# Worksheet 5 Lists in functional programming Answers

Note: specific language syntax (such as Haskell, used in these tasks) will not be expected in an exam.

Here are two lists to use in the Tasks below

**placeList** = ["Ankara", "Baku", "Cairo", "Derby", **"Etna", "Faro"]**

**nameList = ["Anja", "Bjorn", "Cath", "Dirk", "Edna", "Figgis"]**

***See lists.hs***

**Task 1**

1. (a) What is the **head** of **placeList**? “Ankara”

(b) What is the **tail** of **nameList**? ["Bjorn","Cath","Dirk","Edna","Figgis"]

*Note that the head of a list is a single element, the tail is a list,   
which may be an empty list.*

\*Main> let namelist = ["Anja", "Bjorn", "Cath", "Dirk", "Edna", "Figgis"]

\*Main> let placelist = ["Ankara", "Baku", "Cairo", "Derby", "Etna", "Faro"]

\*Main> head placeList

"Ankara"

\*Main> tail nameList

["Bjorn","Cath","Dirk","Edna","Figgis"]

2. (a) What is the **tail** of a list with just one element? An empty list, []

\*Main> tail [1]

[]

(b) In Haskell, a string is defined as a list of characters. What is the **tail** of the **head** of **placeList**?

The head of placeList is a string of chars, “Ankara”. A string of chars is defined as a list of chars. so the answer is “nkara”.

\*Main> tail (head placeList)

"nkara"

(c) What is the **head** of the **tail** of **nameList?**

\*Main> head (tail nameList)

"Bjorn"

3. Use the idea that a string of characters is a list to map the first letters of each element in placeList to a new variable called capitals.

*Remember that in interactive mode, the function* **capitals** *must be defined using “***let***”. Alternatively, the lists and functions can be defined in a text editor, saved and loaded.*

\*Main> let capitals = map head placeList

\*Main> capitals

"ABCDEF"

### Task 2

**placeList** = ["Ankara", "Baku", "Cairo", "Derby", **"Etna", "Faro"]**

**nameList = ["Anja", "Bjorn", "Cath", "Dirk", "Edna", "Figgis"]**

Write Haskell code to achieve each part:

4. (a) Prepend the name “Aaron” to **nameList**

"Aaron**"** : nameList

(or ["Aaron"] ++ nameList)

(b) Append the place “Grimsby” to **placeList**

placeList ++ ["Grimsby"]

(c) Define a new list called **newPlaces** where “Ankara” is replaced by “Athens”

newPlaces = "Athens" : tail placeList (need “let” in interactive mode)

(d) Define a new integer countPlaces which has the value of the number of elements in newPlaces**.**

countPlaces = length newPlaces (need “let” in interactive mode)

5. (a) Evaluate null ( tail [**"**fishpaste**"**])

True

(b) Evaluate **null (tail placeList)**

False

6. Evaluate the following:

(a) tail(tail(tail(placeList)))

["Derby","Etna","Faro"]

(b) head(tail(tail(nameList)))

“Cath”

(c) numList = [3,5,7,9]

null (tail(tail(tail(tail(numList)))))

True

7. Give the result and explain each stage of evaluating the following functional expression:

foldl (+) 10 (map length ["Dave", "Sue", "Alice"])

map length results in **[4, 3, 5]**

Passing it to foldl results in **((10 + 4) + 3) + 5**

Result is **22**

Further reading: <https://www.futurelearn.com/courses/functional-programming-haskell>