

# Задачи

## Увод в програмирането

07 Януари 2019 г.

### Препоръки

- 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 `a4`.  
“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.hackerearth.com/practice/basic-programming/recursion/recursion-and-backtracking/tutorial/>
- Как да дебъгвам ?
  - [Упътване за използване на Visual Studio Debugger](#)

1. “Двоичен часовник”  
(backtracking and/or bit manipulation)

Бинарен часовник има 4 светодиода(LEDs) на върха, които представляват часовете (0-11), а 6-те светодиода на дъното представят минутите (0-59). Всеки светодиод представлява нула или единица, с най-малък бит(least-significant bit) отдясно.



Например, горният двоичен часовник гласи "3:25".

Дадено ви е неотрицателно цяло число  $n$ , което представлява броя на светодиодите, които са включени в момента, се връщат всички възможни времена, които часовникът може да представлява.

For 2: expected output (реда няма значение):

```
["0:03","0:05","0:06","0:09","0:10","0:12","0:17","0:18","0:20","0:24","0:33","0:34","0:36","0:40","0:48","1:01","1:02","1:04","1:08","1:16","1:32","2:01","2:02","2:04","2:08","2:16","2:32","3:00","4:01","4:02","4:04","4:08","4:16","4:32","5:00","6:00","8:01","8:02","8:04","8:08","8:16","8:32","9:00","10:00"]
```

## 2. “Структури”

a. Какво не е наред със следните декларации?

- A. `struct Point ( double x, y )`
- B. `struct Point { double x, double y };`
- C. `struct Point { double x; double y }`
- D. `struct Point { double x; double y; };`
- E. `struct Point { double x; double y; }`

b. Какво не е наред със следните декларации?

- A. `typedef struct { double x; double y } Point;`
- B. `typedef { double x; double y; } Point;`
- C. `typedef struct { double x; double y; };`
- D. `typedef struct { double x; double y; } Point;`

c. Имплементирайте `point_print`, `point_equal`, `point_distance` функции

- d. Дефинирайте тип `Rect` за правоъгълници, които са успоредни на осите в декартова координатна система. Представете правоъгълник от долните лява и горни дясна точки, като използвате горния тип `Point`.
- e. Напишете функция, която връща `true`, ако точка попада в правоъгълник, `false` в противен случай. Използвайте горепосочените типове.
- f. Напишете функция, която връща `true`, ако първият правоъгълник е напълно съдържащи се във втория правоъгълник и `false` в противен случай

## 3. (За Домашна работа) “The last Crusade”

- <https://www.codingame.com/training/medium/the-last-crusade-episode-1>
- <https://www.codingame.com/training/hard/the-last-crusade-episode-2>