Devin Hardy

CS472

Assignment 3

Question 1:

```
// a = min, b = max
int RANDOM(int a, int b)
{
   static random_device dev;
   mt19937 eng{dev()};
   uniform_int_distribution<int> dist{a, b};
   int num = dist(eng);
   return num;
}
```

Question 2:

```
// pass by reference vector
// pass by value size n
void RndmizeVec(vector<int>& vect, int n)
{
   int tempNum = 0;
   int tempPlace = 0;
   for(int i = 0; i <= (n-1); i++)
   {
      tempPlace = RANDOM(i, (n-1));
      tempNum = vect[i];
      vect[i] = vect[tempPlace];
      vect[tempPlace] = tempNum;
   }
```

```
}
```

Question 3:

```
//Function to find target in vector
bool Find(vector<int> vect, int target)
{
   vector<int>::iterator it;
   it = find(vect.begin(), vect.end(), target);
   if(it != vect.end())
      return true;
   else
      return false;
}

vector<int> RandomSample(int n, int m)
{
   vector<int> sample;
   if(m == 0)
      return sample;
   else
   {
      sample = RandomSample(n-1, m-1);
      int i = RANDOM(1, n);
      if(Find(sample, i))
      {
         sample.push_back(n);
      }
      else
      {
         sample.push_back(i);
```

```
            ranCall++;
        }
        return sample;
    }

}
```

Size	5	10	25	50	100
100	5	10	25	35	5
200	5	10	25	46	73
300	5	9	25	47	83
400	5	10	25	49	86
500	5	10	25	49	91
600	5	10	25	48	89
700	5	10	25	47	91
800	5	10	25	46	91
900	5	10	25	47	92
1000	5	9	25	48	92

The claim was that the algorithm would generate less calls to random() if n is significantly larger than m.

For the data I have collected. What was calculated by counting the number of times the random number was added to the Sample list. The closer m was to n generated less calls.