Lab Test 1

General Instructions:

- Time allowed: 120 mins
- Total: 120 marks
- There are 6 questions
- Each Q in this Lab Test 1 is worth 20 marks
- Marks may be deducted for not following Python coding conventions, writing code that is hard to understand (e.g. unclear variable names or poor layout) or writing very inefficient code (e.g. unnecessary lines of code that do not improve clarity)
- You may write on this question paper, and you may keep the question paper
- If you finish early, you are welcome to leave early. However, please raise your hand first, so that the time can be recorded. You are not allowed to resubmit code after you leave. If you submit after you leave the room, you will be given a total score of 0 marks.

Submission Instructions:

- Write your answer for each Q in a separate cell, but all in ONE .ipynb (Jupyter Notebook) file
- Write a comment at the top of each cell, indicating which Q, e.g. "`### Q1`"
- When marking your work, we will assume that all the code for a Q is in only ONE cell, and that no other cell has previously been run
- Submit only ONE .ipynb file
- The name of the zip file should be the same as your ID. For example, if your email address is **jason.chan.2023@smu.edu.sg**, then your file should be called **jason.chan.2023**.ipynb

Q1: Volume of sphere

Write a function called `calculate_volume()` which calculates the volume of a sphere (ball). The function takes in one number as a parameter, which is the radius of the sphere. The function should return the volume of the sphere.

The volume of a sphere is given by this formula:

$$V = \frac{4}{3}\pi r^3$$

where:

- V is the volume of the sphere
- r is the radius of the sphere

Your function should create a variable to store the value of π as 3.14159.

For example:

- `calculate_volume(1)` should return 4.188786666666666
- `calculate_volume(2)` should return 33.51029333333333

You may assume that the function is only ever given non-negative numbers.

Q2: Leap year

Write a function called `is_leap_year()` which determines whether a given year is a leap year. The function takes in one integer as a parameter, which represents a year. The function should return a Boolean value, which should be **true** if the year is a leap year, and **false** otherwise.

Leap years are years that are multiples of 4 (e.g. 2020 and 2024 are leap years, and 2022 and 2023 are not leap years), except those years that are divisible by 100 (e.g. 2100 and 2200 are not leap years). Exceptions to the last rule are years that are divisible by 400 (e.g. 2000 and 2400 are leap years).

You may assume that the function is only ever given non-negative integers.

Q3: Longest suffix

Write a function called `get_longest_suffix()`. The function takes in two strings as parameters. The function should return a new string, which is the longest suffix (series of characters at the end of a word) that the two parameter strings share.

For example:

- `get_longest_suffix('station','stallion')` should return 'ion' (the last 3 letters), since the two words share the same last 3 letters
- `get_longest_suffix('firefighter','fighter')` should return 'fighter'
- `get_longest_suffix('fighter','firefighter')` should also return 'fighter' (the order of the parameters should not matter)
- `get_longest_suffix('python','programming')` should return '' (an empty string, since the characters at the end of the two strings do not match)

Q4: Swap digits

Write a function called `swap_digit_pairs()`. The function takes in one integer as a parameter. The function should return a new integer, where each pair of digits have been swapped.

If there are an odd number of digits, then the left-most digit should remain in the same place.

For example:

- `swap_digit_pairs(123456)` should return 214365 (the 1 swapped with the 2, the 3 swapped with the 4, and the 5 swapped with the 6)
- `swap_digit_pairs(12345)` should return 13254 (the 1 had no digit to swap with, so it remains in the same position)

You may assume that the function is only ever given non-negative numbers, and that no numbers begin with a 0.

Your function should only use the integer data type. If you use any other data type (e.g. strings), then a 5 mark penalty will be imposed.

Q5: Print fan

Write a function called `print_pattern()` which determines a pattern on the screen, depending on the value of the integer parameter n.

Some examples are given below.

If n is 5, your function should print the following:

```
*****    *
 ****   **
  ***  ***
   ** ****
    ******
******
**** **
***  ***
**   ****
*    *****
```

If n is 8, your function should print the following:

```
********       *
 *******      **
  ******     ***
   *****    ****
    ****   *****
     ***  ******
      ** *******
       *********
*********
******* **
******  ***
*****   ****
****    *****
***     ******
**      *******
*       ********
```

Your solution should only contain one pair of nested loops, and only one if-statement inside the inner-most loop, as taught in the lessons. If you do not follow this requirement, then a maximum of 5 marks (not 20 marks) can be awarded for this question.

Q6: Jumble words

Write a function called `jumble_words()` which is given a string as a parameter.

The function jumbles the order of letters in each word, as follows:

- The first and last letter always remain in the same position
- For every other letter, all the characters at even-numbered positions appear in their original order, followed by the characters in the odd-numbered positions.

Some examples of words and their jumbled order are given here:

Before jumbling	**After jumbling**
`"0123456789"`	`"0246813579"`
`"01234"`	`"02134"`
`"Programming"`	`"Pormirgamng"`
`"is"`	`"is"`
`"fun"`	`"fun"`
`"such"`	`"scuh"`

The function returns a list (not a string), which contains all the jumbled words.

The order of the words should still be maintained. Only the order of letters in the words should be jumbled.

For example:

- `jumble_words("Programming is really fun")` should return the list ['Pormirgamng', 'is', 'ralely', 'fun']
- `jumble_words("Python is such an interesting language")` should return the list ['Ptoyhn', 'is', 'scuh', 'an', 'itrsineetng', 'lnugagae']

You may assume that if there is a space in the given string, then there will be a word on both the left and right side of that space.

You may also assume that the string only contains characters and space. So there will not be any other symbol (other than space) that is used to separate words.

Your function must NOT use the split() method on strings. If you use the split() method, then a 10 mark penalty will be imposed.

There are no more questions