CS115 – Lab 12: Imperative Programming, Modularization, and The Future

Spring 2020

Question 1: Bigger half

Exercise

Write a function `(larger-half L)` that consumes a `(listof Num)`. It returns either the list containing all the positive numbers, or the list containing all the negative numbers, whichever is longer.

If the two lists are of equal length, return the positive list.

```
(larger-half (list 2 4 -4 -6 -1)) => (list -4 -6 -1)
(larger-half (list 2 4 -4 -6 -1 7)) => (list 2 4 7)
```

! Remember, use `local` so you don't compute any value more than once!

Question 2: We Try Harder

Give `M`, a `listof Num`, the following expression will return its largest value:

```
(foldr max (first M) (rest M))
```

Exercise

Write a function `(second-largest L)` that consumes a `listof Num` of length at least 2, and returns the second largest value in it.

For example, `(second-largest (list 21 5 8 4 47 2 6 7)) => 21` `(second-largest (list 9 5)) => 5`

! Do not sort the list, using `sort`, or otherwise.

! You may assume there are no duplicates in `L`.

Hint

Find the largest value, then get a list of values that are not the largest value.

Question 3: Now we're even

Exercise

Write a function `(add-sum-evens-to-odds L)` that consumes a `listof Int`, and returns the list where the sum of all even values has been added to all odd values.

For example, `(add-sum-evens-to-odds (list 1 2 3)) => (list 3 2 5)`

```
(add-sum-evens-to-odds (list 4 1 2 3)) => (list 4 7 2 9)
(add-sum-evens-to-odds (list 1 3 5)) => (list 1 3 5)
```

Question 4: Bibliography

Exercise

Write a function `(reorder-name name)` that consumes a `Str`. This will contain two words, a first name, and a last name, separated by a space. The function returns a `Str` containing the last name, a comma, then the first name. For example, `(reorder-name "Raymond Shaw") => "Shaw, Raymond"`

```
(reorder-name "Otho Sackville-Baggins") => "Sackville-Baggins, Otho"
(reorder-name "Jean-Luc Picard") => "Picard, Jean-Luc"
```