Exercise 2: Skip Counting [50 marks]

Problem Statement

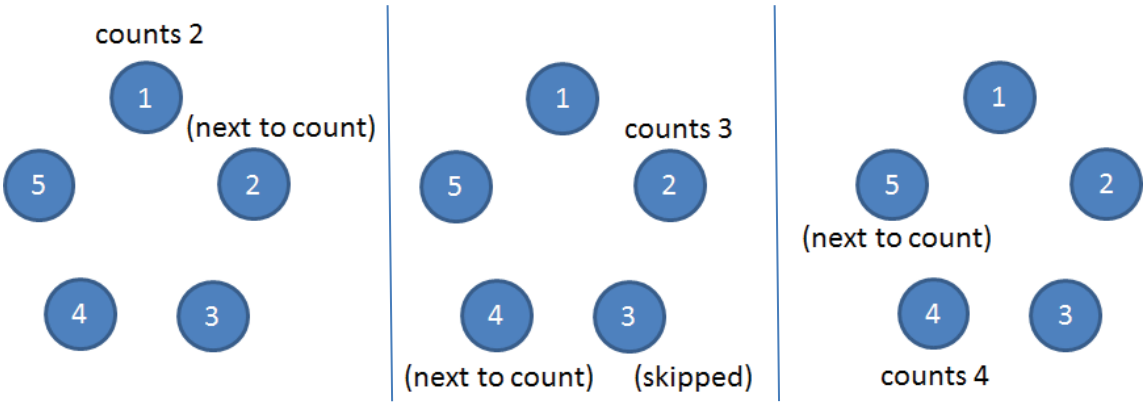
It is the eve of Chinese New Year and the Tan family is having their reunion dinner together as usual. This year, Mr. Tan is planning to give a big red packet to one of his 5 children but he has problem in deciding which child should get it. Therefore, he asks the children to sit around the table and play a game called skip counting.

The rules of the game are as follows:

- The five seats around the table are numbered from 1 to 5 in clockwise direction.
- A starting position, a lower bound and an upper bound are chosen by Mr. Tan.
- The children count the integers from the lower bound to the upper bound (both inclusive) one by one starting from the child at the starting position in clockwise direction. (Since the children are seated around the table, after position 5 comes position 1.)
- If the number being counted contains *n* odd digits, the next *n* positions are skipped. (Do take note that any number, regardless of whether it is odd or even, may contain odd digits. For example, 36 contains 1 odd digit, which is 3, while 35 contains 2 odd digits, which are 3 and 5.
- After the counting is done, the next child to count is the winner of the game.

For example, if the starting position is 1 and the lower bound and upper bound are 2 and 4, respectively. The game proceeds as follows:

- The child at position 1 starts by counting 2. Since 2 does not contain any odd digits, the next child to count is at position 2.
- The child at position 2 counts 3. Since 3 contains *1* odd digit, the next *1* position (i.e., position 3) is skipped and the next child to count is at position 4.
- The child at position 4 counts 4. Since 4 does not contain any odd digit, the next child to count is at position 5.
- At this point, there are no more numbers to count and the child at position 5 gets the red packet.



You are to write a program to play this game and decide which child gets the red packet given a starting position, a lower bound and an upper bound.

Write on the skeleton file **skip.c** given to you. You need to include two functions:

- **`int skipCount(int startPos, int lower, int upper)`**

  This function takes in a starting position `startPos`, a lower bound `lower` and an upper bound `upper`. It returns the ending position (i.e., the next position to count after the skip counting is done).

- **`int countOddDigits(int num)`**

  This function takes in an integer and counts the number of odd digits in it.

You may define additional functions as needed.

You may assume that the inputs are valid: 1) the starting position is an integer between 1 and 5 (both inclusive), 2) the lower bound and the upper bound are both positive integers, and 3) the lower bound is not greater than the upper bound.

Sample Runs

Three sample runs are shown below with user input highlighted in **bold**.

```
Enter starting position: 1
Enter lower bound and upper bound: 2 2
The ending position is 2.
```

```
Enter starting position: 1
Enter lower bound and upper bound: 3 3
The ending position is 3.
```

```
Enter starting position: 5
Enter lower bound and upper bound: 1 2
The ending position is 3.
```