

Recursion exercises

All functions must be realized recursively, without auxiliary functions, global variables or loops of any kind!

1. Write a computer function to factorial of a number. (Reminder)- A factorial is the product of all numbers up to that number)

```
int factorial(int n);
```

2. Write a function that returns the value of the largest number in an array that contains int variables, the function takes an array and its size.

```
int biggestInArray( int arr[], int size);
```

3. Write a function that takes an index value in the Fibonacci series and returns the value of the number at that position in the series.

```
int fibonacci(int n);
```

4. Write a function that takes an array of int variables and returns 1 If the array contains only positive numbers, the function returns 0 if the array contains at least one negative number. For example, for the array -2,3,5,7,8 the function will return 1.

For the array -4,5,-2,7,8 the function will return 0.

```
int isArrayPositive(int arr[], int size);
```

5. Write a function that takes a string and another character, the function will return 1 if the character is in the string, the function will return 0 if the character is not in the string.

```
int isCharExist(char* str, char c);
```

6. Write a function that takes a string and an additional character, the function returns the number of occurrences of the character in the string.

```
int countSameChars(char* str, char c);
```

7. Write a function that performs a power operation between the value of n (the holding base) and the value m (the exponent of the hold).

```
int powerOfANumber(int n, int m);
```

8. Write a function that takes two int variables, returns 1 If n2 is a divisor of n1 without a remainder, the function returns 0 if n2 is not a divisor of n1.

For example, for n1=45 and n2=5 the function will return 1, for n1=51 and n2=5 the function will return 0.

```
int divider(int n1, int n2);
```

9. Write a function that takes an int variable and returns its binary value in a double variable.
For example, for the number 5555, the value 1010110110011 will be returned.

```
double decToBinary(int num)
```

10. Write a function that takes a string and returns the amount of times the ' (space) character appears in the string.

```
int countBlanks(char* str);
```

11. Write a function that takes an int variable and returns the same digits as the number contains.

```
int countDigits(int num);
```

12. Write a function that takes 2 int variables, the variable num is a number to check for is a prime number, the div variable is a division value starting with the number 2.

The function returns 1 if the number is prime, the function returns 0 if the number is not prime.

For example, for the number 13 the function will return 1, for the number 15 the function will return 0.

```
int isPrime(int num, int div);
```

13. Write a function that takes 2 variables of type int and returns the value of the largest common divisor of both.

For example, for n1=100 and n2=80 the value 10 is returned, for n1=75 and n2=25 the value 25 is returned.

```
int biggestDivider(int n1, int n2);
```

14. Write a function that takes an array of int variables and the size of the array, returns 1 If the sum of any two adjacent elements in an array is a product of 3, the function returns 0 if there is a pair of adjacent elements in an array whose sum is not a product of 3.

For example, for the array 10,41,1,5,22 the function returns 1 because 10+41=51, 41+1=42, 1+5=6, 5+22=27—the total sums of adjacent elements are a product of 3.

For the array 3,6,30,5,7 the function will return 0 because 30+5=35 is not a product of 3.

```
int multiply3(int arr[], int size);
```