ENG101/300/Assignment 5/March 12, 2003
Due: 2nd lab session, week of March 17, 2002

Hail to the vectors!

This is a programming assignment that requires you to construct algorithms using vectors and to write functions that use them and return them. It is good practice for both functions and vectors. There are six questions on this assignment. Do questions 1–6, each worth (100/6)% of the assignment grade.

Here's the winning strategy:

- For each question, download the file that contains the associated function stub and the main routine from the course web page. Your GSI will show you how to do this if you do not already know. The files are named `5.*stub.cpp` where `*` is `1,2,3,4,5,6`.
- Compile the file with the function stub and verify that it compiles and produces a load module. (They all have been tested and all should compile error free on a Linux machine with the g++ compiler. Some compilers issue warning messages related to empty function stubs, but no errors.)
- Look at the main routine and understand its operation, what it is trying to do.
- Run the executable to make sure that it seems to be executing. Don't expect reasonable results yet! Careful with questions 3) and 5). They may go into what seems like an infinite loop. If this happens, type Control-C to get out of it. Or, they may cause the program to "crash", producing a "segmentation fault". (These behaviors are expected.)
- Now start working on the function until your program works. Test your code with simple inputs.
- Make sure to answer the questions posed in the comment lines in the first 3 problems.
- Questions 1–3 are "easy". Question 4 and 5 are somewhat challenging and question 6 will take some effort to get right.

...please turn over

1. The inner product

Complete the program started below. You must complete the function innerProduct() that computes the inner product of two vectors, x and y. The inner product is the sum $x_0y_0 + x_1y_1 + ... + x_{n-1}y_{n-1}$ where x_i is the $i+1$'th element of x and the number of elements in the vector is n. Here is a function stub and a main routine that you must use to test your function:

```
//File 5.1stub.cpp
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

float innerProduct(const vector<float>& a, const vector<float>& b)
{  //Why did the function use the const qualifiers in the parameter list
   //and pass the vectors by reference?
   //Answer the question and insert your code after this line.
}

int main(void)
{  int vectorLength = 10;

   vector<float> x (vectorLength);
   cout << " x: ";
   for (int i = 0; i < vectorLength; i = i + 1)
   {  x[i] = rand()%100;
      cout << "\t" << x[i];
   }
   cout << "\n";

   vector<float> y(vectorLength);
   cout << " y: ";
   for (int i = 0; i < vectorLength; i = i + 1)
   {  y[i] = rand()%10;
      cout << "\t" << y[i];
   }
   cout << "\n";

   cout << "Inner product is: " << innerProduct(x,y) << "\n";

   return 0;
}
```

In the comments of your function, answer the question: Why did the function use the const qualifiers in the parameter list and pass the vectors by reference?

2. Maximum and minimum

Complete the program started below. You must complete the function minMax() that returns both the minimum and the maximum elements of the vector. Here is a function stub and a main routine that you must use to test your function:

```
//File 5.2stub.cpp
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

void minMax(const vector<int>& c, int& min, int& max)
{  //Why were min and max given as "reference parameters"?
   //Answer the question and insert your code after this line.
}

int main(void)
{  int vectorLength = 8;

   vector<int> a(vectorLength);
   cout << " a: ";
   for (int i = 0; i < vectorLength; i = i + 1)
   {  a[i] = rand();
      cout << "\t" << a[i];
   }
   cout << "\n";

   int minimum, maximum;
   minMax(a, minimum, maximum);

   cout << "The minimum array element is: " << minimum << "\n";
   cout << "The maximum array element is: " << maximum << "\n";

   return 0;
}
```

In the comments of your function, answer the question: “Why were min and max given as ‘reference parameters’?”

3. Nothing negative

Complete the program started below. You must complete the function `nothingNegative()` that accepts a vector called `bothSigns` containing positive and negative integers and returns a vector of equal or lesser length that contains only the non-negative integers of the vector `bothSigns`. Here is a function stub and a main routine that you must use to test your function:

```
//File 5.3stub.cpp
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

vector<int> nothingNegative(const vector<int>& a)
{  //Why can this program, when compiled with just the function stub,
   //go into what looks like an infinite loop?
   //Answer the question and insert your code after this line.
}

int main(void)
{  int vectorLength = 10;

   vector<int> bothSigns(vectorLength);
   cout << " Input vector: ";
   for (int i = 0; i < vectorLength; i = i + 1)
   {  bothSigns[i] = rand()%201 - 100;
      cout << "\t" <<  bothSigns[i];
   }
   cout << "\n";

   vector<int> noNegs = nothingNegative(bothSigns);

   cout << " Output vector: ";
   for (int i = 0; i < noNegs.size(); i = i + 1)
   {  cout << "\t" <<  noNegs[i];
   }
   cout << "\n";

   return 0;
}
```

In the comments of your function, answer the question: “Why can this program, when compiled with just the function stub, go into what looks like an infinite loop, or crashes?”

4. Sort this out

Complete the program started below. You must complete the function sort() that has the void return type. This function accepts a vector called unsorted containing random int's and returns the vector with its elements sorted in ascending order. Here is a function stub and a main routine that you must use to test your function:

```
//File 5.4stub.cpp
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

void sort(vector<int>& a){}

int main(void)
{  cout << "Input the minimum and maximum random int: ";
   int minRange, maxRange;
   cin >> minRange >> maxRange;

   cout << "Input the length of the vector: ";
   int vectorLength;
   cin >> vectorLength;

   vector<int> unsorted(vectorLength);
   cout << " Vector beforehand: \t";
   for (int i = 0; i < vectorLength; i = i + 1)
   {  unsorted[i] = minRange + rand()%(maxRange - minRange + 1);
      cout << unsorted[i] << " ";
   }
   cout << "\n";

   sort(unsorted);

   cout << " Vector afterhand: \t";
   for (int i = 0; i < unsorted.size(); i = i + 1)
   {  cout << unsorted[i] << " ";
   }
   cout << "\n";
   return 0;
}
```

5. Primes vector(20%)

Complete the program started below. You must complete the function getPrimes() that has the vector<int> return type. This function accepts an int called NPrimes that is the number of consecutive prime numbers starting with 2. The return vector contains these prime numbers in order.
Feel free to use your algorithm from the last assignemnt to do this. Here is a function stub and a main routine that you must use to test your function:

```
//File 5.5stub.cpp
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

vector<int> getPrimes(const int NPrimes){}

int main(void)
{  cout << "Input the number of primes to find: ";
   int NPrimes;
   cin >> NPrimes;

   vector<int> primes = getPrimes(NPrimes); //Get the primes
   cout << "The first " <<  primes.size() << " prime numbers: ";

   for (int i = 0; i < primes.size(); i = i + 1)
   {  cout << primes[i] << " ";
   }
   cout << "\n";
   return 0;
}
```

6. Nothing negative

Complete the program started below. You must complete the function `noRepeats()` that has the `void` return type. This function accepts a vector called `bothSigns` containing random `int`'s and eliminates any values in it that are repeated. You are not allowed to declare any new vectors! (That's what makes this question tough!) Here is a function stub and a main routine that you must use to test your function:

```
//File 5.6stub.cpp
#include <iostream>
#include <vector>
#include <cstdlib>

using namespace std;

void noRepeats(vector<int>& a){}

int main(void)
{  cout << "Input the minimum and maximum random int: ";
   int minRange, maxRange;
   cin >> minRange >> maxRange;

   cout << "Input the length of the vector: ";
   int vectorLength;
   cin >> vectorLength;

   vector<int> hasRepeats(vectorLength);
   cout << " Vector beforehand: \t";
   for (int i = 0; i < vectorLength; i = i + 1)
   {  hasRepeats[i] = minRange + rand()%(maxRange - minRange + 1);
      cout << hasRepeats[i] << " ";
   }
   cout << "\n";

   noRepeats(hasRepeats);

   cout << " Vector afterhand: \t";
   for (int i = 0; i < hasRepeats.size(); i = i + 1)
   {  cout << hasRepeats[i] << " ";
   }
   cout << "\n";
   return 0;
}
```