WORKSHEET

A13.5 Map, filter and fold

Functional programming · A level · AQA 7517 4.12.2.1, Eduqas A500QS 1.4 · about 25 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

The three higher-order functions, foldl and foldr, and a pipeline over a log of sensor readings.

- The log
- map: change every item
- filter: keep some items
- fold: combine everything into one value
- Putting them together
- On the robot

Questions

6 marks in all

1. What does this program print? [1 mark]

```
cms = [12, 45, 30, 61]
print(list(map(lambda cm: cm * 10, filter(lambda cm: cm > 25, cms))))
```

2. What does filter do? [1 mark]

- ☐ **A** Produces a new list containing exactly the items of the original that match a given condition
- ☐ **B** Applies a function to every item and returns the list of results
- ☐ **C** Reduces a list to a single value with a combining function
- ☐ **D** Removes the first item of a list

3. What is the value of foldl (-) 10 [1,2,3]? [1 mark]

4. What is the value of foldr (-) 0 [5,3,1]? [1 mark]

5. What does this program print? [1 mark]

```
from functools import reduce

log = [(0, 40), (90, 15), (180, 62), (270, 33)]
far = reduce(lambda a, b: a if a[1] >= b[1] else b, log)
print(far[0])
```

6. What is the result of the Haskell expression map (*2) [1,2,3]? Write it as a Haskell list. [1 mark]

The task: analyse the sweep

Analyse `sweep.csv` (above) with map, filter and fold. There must be no `for` or `while` anywhere in the program (so no comprehensions either), and no `min`, `max` or `sum`: use `map`, `filter` and `reduce` instead. - `parse(line)` takes one line of the file, such as `"2.4,60,24.5"`, and returns a tuple `(time, heading, cm)` with `time` a float, `heading` an int and `cm` a float. - Read the file, skip the header line, and map `parse` over the rest. - Keep only the **valid** readings, those under 400 cm, and print `valid: <how many>`. - Of the valid readings, print the headings of those over 40 cm, in the order they are in the file, as `clear: 0 30 ...` separated by spaces. - Fold the valid readings to find the nearest, and print `nearest: <cm> cm at <heading>`. - Fold the valid readings to their total, and print `mean: <mean cm>` rounded to one decimal place. The robot does not move.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
from functools import reduce

f = open("sweep.csv", "r")
lines = f.read().splitlines()
f.close()

for line in lines[1:]:
    print(line)
```

Plan your program here, then type it in and press Run.

Do it on the robot

www.bugbotlab.com/learn/a13-5-map-filter-and-fold/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Write `foldr (-) 0 [5,3,1]` and `foldl (-) 0 [5,3,1]` with brackets, work out both, then check the left one with `reduce`.
2. Use one fold, and no `len`, to count how many valid readings are under 20 cm.
3. Write your own `my_map(f, xs)` and `my_filter(p, xs)` using recursion and no loops.