Question 1

Prove that the sum of the n first odd positive integers is n^2, i.e., 1 + 3 + 5 + · · · + (2n − 1) = n^2

Let A and B be two events. Suppose that the probability that neither A or B occurs is 2/3. What is the probability that one or both occur?

Question 2

Let A be the following event:
If given two dice, what is the probability that the sum of the two numbers rolled will be at least 9?

Let B be the following event:
If given two dice, what is the probability that the two numbers rolled will be the same?

What is the probability of B given A?

Question 3: Counting

There are 6 chipmunks and 7 zebras in a ballroom dancing class. If 4 chipmunks and 4 zebras are chosen and paired off, how many pairings are possible? (Note: each zebra and chipmunk is considered a unique, separate being).

Question 4: Pointers

Determine the output of the following code:

```
int f(int* n, int m){
      *n = 10;
      m = 10;
      return *n + m;
}



int main(){
      int n = 5;
      int m = 5;
      int res = f(&n, m);
      cout << res + n + m << endl;
}
```

Question 5: Counting

There are 100 balls in a bucket:

- 30 red balls numbered 1, 2, 3, …, 30.
- 70 green balls numbered 1, 2, 3, …, 70.

In how many ways can you pick 20 balls, such that there will be exactly *k* red balls (without replacement)?

Explain your answer.

Question 6: Expectation

In the following game, a fair coin is tossed until either a head comes up or four tails come up. Let X be the random variable that denotes the number of tosses made in the game.

a. Find the distribution of X. That is, for each possible value of X, say what is the probability X would get that value.
b. What is E[X]? That is, find the expected value of X.

Explain your answers.

Question 7: Algorithm Analysis

a)

```
void f(int* n, int* m, int n_size, int m_size) {
    for (int i = 0; i < n_size; i++) {
         for (int j = 0; j < m_size; j++) {
           // Some O(1) operation here
           }
     }
}
```

b)

```
void f(int n){
   for (int i = 0; i < n; i++) {
       for (int j = 0; j < i; j++) {
        //O(1) work here
       }
   }
}
```

c)

```
void f(int n){
   for (int i = n; i >= 0; i /= 2) {
       for (int j = 0; j < 1000; j++) {
           //O(1) operations
       }
   }
}
```

Question 8: Coding

Given an array of numbers, write a function to move all 0s to the end of the array while maintaining the relative order of the non-zero elements. Do this **in-place**. This should run in $\Theta(n)$.
Example: [0,2,0,1,0] -> [2, 1, 0, 0, 0]

Given an array of nums, find the length of the longest sequence of zeroes **recursively**. You can use the `std::max` function.
Example: `maxZeroLength([0,0,1,0,0,0], 6, 0) = 3`