

MSBA7001 Assignment 1

Module 1, 2024-25
HKU Business School

Contents

Instructions.....	1
Q1 – star matrix	2
Q2 – number pattern	2
Q3 – quiz scores.....	4
Q4 – ticket discount	5
Q5 – distinct substring	5
Q6 – same frequency by one removal.....	6
Q7 – rotate a letter.....	6

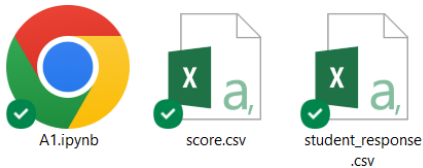
Instructions

1. 7 questions, 2pts each. 14pts in total.
2. For every question, create an output heading, execute the required codes, and show your outputs. See an example below:

Q1 output

1	star(5, 7)
1	star(7, 11)

3. Zero points if outputs are incorrect.
4. Keep script and data files in the same fold. Use relative file path. Name your script “A1.ipynb”.



5. On Moodle, submit the script and all other i/o data files as shown in #4.
6. Due at 11:59pm, Sept 10 (Tuesday).

Q1 – star matrix

Define a function called `star(num1, num2)` that prints a `num1 x num2` matrix of “stars”. Stars are separated by one space. See the following 5 x 7 matrix and 7 x 11 matrix as examples.

Both `num1` and `num2` should be odd integer numbers greater than 2. For simplicity, assume that valid arguments are always passed to the function.

In **two separate** cells, execute the following codes and show your outputs. The correct outputs are presented below for your reference.

```
star(5, 7) # Example 1
* * * *
 * * *
* * * *
 * * *
* * * *
```

```
star(7, 11) # Example 2
* * * * * *
 * * * * *
* * * * * *
 * * * * *
* * * * * *
 * * * * *
* * * * * *
 * * * * *
* * * * * *
```

Q2 – number pattern

Define a function called `pattern(start, total)` that takes two positive integer numbers and prints out a box of two number patterns that are stacked up horizontally. See the example of `pattern(5, 8)` below. The first number indicates the initial value of the left-pattern. There are five “5”s on the first row. The value decreases down the rows until it becomes 1. The difference between the second number and the first number (i.e., $8-5=3$) gives the initial value of the right-pattern. There are three “3”s on the first row. The value increases down the rows. On each row, the left-pattern and the right-pattern is separated by one space.

```

First number → 55555 333 ← Difference between
                  4444 4444 the second number
                  333 55555 and the first number
Left-pattern    22 666666 Right-pattern
                  1 777777

```

Notes:

1. For simplicity, assume that you always pass two integer numbers to the arguments when calling the function.
2. However, both the integer numbers should be between 1 and 9 (inclusive), and the 2nd number is greater than the 1st one. If this rule is violated, print out a reminder. See Examples 1, 2, & 3.
3. For any valid arguments, print out the correct patterns. See Examples 4, 5, & 6.
4. You may define and call other functions as part of the final function `pattern`.

In **six separate** cells, execute the following codes and show your outputs. The correct outputs are presented below for your reference.

```
pattern(6, 1) # Example 1
***REMINDER: Please make sure the two arguments are between 1 and 9,
and that the 2nd argument is greater than the 1st one.***
```

```
pattern(0, 7) # Example 2
***REMINDER: Please make sure the two arguments are between 1 and 9,
and that the 2nd argument is greater than the 1st one.***
```

```
pattern(5, 12) # Example 3
***REMINDER: Please make sure the two arguments are between 1 and 9,
and that the 2nd argument is greater than the 1st one.***
```

```
pattern(1, 3) # Example 4
1 22
```

```
pattern(5, 8) # Example 5
55555 333
4444 4444
333 55555
22 666666
1 7777777
```

```
pattern(8, 9) # Example 6
88888888 1
7777777 22
666666 333
55555 4444
4444 55555
333 666666
22 7777777
1 88888888
```

Q3 – quiz scores

Use the csv file “student_response.csv”. This data file is a record of 55 students’ answers to a quiz of 12 multiple choice questions. For example, student S1’s answer is ACBCCDABDABD.

student_name	answer
S1	ACBCCDABDABD
S2	ACDCDCDAADCD

The correct solution is **ADBCCBABCCAD** with each question valued at 1 point. Therefore, student S1 earns 7 points (the ones with an underscore are correct: ACBCCDABDABD). Note that several students’ answers have a different format that needs to be cleaned up:

S6	A B B C C B B C C C A D
S15	1.D 2.D 3.B 4.B 5.C 6.A 7.B 8.C 9.C 10.C 11.A 12.D
S26	B:C:B:C:C:D:B:C:B:B:A:D
S34	A-A-D-B-C-D-C-C-B-C-C-C
S44	1-A 2-B 3-B 4-C 5-B 6-D 7-C 8-C 9-C 10-D 11-C 12-C

Your tasks are to **clean up** (task 1) the answers when necessary, **calculate** (task 2) all 55 students’ scores and **store** the result in an appropriate data structure called **final_result**, and **write** (task 3) the result to a csv file named “score.csv”. It should have the following header: student_name, clean_answer, score. See a snapshot below:

student_name	clean_answer	score
S1	ACBCCDABDABD	7
S2	ACDCDCDAADCD	3
S3	ADDCCBBDCCBD	8
S4	ABCCCBACACAD	8
S5	ACCBBCACBBAC	3
S6	ABBCCBBCCCAD	9
S7	ACCCDDBAADAD	4

In your notebook, **print out** (task 4) all the “odd-numbered” (i.e., S1, S3, S5, etc.) students’ names, clean answers, and scores. There is no requirement on the printout format.

Q4 – ticket discount

An airline company is running the following promotional offers based on its customer's last name and the original ticket price. Note that offers can be accrued.

Offer	Condition on last name	Condition on original ticket price	Discount
1	1 st character of last name is 'D' or 'd'		\$10 discount
2	2 nd character of last name is 'I' or 'i'	Original price >= \$1000	\$100 discount
3	3 rd character of last name is 'R' or 'r'	Original price >= \$5000	\$500 discount
		\$500 <= Original price < \$5000	\$300 discount

Build a function called `discount(LaName, OriPrice)` that takes the last name and the original ticket price, and returns the final price after applying eligible discounts.

`LaName` should be a non-empty string and `OriPrice` should be a positive number. For simplicity, assume that valid arguments are always passed to the function.

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

```
discount('Wong', 10000) # hint: no offer applies
```

```
discount('d', 15) # hint: offer 1 applies
```

```
discount('Li', 4560) # hint: offer 2 applies
```

```
discount('Ding', 6000) # hint: offers 1 & 2 apply
```

```
discount('DIREN', 5800) # hint: offers 1 & 2 & 3 apply
```

Q5 – distinct substring

Build a function called `DistSub(text, k)` that finds the distinct substrings of length `k` in the `text`. Note that the distinct substrings should not be case-sensitive (see the last example). Print out all valid substrings in capital form, separated by a tab (`'\t'`).

`text` should be a string with its length greater than 2 and `k` should be an integer greater than 1. For simplicity, assume that valid arguments are always passed to the function.

In **five separate** cells, execute the following codes and show your outputs. The correct outputs are presented below for your reference.

```
DistSub('ABCAB', 8)
There is no such substring.
```

```
DistSub('ABCAB', 5)
```

There is only one distinct substring in ABCAB, which is itself.

```
DistSub('ABCAB', 3)
```

There are 3 distinct substrings in ABCAB. They are:

```
ABC  BCA  CAB
```

```
DistSub('ABCAB', 2)
```

There are 3 distinct substrings in ABCAB. They are:

```
AB   BC   CA
```

```
DistSub('ABCAb', 2)
```

There are 3 distinct substrings in ABCAB. They are:

```
AB   BC   CA
```

Q6 – same frequency by one removal

Build a function called `samefreq(text)` that finds whether all other elements in the `text` share the same frequency after removing only one element. Return True or False.

In **three separate** cells, execute the following codes and show your outputs. The correct outputs are presented below for your reference.

```
samefreq('CCCCDDDEEE') # remove letter C will do
True
```

```
samefreq('xxxxxyzz')
False
```

```
samefreq('abcdefgg') # remove letter g will do
True
```

Q7 – rotate a letter

Rotating a letter means to shift it through the alphabet, wrapping around to the beginning if necessary, so 'A' rotated by 3 is 'D' and 'Z' rotated by 1 is 'A'.

Build a function called `rotate_letter(letter, n)` that accepts a `letter` and an integer `n`, and returns a new letter that is rotated by the amount of `n`. If the arguments are invalid, return a reminder.

In **eight separate** cells, execute the following codes and show your outputs. The correct outputs are presented below for your reference.

```
rotate_letter('t', 3)
'w'
```

```
rotate_letter('Y', 5)
'D'
```

```
rotate_letter('k', 20.1)
'***REMINDER: Your input is not valid.***'
```

```
rotate_letter('abc', 19)
'***REMINDER: Your input is not valid.***'
```

```
rotate_letter('D', -4)
'Z'
```

```
rotate_letter('?', 8)
'***REMINDER: Your input is not valid.***'
```

```
rotate_letter(7, 109)
'***REMINDER: Your input is not valid.***'
```

```
rotate_letter(20, 'k')
'***REMINDER: Your input is not valid.***'
```

Hint: `ord` is a built-in function which converts a character to a numeric value, and `chr` is a built-in function which converts a numeric value to a character. For example:

```
ord('a')
97
```

```
chr(65)
A
```