

MSBA7001 Assignment 2

Module 1, 2023-24
HKU Business School

Contents

Instructions.....	1
Q1 – six-digit consecutive numbers in pi	2
Q2 – HK fitness track	2
Q3 – CE policy address.....	3
Q4 – password generator	4

Instructions

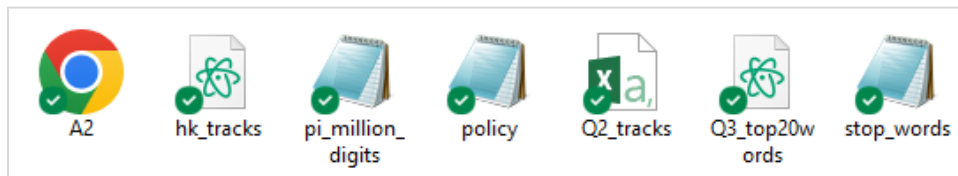
1. 4 questions, 5pts each, 20pts in total.
2. For every question, create an output heading, execute the required codes, and show your outputs. See an example below:

Q1 output

```
In [ ]: 1 print(len(sixdigit))
```

```
In [ ]: 1 print(sixdigit)
```

3. Keep codes and data files in the same fold. Use relative file path.



4. Partial points even if outputs are incorrect.
5. Save your codes in a Jupyter Notebook file named “A2.ipynb”
6. On Moodle, submit the ipynb and all other i/o data files as shown in #3.
7. Due at 11:30am, Sept 21 (Thursday).

Q1 – six-digit consecutive numbers in pi

Read from the text file “pi_million_digits.txt”, which includes the first million digits of pi. Write a program to find out all six-digit consecutive numbers (such as 123456, 345678, and 789012) in pi. Store these numbers as strings in a list called `sixdigit`.

In **two separate** cells, execute the following codes and show your outputs.

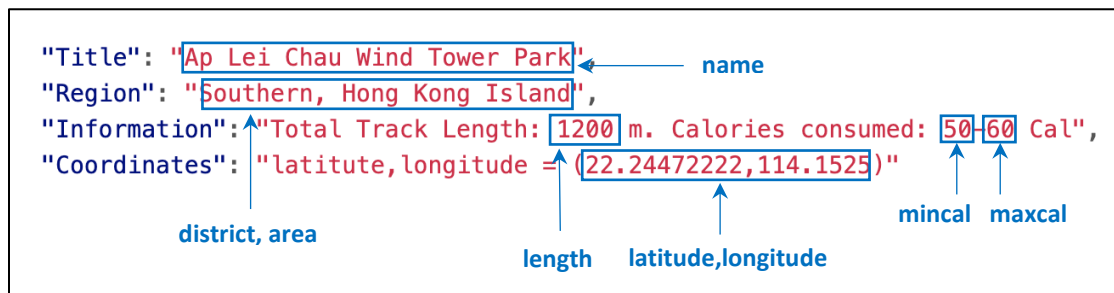
```
print(len(sixdigit))
```

```
print(sixdigit)
```

Q2 – HK fitness track

Note: this question is modified from a 2019 exam question.

Across Hong Kong, there are 34 fitness walking tracks. Detailed track data is stored as key-value pairs in the file “hk_tracks.json”. Your task is to extract all tracks’ data. See an example track from the file:



Notes:

1. From “Title”, extract each track’s name.
2. From “Region”, extract the district and area in which each track is located.
3. From “Coordinates”, extract each track’s latitude and longitude.
4. From “Information”, extract each track’s length, minimum calories (as mincal), and maximum calories (as maxcal).

Each track’s information should be stored in a list like below. Store all tracks’ information in a list of lists called `tracks`.

```
['Ap Lei Chau Wind Tower Park', 'Southern', 'Hong Kong Island', 22.24472222, 114.1525, '1200', '50', '60']
```

In **three separate** cells, execute the following codes and show your outputs.

```
print(len(tracks))
```

```
print(tracks[:5])
```

```
print(track[-5:])
```

Finally, use the `csv` module to write `tracks` to a csv file named “Q2_tracks.csv”. The header should be: “name, district, area, latitude, longitude, length, mincal, maxcal”. It has 35 rows (including the header). Your csv file should look like this when opened in Excel.

name	district	area	latitude	longitude	length	mincal	maxcal
Ap Lei Cha	Southern	Hong Kong	22.24472	114.1525	1200	50	60
Sun Yat Se	Central an	Hong Kong	22.2904	114.1438	2430	95	115
Hong Kong	Central an	Hong Kong	22.27714	114.1638	1200	45	50
Wan Chai	Wan Chai	Hong Kong	22.27547	114.176	321	10	15

Q3 – CE policy address

Note: this question is modified from a 2018 exam question.

On Oct 10th 2018, the Chief Executive of the HKSAR, Mrs. Carrie Lam, gave a policy address to the Legislative Council. The policy address lays out future plans for housing, health care and more.

The entire speech is saved in the text file “policy.txt”. Your goal is to find the top 20 most frequent words in the file and store them in a dictionary called `top20words`. The keys should be the capitalized words and the values should be their corresponding frequencies. Save the result in a descending order based on the frequency and then the word (if frequencies are the same). For example, the following are the most frequent and the least frequent words.

```
{'GOVERNMENT': 60, 'CHIEF': 13}
```

Notes:

1. **Punctuation in words:** “government’s” is considered the same as “government”.
2. **Singular and plural:** “government” and “governments” are considered as two different words.
3. **Hyphen/underscore in words:** “my-government” is considered as two different words; “my_government” is considered as two different words.
4. **Word variations:** “you”, “your” and “yours” are considered as three different words; “govern”, “governing”, and “governed” are considered as three different words; “government” and “governmental” are considered as two different words.
5. **Numbers:** ignore all numbers such as 2.8, 10, 2019.
6. **Stop words:** they refer to the most common words in a language such as “the,” “from,” and “are” in English. These words appear to be of little value. The text file “stop_words.txt” includes a list of common English stop words. Your `top20words` should not include any stop words listed in the file.

In **two separate** cells, execute the following codes and show your outputs.

```
print(len(top20words))
```

```
print(top20words)
```

Finally, write `top20words` to a json file named “`Q3_top20words.json`”. The first and the last items are shown below for your reference.

```
{
    "GOVERNMENT": 60,
    .....
    "CHIEF": 13
}
```

Q4 – password generator

Define a function called `password_gen(num)` that takes an integer number `num` and returns a valid password of length `num`. A valid password must

- Be created at random
- Have a length between 8 and 16 (both inclusive)
- Consist of only the following elements:
 1. at least one lowercase letter
 2. at least one uppercase letter
 3. at least one number between 0 and 9 (both inclusive)
 4. at least one special character in `_ $! ?` (they are underscore, dollar sign, exclamation mark, and question mark)

Notes:

- There is no additional requirement on the validity of the password.
- If the argument value is not an integer number between 8 and 16, the function should produce a warning message (see examples 1, 3, 6).
- You may define and call other functions as part of `password_gen(num)`.
- *Hint:* The `string` module may come in handy. Import it and explore its methods.

In **six separate** cells, execute the following codes and show your outputs. Sample outputs are presented below for your reference. Obviously, the passwords you generate should be most certainly different from the samples.

```
password_gen('8')           # example 1
'WARNING: please enter an integer number between 8 and 16.'
```

```
password_gen(8)             # example 2
'3?zvEEb5'
```

```
password_gen(9.6)          # example 3  
'WARNING: please enter an integer number between 8 and 16.'
```

```
password_gen(10)           # example 4  
'5uuCRg!U_v'
```

```
password_gen(13)           # example 5  
'HEd$YTDpKs4Us'
```

```
password_gen(20)           # example 6  
'WARNING: please enter an integer number between 8 and 16.'
```