3 Name your Jupyter Notebook as

`TASK3_<your name>_<centre number>_<index number>.ipynb`

The task is to write a function that takes a sequence of characters that represents a quantity of data and unit, and translates this quantity to a different unit.

The basic unit of data is the byte (B):

- A kilobyte (KB) is 10^3 bytes
- A megabyte (MB) is 10^6 bytes
- A gigabyte (GB) is 10^9 bytes
- A terabyte (TB) is 10^{12} bytes.

For example, 8KB has 8000 bytes.

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 3.1

Write a function called `task3_1(quantity_of_data)` that:

- takes a string, `quantity_of_data`
- tests that the given string is a sequence of digits followed by one of the four approved units shown above (KB, MB, GB, TB).
- returns and displays either:
  - the actual number of bytes represented by the input string

  or
  - the error message, `"invalid data"`. [5]

Test the function fully with suitable test data, including all four approved units.

For example,

```
task3_1("8KB")
```

should return and display `8000` [3]

Task 3.2

Companion units are also defined in terms of powers of 2. These have similar abbreviations, as shown:

- A kibibyte (KiB) is 2^{10} bytes
- A mebibyte (MiB) is 2^{20} bytes
- A gibibyte (GiB) is 2^{30} bytes
- A tebibyte (TiB) is 2^{40} bytes.

Write a second function `task3_2(quantity_of_data)` that:

- takes a string, `quantity_of_data`
- tests that the given string is a sequence of digits followed by one of the eight approved units (KB, KiB, MB, MiB, GB, GiB, TB, TiB)
- returns and displays either:
    - the number of bytes represented by the input string

    or
    - the error message, `"invalid data"` [5]

Test the function fully with suitable test data, including all eight approved units.

For example,

```
task3_2("2MiB")
```

should return and display `2097152` [3]

Task 3.3

Write a third function, `task3_3(quantity_of_data, target_unit)` that:

- takes two strings, `quantity_of_data` and `target_unit`
- tests that `target_unit` is one of the eight approved units from task 3.2
- uses your function `task3_2` to generate the actual number of bytes represented by `quantity_of_data`
- converts the generated number of bytes into `target_unit`
- returns and displays either:
    - the `quantity_of_data` in terms of the `target_unit`

    or
    - the error message, `"invalid data"` [4]

Test the function with **three** suitable sets of values.

For example,

```
task3_3("512MiB", "GiB")
```

should return and display `0.5` [3]

Save your Jupyter Notebook for Task 3.