GEBZE
TECHNICAL UNIVERSITY

# INF211

## LABORATORY LEAFLET

FOR STUDENTS

## LABORATORY-3
## Functions

| Exercises | Explanations |
|---|---|
| 1 | A function finding and returning minimum of the four numbers |
| 2 | A function finding and returning the only character that does not have a duplicate in a given string |
| 3 | Ascending ordered string |
| Tasks | Explanations |
| 1 | Round to the nearest |
| 2 | Recurse with me |
| 3 | Similarity |

**Exercise 1: Write a function named "`functionE1`" that expects four parameters as numbers and returns the minimum of these numbers.**

- Function parameter names do not matter, but the function should expect 4 parameters.
- inputs : a, b, c, d ∈ R
- output : float

**The output must be as follows:**

```
 >>> functionE1(2.3, -2.4, -1, 4.3)

-2.4
```

**Exercise 2: Write a function named "`functionE2`" that will find and return the only character that does not have a duplicate in a given string. The returned character needs to be a string.**

- There can be multiple copies of a given character.
- The given string will have exactly **one character** without a pair.
- String will not be ordered, can be any order.
- Function parameter name does not matter, but function should expect 1 parameter.
- `inputs : a = { x: x ∈printable characters except whitespaces and len(x) ∈[1:200]}`
- output : string

**The output must be as follows:**

```
>>> functionE2('12a31a333')

'2'

Explanation: There are four 3's, two 1's and two a's. Only 2
does not have a pair.

>>> functionE2('7b3b17C3C1b7z3')

'z'

Explanation: There are three 7's, three 3's, three b's, 2 C's
and two 1's. Only z does not have a pair.
```

```
>>> functionE2('t')

't'

>>> functionE2('00631543146')

'5'
```

**Exercise 3: Write a function named “`functionE3`” that will find and return if the characters are in ascending order. (abcde...z).**

- The letters do not need to be consecutive, look only for ascending order. (i.e. 'az' is fine)
- Same letter might repeat multiple times.
- `inputs : a = { x: x ∈lowercase English letters and len(x) ∈ [1:200]}`
- output : boolean

**The output must be as follows:**

```
>>> functionE3('abcd')

True

>>> functionE3('abbbbc')

True

>>> functionE3('bbcdd')

True

>>> functionE3('bacd')

False


>>> functionE3('ra')

False

>>> functionE3('abcdefzg')
```

```
False
```

**Task 1: Write a function named “`functionT1`” that expects two parameters as numbers, rounds the first number closer to the second number and returns the result.**

- You should always round the first number.
- Function parameter names do not matter, but the function should expect 2 parameters.
- In the case of indecisivity, round up.
- inputs : a, b ∈ R
- output : integer

**The output must be as follows:**

```
>>> functionT1(2.6, 1)

2

>>> functionT1(2.6, 2)

2

>>> functionT1(2.6, 3)

3

>>> functionT1(3, -10)

3

>>> functionT1(2.6, 5)

3

>>> functionT1(-0.7, 10)

0

>>> functionT1(-0.7, -10)

-1
```

```
>>> functionT1(0.7, 0.6)

1

Explanation: between two numbers (0, 1), 1 is closer to 0.6.

>>> functionT1(0.7, 0.9)

1

>>> functionT1(0.7, 0.5)

1

Explanation: between two numbers (0, 1), both 0 and 1 have the
same distance to 0.5, this is a case of indecisivity, so we round
up to 1.
```

**Task 2 : Write a function named “`functionT2`” that will find and return the number with the given row `(i-th)` and column `(j-th)` index using the following rules.**

- `i`, and `j` show the row and column indexes respectively.
- `f(i, j)` function is given as: `f(i,j) = f(i-1, j-1) + f(i-1, j)`
- `f(i=1, j=1) = 1`
- `f(i, j=1) = 3` for `i > 1`
- `f(i, j=i) = 2 for i > 1`
- `j <= i`

- Function parameter names will be row, and column. Make sure to have these for any credit.
- inputs : **row, column** ∈ [1:100]
- output : integer

**The output must be as follows:**

```
>>> functionT2(1, 1)

1

>>> functionT2(4, 1)

3

>>> functionT2(9, 1)

3

>>> functionT2(row=5, column=5)

2

>>> functionT2(8, 8)

2

>>> functionT2(12, 7)

1134
```

**Task 3: Write a function named “`functionT3`” that takes two parameters as strings and returns the number of characters that appear in the same index.**

- If they share no characters in the same index, return 0
- One string might be longer than the other, so take necessary precautions.
- Function parameter names do not matter, but function should expect 2 parameters.
- inputs: `s1, s2 = {s1, s2 : printable characters except whitespaces and len(s1), len(s2) ∈ [1, 200] }`
- outputs: integer

**The output must be as follows:**

```
>>> functionT3('tarkan', 'gurkan')

4

>>> functionT3('kesit', 'telas')

1

Explanation: both strings have only character e in common in the
same index.

>>> functionT3('ke@', 'telas')

1

Explanation: both strings have only character e in common in the
same index.
```

```
>>> functionT3('k', 'tekkk')

0

>>> functionT3('telas', 'ke')

1

>>> functionT3('!telas', 'k')

0
```