CSC112 Spring 2011
Fundamentals of Computer Science
Practice Problems

Note: Some of these problem are too long (time-wise) to include on a 50-minute exam. Also, there are too many problems here to complete in a 50-minute exam. But, they review the material, and help prepare you for the exam.

1. Write a C++ function to accept an array of characters (C-style character strings, terminated by the null character `'\0'`) and return the length of the string. *Hint:* Count the number of characters up to the character `'\0'`. The function header should be:

   ```
   int string_length( char * s )
   ```

   Try this exercise again, with the function header:

   ```
   int string_length( char s[] )
   ```

2. Describe how to implement a 2-D array using a pointer to a pointer. Suppose the declaration of an array is:

   ```
   int m, n       ;
   double ** table ;
   ```

   Write a segment of code to allocate memory for a 2-D array with m rows and n columns.

3. A *Hilbert matrix* is a matrix whose entries are given by:

   $$H_{i,j} = \frac{1}{i+j+1}$$

   where $i, j \in \{0, 1, 2, ..., n\}$. Write a segment of code to initialize a 7×7 Hilbert matrix.

4. (**Slightly harder problem**) Write a C++ function to accept two arrays of characters (C-style character strings), `s` and `t`. Find the first occurrence of substring `t` in string `s`. Your function should return a (char) pointer to where the match occurs in string `s`. If no match is found, return the NULL pointer. The function header should be:

   ```
   char * find_string( char *s, char * t )
   ```

5. Write a C++ function to accept the name of a file containing binary integers, and compute the sum of the numbers in the file. The function header should be:

   ```
   int file_sum( char * fname )
   ```

6. Write a C++ function to accept the name of a text file and count the number of sentences in that file. Assume every sentence ends with a period, and there are no other uses of the period character in the file. I.e., read the file one character at a time and count the number of periods. The function header should be:

```
int count_periods( char * fname )
```

7. Give the declaration of a C++ structure (named `struct polynomial`). to implement a polynomial. You may assume that any polynomial that we will use is of degree less than 40. An example of a polynomial is given by:

$$p(x) = 2 + x - 3x^2 + 5x^3 + x^4 - 2x^5$$

Hint: A polynomial can be represented by a list of its coefficients.

8. Using your declaration from problem 7, write a C++ function to evaluate the polynomial at x. The function header should be:

```
double polynomial_eval(struct polynomial & p, double x)
```

9. Suppose the following structure is used to implement an un-ordered list integers:

```
const int max_nums = 1024 ;
struct numlist {
    int n ;                 // n is the number of items on the list
    int nums[ max_nums ] ;
} ;
```

Write a C++ function to find a particular item on the list. Your function should return the position on the list where the item is found; return -1 if the item is not on the list. The function header should be:

```
int find_num( struct numlist & nu, int item )
```

10. Use your result (the position number) from problem 9 to delete the item from the list. Write a C++ function to do this. The function header should be:

```
void delete_num( struct numlist & nu, int position )
```

11. Suppose the following structure is used to implement an alphabetized list of names:

```
const int max_names = 100 ;
struct namelist {
    int n ;                 // n is the number of items on the list
    char * the_names[ max_names ] ;
} ;
```

Write a C++ function to insert a new name in the correct position. The function header should be:

```
void insert_name( struct namelist & nlist, char * new_name )
```

12. What does the following program print ? *Hint:* Draw a picture of memory (boxes labeled a, b, c, p, q, r). Trace the execution of this program line by line, updating your diagram after each line.

```
#include <iostream>
using namespace std ;
int main()
{
   int a, b, c ;
   int * p, * q, * r  ;

   a = 11 ; b = 13 ; c = 17 ;
   p = &a ; q = &b ; r = &c ;
   *q = a ;
   *r = b ;
   *p = c ;
   cout << a << " " << b << " " << c << endl ;
 }
```