Assignment Solution: Recursion -1

Print an increasing-decreasing sequence using recursion
Example:
If n = 5, the output should be:
1 2 3 4 5 4 3 2 1
Ans:

```
#include <iostream>
using namespace std;
// Function to print increasing sequence
void printIncreasing(int current, int n) {
if (current > n) {
return;
}
cout << current << " ";
printIncreasing(current + 1, n);
}
// Function to print decreasing sequence
void printDecreasing(int current) {
if (current < 1) {
return;
}
cout << current << " ";
printDecreasing(current - 1);
}
// Combined function to print increasing and decreasing sequence
void printSequence(int n) {
printIncreasing(1, n);
printDecreasing(n - 1);
cout << endl;

}
int main() {
int n;
cout << "Enter a number: ";
cin >> n;
cout << "Increasing-Decreasing sequence: ";
printSequence(n);
return 0;
```

Write a program to print the sum of all odd numbers from a to b (inclusive) using recursion.
Ans:

```
#include <iostream>
```

```
using namespace std;
// Function to calculate the sum of all odd numbers from a to b using recursion
int sumOddNumbers(int a, int b) {
if (a > b) {
return 0;
}
if (a % 2 != 0) {
return a + sumOddNumbers(a + 1, b);
} else {
return sumOddNumbers(a + 1, b);
}
}
int main() {
int a, b;
cout << "Enter the start number (a): ";
cin >> a;
cout << "Enter the end number (b): ";
cin >> b;
int sum = sumOddNumbers(a, b);
cout << "The sum of all odd numbers from " << a << " to " << b << " is: " << sum << endl;
return 0;
}
```

Given a positive integer, return true if it is a power of 2 using recursion.
Ans:

```
#include <iostream>
using namespace std;
// Function to check if a number is a power of 2 using recursion
bool isPowerOfTwo(int n) {
if (n == 1) {
return true;
}
if (n % 2 != 0 || n == 0) {
return false;
}
return isPowerOfTwo(n / 2);
}
int main() {
int n;
cout << "Enter a positive integer: ";
cin >> n;
if (isPowerOfTwo(n)) {
cout << n << " is a power of 2." << endl;
} else {
cout << n << " is not a power of 2." << endl;
}
return 0;
```

There are n stairs, and a person standing at the bottom wants to climb the stairs to reach the nth stair. The person can climb either 1, 2, or 3 stairs at a time. Write a program to count the number of ways the person can reach the top using recursion.
Ans:

```
#include <iostream>
using namespace std;
// Function to count the number of ways to reach the nth stair
int countWays(int n) {
if (n == 0) {
return 1; // One way to stay at the ground (do nothing)
}

if (n < 0) {
return 0; // No way to go below the ground
}
return countWays(n - 1) + countWays(n - 2) + countWays(n - 3);
}
int main() {
int n;
cout << "Enter the number of stairs: ";
cin >> n;
int ways = countWays(n);
cout << "Number of ways to reach the top: " << ways << endl;
return 0;
}
```