**Problem 1:**

for (int j = 0; j < 15; j++)

{

counter = 0;

binary\_search(a, 0, 14, j);

cout << "Value of v[" << j << "] = " << counter << endl;

}

Value of probe [0] = 4

Value of probe [1] = 4

Value of probe [2] = 3

Value of probe [3] = 4

Value of probe [4] = 2

Value of probe [5] = 4

Value of probe [6] = 3

Value of probe [7] = 4

Value of probe [8] = 1

Value of probe [9] = 4

Value of probe [10] = 3

Value of probe [11] = 4

Value of probe [12] = 2

Value of probe [13] = 4

Value of probe [14] = 3

**Problem 2:**

int sum = 0;

for (int j = 0; j < 15; j++)

{

counter = 0;

binary\_search(a, 0, 14, j);

sum = sum + counter;

cout << "Value of probe[" << j << "] = " << counter << endl;

}

int average = sum / 15;

cout << "Average probes is: " << average << endl;

Average probes is: 3

**Problem 3:**

Value of missing[0] = 4

Value of missing[1] = 3

Value of missing[2] = 2

Value of missing [3] = 3

Value of missing [4] = 1

Value of missing [5] = 3

Value of missing [6] = 2

Value of missing [7] = 3

Value of missing [8] = 0

Value of missing [9] = 3

Value of missing [10] = 2

Value of missing [11] = 3

Value of missing [12] = 1

Value of missing [13] = 3

Value of missing [14] = 2

The number of probes vary depending on how far away it is from the value is from the key. Values outside the boundary always require 2 probes.

**Problem 4:**

Average probes is: 2

**Problem 5/6:**

int revised\_linear\_search\_iterator(const vector<int>& v, int key, int beginLocation)

{

int i = beginLocation;

while (i <= (v.size() - 1))

{

if (v[i] > key)

{

return v.end();

}

else if (v[i] == key)

{

return i;

}

i++;

}

return -1;

}