Main.cpp

```
#include "MyReader.h"
#include <iostream>

using namespace std;

int main()
{
   myReader mr;
   int total;
   double average;

   mr.read();
   mr.output();
   mr.doubleNums();
   mr.sort();
   mr.totalAndAverage(total, average);
   cout << "\nTotal: " << total << ", Average: " << average;
   mr.write(total, average);

   return 0;
}
```

MyReader.h

```
class myReader
{
private:
   int numArray[3];

public:
```

```
    void read();
    void output();
    void doubleNums();
    void sort();
    void totalAndAverage(int &t, double &a);
    void write(int to, double av);
};
```

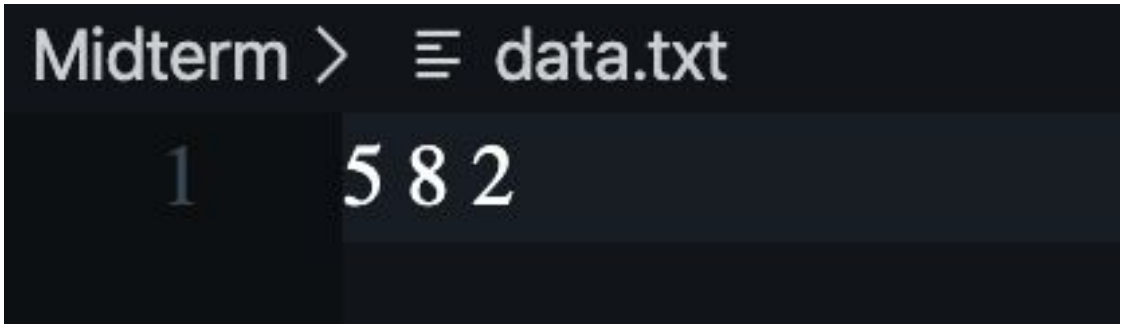
MyReader.cpp

```
#include "MyReader.h"
#include <iostream>
#include <iomanip>
#include <fstream>

using namespace std;

void myReader::read()
{
    ifstream myFile;
    myFile.open("data.txt");
    int tempNum;

    for (int i = 0; i < 3; i++)
    {
        myFile >> tempNum;
        numArray[i] = tempNum;
    }

    myFile.close();
}
```

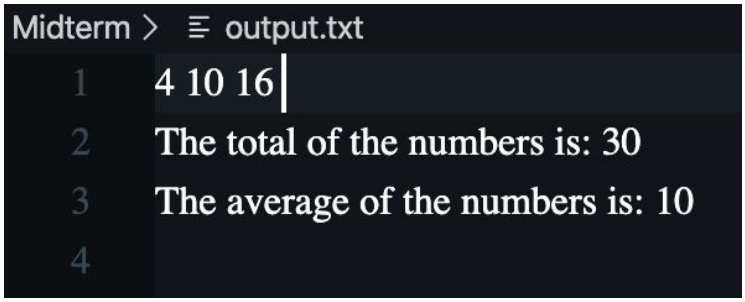
```
void myReader::output()
{
   for (int i = 0; i < 3; i++)
       cout << numArray[i] << ", ";
}

void myReader::doubleNums()
{
   for (int i = 0; i < 3; i++)
       numArray[i] *= 2;
}

void myReader::sort()
{
   int temp;
   for (int i = 0; i < 3; i++)
       for (int j = 0; j < 2 - i; j++)
           numArray[j] > numArray[j + 1]
               ? temp = numArray[j + 1],
                 numArray[j + 1] = numArray[j], numArray[j] =
temp : numArray[j + 1] = numArray[j + 1];
}

void myReader::totalAndAverage(int &total, double &average)
{
   total = 0;
   average = 0;
   int array[2];
```

```
    for (int i = 0; i < 3; i++)
    {
        total += numArray[i];
    }
    average = total / 3;
}

void myReader::write(int total, double average)
{
    ofstream outputFile("output.txt");

    for (int i = 0; i < 3; i++)
        outputFile << numArray[i] << " ";

    outputFile << "\n";
    outputFile << "The total of the numbers is: " << total <<
endl;
    outputFile << "The average of the numbers is: " << average <<
endl;
    outputFile.close();
}
```

Data.txt



Output.txt



Output