# list\_comprehensions.py

"""

CP1404/CP5632 Practical

List comprehensions

"""

names = ["Bob", "Angel", "Jimi", "Alan", "Ada"]

full\_names = ["Bob Martin", "Angel Harlem", "Jimi Hendrix", "Alan Turing",

"Ada Lovelace"]

# for loop that creates a new list containing the first letter of each name

first\_initials = []

for name in names:

first\_initials.append(name[0])

print(first\_initials)

# list comprehension that does the same thing as the loop above

first\_initials = [name[0] for name in names]

print(first\_initials)

# list comprehension that creates a list containing the initials

# splits each name and adds the first letters of each part to a string

full\_initials = [name.split()[0][0] + name.split()[1][0] for name in

full\_names]

print(full\_initials)

# one more example, using filtering to select only the names that start with A

a\_names = [name for name in names if name.startswith('A')]

print(a\_names)

# in lowercase format

lowercase\_full\_names = [name.lower() for name in full\_names]

print(lowercase\_full\_names)

almost\_numbers = ['0', '10', '21', '3', '-7', '88', '9']

# from the above list of strings

numbers = [int(i) for i in almost\_numbers]

print(numbers)

# greater than 9 from the numbers (not strings) you just created

numbers\_greate\_than\_9 = []

for x in range(9,19):

numbers\_greate\_than\_9.append(int(x))

print(numbers\_greate\_than\_9)

# list\_exercises.py

def main():

numbers = get\_number()

print\_numbers(numbers)

authorised\_users()

def get\_number():

numbers = []

for i in range(5):

number = int(input("Number: "))

numbers.append(number)

return numbers

def print\_numbers(numbers):

print("The first number is {}".format(numbers[0]))

print("The last number is {}".format(numbers[-1]))

numbers.sort()

print("The smallest number is {}".format(numbers[0]))

print("The largest number is {}".format(numbers[-1]))

print("The average of the number is {}".format(sum(numbers)/len(numbers)))

def authorised\_users():

print()

usernames = ['jimbo', 'giltson98', 'derekf', 'WhatSup', 'NicolEye', 'swei45', 'BaseInterpreterInterface', 'BaseStdIn', 'Command', 'ExecState', 'InteractiveConsole', 'InterpreterInterface', 'StartServer', 'bob']

name = input("Username: ")

if name in usernames:

print("Access granted")

else:

print("Acess denied")

main()

# list\_warmup.py

numbers = [3, 1, 4, 1, 5, 9, 2]

'''numbers[0] = 3

numbers[-1] = 2

numbers[3] = 1

numbers[:-1] = 3, 1, 4, 1, 5, 9

numbers[3:4] = 1

5 in numbers = Ture

7 in numbers = False

"3" in numbers = False

numbers + [6, 5, 3] = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3]'''

'''1. Change the first element of numbers to "ten" (the string, not the number 10)'''

numbers = [3, 1, 4, 1, 5, 9, 2]

numbers[0] = "ten"

print(f"1. {numbers}")

'''2. Change the last element of numbers to 1'''

numbers = [3, 1, 4, 1, 5, 9, 2]

numbers[-1] = 1

print(f"2. {numbers}")

'''3. Get all the elements from numbers except the first two (slice)'''

numbers = [3, 1, 4, 1, 5, 9, 2]

x = numbers[2:]

print(f"3. {x}")

'''4. Check if 9 is an element of numbers'''

y = 9 in numbers

print(f"4. {y}")

# quick\_picks.py

def main():

picks = number\_of\_picks()

quick\_pick(picks)

def number\_of\_picks():

number = int(input("How many quick picks? "))

return number

def quick\_pick(picks):

import random

for i in range(picks):

constants = []

for x in range(6):

pick\_number = random.randint(1,45)

while pick\_number in constants:

pick\_number = random.randint(1, 45)

constants.append(pick\_number)

constants.sort()

print(" ".join("{:2}".format(number) for number in constants))

main()

# subject\_reader.py

"""

CP1404/CP5632 Practical

Data file -> lists program

"""

FILENAME = "subject\_data.txt"

def main():

data = get\_data()

print(data)

get\_detail(data)

def get\_data():

"""Read data from file formatted like: subject,lecturer,number of students."""

input\_file = open(FILENAME)

subject\_list = []

for line in input\_file:

print(line) # See what a line looks like

print(repr(line)) # See what a line really looks like

line = line.strip() # Remove the \n

parts = line.split(',') # Separate the data into its parts

print(parts) # See what the parts look like (notice the integer is a string)

parts[2] = int(parts[2]) # Make the number an integer (ignore PyCharm's warning)

print(parts) # See if that worked

subject\_list.append(parts)

print("----------")

return subject\_list

input\_file.close()

def get\_detail(data):

print("----------")

for i in data:

print("{} is taught by {} and has {} students".format(i[0],i[1],i[2]))

main()

# total\_income.py

"""

CP1404/CP5632 Practical

Starter code for cumulative total income program

"""

def main():

"""Display income report for incomes over a given number of months."""

incomes = []

number\_of\_months = int(input("How many months? "))

for month in range(1, number\_of\_months + 1):

income = float(input("Enter income for month {} : ".format(month)))

incomes.append(income)

print\_result(incomes,number\_of\_months)

def print\_result(incomes,number\_of\_months):

print("\nIncome Report\n-------------")

total = 0

for month in range(1, number\_of\_months + 1):

income = incomes[month - 1]

total += income

print("Month {:2} - Income: ${:10.2f} Total: ${:10.2f}".format(month, income, total))

main()