

РЕКУРСИЯ (ЧАСТ 2). ФУНКЦИИ ОТ ПО - ВИСОК РЕД.

Задача първа:

Напишете функция `filter`, която приема масив от цели числа, дължината му и булева функция с един аргумент цяло число. В масива, елементите които не изпълняват условието на булевата функция променят стойността си на -1.

Пример:

```
#include <iostream>

bool isEven(int n) {
    return (n % 2) == 0;
}

void filter(array, length, predicate) {
    ...
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    filter(arr, 5, isEven);
    for(int i = 0; i < 5; i++){
        std::cout << arr[i] << " ";
    }
}
```

Изход: -1 2 -1 4 -1

Задача втора:

Напишете функция `map`, която приема масив от цели числа, дължината му и функция, която приема число по референция, и приложете функцията върху всеки елемент на масива.

Пример:

```
#include <iostream>

int plusFive(int& n){
    n += 5;
}
```

```

void map(array, length, func) {
    ...
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    map(arr, 5, plusFive);
    for(int i = 0; i < 5; i++){
        std::cout << arr[i] << " ";
    }
}

```

Изход: 6 7 8 9 10

Задача трета:

Да се дефинира рекурсивна функция, която заменя всяко срещане на цифрата 5 в дадено неотрицателно цяло число с 8.

Задача четвърта:

Да се напише рекурсивна функция, която извършва двоично търсене в предварително сортиран масив от цели числа.

Задача пета:

Имаме лабиринт, с размери $p \times m$, в който проходимите клетки се бележат с 0 а непроходимите с 1. Да се напише програма, която проверява дали съществува път от подадени като аргументи начална и крайна точка.

Задача шеста:

Дадена е квадратна матрица с размери $n \times m$ от цели числа. Да се намери пътят от горния ляв ъгъл до долния десен ъгъл с най - голяма сума от елементите, като имаме право да се движим само надолу и надясно.