

## Домашна работа № 3 по Функционално програмиране

специалност „Компютърни науки“, II курс, I поток, 2021/2022 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 *hw3\_<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=7484>) най-късно до **23:55 ч. на 05.01.2022 г.** (сряда).

*Приятна работа и успех!*

---

**Задача 1.** Докато подготвяли подаръците за Коледа, елфите на дядо Коледа осъзнали, че имат проблем: подаръците трябва да бъдат подредени по специален начин в шейната, за да може тя да лети във въздуха. Всеки подарък има уникален номер – цяло число. Целта е абсолютните стойности на разликите между номерата на всеки два съседни подаръка да заемат всички стойности между 1 и  $n-1$ , където  $n$  е броят на подаръците за нареждане. Въпреки че елфите могат да измислят комбинации от подредби, подаръците са твърде много, за да преценят дали ще успеят да ги наредят по правилен начин от първия път. Помогнете им, като дефинирате предикат `willItFly :: [Int] -> Bool`, който получава наредба, предложена от елфите, и проверява дали тя е валидна.

*Примери:*

```
willItFly [1, 4, 2, 3]      → True      -- |1-4|=3, |4-2|=2, |2-3|=1
```

```
willItFly [1, 4, 2, -1, 6] → False
```

**Задача 2.** Намирането на правилна наредба на подаръците обаче не било единственият проблем. Оказало се, че часовникът на шейната, който показва на дядо Коледа колко време му остава, докато стигне до следващото послушно дете, на което трябва да остави подарък, се е развалил. Вместо да показва времето като комбинация от години, дни, часове, минути и секунди, часовникът показва единствено броя оставащи секунди. Екипът на дядо Коледа отново се нуждае от вашата помощ. Знаейки, че една година има 365 дни и един ден има 24 часа, дефинирайте функция `formatDuration :: Int -> String`, която приема време, представено като брой секунди, и връща времето в описаната комбинация. Ако  $n$  е нула, да се връща "now".

### *Примери:*

`formatDuration 0 → "now"`

`formatDuration 1 → "1 second"`

`formatDuration 62 → "1 minute and 2 seconds"`

`formatDuration 120 → "2 minutes"`

`formatDuration 3600 → "1 hour"`

`formatDuration 3662 → "1 hour, 1 minute and 2 seconds"`

### *Подсказки:*

Полученият израз да е съставен от различни компоненти във вида:

"<естествено\_число> <единица\_време>".

Единицата за време се използва в множествено число, ако числото е по-голямо от 1. Компонентите са разделени със запетая и интервал с изключение на последния компонент, който е разделен с " and ", точно както би било написано на английски език.

По-значимите единици време да се поставят преди по-малко значимите, т.е. вместо "1 second and 1 year" да се извежда "1 year and 1 second".

Да не се появява компонент, ако стойността му е нула, т.е. вместо "1 minute and 0 seconds" да се извежда само "1 minute".

Продължителността, определена от по-малка по значимост единица време, не трябва да бъде по-голяма от която и да е по-значима единица време. Това означава, че функцията не трябва да връща "61 seconds", а "1 minute and 1 second".