# Data Structures and Algorithms

**Practice Exercises** **Recursion**

## Instructions

Solve the following recursion problems. It is highly recommended to perform the tasks using **paper and pencil** before taking them on compiler.

## Tracing Problems

Trace the following recursive functions using **paper and pencil** and also determine their output(s).

Trace it with **print(6);**

```
void print(int n)
{
	if (n > 1)
	{
		cout << n << " ";
		print(n - 1);
		cout << n << " ";
	}
}
```

Trace it with **foo(99);**

```
int foo(int n)
{
	if (n > 100)
		return n - 10;

	return fun( fun(n + 11) );
}
```

Trace it with `int x = 15;` **bar(5, &x);**

```
int bar( int n, int *p )
{
	int t, f;

	if ( n <= 1 )
	{
		*p = 1;
		return 1;
	}

	t = bar ( n - 1, p );
	f = t + *p;
	*p = t;

	return f;
}
```

## Coding Problems

1. Write a recursive function that returns the **sum of first *n* positive integers**.
2. Write a recursive function that returns the **sum of the squares of the first *n* positive integers**.
3. Write a recursive function that returns the **power $a^n$**.
4. Write a recursive function to calculate the **factorial** of a number $a$!.
5. Write a recursive function that returns the **maximum** among the ***n* elements of an integer array**.

**NOTE: - No submission will be accepted after the DUE DATE and TIME.**

B E S T O F L U C K