Lab 05
Data Structures
BS DS Fall 2024 Morning/Afternoon

Task 1

Write the recursive function ***dec2oct(n)*** to return the octal number equivalent to its integer parameter. Test your function using the following driver.

```
int main() {
   int n=69;
   cout << "Octal equivalent of " << n<<" is : "<<dec2oct(n) << endl;
   n = 389;
   cout << "Octal equivalent of " << n<<" is : "<<dec2oct(n) << endl;
   return 0;
}
```

Task 2

A digit string is good if the digits (0-indexed) at even indices are even and the digits at odd indices are prime.

Sr #	**Input**	**Expected Output**	**Description**
1	02468	Not Good	Number 6 at index 3 is not a prime number.
2	23478	Good	All digits at even indices are even, and all digits at odd indices are prime. Therefore, the digit string is considered "good".
3	224365	Good	All digits at even indices are even, and all digits at odd indices are prime. Therefore, the digit string is considered "good".

Write a recursive function which takes a digit string as input and identify the given digit string is a Good or Not.

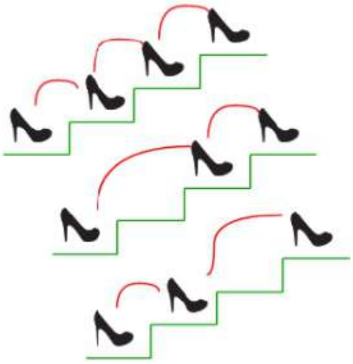
bool isGoodNumber(const string &s, int index = 0)

Use the following drive to test your function.

```
int main() {
   string digit_strings[] = {"02468", "23478", "224365"};
   for (string digits : digit_strings) {
      bool is_good = isGoodNumber(digits, 0);
      cout << "Digit string: " << digits
          << " is " << (is_good ? "good" : "not good") << endl;
   }
   return 0;
}
```

Task 3

You're standing at the base of a staircase and are heading to the top. A small stride will move up one stair, a large stride advances two. You want to count the number of ways to climb the entire staircase based on different combinations of large and small strides. For example, a staircase of three steps can be climbed in three different ways: via three small strides or one small stride followed by one large stride or one large followed by one small. A staircase of four steps can be climbed in five different ways (enumerating them is an exercise left to reader :-).



Write the recursive function **int countWays(int numStairs)** that takes a positive **numStairs** value and returns the number of different ways to climb a staircase of that height taking strides of one or two stairs at a time.

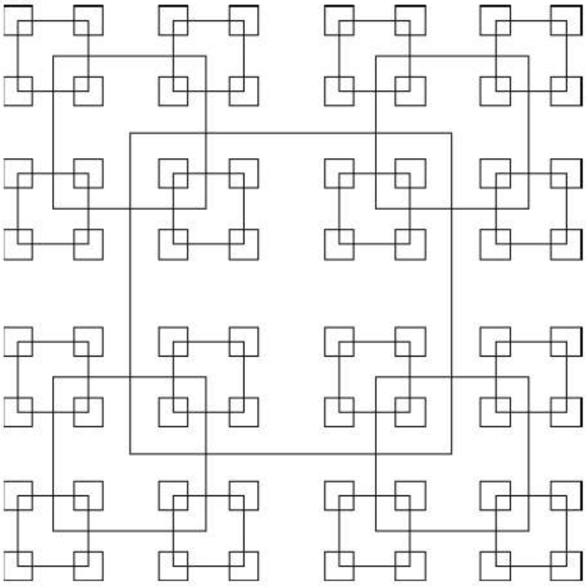
Task 4

Geometrically, any square has a unique, well-defined centre point. On a grid this is only true if the sides of the square are an odd number of points long. Since any odd number can be written in the form 2k+1, we can characterize any such square by specifying k, that is we can say that a square whose sides are of length 2k+1 has size k. Now define a pattern of squares as follows.

1. The largest square is of size k (that is sides are of length 2k+1) and is centred in a grid of size 1024 (that is the grid sides are of length 2049).
2. The smallest permissible square is of size 1 and the largest is of size 512, thus 1 ≤512≤k .
3. All squares of size $k > 1$ have a square of size ***k div 2*** centred on each of their 4 corners. (Div implies integer division, thus 9 div 2 = 4).
4. The top left corner of the screen has coordinates (0,0), the bottom right has coordinates (2048, 2048).

Hence, given a value of k, we can draw a unique pattern of squares according to the above rules.

Furthermore any point on the screen will be surrounded by zero or more squares. (If the point is on the border of a square, it is considered to be surrounded by that square). Thus if the size of the largest square is given as 15, then the following pattern would be produced.



Write a program that will read in a value of k and the coordinates of a point, and will determine how many squares surround the point.

Input and Output

Input will consist of three integers, first integer shows the value of k and other two integers shows the coordinates of a point.

Output will consist of a integer which shows the number of squares containing the specified point.

Sample input 1

500 113 941

Sample output 1

5

Sample input 2

300 100 200

Sample output 2

0

Sample input 3

300 1024 1024

Sample output 3

1