

## Домашна работа № 1 по Функционално програмиране

специалност „Информационни системи“, I курс, 2022/2023 учебна година

---

Решенията трябва да са готови за автоматично тестване. Важно е програмният код да бъде добре форматиран и да съдържа коментари на ключовите места. Предайте решенията на всички задачи в *един* файл с наименование *hw1\_<FN>.hs*, където *<FN>* е Вашият факултетен номер.

Домашните работи се предават като изпълнение на съответното задание в курса по ФП в Moodle (<https://learn.fmi.uni-sofia.bg/course/view.php?id=9029>) най-късно до **23:55 ч. на 30.03.2023 г.** (четвъртък).

*Приятна работа и успех!*

---

### Задача 1

Естественото число  $n$  е „специално спрямо степен  $k$ “, ако е по-голямо от 1 и сборът на цифрите на  $n$ , повдигнати на степен  $k$ , е равен на  $n$ .

Например първото специално спрямо степен 3 число е 153 ( $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$ ). Следващото число, което има това свойство при степен 3, е 370.

Да се дефинира функция `eqSumPowDig :: Int -> Int -> Int`, която приема естествено число, задаващо горна граница – `hMax`, и естествено число, задаващо степента – `power`, и намира сумата на числата, които са специални спрямо степента `power` и са по-малки от `hMax`. При подадени невалидни аргументи да се извеждат грешки с подходящ текст. **Да се реализира линеен итеративен процес!**

*Примери:*

```
eqSumPowDig 100 2 → 0
eqSumPowDig 1000 2 → 0
eqSumPowDig 2000 2 → 0
eqSumPowDig 200 3 → 153
eqSumPowDig 370 3 → 523
eqSumPowDig 370 3 → 523
eqSumPowDig 400 3 → 894
eqSumPowDig 500 3 → 1301
eqSumPowDig 1000 3 → 1301
eqSumPowDig 1500 3 → 1301
```

## Задача 2

Число е „седемкратно“, ако е или степен на 7, или сбор от уникални степени на 7.

Да се дефинира функция `getNthSevenlikeNum :: Int -> Int`, която намира  $n$ -тото седемкратно число. При подадено неположително число да се извежда грешка с подходящ текст.

*Примери:*

```
getNthSevenlikeNum 1 → 1  
getNthSevenlikeNum 2 → 7  
getNthSevenlikeNum 3 → 8  
getNthSevenlikeNum 4 → 49
```