Problem 1. Short answers.

(a) [7 points] what will the following code output?

```
int main(){

 vector<int> x;

 cout << x.size() <<endl;

 for(int i=0; i<5; i++){
 x.push_back(i);
 }

 for( int i=0; i<x.size(); i++){
 cout << x[i] <<" ";
 }
 cout << endl;
 cout << x.size() <<endl;
 return 0;
}
```

Output:

0

0 1 2 3 4

5

(b) [3 points] explain why the following code segment leads to memory leak?

```
int *ptr;
for(int i = 0; i < 100; i++){
      ptr = new int[5];
}
delete[] ptr;
```

It leads to memory leak because the allocated memory wasn't deleted within the scope

(c) [6 points] what is the output of the following C++ code?

```
int myArray[5] = {-3, 5, 10, -1, 4};
int *myPointer = myArray;
cout << myArray[2] << endl;

myArray[4] = myArray[0];
cout << myPointer[4] << endl;

myPointer[3] *= myArray[3];
```

```
cout << myArray[3] << endl;
```

Output:

10

-3

1

(d) [4 points] the following code snippet calculates the running sum of a vector of integers. For example, the running sum of the sequence { a, b, c, d, …} is {a, a+b, a+b+c, a+b+c+d, …}. What is the error in the code, and how would you fix it? (Note: Assume it is not related to syntax or any missing #includes).

```
vector<int> v = {1, 2, 3, 4, 5};
vector<int> runningSum(v.size(), 1);
/* Initializes runningSum to zeros with the same size
as v */


for (int i=1; i<v.size(); i++) {
      runningSum[i] = runningSum[i-1] + v[i];
}
```

It won't properly compute the runningSum because the code in the for loop takes the previous term of runningSum[] and adds it to the "next" term of v[], but in the first iteration the runningSum[0] is equal to 0 instead of 1 which messes up the rest of the sum. So setting the first term of runningSum[] equal to the first term of v[] would make the runningSum work. This can be done before the for loop. Another solution would be to initialize all the values of runningSum to 1.

(e) [5 points] Explain what this function computes

```
#include <vector>
#include <limits>
using namespace std;
int func(vector<int> &x){

      int ret = numeric_limits<int>::max();
      for( int i=0; i<x.size(); i++ ) {
            if( x[i] < ret ) {
               ret = x[i];
            }
      }
      return ret;
}
```

This function finds the smallest term in an array/vector.

Problem 2. (25 points) dotProduct is a function that computes the sum of the elementwise product of two vectors of doubles and return it:

$$c = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

If the vectors do not have the same number of elements, dotProduct should give an error and exit the program. Otherwise, the function should return the result c as shown above. The following routine should find the dot product of a and b and then print out the result in main. Report your result in the write-up. Please submit your .cpp file as "yourLastName_hw8_prob2.cpp".

```
#include <iostream>
#include <vector>
using namespace std;

double dotProduct( vector &a, vector &b){

        // [Your code here]
}

int main(){
vector a = {2.1,2.2,4.1};
vector b = {-3.0,0.0,-0.25}; // you should not assume a or b always
have a size 3 //

        //[Your code here]

        return 0;
}
```

Output:
The dot product of the two vectors is -7

Problem 3. (25 points) Write a function that takes an input argument vector &x and returns the mean of all entries in x. Here the mean x is defined as:

$$\bar{x} = \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{N}$$

where xi is the ith entry in x and N is the total number of entries in x.

Write another function that takes an input argument vector &x and returns the standard deviation of all entries in x. Here the standard deviation s is defined as:

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}.$$

where xi is the ith entry in x, �is the mean of all entries in x and N is the total number of entries in x. Write a simple test program to demonstrate that both functions generate the correct results for vector x = {3.5, 5.5, -1.7, 9.6, 0, -2.7, 20.5};
Report your result in the write-up. Please submit your .cpp file as "yourLastName_hw8_prob3.cpp".

Output:

```
The average is 4.95714
The standard deviation is 8.094
```

Problem 4. (25 points) Write a function isSorted that takes an input vector of integers called vec and return true if vec is sorted in increasing order. Write a simple test program to demonstrate that the function returns the correct values for the following vector vec inputs.
Report your result in the write-up. Please submit your .cpp file as "yourLastName_hw8_prob4.cpp".

vec	Returns
{1, 2, 5, 6}	true
{5, 6, 0, 1}	false
{}	true
{10}	true
{10, 10}	true
{10, 10, 20}	true
{10, 10, 20, 5}	false

I was unclear as to whether or not this function called for the user to input the terms of the vector they want to test. I just assumed that you wanted to see a code that tested these vectors to see if the function we created actually tested their increasing order properly. It said simple test program so I just went with that, I'm sorry if that was not the correct way to interpret it.

Output:

```
true
false
```

true
true
true
true
false