**HOMEWORK WEEK 3**

(handout for students)

**TASK 1 (Conditional flow)**

**Question 1**

Create a program that tells you whether or not you need an umbrella when you leave the house.

**The program should:**

1. Ask you if it is raining using input()

2. If the input is 'y', it should output 'Take an umbrella'

3. If the input is 'n', it should output 'You don't need an umbrella'

rain = input("Is it raining today? Please enter 'y' for yes 'n' for no. ")

if rain in ('y', 'Y'):

    print('Take an umbrella')

elif rain in ('n', 'N'):

    print("You don't need an umbrella")

else:

    print("Invalid input. Good bye!")

**Question 2**

I'm on holiday and want to hire a boat. The boat hire costs £20 + a refundable £5 deposit. I've written a program to check that I can afford the cost, but something doesn't seem right. Have a look at my program and work out what I've done wrong

my\_money = input('How much money do you have? ')

boat cost = 20 + 5

if my\_money < boat\_cost:

print('You can afford the boat hire')

else :

print('You cannot afford the board hire'

**Answer:**

**There are several bugs in this code**

1) variables (boat cost) should be a single word like boat\_cost

2) The values received through input() method appear to be of *string* type. To fix the problem, we need to change my\_money to *int* or *float* type

3) The logic is wrong, if you have more money than the cost of the boat (my\_money >= boat\_cost), then you should be able to hire a boat

4) print method lacks closing bracket at the end

The corrected code below:

my\_money = int(input('How much money do you have? '))

boat\_cost = 20 + 5

if my\_money >= boat\_cost:

    print('You can afford the boat hire')

else :

    print('You cannot afford the board hire')

**Question 3**

Your friend works for an antique book shop that sells books between **1800** and **1950** and wants to quickly categorise books by the century and decade that they were written. Write a program that takes a year (e.g. 1872) and outputs the century and decade (e.g. "Eighteenth Century, Seventies")

#Question 3

year = input("Please enter a year: ")

#century check

def century\_check(year):

    if year[:2] == '18':

        century = 'Nineteenth'

    else:

        century = 'Twentieth'

    return century

#decade check

def decade\_check(year):

    if year[2] == '1':

        decade = 'Tens'

    elif year[2] == '2':

        decade = 'Twenties'

    elif year[2] == '3':

        decade = 'Thirties'

    elif year[2] == '4':

        decade = 'Fourties'

    elif year[2] == '5':

        decade = 'Fifties'

    elif year[2] == '6':

        decade = 'Sixties'

    elif year[2] == '7':

        decade = 'Seventies'

    elif year[2] == '8':

        decade = 'Eighties'

    elif year[2] == '9':

        decade = 'Nineties'

    else:

        decade = 'Aughts'

    return decade

# year range check

def year\_range\_is\_valid(year, year\_min, year\_max):

    return year\_min <= int(year) <= year\_max

# main program

try:

    if year\_range\_is\_valid(year, 1800, 1950):

        century = century\_check(year)

        decade = decade\_check(year)

        print(f'{century} Century, {decade}')

    else:

        raise ValueError

except Exception:

    print("Please enter valid year in range 1800-1950")

**TASK 2 (Lists and Dictionaries)**

**Question 1**

I have a list of things I need to buy from my supermarket of choice.

shopping\_list = [

"oranges",

"cat food",

"sponge cake",

"long-grain rice",

"cheese board",

]

print(shopping\_list[1])

I want to know what the first thing I need to buy is. However, when I run the program it shows me a different answer to what I was expecting? What is the mistake? How do I fix it.

**Answer:** The first thing in the list comes at index 0. So you need to print(shopping\_list[0])

**Question 2**

I'm setting up my own market stall to sell chocolates. I need a basic till to check the prices of different chocolates that I sell. I've started the program and included the chocolates and their prices. Finish the program by **asking the user to input an item and then output its price**.

chocolates = {

'white': 1.50,

'milk': 1.20,

'dark': 1.80,

'vegan': 2.00,

}

choice = input('Please choose chocolate one type: white, milk, dark, vegan. ')

try:

    price = chocolates[choice]

    print(f'The price of {choice} chocolate is ${price}')

except Exception:

    print('There is no such option. Please try again')

**Question 3**

Write a program that simulates a lottery. The program should have a list of seven numbers that represent a lottery ticket. It should then generate seven random numbers. After comparing the two sets of numbers, the program should output a prize based on the number of matches:

· £20 for three matching numbers

· £40 for four matching numbers

· £100 for five matching numbers

· £10000 for six matching numbers

· £1000000 for seven matching numbers

**Answer:** I assumed that the lottery generates 7 unique random numbers in range of 1 to 100 .

# Question 3

import random

lottery\_ticket = {23, 1, 43, 62, 9, 47, 96} #given lottery ticket

#generating list of 7 unique random numbers

rand\_list = []

i = 0

while i < 7:

    num = random.randint(1, 100)

    if num not in rand\_list:

        rand\_list.append(num)

        i += 1

#counting number of matches

match\_count = 0

for num in rand\_list:

    if num in lottery\_ticket:

        match\_count += 1

#prize allocation

prize = '£0'

if match\_count == 3:

    prize = '£20'

elif match\_count == 4:

    prize = '£40'

elif match\_count == 5:

    prize = '£100'

elif match\_count == 6:

    prize = '£10000'

elif match\_count == 7:

    prize = '£1000000'

print(f'You have {match\_count} matches, your prize is {prize}! ')

**TASK 3 (Read and Write files)**

**Question 1**

**You're having coffee/tea/beverage of your choice with a friend that is learning to program in Python. They're curious about why they would use pip. Explain what pip is and one benefit of using pip.**

**Answer:**  PIP is a package installer for Python. It is used to install packages that are not built-in your computer already, or create packages of your own. The friend wants to get an information from some the website and parse it automatically. I can suggest her to write a program that will do it for her, using requests package. The package should be installed using pip install command.

**Question 2**

**This program should save my data to a file, but it doesn't work when I run it. What is the problem and how do I fix it?**

**poem = 'I like Python and I am not very good at poems'**

**with open('poem.txt', 'r') as poem\_file:**

**poem\_file.write(poem)**

**Answer:**

Currently the file is asked to be opened in read mode ‘r’, but it should be opened in write mode ‘w’.

The correct piece of code should look like follows:

poem = 'I like Python and I am not very good at poems'

with open('poem.txt', 'w') as poem\_file:

    poem\_file.write(poem)

**Question 3**

Here is a snippet of Elton John’s song “I’m Still Standing”

**You could never know what it's like**

**Your blood like winter freezes just like ice**

**And there's a cold lonely light that shines from you**

**You'll wind up like the wreck you hide behind that mask you use**

**And did you think this fool could never win?**

**Well look at me, I'm coming back again**

**I got a taste of love in a simple way**

**And if you need to know while I'm still standing, you just fade away**

**Don't you know I'm still standing better than I ever did**

**Looking like a true survivor, feeling like a little kid**

**I'm still standing after all this time**

**Picking up the pieces of my life without you on my mind**

**I'm still standing (Yeah, yeah, yeah)**

**I'm still standing (Yeah, yeah, yeah)**

**Tasks:**

1. Write the lyrics to a new file called song.txt

2. Check that a file has been created successfully.

3. The read lines from this file and print out ONLY those lines that have a word ‘still’ in them.

import os

with open('song.txt', 'w') as file:

    text = '''You could never know what it's like

Your blood like winter freezes just like ice

And there's a cold lonely light that shines from you

You'll wind up like the wreck you hide behind that mask you use

And did you think this fool could never win?

Well look at me, I'm coming back again

I got a taste of love in a simple way

And if you need to know while I'm still standing, you just fade away

Don't you know I'm still standing better than I ever did

Looking like a true survivor, feeling like a little kid

I'm still standing after all this time

Picking up the pieces of my life without you on my mind

I'm still standing (Yeah, yeah, yeah)

I'm still standing (Yeah, yeah, yeah)'''

    file.write(text)

if os.access('song.txt', os.F\_OK):

    print('The file has been created successfully ')

try:

    with open('song.txt', 'r') as file:

        [print(line) for line in file.readlines() if 'still' in line]

except FileNotFoundError:

    print("The file 'song.txt' does not exist. ")

**TASK 4 (API)**

**Question 1**

In this session you used the Pokémon API to retrieve a single Pokémon.

I want a program that can retrieve multiple Pokémon and save their names and moves to a file.

Use a list to store about 6 Pokémon IDs. Then in a for loop call the API to retrieve the data for each Pokémon. Save their names and moves into a file called 'pokemon.txt'

import requests

pokemon\_ID\_1 = input('What is pokemon\_1 ID? ')

pokemon\_ID\_2 = input('What is pokemon\_2 ID? ')

pokemon\_ID\_3 = input('What is pokemon\_3 ID? ')

pokemon\_ID\_4 = input('What is pokemon\_4 ID? ')

pokemon\_ID\_5 = input('What is pokemon\_5 ID? ')

pokemon\_ID\_6 = input('What is pokemon\_6 ID? ')

pockemon\_ids = [pokemon\_ID\_1, pokemon\_ID\_2, pokemon\_ID\_3, pokemon\_ID\_4, pokemon\_ID\_5, pokemon\_ID\_6]

with open('pokemon.txt', 'w') as file:

    for pokemon\_num in pockemon\_ids:

        url = 'https://pokeapi.co/api/v2/pokemon/{}/'.format(pokemon\_num)

        r = requests.get(url)

        pokemon\_data = r.json()

        moves = pokemon\_data['moves']

        moves\_str = ', '.join([move['move']['name'] for move in moves])

        file.write('pokemon name is {name}  '.format(name=pokemon\_data['name']))

        file.write('pokemon moves are: {moves\_str} \n'.format(moves\_str=moves\_str))

**Question 2 (optional)**

Here is a link to a really cool API:<https://opentdb.com/>

Answer the following questions:

· What is the name of this API?

· What does it do?

· Example URL to make a call to this API?

· Example output?