Programming Language I
(Spring 2016)

Sheet 2: Functions, Recursion and Arrays

1. **(Find the Error)** Find the error in each of the following program segments and correct the error:

a) #include <iostream>;

b) arraySize = 10; // arraySize was declared const

c) Assume that int b[10] = {};

```
for ( int i = 0; i <= 10; ++i )
      b[ i ] = 1;
```

d) Assume that: int a[3];

```
cout << a[ 1 ] << " " << a[ 2 ] << " " << a[ 3 ] << endl;
```

e)
```
int g( void )
{
        cout<<"Inside function g\n" ;
        int h( void )
        {
         cout<<"Inside function h\n" ;
        }
}
```

f)
```
int sum( int x, int y )
   {
   int result;
   result = x + y;
   }
```

g)
```
int sum( int n )
   {
       if( n == 0 ) {
         return 0;
       }
      else{
         n + sum( n - 1 );
      }
   }
```

h)
```
void f( float a );
{
            Float a;
            cout<< a ;
}
```

i)
```
void product( void )
         {
           Int a, b, c, result;
           cout<<"Enter three
              integers: "
           cin>> a >> b >>c ;
           result = a * b * c;
           cout<<"Result is "<<
              result;
           return result;
           }
```

j)
```
// function prototype
float cube( float );

// function definition
cube( float number )
{
  return number * number * number;
}
```

2. Write a function using recursion to print numbers from n to 0

3. For each of the following sets of integers, write a single statement that will print a number at random from the set.
   a) 2, 4, 6, 8, 10.
   b) 3, 5, 7, 9, 11.
   c) 6, 10, 14, 18, 22.

4. **(Distance Between Points)** Write function distance that calculates the distance between two points (x1, y1) and (x2, y2). All numbers and return values should be of type double.
   The distance formula is Distance = $\mathrm{sqrt}\ ((x2-x1)^2+ (y2-y1)^2)$

5. **Greatest Common Divisor**, The greatest common divisor (GCD) of two integers is the largest integer that evenly divides each of the two numbers.
   a. Write function **gcd** that returns the greatest common divisor of two integers.
   b. Write a recursive function **gcd** that returns the greatest common divisor of x and y. The **gcd** of x and y is defined recursively as follows: If y is equal to 0, then **gcd**(x, y) is x; otherwise **gcd**(x, y) is **gcd**(y, x % y) where % is the remainder operator.

6. **Recursive Exponentiation**, Write a recursive function
   power(base, exponent) that when invoked returns $\mathrm{base}^{\mathrm{exponent}}$
   For example, power(3, 4) = 3 * 3 * 3 * 3.
   Assume that exponent is an integer greater than or equal to 1. Hint: The recursion step would use the relationship
   $$\mathrm{base}^{\ \mathrm{exponent}} = \mathrm{base} * \mathrm{base}^{\mathrm{exponent}-1}$$
   and the terminating condition occurs when exponent is equal to 1 because base^1 = base

7. Write a recursive function that finds and returns the minimum element in an array, where the array and its size are given as parameters.

8. Write a recursive function that takes in one argument n and computes Fn, the nth value of the Fibonacci sequence. Recall that the Fibonacci sequence is defined by the relation
   Fn = Fn−1 + Fn−2 where F0 = 0 and F1 = 1
   Trace the function calling when applied to F5.

9. Write C++ statements to accomplish each of the following:
   a. Display the value of element 6 of character array f.

b. Input a value into element 4 of one-dimensional floating-point array b.
c. Initialize each of the 5 elements of one-dimensional integer array g to 8.
d. Total and print the elements of floating-point array c of 100 elements.

10. Write a program to copy array a into the first portion of array b. Assume double a[11], b[34];

11. Write a function that, given an array of integers and an integer n, reverses the first n elements of the array.

12. Write a program to read names and phone numbers into two arrays. Request a name and print the person's phone number. Use at least one function.

13. The number 27472 is said to be **palindromic** since it reads the same forwards or backwards. Write a function that, given an integer n, returns 1 if n is palindromic and 0 if it is not.

14. Use a **one-dimensional array** to solve the following problem. Read in 20 numbers, each of which is between 10 and 100, inclusive. As each number is read, validate it and store it in the array only if it isn't a duplicate of a number already read. After reading all the values, display only the unique values that the user entered.

15. Write a function **indexOfthat**, given a string s and a character c, returns the position of the first occurrence of c in s. If c is not in s, return -1. For example**, indexOf**("brother",'h') returns 4 but
indexOf("brother", 'a') returns -1.

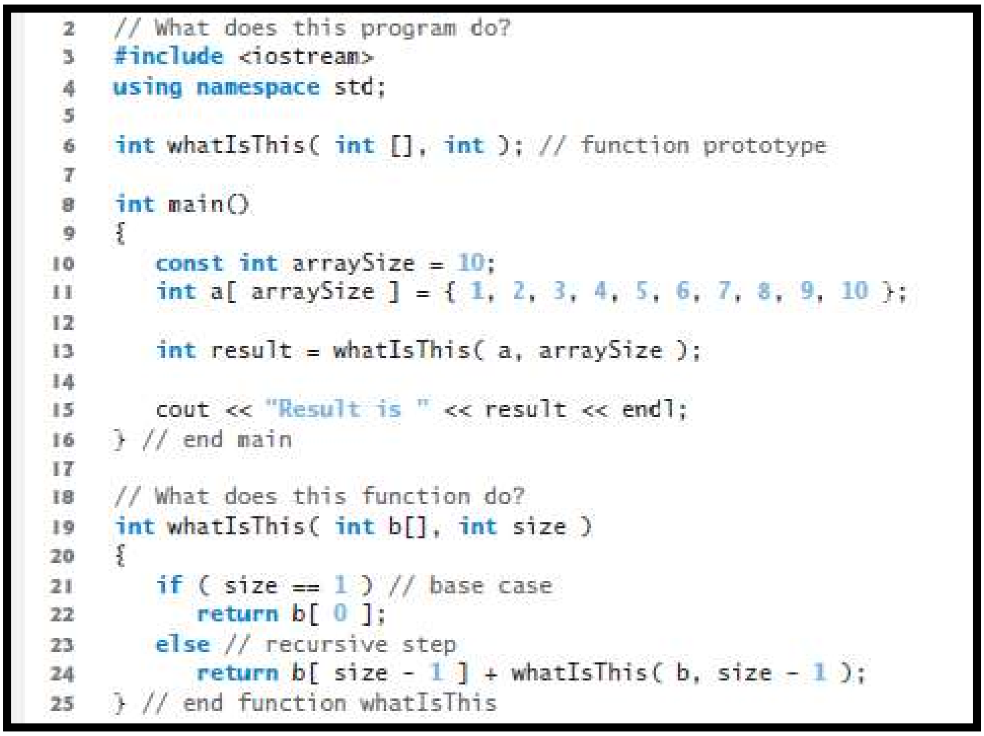
16. In an **election**, there are seven candidates. Each voter is allowed one vote for the candidate of his/her choice. The vote is recorded as a number from 1 to 7. The number of voters is unknown beforehand but the votes are terminated by a vote of 0. Any vote that is not a number from 1 to 7 is an invalid (spoilt) vote. Write a program to read the data in an array and evaluate the results of the election. Your output should specify the total number of votes, the number of valid votes, and the number of spoilt votes.

17. Write function **Merge** that merges two given sorted arrays into one sorted array. Example: Merge({1,2,4,15},{2,3,6,30},{}) the output ,which should be placed in that last parameter, is {1,2,2,3,4,6,15,30}

18. Write a function Remove that removes the values of specific range, specified by starting index and last index, in an array. Example Remove({9,3,5,4,7,5},1,3)
    The sent array {9,3,5,4,7,5} should be modified into {9,7,5,0,0,0}

19. Write a recursive function **recursiveMinimum** that takes an integer array, a starting subscript and an ending subscript as arguments, and returns the smallest element of the array. The function should stop processing and return when the starting subscript equals the ending subscript.

20. What does the following program do?



```
// What does this program do?
#include <iostream>
using namespace std;

int whatIsThis( int [], int ); // function prototype

int main()
{
   const int arraySize = 10;
   int a[ arraySize ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

   int result = whatIsThis( a, arraySize );

   cout << "Result is " << result << endl;
} // end main

// What does this function do?
int whatIsThis( int b[], int size )
{
   if ( size == 1 ) // base case
      return b[ 0 ];
   else // recursive step
      return b[ size - 1 ] + whatIsThis( b, size - 1 );
} // end function whatIsThis
```