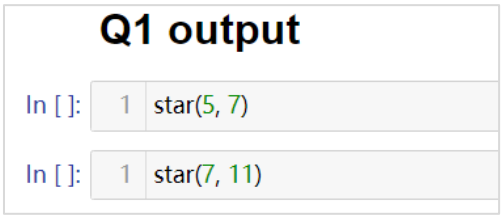
MSBA7001 Assignment 1

Module 1, 2023-24
HKU Business School

Contents

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:



3. Zero points if outputs are incorrect.
4. Save your codes in a Jupyter Notebook file named "A1.ipynb"
5. On Moodle, submit the ipynb file.
6. Due at 11:30am, Sept 14 (Thursday).

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.

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 – smallest and biggest input

Define a function called minnmax() that repeatedly accepts an integer number from the user until the user enters "done". Print out the smallest and the biggest numbers the user has entered.

Notes:

1. If the user enters anything other than an integer number or "done", print out a reminder. See Example 1.
2. If the user enters "done" on first input, print out a message and end the program. See Example 2.
3. **DO NOT** use built-in functions max and min.

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

```
minnmax() # Example 1
Enter an integer number or done: 5
Enter an integer number or done: -2
Enter an integer number or done: hku
***REMINDER: please enter an integer number or done***
Enter an integer number or done: 3.6
***REMINDER: please enter an integer number or done***
Enter an integer number or done: 11
Enter an integer number or done: 5.8
***REMINDER: please enter an integer number or done***
Enter an integer number or done: -9
Enter an integer number or done: 0
Enter an integer number or done: 8
Enter an integer number or done: done

The smallest number you entered is: -9
The biggest number you entered is: 11
```

```
minnmax() # Example 2
Enter an integer number or done: done
***No number received, bye***
```

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	1st character of last name is `'D'` or `'d'`		$10 discount
2	2nd character of last name is `'I'` or `'i'`	Original price >= $1000	$100 discount
3	3rd 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 discounted price.

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).

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 – verify ISBN-10

The International Standard Book Number (ISBN) is a numeric commercial book identifier. The ISBN-10 format has 10 digits (0 to 9) with the last digit being either a number or an X (e.g., 1-234-56789-X). Denote each digit by d_i, where $i = 1, \dots, 10$. The validity of ISBN-10 can be checked by the following:

$$\sum_{i=1}^{10} (11 - i) \cdot d_i \ \% \ 11 = 0$$

In the case the last digit is an X, then its value is 10. For example, 3-598-21507-X is a valid ISBN-10 according to the equation. Note that there are hyphens in ISBN-10 and they do not play a role.
$10 \times 3 + 9 \times 5 + 8 \times 9 + 7 \times 8 + 6 \times 2 + 5 \times 1 + 4 \times 5 + 3 \times 0 + 2 \times 7 + 1 \times 10 = 264$

Build a function called `isbn_check(isbn)` that checks whether a given ISBN-10 is valid. Return `True` if it's valid, `False` otherwise.

`isbn` should follow its standard format. For simplicity, assume that valid arguments are always passed to the function.

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

```
isbn_check('3-598-21507-X')
```

```
isbn_check('3-598-41508-8')
```

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, print out 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
```