

## Задачи:

1. Напишете функция, **factorial** (int n), която намира n! рекурсивно.

Примерен вход и изход: *factorial(5) -> 120*

2. Напишете функция **fib** (int n), която намира n-то число на Фибоначи рекурсивно.

Примерен вход и изход: *fib (9) -> 34, fib (20) -> 6765*

3. Напишете рекурсивна функция **power** (int n, int k), която повдига n на степен k.

Примерен вход и изход: *power(5,3) -> 125, power(2,5) -> 32*

4. Да се напише рекурсивна функция **isElementInArray** (double\* arr, int size, double element), която приема масив от дробни числа, размера му и елемент- дробно число и проверява дали елементът принадлежи на масива.

Примерен вход и изход: *isElementInArray(5, {1.2, 2.6, 0.8, -3.6, 4}, 0.8) -> true*

*isElementInArray(5, {1.2, 2.6, 0.8, -3.6, 4}, 1.8) -> false*

5. Да се напише рекурсивна функция double **minRec** (int n, double\* arr), която по подаден масив от дробни числа и размера му, намира неговия минимален елемент.

Примерен вход и изход: *min(5, {1.2, 2.6, 0.8, -3.6, 4}) -> -3.6*

6. Да се напише рекурсивна функция, **isAscending** (int\* arr, int size) която по подаден масив и размера му, проверява дали той е подреден във възходящ ред.

Примерен вход и изход: *isAscending(5, {1, 2, 0, -3, 4}) -> false*

*isAscending(5, {1, 2, 3, 4, 5}) -> true*

7. Напишете рекурсивна функция **sum** (int\* arr, int size, int start), която по подаден масив и размера му намира сумата на неговите елементи.

Примерен вход и изход: *sum (5, {1, 2, 0, -3, 4}, 0) -> 4*

*sum (5, {1, 2, 3, 4, 5}, 0) -> 15*

8. Напишете рекурсивна функция **sumInInterval** (int a, int b), която намира сумата на числата в интервала [a, b].

Примерен вход и изход: *sumInInterval* (3, 6) -> 18

9. \* Да се напише рекурсивна функция **binarySearch**(int\* arr, int len, int start, int end, int element), която търси двоично елемент в сортиран масив.

Примерен вход и изход: **binarySearch** ({1,2,3,4,5,6,7,8}, 8, 0,7,6) -> 18