would be the top of the of either the left or right heap depending on weather or not the value last added was more then the median before it or not. These medians where then printed out, along with the value of what was added to the stream before the median was calculated.

Input/Output Screen Shots:

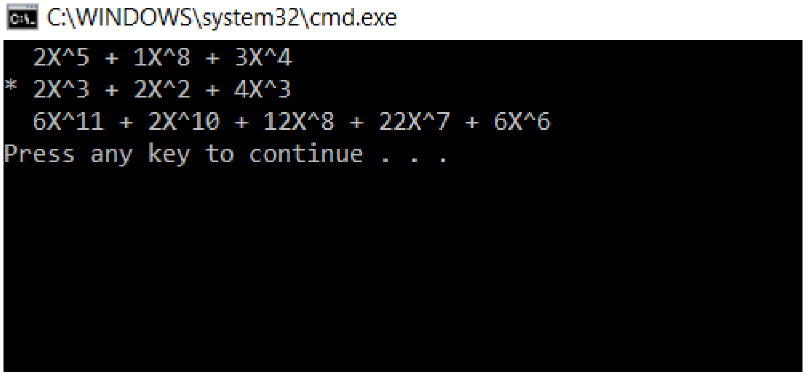
Part 2:


Polynomials - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Toolbox
Polynomials.cpp
Polynomials
(Global Scope)
90 %
Output
Error List
Output

```
int main()
{
    vector<vector<int>> Productn;
    vector<int> Variable;
    Variable.push_back(5);
    Variable.push_back(2);
    Productn.push_back(Variable);
    Variable.clear();
    Variable.push_back(8);
    Variable.push_back(1);
    Productn.push_back(Variable);
    Variable.clear();
    Variable.push_back(4);
    Variable.push_back(3);
    Productn.push_back(Variable);
    Variable.clear();
    sparseProduct P1(Productn);
    Productn.clear();
    Variable.push_back(3);
    Variable.push_back(2);
    Productn.push_back(Variable);
    Variable.clear();
    Variable.push_back(2);
    Variable.push_back(2);
    Productn.push_back(Variable);
    Variable.clear();
    Variable.push_back(3);
    Variable.push_back(4);
    Productn.push_back(Variable);
    sparseProduct P2(Productn);
    cout << "   ";
    P1.print();
    cout << "* ";
    P2.print();
    P1.PolyMaker(P2.product);
    cout << "   ";
    P1.print();
    return 0;
```

Input values of first 5 and 2, then 8 and 1, then 4 and 3 for the first polynomial

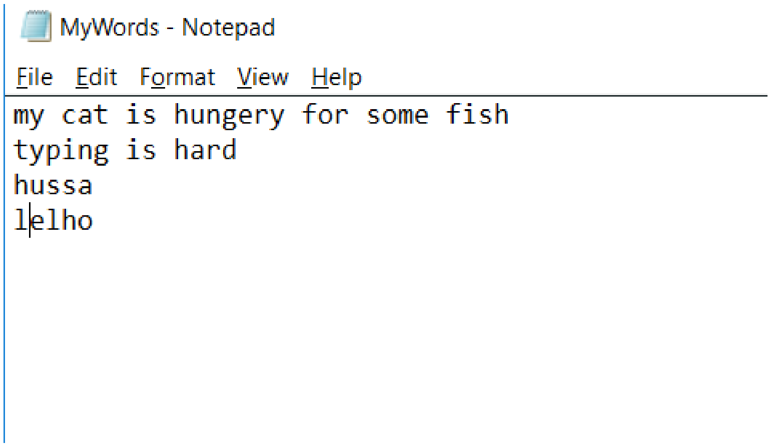
Input values of first 3 and 2, then 2 and 2, then 3 and 4 for the second polynomial



```
  2X^5 + 1X^8 + 3X^4
* 2X^3 + 2X^2 + 4X^3
  6X^11 + 2X^10 + 12X^8 + 22X^7 + 6X^6
Press any key to continue . . .
```

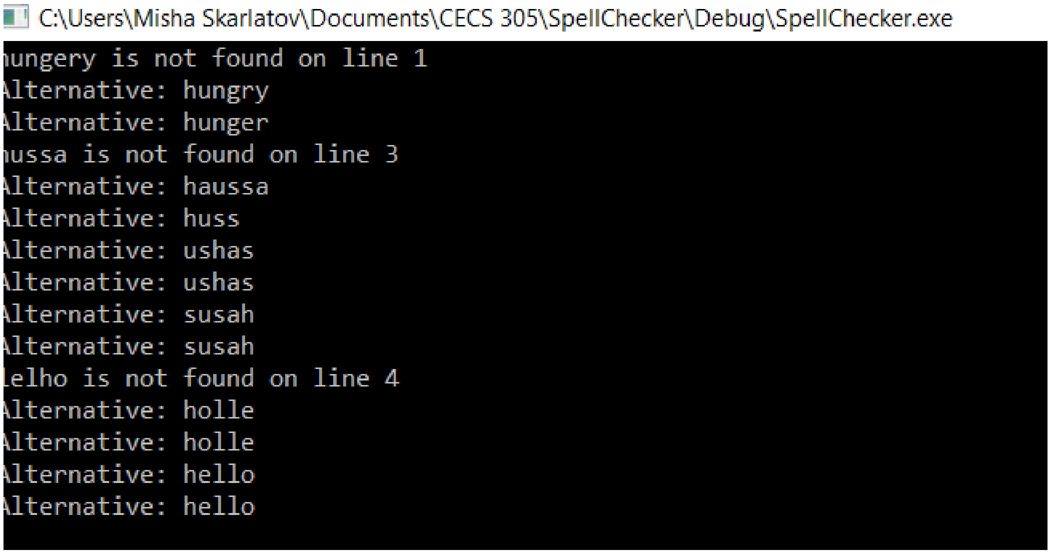
Output of both polynomials and their combined value

Part3:



```
my cat is hungery for some fish
typing is hard
hussa
lelho
```

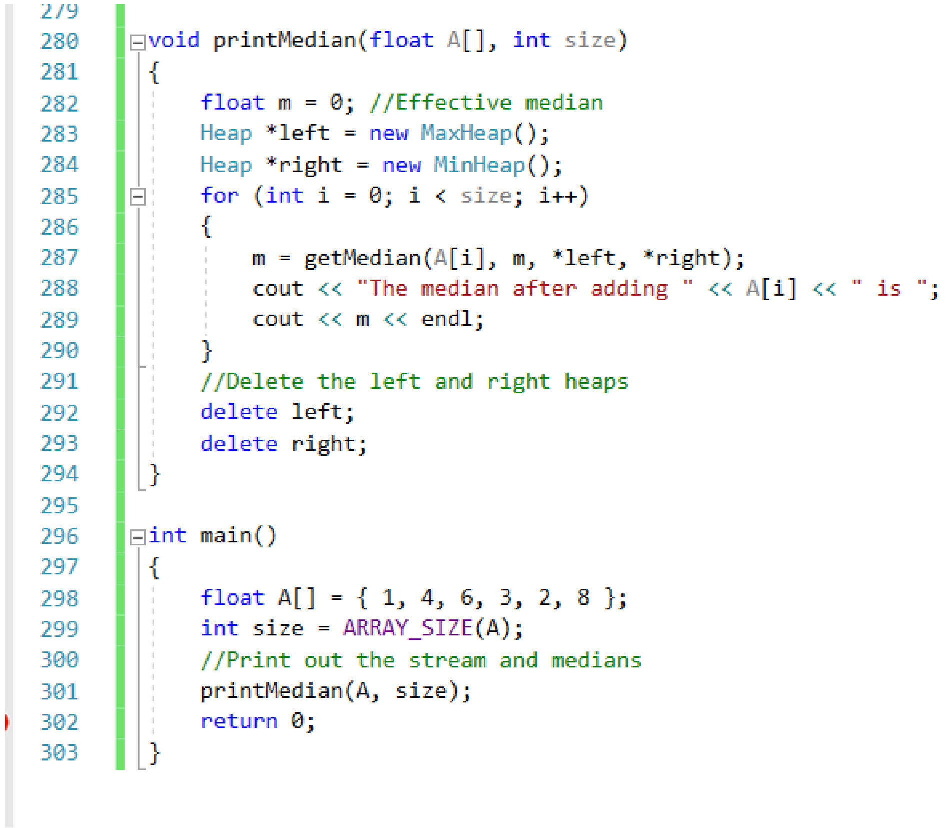
Inputted text file called MyWords.txt (no punctuation was added because if messed up the check)


C:\Users\Misha Skarlatov\Documents\CECS 305\SpellChecker\Debug\SpellChecker.exe

```
hungery is not found on line 1
Alternative: hungry
Alternative: hunger
hussa is not found on line 3
Alternative: haussa
Alternative: huss
Alternative: ushas
Alternative: ushas
Alternative: susah
Alternative: susah
lelho is not found on line 4
Alternative: holle
Alternative: holle
Alternative: hello
Alternative: hello
```

Output that shows the words that where misspelled and on what line they where misspelled as well as possible alternatives to the words that are similar but not exactly the misspelled word

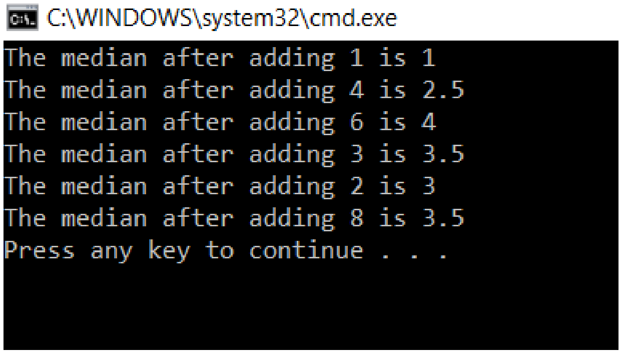
Part 4:



```
void printMedian(float A[], int size)
{
    float m = 0; //Effective median
    Heap *left = new MaxHeap();
    Heap *right = new MinHeap();
    for (int i = 0; i < size; i++)
    {
        m = getMedian(A[i], m, *left, *right);
        cout << "The median after adding " << A[i] << " is ";
        cout << m << endl;
    }
    //Delete the left and right heaps
    delete left;
    delete right;
}

int main()
{
    float A[] = { 1, 4, 6, 3, 2, 8 };
    int size = ARRAY_SIZE(A);
    //Print out the stream and medians
    printMedian(A, size);
    return 0;
}
```

Part of the code that show the creation and insertion of Array A with the values (1,4,6,3,2,8)



```
The median after adding 1 is 1
The median after adding 4 is 2.5
The median after adding 6 is 4
The median after adding 3 is 3.5
The median after adding 2 is 3
The median after adding 8 is 3.5
Press any key to continue . . .
```

Output that shows the median and the number that was added to the stream right before calculation