

Задачи

Увод в програмирането

17 декември 2018 г.

Правила

- Правилното декомпозиране на функции е много важно. Вашите функции трябва да са кратки (по-малко от 20 реда!) И всяка трябва да изпълнява една единствена, ясна задача.
- Имената на променливи трябва да са описателни – да обясняват за какво служи дадената променлива. Примерно за име на човек подходящо име е `personName`, а неподходящо име е `a1`.
“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live” - Martin Golding

Допълнителни Материали

- Графични визуализации на различни алгоритми:
 - <https://visualgo.net/en>
- Указатели:
 - <https://www.learncpp.com/cpp-tutorial/67-introduction-to-pointers/>
- Рекурсия:
 - <https://www.programiz.com/cpp-programming/recursion>
- Как да дебъгвам ?
 - [Упътване за използване на Visual Studio Debugger](#)
 - [Ако сте любопитни, още по-подробни упътване](#)
 - [Ако сте прекалено много любопитни - как да дебъгвате с GDB](#)

Задачи

1. !!!!!!! Ако не сте решили задачите от предишни пъти решете тях !!!!!!!
 - <https://github.com/tozka/fmi-kn-8/>

2. “Debugging”

Имате програмки със задачи с бъгове - `bug1.cpp` до `bug7.cpp` :

<https://github.com/tozka/fmi-kn-8/tree/master/up-ex10>

Намерете бъговете на всяка от тях и ги оправете да работят правилно. Използвайте дебъгер където можете.

3. “SentenceMaker”

Създайте функция, която изкарва всички възможни изречения (комбинации на думи от всеки ред) при дадена матрица от думи, например

"what", "why"

"red", "wooden", "pretty"

"house", "man"

Изход: what red house, what wooden house, what pretty house, what red man, what wooden man
(Отделна функция трябва да принтира изреченията)
(Всеки ред може да има произволен брой думи, но по-малък от 30)
(Опростен вариант: изхода е: what red house, why wooden man, pretty)
(Опростен вариант: матрицата е NxM т.е всеки ред има еднакъв брой думи)
(Сложен вариант: Изходния ред на изречения трябва да е като в примера - пробвайте със и без рекурсия)

4. "BinarySearch Next"

Упражнявали сте двоично търсене използвайки [рекурсия](#). Техническа всяка задача с рекурсия може да бъде решена без.

Ако не помните Двоично търсене прочете <https://www.codingame.com/learn/binary-search>

А) Решете я без рекурсия (т.е с цикли)

Б) Помогнете на Батман да спаси Готъм !

<https://www.codingame.com/ide/puzzle/shadows-of-the-knight-episode-1>

за тези които искат повече предизвикателство (за домашна работа) могат да пробват да решат :

<https://www.codingame.com/ide/puzzle/shadows-of-the-knight-episode-2>

5. "Queens"

Дадена е шахматна дъска 8 x 8. На произволни места са поставени бялата и черната царици. Напишете функция, която проверява дали ако бялата е на ход може да вземе черната. Решете задачата с рекурсия.

6. "Treasure"

Лабиринт се представя като квадратна матрица от символи с размери 10x10, където празна клетка се представя с интервал, непроходима клетка се представя със символа '#', а клетката означена със символа '\$' е безценно съкровище. Юнакът влиза в лабиринта от най-горната лява клетка и на всяка стъпка може да избира да продължи на изток, запад, север или юг, но след като мине през някоя клетка, след него таванът се срутва и клетката става непроходима. Да се напише програма, която по дадена карта, определя дали е възможно юнакът да намери съкровището.