

**Домашна работа №2 по Функционално програмиране**  
**Специалност Информационни системи, 1-ви курс, 1-ви поток**  
**2016/2017 учебна година**

Домашното се състои от 3 задачи. Крайният срок за предаване е **неделя, 02.04.2017 г.**

Решенията **трябва** да се компилират, защото ще се тестват автоматично. Важно е работата да е добре форматирана и да има коментари, където е нужно.

Предайте решенията и на 3-те задачи в един файл, който трябва да е именован **hw1.<FN>.hs**, където <FN> е вашият факултетен номер.

*Приятна работа и успех!*

### **Задача 1. Сума на числа**

Дефинирайте функцията **sumNumbers str**, която приема символен низ **str** и връща сумата на всички числа, които се съдържат в него.

**Примери:**

`sumNumbers "a123b2c56" → 181`

`sumNumbers "a1b2c3" → 6`

### **Задача 2. Питагорови тройки**

Целите числа  $a$ ,  $b$  и  $c$  наричаме Питагорови тройки, ако  $a < b < c$ ,  $a^2 + b^2 = c^2$ , т. е. ако са съответно катетите и хипотенузата на правоъгълен триъгълник. Пример за такава тройка са числата 3, 4 и 5, които образуват т. нар. египетски триъгълник.

Дефинирайте функцията **pythagoreanTriples p**, която връща всички питагорови тройки, чийто съответващ триъгълник е с периметър по-малък от целочисления параметър **p**.

**Примери:**

`pythagoreanTriples 13 → [(3,4,5)]`

`pythagoreanTriples 50 → [(3,4,5),(6,8,10),(5,12,13),(9,12,15),(8,15,17),(12,16,20)]`

### **Задача 3. Кодиране на съобщения посредством субституционен шифър**

Субституционният шифър (Цезаровият шифър) е един от най-старите методи за криптиране на съобщения. Този алгоритъм се използва до началото на компютърната ера и съвременната криптография като дори известната машина Енигма работи посредством наслагването на няколко субституционни шифъра един след друг.

Идеята на алгоритъма е следната: разполагаме с речник от двойки символи `srcChar` -> `dstChar` (който в настоящата задача ще моделираме, посредством списък от двойки). Кодирането на символа `srcChar` се случва като просто го заместим със съответстващия му символ `dstChar`.

Дефинирайте следните функции:

1. Подготовка на входното съобщение: дефинирайте функцията **`normalizeMessage str`**, която приема символния низ **`str`** и го нормализира по следните правила:
  - a. Всички малки букви стават главни.
  - b. Ако съобщението съдържа цифрови символи, функцията връща грешка.
  - c. Всички останали символи се игнорират.
2. Кодиране на символ: дефинирайте функцията **`encodeSymbol dct c`**, която приема речник от двойки символи **`dct`** и символ за кодиране **`c`** и връща съответстващият на **`c`** символ от речника.
3. Дефинирайте функцията **`encodeMessage dct message`**, която приема речник от двойки символи **`dct`** и символен низ **`message`** и първо го нормализира, и връща кодираното съобщение.
4. Дефинирайте функцията **`reverseEncoding dct`**, която приема речник от двойки символи **`dct`** и връща речник от двойки символи, в който символите на всяка двойка от **`dct`** са обърнати.
5. Дефинирайте функцията **`decodeMessage dct encoded`**, която приема символния низ **`encoded`**, който представлява съобщение кодирано посредством речника **`dct`** и го декодира. За целта може да използвате факта, че разкодирането на съобщение кодирано с **`dct`** е еквивалентно на кодирането му посредством обърнатия на **`dct`** речник.

**Примери:**

```
dct = zip ['A' .. 'Z'] ([ 'F' .. 'Z' ] ++ [ 'A' .. 'E' ])
```

```
dct → [('A','F'), ('B','G'), ('C','H'), ('D','I'), ('E','J'), ('F','K'), ('G','L'), ('H','M'), ('I','N'), ('J','O'), ('K','P'), ('L','Q'), ('M','R'), ('N','S'), ('O','T'), ('P','U'), ('Q','V'), ('R','W'), ('S','X'), ('T','Y'), ('U','Z'), ('V','A'), ('W','B'), ('X','C'), ('Y','D'), ('Z','E')]
```

```
message = "The quick brown fox jumps over the lazy dog"
```

```
normalizeMessage message → "THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG"
```

```
encodeMessage dct message → "YMJVZNHPGWTBSKTCOZRUXTAJWYMJQFEDITL"
```

```
reverseEncoding dct → [('F','A'), ('G','B'), ('H','C'), ('I','D'), ('J','E'), ('K','F'), ('L','G'), ('M','H'), ('N','I'), ('O','J'), ('P','K'), ('Q','L'), ('R','M'), ('S','N'), ('T','O'), ('U','P'), ('V','Q'), ('W','R'), ('X','S'), ('Y','T'), ('Z','U'), ('A','V'), ('B','W'), ('C','X'), ('D','Y'), ('E','Z')]
```

```
decodeMessage dct (encodeMessage dct message) →
```

```
"THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG"
```