

## Ch10 Solutions Part 1

March 20, 2025

**Exercise 4.1** The Euclidean distance between two points  $(a,b)$  and  $(c,d)$  is defined as:  $\sqrt{(a-c)^2 + (b-d)^2}$ . Write a Python code that reads  $a, b, c, d$  from the user, calculates the Euclidean distance, and prints the result.

```
[ ]: a = float(input("Enter a number:"))
      b = float(input("Enter another number:"))
      c = float(input("Enter a third number:"))
      d = float(input("Enter a fourth number:"))
      distance = ((a - c) ** 2 + (b - d) ** 2) ** 0.5
      print(f"The distance between ({a}, {b}) and ({c}, {d}) is {distance}")
```

The distance between (5.0, 3.0) and (2.0, 10.0) is 7.615773105863909

**Exercise 4.2** To convert Celsius to Fahrenheit, we multiply Celsius by 9/5 and then add 32. Write a Python code that reads a Celsius value and prints the equivalent Fahrenheit.

```
[ ]: c = float(input("Enter a celsius:"))
      f = c * 9 / 5 + 32
      print(f"{c}°C is {f}°F")
```

10.0°C is 50.0°F

The formula for compound interest is the following:

$$A = P \left(1 + \frac{r}{n}\right)^{nt}$$

where:

- $A$  = final amount
- $P$  = initial principal value
- $r$  = annual interest rate
- $n$  = the number of times interest is applied per year
- $t$  = the number of years

Write a Python code that reads  $P, r, n, t$  from the user and prints  $A$  as the result.

```
[ ]: P = float(input("initial principal value:"))
      r = float(input("annual nominal interest rate:"))
      n = float(input("number of times that interest is applied per year:"))
      t = int(input("number of years:"))
```

```
A = P * (1 + r / n) ** (n * t)
print(f"The final amount after {t} years is {A}")
```

The final amount after 10 years is 1343.9163793441223

**Exercise 4.3** A sphere of radius  $R_1$  is made of plastic. There is a sphere-shaped hole of radius  $R_2$  ( $R_2 < R_1$ ) in the plastic sphere.

Write a Python code that reads  $R_1, R_2$  from the user, and prints the volume of plastic used.

```
[ ]: from math import pi
r1 = float(input("Enter the first radius:"))
r2 = float(input("Enter the second radius:"))
volume = 4 / 3 * pi * (r1 ** 3 - r2 ** 3)
print(f"The volume of sphere is {volume}")
```

The volume of sphere is 3665.1914291880917

**Exercise 5.1** Implement the while statements examples for average and standard deviation using a for statement.

```
[2]: L = [10 , -4, 4873 , -18]
total = 0
for x in L:
    total = total + x
avg = total / len (L)
print (avg )
```

1215.25

```
[ ]: L = [10 , 20, 30, 40]
N = len (L)

total = 0
for x in L:
    total = total + x
avg = total / N

total = 0
for x in L:
    total = total + (x - avg) ** 2
12
std = ( total / N) ** 0.5
print ("Avg & Std of the list are :", avg , std )
```

Avg & Std of the list are : 25.0 11.180339887498949

**Exercise 5.2** Implement the following for statement examples using a while statement.

- Words with vowels and consonants.

b. Run-length encoding.

```
[ ]: # a. Words with vowels and consonants.
mixed = ["lorem","ipsum","dolor","sit","amet",
↪,"consectetur","adipiscing","elit",\
↪
↪"sed","do","eiusmod","tempor","incididunt","ut","labore","et","dolore","magna","aliqua"]
vowels = []
consonants = []
i = 0
N = len ( mixed )
while i<N:
    word = mixed [i]
    i += 1
    if word [0] in ['a','e','i','o','u']:
        vowels += [ word ]
    else:
        consonants += [ word ]
print ("Starting with consonant :", consonants )
print ("Starting with vowel :", vowels )
```

```
Starting with consonant : [' lorem ', ' ipsum ', ' dolor ', ' sit ', '
consectetur ', ' adipiscing ', ' elit ', ' sed ', 'do', ' eiusmod ', ' tempor ',
' incididunt ', ' labore ', ' dolore ', ' magna ', ' aliqua ']
Starting with vowel : ['amet ', ' ut', ' et']
```

```
[ ]: # b. Run-length encoding.
text = "aaaaaaxxxmyyyaaaassssssstttuivvvv"
code_list = []
last_character = text [0]
count = 1
# Go over each character except for the first
N = len ( text [1:])
i = 0
while i<N:
    curr_character = text [1:] [ i]
    i += 1
    # If curr_character is equal to last_character , we found a duplicate
    if last_character == curr_character:
        count += 1
    else:
        # We have finished a sequence of same characters : Save the count and
        # reinitialize last_character and count accordingly
        code_list += [ last_character if count ==1 else [ last_character ,\
↪count ]]
        count = 1
        last_character = curr_character
# handle the last_character here :
```

```
code_list += [ last_character if count ==1 else [ last_character , count ]]
print ( code_list )
```

```
[' ', ['a', 6], ['x', 4], 'm', ['y', 3], ['a', 4], ['s', 9], ['t', 3], 'u', 'i',
['v', 4], ' ']
```

**Exercise 5.3** Write a Python code that removes duplicate items in a list. For example, [12, 3, 4, 12] should be changed to [12, 3, 4]. The order of the items should not change.

```
[ ]: L = list ("aaaaaadadreeseaess")
N = len (L)
i = 0
while i<N:
    j = i+1
    while j<N:
        if L[i]==L[j]:
            del L[j]
            N -= 1
        else :
            j += 1
    i += 1

print (L)
```

```
[' ', 'a', 'd', 'r', 'e', 's']
```

**Exercise 5.4** Write a program that takes a list of strings and prints the frequency of each word in the list. Hint: Make use of a dictionary. Example: For the following list,

```
["apple","banana","apple","cherry","orange","banana",\
"cherry","elderberry","orange","fig","fig","fig"]
```

Your code should print the following:

Frequency of words:

```
apple: 2
banana: 2
cherry: 2
orange: 2
elderberry: 1
fig: 3
```

```
[ ]: words = ["apple","banana","apple","cherry","orange","banana",\
              "cherry","elderberry","orange","fig","fig","fig"]

frequency = {}
for word in words :
    if word in frequency :
        frequency [ word ] += 1
```

```

    else :
        frequency [ word ] = 1

print ("Frequency of words :")
for word , count in frequency.items (): # the way to access all
    print (f"{ word }: { count }") # the dictionary element

```

```

Frequency of words :
apple: 2
banana: 2
cherry: 2
orange: 2
elderberry: 1
fig: 3

```

**Exercise 5.5** You are given a dictionary of food and price information, stored under a variable Menu. Here is an example:

```

python    Menu = {"soup":22.5,"fries":15.0,"coke":10.0, "beer":30.0,
"bread":5.0,"hamburger":65.0,"scrambled egg":27.5,          "coffee":18.0,"pastry":23.5,"large
pizza":70.0,          "medium pizza":40.0,"small pizza":25.0,"salad":7.75}

```

Write a program that takes a list of items (from the menu), i.e., the bill as:

```

```python
["hamburger","fries","beer","medium pizza","beer"]
```

```

and then, prints the sum as follows:

```

```python
180.0
```

```

Now, modify your program so that the same bill above can be entered as:

```

```python
["hamburger","fries", (2,"beer"),"medium pizza"]
```

```

```

[ ]: Menu = {"soup": 22.5 , "fries": 15.0 , "cok": 10.0 , "beer": 30.0 , \
            "bread": 5.0 , "hamburger": 65.0 , "scrambled egg": 27.5 , \
            "coffee": 18.0 , "pastry": 23.5 , "large pizza": 70.0 , \
            "medium pizza": 40.0 , "small pizza": 25.0 , "salad": 7.75}

bill = ["hamburger","fries","beer","medium pizza","beer"]

total_price = sum ( Menu [ item ] for item in bill )

```

```
print ( total_price )
```

180.0

```
[ ]: Menu = {"soup": 22.5 , "fries": 15.0 , "cok": 10.0 , "beer": 30.0 , \
            "bread": 5.0 , "hamburger": 65.0 , "scrambled egg": 27.5 , \
            "coffee": 18.0 , "pastry": 23.5 , "large pizza": 70.0 , \
            "medium pizza": 40.0 , "small pizza": 25.0 , "salad": 7.75}

bill = ["hamburger", "fries", (2, "beer"), "medium pizza", "beer"]

total_price = sum ( [ item [0]* Menu [item [1]] if type ( item ) == tuple else
    ↪ Menu [ item ] for item in bill ] )
print ( total_price )
```

210.0

**Exercise 5.6** Write a rectangle that asks the user width and height and that draws a rectangle of 'x':

```
xxxxxxxxxxxxx
x             x
x             x
x             x
xxxxxxxxxxxxx
```

Check that only positive value ( $> 0$ ) can be entered: refuse incorrect inputs and give the user the opportunity to reenter.

Tip: you can print a character without a newline with

```
print("x",end="")
```

```
[20]: width = -1
while width < 0:
    width = int(input("Enter the width as a positive integer:"))
    if width < 0:
        print ("Please enter a positive integer")

height = -1
while height < 0:
    height = int(input("Enter the heighth as a positive integer:"))
    if height < 0:
        print ("Please enter a positive integer")

print(f"Printing a {width}x{height} rectangle:")

for i in range ( height ):
    for j in range(width):
```

```
    if j == 0 or j == width - 1 or i == 0 or i == height - 1:
        print ("*", end="")
    else:
        print (" ", end="")
print ()
```

Printing a 20x5 rectangle:

```
*****
*                               *
*                               *
*                               *
*****
```