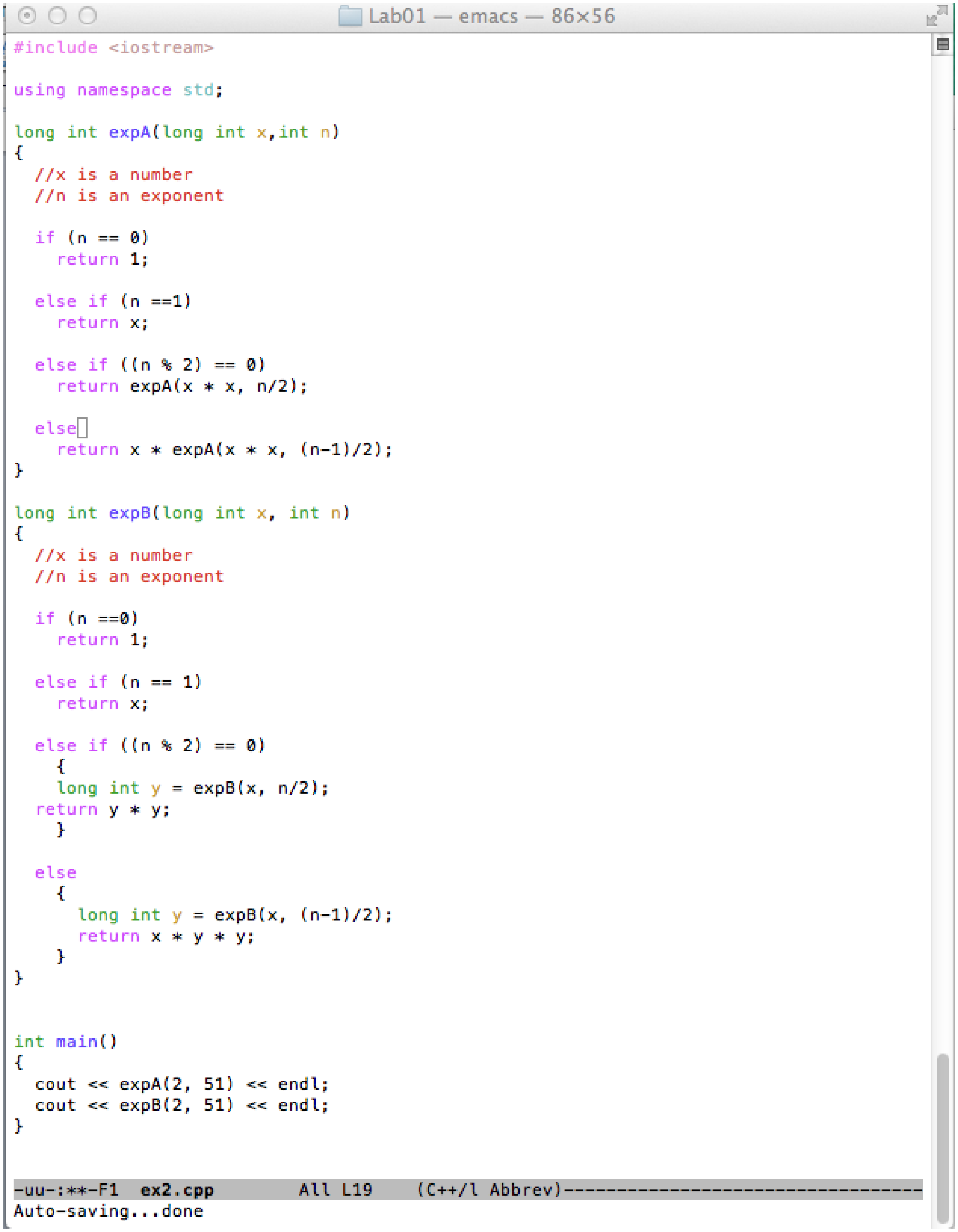
Mahesh Gautam
CS 315
Fall 2014
Lab 01

Exercises:

1) All values resulting from the equation **‘3n+1’** will halt the call to function f(n).
   Example:
   When n = 1; 3n+1 = 4
   So, return f(n-3) + 1 would return
   f(1) + 1
   and this will halt the program.

Please proceed to next page for Exercise #2

2) Code:



```
#include <iostream>

using namespace std;

long int expA(long int x,int n)
{
  //x is a number
  //n is an exponent

  if (n == 0)
    return 1;

  else if (n ==1)
    return x;

  else if ((n % 2) == 0)
    return expA(x * x, n/2);

  else
    return x * expA(x * x, (n-1)/2);
}

long int expB(long int x, int n)
{
  //x is a number
  //n is an exponent

  if (n ==0)
    return 1;

  else if (n == 1)
    return x;

  else if ((n % 2) == 0)
    {
    long int y = expB(x, n/2);
  return y * y;
    }

  else
    {
      long int y = expB(x, (n-1)/2);
      return x * y * y;
    }
}


int main()
{
  cout << expA(2, 51) << endl;
  cout << expB(2, 51) << endl;
}


-uu-:**-F1  ex2.cpp        All L19     (C++/l Abbrev)-----------------------------------
Auto-saving...done
```

2^{51} = 2251799813685248

3)

a) f (1000)

= 1 + f(500)

= 1 + 1 + f(250)

= 1 + 1 + 1 + f(125)

= 1 + 1 + 1 + 2 + f(62)

= 1 + 1+ 1 + 1 + 2 + f(31)

= 1 + 1 + 1 + 1 + 2 + 2 + f(15)

= 1 + 1 + 1 + 1 + 2 + 2 + 2 + f(7)

= 1 + 1 + 1 + 1 + 2 + 2 + 2 + 2+ f(3)

= 1 + 1 + 1 + 1 + 2 + 2 + 2 + 2+ 2 + f(1)

= 1 + 1 + 1 + 1 + 2 + 2 + 2 + 2+ 2 + 0

= 14

b) f (2^k)?

= K

The answer is 'K' because f is always even and also 'k' is a power of 2. So, 2 keeps multiplying itself times 'K'.

c) f (2^k-1)?

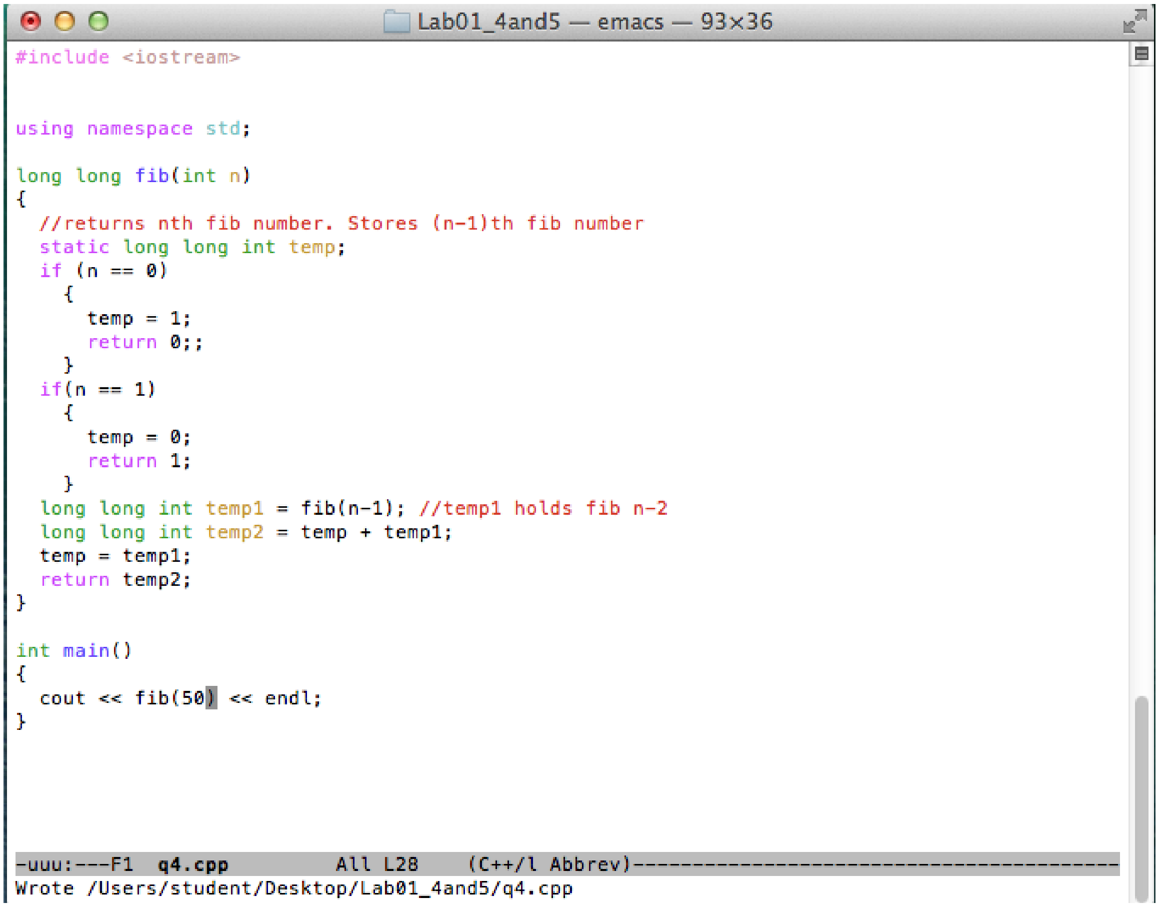
= 2K -2

= 2(k-1)

4) F(n) = F(n-1) + F(n-2)

- How many times is a call for F(1) made when computing F(10)?
  F(1) is called **55** times when computing F(10)

- Write a recursive function for computing the nth Fibonacci number that avoids redundancy.
  = see code below (screenshot)

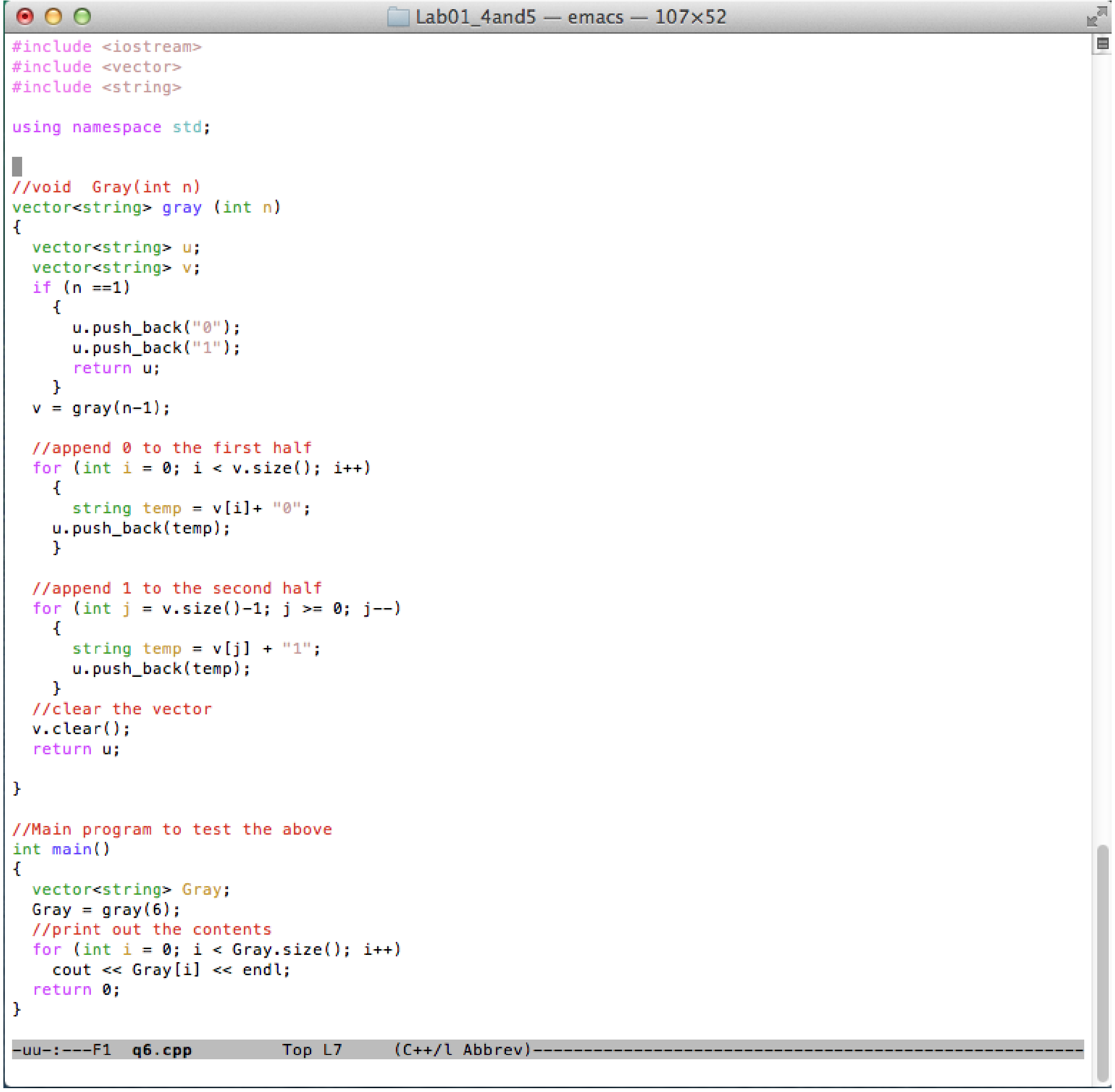


```
#include <iostream>


using namespace std;

long long fib(int n)
{
  //returns nth fib number. Stores (n-1)th fib number
  static long long int temp;
  if (n == 0)
    {
      temp = 1;
      return 0;;
    }
  if(n == 1)
    {
      temp = 0;
      return 1;
    }
  long long int temp1 = fib(n-1); //temp1 holds fib n-2
  long long int temp2 = temp + temp1;
  temp = temp1;
  return temp2;
}

int main()
{
  cout << fib(50) << endl;
}
```

```
-uuu:---F1  q4.cpp        All L28    (C++/l Abbrev)------------------------------------------
Wrote /Users/student/Desktop/Lab01_4and5/q4.cpp
```

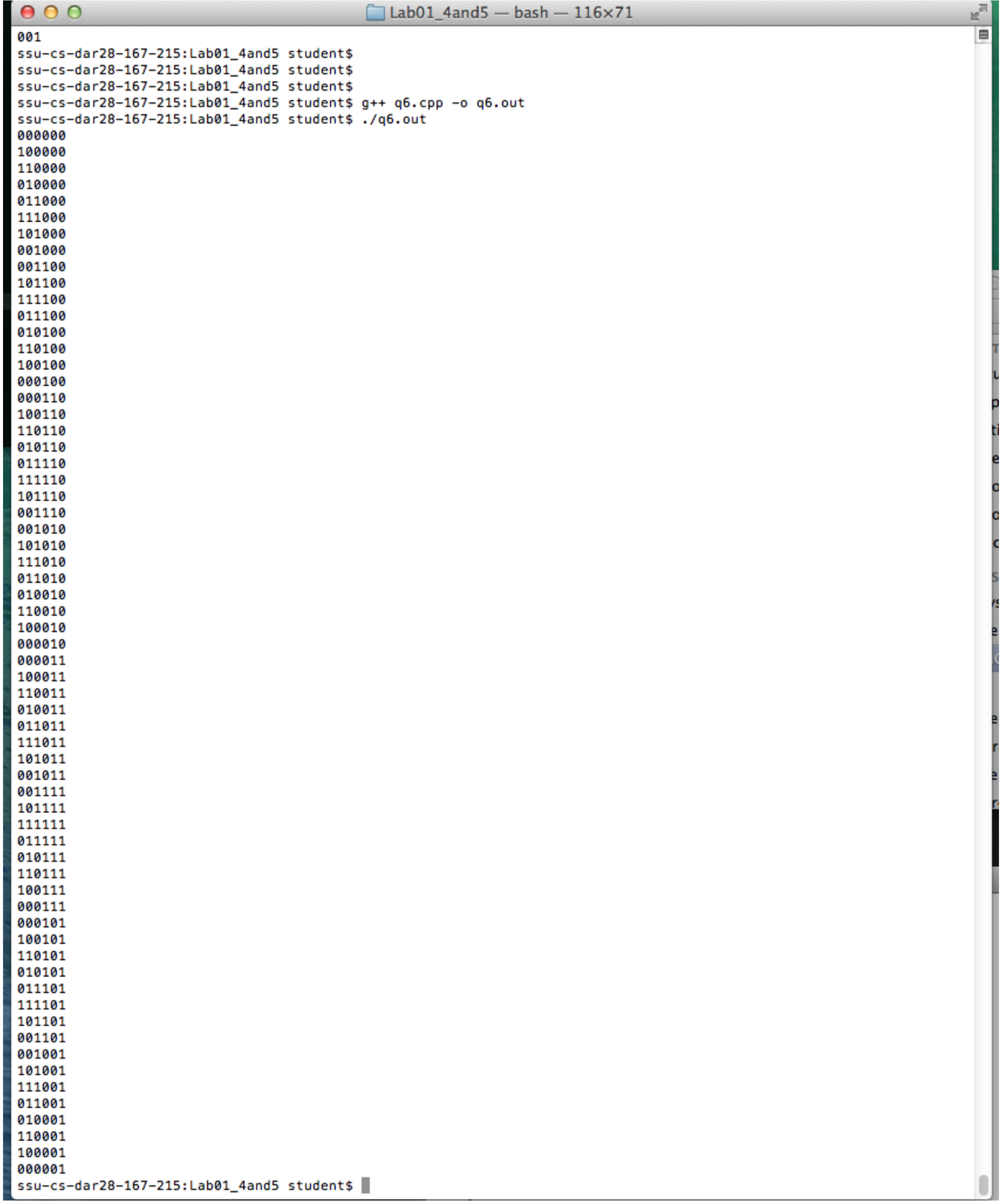
- Use your function to compute f(50).
  f (50) = 12586269025

5) Code:


Lab01_4and5 — emacs — 107×52
-uu-:---F1 q6.cpp Top L7 (C++/l Abbrev)

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;


//void  Gray(int n)
vector<string> gray (int n)
{
  vector<string> u;
  vector<string> v;
  if (n ==1)
    {
      u.push_back("0");
      u.push_back("1");
      return u;
    }
  v = gray(n-1);

  //append 0 to the first half
  for (int i = 0; i < v.size(); i++)
    {
      string temp = v[i]+ "0";
    u.push_back(temp);
    }

  //append 1 to the second half
  for (int j = v.size()-1; j >= 0; j--)
    {
      string temp = v[j] + "1";
      u.push_back(temp);
    }
  //clear the vector
  v.clear();
  return u;

}

//Main program to test the above
int main()
{
  vector<string> Gray;
  Gray = gray(6);
  //print out the contents
  for (int i = 0; i < Gray.size(); i++)
    cout << Gray[i] << endl;
  return 0;
}
```

Output when n = 6



```
001
ssu-cs-dar28-167-215:Lab01_4and5 student$
ssu-cs-dar28-167-215:Lab01_4and5 student$
ssu-cs-dar28-167-215:Lab01_4and5 student$
ssu-cs-dar28-167-215:Lab01_4and5 student$ g++ q6.cpp -o q6.out
ssu-cs-dar28-167-215:Lab01_4and5 student$ ./q6.out
000000
100000
110000
010000
011000
111000
101000
001000
001100
101100
111100
011100
010100
110100
100100
000100
000110
100110
110110
010110
011110
111110
101110
001110
001010
101010
111010
011010
010010
110010
100010
000010
000011
100011
110011
010011
011011
111011
101011
001011
001111
101111
111111
011111
010111
110111
100111
000111
000101
100101
110101
010101
011101
111101
101101
001101
001001
101001
111001
011001
010001
110001
100001
000001
ssu-cs-dar28-167-215:Lab01_4and5 student$ 
```