



Degree in Software Engineering – Computing Basics

Solution to Unit 2.7 Exercises: Structured Data Types (I)

This document includes the solution to the exercises of the document “Unit 2.7 Exercises: Structured Data Types (I)”. It is recommended that you try doing the exercises without looking at the solutions first, and then you check your answers. Please note there might be multiple solutions to the same problem.

Exercise 1

Proposed solution:

```
def num_even(a):  
    count = 0  
    for i in a:  
        if isinstance(i, int) and i > 0 and i % 2 == 0:  
            count = count + 1  
    return count  
  
a = [-2, 6, -6, 7, 1, 0, 4, 12]  
n = num_even(a)  
print("The list contains {} even natural numbers".format(n))  
print("After calling the function, the list is {}".format(a))
```

Exercise 2

Proposed solution:

```
def replace_even(a):  
    count = 0  
    for i in range(len(a)):  
        if isinstance(a[i], int) and a[i] > 0 and a[i] % 2 == 0:  
            count = count + 1  
            a[i] = -a[i]  
    return count  
  
a = [-2, 6, -6, 7, 1, 0, 4, 12]  
n = replace_even(a)  
print("The list contained {} even natural numbers ".format(n))  
print("After calling the function, the list is {}".format(a))
```



Exercise 3

Proposed solution:

```
def replace_even(a):
    result = []
    for i in range(len(a)):
        if isinstance(a[i], int) and a[i] > 0 and a[i] % 2 == 0:
            result.append(a[i])
            a[i] = -a[i]
    return result

a = [-2, 6, -6, 7, 1, 0, 4, 12]
b = replace_even(a)
print("{} elements were replaced, and they were {}".format(len(b), b))
print("After calling the function, the list is {}".format(a))
```

Exercise 4

Proposed solution:

```
def fibonacci(n):
    result = []
    if n > 0:
        result.append(0)
    if n > 1:
        result.append(1)
    for i in range(2, n):
        result.append(result[-1] + result[-2])
    return result

n = int(input("Input number of Fibonacci numbers to show: "))
while n < 0:
    print("Number of elements to show must be greater than or equals to 0")
    n = int(input("Input number of Fibonacci numbers to show: "))

for i in fibonacci(n):
    print(i, end=" ")
print()
```



Exercise 5

Proposed solution:

```
def intersection(a, b):
    result = []
    for i in a:
        if i not in result and i in b:
            result.append(i)
    return result

a = [1, 6, 6, 5, 2, 8]
b = [2, 8, 8, 6]
c = intersection(a, b)
print("The elements present in both lists are {}".format(c))
```

Exercise 6

Proposed solution:

```
def avg_and_stddev(a):
    if len(a) == 0:
        return None, None

    avg = sum(a) / len(a)
    stddev = 0
    for i in a:
        stddev = stddev + (i - avg) ** 2
    stddev = (stddev / len(a)) ** (1 / 2)

    return avg, stddev

n = int(input("Input population size: "))
while n <= 0:
    print("Population size must be greater than 0")
    n = int(input("Input population size: "))

a = []
for i in range(n):
    value = float(input("Input value {}: ".format(i + 1)))
    a.append(value)

avg, stddev = avg_and_stddev(a)
print("The average is {:.2f} and the std dev is {:.2f}".format(avg, stddev))
```