Introduction

Knightro has been assigned a series of computational tasks by the UCF College of Sciences. Each task involves solving a problem using **recursion**, and Knightro needs your help to complete them efficiently. Your mission is to implement recursive solutions for the following challenges:

Mission 1: Counting Trailing Zeros in Factorial

Knightro is analyzing factorial-based data from UCF's math archives. He needs to count how many zeros appear at the end of $n!$.

The current iterative solution is:

```
int zeros(int n) {
    int res = 0;
    while (n != 0) {
        res += n / 5;
        n /= 5;
    }
    return res;
}
```

Task: Rewrite this function recursively:

```
int zeros(int n);
```

Mission 2: Palindrome Check for Knightro's Secret Codes

Knightro receives encrypted messages that must be verified for symmetry. A message is valid if it reads the same forward and backward.

Task: Write a recursive function:

```
int isPalindrome(char *str, int left, int right);
```

Return 1 if the string is a palindrome, 0 otherwise.

Mission 3: Finding Max/Min in Knightro's Data Logs

Knightro is scanning sensor logs from UCF's Pegasus satellites. He needs to find the maximum and minimum values in a dataset using recursion.

Task: Write two recursive functions:

```
int findMax(int arr[], int start, int end);
int findMin(int arr[], int start, int end);
```

Mission 4: Search in Knightro's Sorted Archives

Knightro is preparing to search through UCF's digitized student archives. These records are sorted by ID numbers, and Knightro needs a recursive tool to locate a specific entry efficiently.

Task: Write a recursive function that takes in a sorted array `numbers`, two integers `low` and `high` representing the range of indices to search within, and another integer `value` representing the target record ID.

The function should return the index where `value` is found between `low` and `high`, inclusive. If `value` is not found in that range, return `-1`.

```
int searchRecord(int numbers[], int low, int high, int value);
```