Instruction to candidates:

Your program code and output for each of Task 1 to N should be saved in a single `.ipynb` file using Jupyter Notebook. For example, your program code and output for Task 1 should be saved as:

`Task1_<your name>_<centre number>_<index number>.ipynb`

Make sure that each of your `.ipynb` files shows the required output in Jupyter Notebook.

1 Name your Jupyter Notebook as

`Task2_<your name>_<centre number>_<index number>.ipynb`

In a school, students are identified by student numbers. These numbers are stored in a hash table which uses the hashing function

`Address ← StudentNumber MOD X`

The hash table is implemented as a one-dimensional array with elements indexed `0` to `(X-1)`.

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 2.1
          Program code

          Output:
```

Task 2.1

Write program code to:

- read student numbers from a text file and store them in a hash table, where `X` is to be set to the value of `12`.
- print out the contents of the hash table in the order in which the elements are stored in the array.

For the purpose of this sub-task, assume different student numbers will hash to different addresses (no collisions).

Use `KEYS.TXT` to test your program code.

[7]

Task 2.2

Linear probing is a method to handle collisions. In linear probing, a collision is resolved by searching sequentially from the hashed address for an empty location and storing the data at this empty location. If the end of the table is reached, the search for an empty location is continued from the start of the table.

Amend your code in **Task 2.1** to handle collisions using linear probing.

Use `KEYS2.TXT` to test your amended program code.

[4]

Task 2.3

Write additional code to:

- take as input a student number
- search the hash table and output the address (index number) of the hash table where the student number was found.

Use `KEYS2.TXT` to test your code.

Run the program three times with the following inputs:

- `15`
- `23`
- `88`

[7]

Save your Jupyter Notebook for Task 2.