CPSC 122
Test 1

Name Key

Regarding questions with pre/post conditions:

- The pre condition is what must be true for the function to rum
- The post condition describes what is true after the function has run successfully

Each question is worth 6 points

1. Write the following function:
pre: *limit* is an integer >= 2
post: returns the sum of the integers in the closed interval [1..limit].
int sum(int limit)

```
{
    int ct=0
    for(int i=1; i<=limit; i++)
        ct = ct + i;
    return ct;
}
```

limit	ct	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

2. Write the following function:
pre: *stuff* is an integer array, *size* is the number of positions in the array, *target* is what you are looking for
post: returns the index of *target* if *target* is in the array, -1 otherwise
int linearSearch(int stuff[], int size, int target)

```
{
    int i = 0;
    while(i < size)
    {
        if (stuff[i] == target)
            return i;
        i++; -1;
    }
    return -1;
}
```

3. Write the following function
pre: *num* is an integer >= 0
post: returns the factorial of num. Example: fact(0) = 1, fact(3) = 3 * 2 * 1
int fact(int num)

```
{
    int f=1;
    while(num > 1)
    {
        f=f* num;
        num--;
    }
    return f;
}
```

num	f
5	1
4	5
3	20
2	60
1	120

4. Write the following function
pre: num is a positive integer
post: returns true if *num* is even, false otherwise
bool isEven(int num)

```
{
  if (num%2==0)
    return true;
  return false;
}
```

5. Write the following function by invoking the function you wrote for problem 4
pre: *stuff* is an array containing *size* integers
post: returns the number of odd integers in the array.
int odds(int stuff[], int size)

```
{
  int ct=0;
  for(int i=0; i<size; i++)
    if(!isEven(stuff[i]))
      ct++;
  return ct;
}
```

6. What is the Big O complexity of the algorithm necessary for problem 4

O(1)

7. What is the Big O complexity of the algorithm necessary of problem 5

O(N)

8. Write the following function
pre: *num* is an integer >=2
post: returns true if *num* is prime, false otherwise
Note: Use the less efficient method
bool isPrime(int num)

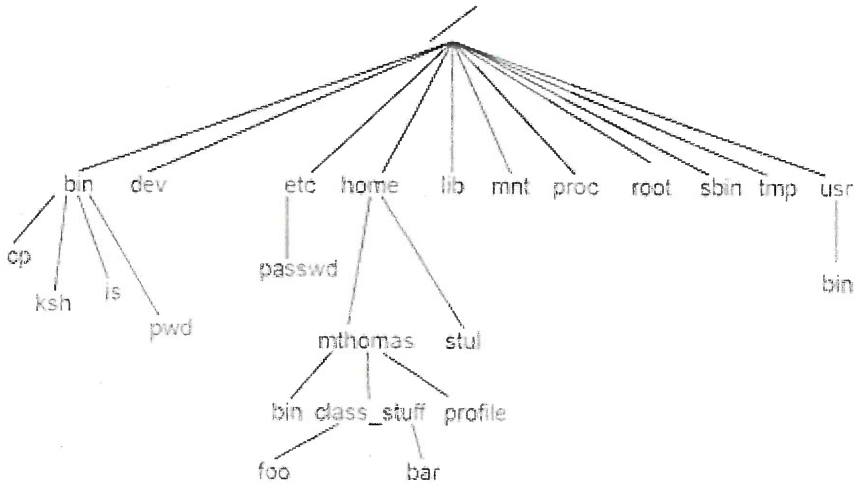
```
{
  int i=2;
  while(i<num)
  {
    if(num%i==0)
      return false;
    ++i;
  }
  return true;
}
```

O(n²) Selection sort

9 - 11. We discussed several Big O complexity classes and their associated algorithms. Fill in the following table with three of the classes and example algorithms.

Complexity Class	Algorithm
O(1)	dictionary / hash table
O(√n)	is-prime
O(log n)	binary Search
O(n)	linear search

Your username is *mthomas*. You have just logged on to the system. Every item listed below is a directory. Write command line Linux commands to do what is requested.



12. Cause everything in the current directory to be displayed on the screen

ls

13. With a single command, make *bar* the current directory.

cd class_stuff/bar

14. Assume *bar* is the current directory and that it contains a file x.cpp. Delete it..

rm x.cpp

15. Write code using argc and argv to display all of the command line arguments in the code snippet below.

```
#include <iostream>
using namespace std;
int main(argc, char* argv[])
{
for(int i=0; i<argc; i++)
    cout << argv[i] << endl;

}
```

Bonus (+6)

Think of the following integers as being stored in an array. Show the contents of the array after 2 passes of selection sort. We are sorting smallest to largest.

7
12
1
3
5