Lab10 Recursion - Solution

Exercise 1: Sum of all integers from 1 to n (30 min)

Write a *recursive* function to calculate the **sum of all integers from 1 to n**..Then write a *non-recursive version* and compare it with the recursive version.

1. What happen to the recursive case of recursive function when converting it into non-recursive version?
2. What happen to the base case of recursive function when converting it into non-recursive version?

Exercise 2: Power of x (30 min)

Write a *recursive* and a *non-recursive function* to calculate the power of a number where power(n, x) will return n to the **power of x**.

Exercise 3: Fibonacci (30 min)

The **Fibonacci sequence** a(1), a(2), a(3), ..., a(n), ... is defined by

a(1) =1
a(2) = 1
a(n) = a(n-1) + a(n-2), for all n > 2
This generates the sequence

1, 1, 2, 3, 5, 8, 13, 21, ...

Write a *recursive* version of **Fibonacci function** that computes the Fibonacci number corresponding to its positive integer argument, so that, for example, fibonacci(7) == 13.

Exercise 4: Reverse the characters in a string (30 min)

Write a *recursive* function to **reverse the characters in a string**. The prototype of the function looks like this:

```
void reverse(char *p1, char *p2);
```

You can test your program with the following `main()` function:

```
int main()
{
    char str[] = "Hello World";
    cout << str << endl;  // prints Hello World
    reverse(&str[0], &str[strlen(str)-1]); //pass by ref to call ptr
fn
    cout << str << endl;  // prints dlroW olleH

   return 0;
}
```

Extra Exercises

1. Write a *recursive* function to **change the letters in a string to uppercase**.

2. Write a *recursive* function `isPalindrome` to **check whether a given string is a palindrome or not**.

3. Implement the **Tower of Hanoi algorithm** presented in the lecture slides (Carano's textbook, pg 100-103) using a recursive C++ program in such a way that the instructions for moving disks are displayed.

4. Try programming problem No. 4 in Carano's textbook (pg213-25) to solve the problem of **finding a way through a maze**.