

% Name your Jupyter Notebook as

TASKF_<your name>_<centre number>_<index number>.ipynb

A credit card number typically has 16 digits: 15 digits with an extra check digit added to the right-hand side.

The check digit is calculated using the following algorithm on the left-most 15 digits of a credit card number:

1. starting from the left, the first digit is location number 1
2. double all digits in the odd numbered positions
3. sum all the digits, including both the unchanged digits (i.e. those in the even numbered positions) as well as those doubled (e.g. 12 contributes to 1 + 2)
4. the check digit is the value between 0 and 9 that must be added to the sum to make the result exactly divisible by 10.

For example, given the 15 digits 540012620081719

Step 1: 5 4 0 0 1 2 6 2 0 0 8 1 7 1 9
Step 2: 10 4 0 0 2 2 12 2 0 0 16 1 14 1 18
Step 3: (1+0) + 4 + 0 + 0 + 2 + 2 + (1+2) + 2 + 0 + 0 + (1+6) + 1 + (1+4) + 1 + (1+8)
 Sum = 37
Step 4: Check digit = 3 since 37 + 3 = 40, which is divisible by 10.

For each of the sub-tasks, add a comment statement at the beginning of the code, using the hash symbol '#' to indicate the sub-task the program code belongs to, for example:

```
In [1] :    # Task 3.1  
          Program code
```

Output:

Task %1

Write a function `taskF_1(input_value)` that returns an integer.

The function should:

- validate that the parameter `input_value` is either an integer, or a string containing a valid integer
- check that it is 15 digits in length
- return -1 if the value received is invalid for any reason
- calculate the check digit for the given first 15 digits of the credit card number
- return the calculated check digit.

[6]

Task %2

Create **four** tests that should fully test your function. Ensure that each test validates the inputs accurately and returns the correct expected result.

Each of the four tests involves a pair of data items: the input value and its expected result.

Your four input values should be:

- a string containing just a valid integer
- a valid integer
- a string containing characters that are not digits
- a value of the incorrect length.

Test your function with your four input values by calling it using the following statement:

```
print(taskF_1(input_value) == expected)
```

The four statements should all print `True`.

For example:

```
print(taskF_1("540012620081719") == 3)
```

Do not re-use the example provided above.

[4]

Task %3

A full 16-digit credit card number can be validated by removing the check digit, calculating the check digit from the remaining 15 digits and comparing it to the removed digit. It is also given that a credit card number is only valid if it starts with a digit 3, 4, 5 or 6.

Write a function `taskF_3(input_value)` that returns a Boolean.

The function should:

- validate that the parameter `input_value`
 - is either an integer, or a string containing a valid integer
 - is 16 digits in length
 - starts with a digit 3, 4, 5 or 6
- return `False` if the value received is invalid for any reason
- remove the digit on the right-hand side
- use your function from **Task %1** to calculate the check digit from the remaining 15 digits
- return a Boolean representing the comparison between the calculated check digit and the original removed digit.

[6]