

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА
САМОПОДГОТОВКА
ПО
Структури от данни и програмиране
*Привеждане на рекурсивни решения към
решения със стек*

email: kalin@fmi.uni-sofia.bg

17 октомври 2017 г.

Упътване: Решете задачите с рекурсия и след това преобразувайте решението в решение със стек.

1. (*) Да се дефинира функция за намиране на стойността на полинома на Ермит $H_n(x)$ (x е реална променлива, а n неотрицателна цяла променлива), дефиниран по следния начин:

$$H_0(x) = 1$$

$$H_1(x) = 2x$$

$$H_n(x) = 2xH_{n-1}(x) + 2(n-1)H_{n-2}(x), n > 1,$$

за дадени n и x с използване на стек.

2. Нека е дадена абстрактна шахматна дъска с размери $n \times n$, $4 \leq n \leq 8$ и число k , $0 \leq k \leq n$. Казваме, че разположени на дъската k коня образуват “валидна конфигурация”, ако никоя фигура не е поставена на поле, което се “бие” от друга фигура според съответните шахматни правила.

Да се дефинира клас `KnightConfig`, представящ “конфигуратор” на шахматни коне. Конструкторът на класа инициализира конфигуратора с числата n и k . Класът позволява “обхождането” една по една на всички валидни конфигурации за дадените параметри,

по подобие на `forward` итератор на структура от данни. Класът да притежава следните методи:

- `void KnightConfig::printCurrentConfig()`: Отпечатва текущо намерената конфигурация. Пример за отпечатана конфигурация с $n = 5, k = 2$:

```
- - - - -  
- _ H _ -  
- - - - -  
- _ _ _ H  
- - - - -
```

- `void KnightConfig::findNextConfig()`: Намира следваща конфигурация.
- `bool KnightConfig::noMoreConfigs()`: Показва дали всички възможни конфигурации са вече изчерпани.