Spring, 1995 Name: ______________________
My TA's name is: ______________________ My Section Day and Time : ____________

Com S 228 — Introduction to Data Structures

Test on C++

This test has 3 questions and pages numbered 1 through 5.

Reminders

This test is open notes; you may also use both of the course texts, but no other books.

If you need more space, use the back of a page. Note when you do that on the front.

This test is timed. We will not grade your test if you try to take more than the time allowed. Therefore, before you begin, please take a moment to look over the entire test so that you can budget your time.

Indentation is important to us for "clarity" points; if your code is sloppy or hard to read, you will lose points. To be sure of clarity, please indent and use brackets as described in class. Also for helping functions, include PRE, MODIFIES, and POST comments (as described in class) if the function is more than a line or so long.

Correct syntax also matters. Check your code over for syntax errors. You will lose points if your code has syntax errors.

Subset of C++ You May Use

Unless otherwise stated in a problem, when solving problems you may only use standard features of the language that we discussed in class, and helping functions that you define yourself. So for headers, you may only use macros, objects, and functions declared in the following header files.

```
<limits.h>    <iomanip.h>    <iostream.h>    "prompt.h"    "pretend_bool.h"
```

Parts of C++ You May *Not* Use

Unless otherwise stated in a problem, you are prohibited from using recursion and also prohibited form using the following C++ keywords.

```
asm        catch      class      delete
friend     goto       operator   template
throw      try        virtual
```

1. (15 points) What is the output of the following program?

```
// prob1.C
#include <iostream.h>

int main()
{
  cout << "a. (5 points) " << 10 + 50 * 3 - 10 << endl;

  int bvar = 100;
  cout << "b. (5 points) ";
  cout << (bvar -= 20) << endl;

  int cvar = 4;
  cvar++;
  cout << "c. (5 points) ";
  if ( (cvar > 4 || 22 < 8 - 3) && !(5 == 3 * 1 + 2) ) {
    cout << "first one\n";
  } else {
    cout << "the second\n";
  }
  cout << "The end." << endl;

  return 0;
}
```

2. (30 points) Implement the following function specification.

```
int NumBiggerThan(const int arr[], int size, int n)
  // PRE: size >= 0 && 0 .. size-1 are legal indexes of arr
  // POST: FCTVAL is the number of elements in arr that are
  // strictly larger than n.
```

For example, the output of

```
  int myArr[10] = {-1, 2 , -3, 4, -5, 6, -7, 8, -9, 10};
  int firstResult = NumBiggerThan(myArr, 10, 0);
  int secondResult = NumBiggerThan(myArr, 7, -1);
  cout << "firstResult = " << firstResult << " and ";
  cout << "secondResult = " << secondResult << endl;
```

should be

```
firstResult = 5 and secondResult = 3
```

3. (35 points) There is space to write your answer to this problem on the next page.

   A linguist wants you to write a program to compute the percentage of lower-case vowels in an input text. For purposes of this problem the vowels are considered to be the letters: `a, e, i, o, u`. The program should print the prompt `Type text, end with EOF` on a line, then read characters form standard input (cin) up to end-of-file. It should print the percentage of non-whitespace characters that are vowels; in other words, it should ignore blanks, tabs, and other whitespace. If no non-whitespace characters are input, the program prints the line `No input.`. Otherwise the program prints a line as in the sample below with the percentage. (The main program should always return 0 to the operating system.)

   The formula for the percentage is $100 \times V/(V+O)$, where V is the number of characters `a, e, i, o, and u` seen, and O is the number of other non-whitespace characters). (You needn't worry about the number of digits printed in the output, and you may assume that less than 10^9 non-whitespace characters are input. You may assume that no errors happen on input.)

   The following is a sample session with the program (`^D` indicates where the user typed end-of-file).

```
Script started on Wed Feb  8 16:53:53 1995
% ./prob3.exe
Type text, end with EOF
aeiou euio a aa
^D
Percentage vowels (a,e,i,o,u) is: 100%
% ./prob3.exe
Type text, end with EOF
xyzzy nnn
^D
Percentage vowels (a,e,i,o,u) is: 0%
% ./prob3.exe
Type text, end with EOF
^D
No input.
% ./prob3.exe
Type text, end with EOF
Practice yourself, for heaven's sake,
in little things;
and thence proceed to greater.
^D
Percentage vowels (a,e,i,o,u) is: 35.1351%
% ./prob3.exe
Type text, end with EOF
Epictetus (Discourses, IV, i)
^D
Percentage vowels (a,e,i,o,u) is: 30.7692%
% exit
script done on Wed Feb  8 16:54:59 1995
```

// space for the answer to the previous problem