Introduction to Computer Science
HW #5
Due: 2014/05/27

Homework Rules:

Hand-written homework can be handed in in class. Otherwise, you may contact the TA in advance and then bring the hardcopy to the TA in BL421.

As for the programming part, you need to upload it to CEIBA before the deadline. The file you upload must be a **.zip** file that contains the following files:

README.txt

HW01_b03901XXX (a folder that contains all .cpp & .h as required),

1. Do not submit executable files (.exe) or objective files (.o, .obj). Files with names in wrong format will not be graded. You must **remove any system calls**, such as system ("pause"), in your code if you use it.
2. In README.txt, you need to describe which compiler you used in this homework and how to compile it (if it is in a "project" form).
3. In your .cpp files, we suggest you write comments as detailed as you can. If your code does not work properly, code with comments earns you more partial credits.

Programming Problem I (50%):

Write a class **MyVector** for vector operations. You need to write:

(1) Default constructor
(2) Copy constructor
(3) Constructor that take one integer argument.
(4) Destructor
(5) Copier
(6) Getter for length
(7) Getter and setter for data[]
(8) Ostream (for cout)
(9) Operator+ and operator* (inner product)
(10) resize() (content in data[] can be destroyed)

Save your code in **myvector.h** and **myvector.cpp**. To get full credit, you need to

(1) Use initialization list whenever appropriate.
(2) Use const whenever appropriate.
(3) Header file only contain function prototype. Leave the function body to cpp

(which is not always a good thing, but let's practice here).

(4) Handle memory appropriately. Any use of uninitialized values, or un-released memory causes point deduction.

Below is a sample test for your code:

MyVector a; // test for default constructor	**Output:**
cout << a << endl;	()
a.resize(3); a.set(0, 0.0).set(1, 0.1).set(2, 0.2);	
cout << a << endl;	(0,0.1,0.2)
cout << a.get(1) << endl;	0.1
cout << a.getLength() << endl;	3

MyVector b(2), c(1); // test if operator= deals with resizing
c = b = a; // test copier	

cout << b << endl;	(0,0.1,0.2)
cout << c << endl;	(0,0.1,0.2)
MyVector d(c); // test copy constructor	
cout << d << endl;	(0,0.1,0.2)
cout << c*d << endl; // inner product	0.05
cout << a+b+c << endl;	(0,0.3,0.6)

Programming Problem II (25%):

Write a **prolog** program to compute $\sqrt{a}$ by the following iteration:

Introduction to Computer Science
HW #5
Due: 2014/05/27

$$x_m \leftarrow \frac{x_{m-1} \cdot x_{m-1} + a}{2 \cdot x_{m-1}}$$

You need to write the function (save your code in **"sroot.pl"**):

- ***sroot*(*X,M,SX*):** compute the square root (*SX*) of *X* up to *m* iterations. Initial guess is 1. In other words, sroot(*X*, 1, 1) is true for any *X*.

Some more examples are listed below:

sroot(2,1,X)	X=1
sroot(2,2,X)	X=1.5
sroot(2,3,X)	X=1.4166666666666667
sroot(2,4,X)	X=1.4142156862745099
sroot(2,5,X)	X=1.4142135623746899

Programming Problem III (25%):

Write a **prolog** program to compute gcd (greatest common divisor).

You need to write the function (Save your code in "**gcd.pl**"):

- ***gcd*(*A,B,G*):** compute gcd (*G*) of *A* and *B*.

Some more examples are listed below:

gcd(23,17,G)	G=1
gcd(24,18,G)	G=6
gcd(24,0,G)	G=24 (**Hint:** base case)

BONUS (5%) Template Wrapper for MyVector:

Learn C++ template by yourself.

Wrap MyVector as TVector (**"tvector.h"**). Below is a sample test how TVector works:

```
TVector<3> a,b,c;
b.set(0,0).set(1,1).set(2,2);
c = b;
a = b+c;
cout << a;
```

To get full credit,

(1) it is a wrapper, not a full rewrite. So if I change something in MyVector, the

behavior of TVector changes accordingly.

(2) Now resize() does not make any sense. So you do not allow user to use it.

(3) Of course, we will not test assignment from TVector<2> to TVector<3>.

How to submit:

Compress all your files into one single file and then submit electronically via Ceiba by the due date.