Task 1: Table Duplication (30%)

Task Description

Write a program that allows a user to display and duplicate tables. At the beginning of the program, some tables are imported from a separate module `table_data.py` that you must not modify.

> Use the module `tabulate` to print tables in the expected format.

Main menu

The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
```

If the user inputs 1, 2, 3 or 0, it runs the corresponding menu option.

> Assume that the user and the automated tests input a correct choice in the main menu.

> We recommend you create a function for each menu option.

1. List tables

This menu option prints a table that lists the tables that currently exist in the program.

```
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      2          3       4
      3          3       6
```

After the header, there is one row per table, with its index, and the number of columns and rows of that table.

At the beginning of the program, all tables from the module `table_data` are shown in the list, which then looks exactly as shown above.

This list is updated as tables change. See example 1.

2. Display table

This menu option prints a table given its index.

The program prints `Choose a table index (to display):`. The user then selects a table by its index (as listed by menu option 1). See below.

```
Choose a table index (to display):
1
  Student ID  First Name    Last Name    Grade Code
------------  ------------  -----------  ------------
      798154  Brynhildr     Blakeley     N
      134789  Felix         Li           N
      798951  Paityn        Summers      P
      465120  Turnus        Elliot       C
      963245  Alysia        Jervis       D
      469120  Muhammad      Saad         HD
```

If an incorrect table index is provided, the program prints `Incorrect table index. Try again.`, and prints `Choose a table index (to display):` again, until a correct index is provided. See example 2.

> For menu option 2 and future menu options, assume that, when asked to input a table index, the user inputs an integer.

3. Duplicate table

Similar to menu option 2, this menu option repeatedly asks the user to select a table, then creates a copy of that table. The index of the new table is the smallest index that has not previously been assigned to a table.

```
Choose a table index (to duplicate):
1
```

See example 3.

Examples

User inputs are in **bold font** below.

Example 1

```
================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
================================
```

```
1
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      2          3       4
      3          3       6
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
0
```

Example 2

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
2
Choose a table index (to display):
4
Incorrect table index. Try again.
Choose a table index (to display):
3
Rabbit      Birth year  Favorite treat
--------  ------------  ----------------
Rabbit_1          2022  Carrots
Rabbit_2          2023  Celery
Rabbit_3          2023  Broccoli
Rabbit_4          2024  Cabbage
Rabbit_5          2022  Lettuce
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
0
```

Example 3

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
```

```
3
Choose a table index (to duplicate):
1
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
1
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      2          3       4
      3          3       6
      4          4       7
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
2
Choose a table index (to display):
4
  Student ID  First Name    Last Name    Grade Code
------------  ------------  -----------  ------------
      798154  Brynhildr     Blakeley     N
      134789  Felix         Li           N
      798951  Paityn        Summers      P
      465120  Turnus        Elliot       C
      963245  Alysia        Jervis       D
      469120  Muhammad      Saad         HD
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
0. Quit.
==================================
0
```

Task 2: Table Creation and Deletion (40%)

Task Description

Write a program that, on top of the previous options, allows a user to create and delete a table.

Main menu

The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
```

If the user inputs 1, 2, 3, 4, 5, or 0, it runs the corresponding menu option.

4. Create table

This menu creates a table from an existing table by selecting some of its columns in a certain order.

It repeatedly asks the user `Choose a table index (to create from):` until a correct index is input.

It then asks the user `Enter the comma-separated indices of the columns to keep:`.

Each index must be between 0 and the current number of columns of that table, minus one.

The order of the indices determines the order of the columns of the created table.

The index of the new table is the smallest index that has not previously been assigned to a table.

See example 1.

> Assume that the user and test inputs are well-formatted (e.g. `3,1`, without spaces) and the set of indices is valid.

5. Delete table

This menu option repeatedly asks the user to select a table by its index (as shown by menu option 1) with `Choose a table index (for table deletion):`, then deletes the table.

Note that deleting a table does not change the indices of other tables. See example 2.

Examples

User inputs are in **bold font** below.

Example 1

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
4
Choose a table index (to create from):
6
Incorrect table index. Try again.
Choose a table index (to create from):
1
Enter the comma-separated indices of the columns to keep:
3,1
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
2
Choose a table index (to display):
4
Grade Code    First Name
------------  ------------
N             Brynhildr
N             Felix
P             Paityn
C             Turnus
D             Alysia
HD            Muhammad
==================================
Enter your choice:
```

```
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
0
```

Example 2

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
5
Choose a table index (for table deletion):
2
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
1
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      3          3       6
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
2
Choose a table index (to display):
2
Incorrect table index. Try again.
Choose a table index (to display):
1
  Student ID  First Name    Last Name    Grade Code
------------  ------------  -----------  ------------
      798154  Brynhildr     Blakeley     N
      134789  Felix         Li           N
      798951  Paityn        Summers      P
      465120  Turnus        Elliot       C
```

```
      963245  Alysia        Jervis       D
      469120  Muhammad      Saad         HD
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
5
Choose a table index (for table deletion):
2
Incorrect table index. Try again.
Choose a table index (for table deletion):
3
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
0. Quit.
==================================
0
```

Task 3: Column Deletion and Table Restoration (30%)

Task Description

Write a program that, on top of the previous options, allows a user to delete a column and restore a table.

Main menu

The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
```

If the user inputs 1, 2, 3, 4, 5, 6, 7 or 0, it runs the corresponding menu option.

6. Delete column

This menu option allows the user to delete a column from a table.

Similar to other menu options, the program repeatedly prints `Choose a table index (for column deletion):` until a correct index is input.

Then the program prints `Enter the index of the column to delete:`.

The user then inputs one column index, between 0 and the current number of columns of that table, minus one.

That column is then deleted, which is for example reflected in menu option 1 and 2.

See example 1.

> Assume that the column index is valid.
>
> Deleting a column in a table does not affect any other table.

7. Restore table

This menu option allows a user to restore a table that has been deleted via menu option 5.

It prompts `Choose a table index (for restoration):` and reads a table index from the user.

If the table index is invalid (either the table exists and has not been deleted, or has never existed), then the program prints `Incorrect table index. Try again.`, and iterates until a valid index has been input.

The deleted table is then restored with the index it had prior to deletion.

> Assume that this menu option is only used if a table has been deleted.

Examples

User inputs are in **bold font** below.

Example 1

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
1
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      2          3       4
      3          3       6
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
6
```

```
Choose a table index (for column deletion):
6
Incorrect table index. Try again.
Choose a table index (for column deletion):
2
Enter the index of the column to delete:
1
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
1
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      2          2       4
      3          3       6
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
2
Choose a table index (to display):
2
First Name    Rabbit
------------  --------
Brynhildr     Rabbit_1
Turnus        Rabbit_2
Jamaluddin    Rabbit_5
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
0
```

Example 2

```
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
5
Choose a table index (for table deletion):
2
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
7
Choose a table index (for restoration):
1
Incorrect table index. Try again.
Choose a table index (for restoration):
2
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
1
  Index    Columns    Rows
-------  ---------  ------
      0          5       6
      1          4       7
      2          3       4
      3          3       6
==================================
Enter your choice:
1. List tables.
2. Display table.
3. Duplicate table.
4. Create table.
5. Delete table.
6. Delete column.
7. Restore table.
0. Quit.
==================================
0
```

Task 4: Breeding Rabbytes (40%)

You receive the text "Yo. Fibo here. We're classing up the breeding program. You in?"

Your boy Fibonacci is counting on you. Of course you're in!

Task Description

Write a program that allows the user to interact with a console menu to list, breed two rabbytes, and simulate the passage of time.

> Requirement: to represent the rabbytes, you must create and use at least one class in your program. This holds for the next tasks.
>
> Write your program from scratch, without reusing previous rabbyte work. There's almost nothing in common. This holds for the next tasks.

Main menu

The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
```

If the user inputs 1, 2, 3 or 0, it runs the corresponding function.

> Assume that a correct index is input in the main menu.

Initial Rabbit population

At the beginning of the program, initialise the rabbit population as shown in the table below.

Each rabbit has an ID, a sex (male or female), an age, two parents (they may be unknown). Female rabbytes can be pregnant.

You may need to store additional information of your choosing to write the program.

> *Initialising* the rabbit population does not necessarily means these initial rabbytes need to be created in an `__init__` method!

1. List Rabbytes

Upon inputting `1`, the program will then print a table of all rabbytes.

At the beginning of the execution of the program, the table is as shown below.

```
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                No
   2  M          2                No
   3  F          3                No
   4  M          4                No
   5  F          5                No
   6  M          6                No
====  =====  =====  =====  =====  ==========
```

If the dad or the mom of a rabbit is known, their ID is shown in the corresponding column.

> Use the module `tabulate` to print the list of rabbytes in the expected format.

2. Breed a Rabbit

Upon inputting `2`, the program will then ask the user for two rabbytes, who will then be bred. See example below.

```
Which rabbytes do you want to breed?
First parent:
1
Second parent:
2
```

> Assume the user will input two integers.

A certain number of checks are performed to ensure that the breeding is possible:

1. Each input must be a rabbit ID. Otherwise, `Rabbit not found. Try again.` is printed, and the parent is asked for again. See example 1.
2. If it is a valid rabbit, other checks are performed, in this order:
    1. If the age of the rabbit is 0, the error is `Rabbit too young.`.
    2. When the second parent is input, if the sex of both parents is the same, then the error is `Rabbit cannot be male.`, or `Rabbit cannot be female.`.
    3. If the rabbit is already pregnant, the error is `Rabbit cannot be pregnant.`.

Multiple errors are given on the same line if there are multiple errors with the input. See example 2.

> Assume menu option 2 is used only if there is valid pair of rabbytes that can breed.

Once a valid pair of rabbytes is given, the female one becomes pregnant. See example 2.

3. Wait 1 year

Upon inputting `3`, the program will then age all rabbytes by 1 year, and all pregnant rabbytes will give birth to a new 0-year old kitten.

Kittens are born in the order of their mother's ID (smallest first). The ID of a kitten is the smallest non-negative integer currently not-assigned to another rabbit.

The sex of a newborn kitten is male if the ID of the father is smaller than the ID of the mother, female otherwise.

See example 2.

Examples

User inputs are in **bold font** below.

Example 1

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
-1
Rabbit not found. Try again.
First parent:
1
Second parent:
10
Rabbit not found. Try again.
Second parent:
2
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
```

```
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                Yes
   2  M          2                No
   3  F          3                No
   4  M          4                No
   5  F          5                No
   6  M          6                No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
0
```

Example 2

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                No
   2  M          2                No
   3  F          3                No
   4  M          4                No
   5  F          5                No
   6  M          6                No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
0
Rabbit too young. Try again.
First parent:
1
Second parent:
```

```
3
Rabbit cannot be female. Try again.
Second parent:
2
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                Yes
   2  M          2                No
   3  F          3                No
   4  M          4                No
   5  F          5                No
   6  M          6                No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
2
Second parent:
1
Rabbit cannot be pregnant. Try again.
Second parent:
0
Rabbit too young. Rabbit cannot be male. Try again.
Second parent:
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                Yes
   2  M          2                No
   3  F          3                Yes
   4  M          4                No
   5  F          5                No
   6  M          6                No
====  =====  =====  =====  =====  ==========
```

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          1                No
   1  F          2                No
   2  M          3                No
   3  F          4                No
   4  M          5                No
   5  F          6                No
   6  M          7                No
   7  F          0  2      1      No
   8  M          0  2      3      No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
0. Quit.
==================================
0
```

Task 5: Rabbyte (step-)siblings (30%)

"We're not going to make the same mistakes as last time", says Fibonacci.

"We'll want to avoid breeding rabbytes with their siblings and step-siblings."

You wonder what happened last time, but you know better than to ask.

Task Description

Improve upon the program you wrote in the previous task by adding a menu option to find siblings, and another one to find step-siblings.

Two rabbytes are siblings if they have the same two parents. They are half-siblings if they share a single parent.

For a given rabbit `x` and their half-sibling `y`, a step-sibling is defined as a rabbit whose parent is also a parent of `y` (but not one of `x`'s).

For example, if we had:

```
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          1                No
   1  F          1                No
   2  M          2                No
   3  F          3                No
   4  M          4      0      1  No
   5  F          5      0      3  No
   6  M          6      2      3  No
====  =====  =====  =====  =====  ==========
```

then `4` and `5` are half-siblings, as their dad is `0`, and `6` is a half-sibling of `5`, and a step-sibling of `4`.

Main menu

The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
```

Assume that a correct index is input in the main menu.

4. Find siblings

Upon inputting `4`, the program will then prompt `Rabbit to find siblings of:`, read a rabbit ID `<x>` from the user, and finally print `The siblings of Rabbit <x> are:`, followed by the sorted IDs of the siblings, one per line. See example 1.

Similar to option menu 2, the user is prompted again if the ID input does not correspond to a rabbit. See example 2.

5. Find step-siblings

Option menu 5 is very similar to option menu 4.

Upon inputting `5`, the program will then prompt `Rabbit to find step-siblings of:`, read a rabbit ID `<x>` from the user, and finally print `The step-siblings of Rabbit <x> are:`, followed by the sorted IDs of the step-siblings, one per line. See example 3.

Similar to option menu 2 and 4, the user is prompted again if the ID input does not correspond to a rabbit.

Examples

User inputs are in **bold font** below.

Example 1

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
1
Second parent:
2
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
```

```
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
2
Second parent:
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          1                No
   1  F          2                No
   2  M          3                No
   3  F          4                No
   4  M          5                No
   5  F          6                No
   6  M          7                No
   7  F          0  2      1      No
   8  M          0  2      3      No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
1
Second parent:
2
==================================
Enter your choice:
1. List Rabbytes.
```

```
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          2                No
   1  F          3                No
   2  M          4                No
   3  F          5                No
   4  M          6                No
   5  F          7                No
   6  M          8                No
   7  F          1  2      1      No
   8  M          1  2      3      No
   9  F          0  2      1      No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
4
Rabbit to find siblings of:
7
The siblings of Rabbit 7 are:
9
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
0
```

Example 2

```
==================================
```

```
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
4
Rabbit to find siblings of:
-1
Rabbit not found. Try again.
Rabbit to find siblings of:
10
Rabbit not found. Try again.
Rabbit to find siblings of:
5
The siblings of Rabbit 5 are:
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
0
```

Example 3

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
1
Second parent:
2
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
2
```

```
Second parent:
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
1
Second parent:
4
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          2                No
   1  F          3                No
   2  M          4                No
   3  F          5                No
   4  M          6                No
   5  F          7                No
   6  M          8                No
   7  F          1  2      1      No
   8  M          1  2      3      No
   9  F          0  4      1      No
====  =====  =====  =====  =====  ==========
```

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
5
Rabbit to find step-siblings of:
8
The step-siblings of Rabbit 8 are:
9
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
0. Quit.
==================================
0
```

Task 6: Breed all Rabbytes (30%)

"We've got to crank it up a notch", explains Fibonacci.

"No matter how careful I am, my students always end up misusing my program. We need it to be foolproof. Fool-, proof!"

You oblige, reluctantly though, as you know that Fibonacci's students will always be one step ahead of him. When will he learn?

Task Description

Improve upon the program you wrote in the previous task by adding a menu option to implementing a safe "breed all" menu option, which finds pairs of rabbytes to breed which are a certain parental distance from each other.

For example, a rabbit and their kitten are at distance 1.

Two siblings, half-siblings, or a grandparent and their grandchild are at distance 2.

A rabbit and their aunt or uncle are at distance 3.

A rabbit and their cousin or step-sibling are at distance 4.

Therefore, the "breed all" menu option will not allow breeding between two rabbytes unless the distance between them is at least 5.

> Use the module `networkx` to represent the parental relationships between rabbytes as a graph and to find the distance between them.

Main menu

The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
```

> Assume that a correct index is input in the main menu.

6. Breed all

Upon inputting `6`, the program attempts to breed rabbytes by order of ID.

For each rabbit `x`, it looks for another rabbit `y` it can breed with, at distance at least 5.

In case there are multiple rabbits at distance at least 5, then `y`, the rabbit selected for breeding is

1. the closest in age to `x`, and, if there is another tie,
2. the one with the smallest ID out of the remaining ones.

As before, an underage or pregnant rabbit cannot breed.

For each pair thus found, the female becomes pregnant, and the program prints `Breeding Rabbytes <x> and <y>.`.

See examples 1 and 2.

Examples

User inputs are in **bold font** below.

Example 1

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
6
Breeding Rabbytes 1 and 2.
Breeding Rabbytes 2 and 3.
Breeding Rabbytes 4 and 5.
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
```

```
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                Yes
   2  M          2                No
   3  F          3                Yes
   4  M          4                No
   5  F          5                Yes
   6  M          6                No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
0
```

Example 2

```
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
2
Which rabbytes do you want to breed?
First parent:
6
Second parent:
1
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          0                No
   1  F          1                Yes
   2  M          2                No
```

```
   3  F          3                No
   4  M          4                No
   5  F          5                No
   6  M          6                No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
6
Breeding Rabbytes 2 and 3.
Breeding Rabbytes 4 and 5.
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          1                No
   1  F          2                No
   2  M          3                No
   3  F          4                No
   4  M          5                No
   5  F          6                No
   6  M          7                No
   7  F          0  6      1      No
   8  M          0  2      3      No
   9  M          0  4      5      No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
```

```
6. Breed all.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
6
Breeding Rabbytes 0 and 1.
Breeding Rabbytes 2 and 5.
Breeding Rabbytes 3 and 4.
Breeding Rabbytes 4 and 7.
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
3
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
1
====  =====  =====  =====  =====  ==========
  ID  Sex      Age  Dad    Mom    Pregnant
====  =====  =====  =====  =====  ==========
   0  M          4                No
   1  F          5                No
   2  M          6                No
   3  F          7                No
   4  M          8                No
   5  F          9                No
```

```
   6  M         10                No
   7  F          3  6      1      No
   8  M          3  2      3      No
   9  M          3  4      5      No
  10  M          0  0      1      No
  11  F          0  4      3      No
  12  M          0  2      5      No
  13  M          0  4      7      No
====  =====  =====  =====  =====  ==========
==================================
Enter your choice:
1. List Rabbytes.
2. Breed a Rabbit.
3. Wait 1 year.
4. Find siblings.
5. Find step-siblings.
6. Breed all.
0. Quit.
==================================
0
```

Task 7: Bakery City Distance (30%)

Task Description

Write a program that allows the user to input two cities by their ID, and outputs the geodesic distance between the two cities, rounded up to the nearest whole kilometre.

> The city data is provided in the `city_data.py` file given in the scaffold. Do not edit this file.
>
> Unlike many of our previous tasks, you do not need to implement a menu in this task.

Cities

The bakery vehicles travel only between certain cities.

Not all cities in the file `city_data.py` will be available in our program.

The available cities are those that, in the file `city_data.py`, either:

- are a "primary" capital, which means that under the header capital, the data is "primary", or
- have a population of at least 5 million people.

For example, it is a possible for vehicles to go to

- Canberra or Kuala Lumpur, because they are a "primary" capital, and
- Delhi or New York, as, although they are not "primary" capitals, they have a large enough population.

Examples of cities that do not make the cut are Melbourne and Sydney.

> This holds for the following tasks, which means, for example, that vehicles will not be able to depart, arrive or even transit via Melbourne or Sydney

Program execution

Starting Output

At the beginning of its execution, the program prints

```
Delivering baked goods to 363 cities.
```

which indicates that 363 cities meet the criteria.

Distance between cities

After the starting output has been printed, the program prints the prompt

```
Enter the ID of the departure city:
```

upon which the user enters a city ID (e.g. `1392685764` for Tokyo, see city_data.py).

The program then prints

```
Enter the ID of the destination city:
```

upon which the user enters a city ID (e.g. `1360771077` for Jakarta, see city_data.py).

The program then prints

```
Distance between Tokyo and Jakarta:
5777 km
```

and terminates. See examples.

> Use the module `geopy` to compute the geodesic distance between two cities based on their latitude and longitude.
>
> Assume the input city IDs input by the user are correct, and correspond to one of the 363 cities.

Examples

User inputs are in **bold font** below.

Example 1

```
Delivering baked goods to 363 cities.
Enter the ID of the departure city:
1392685764
Enter the ID of the destination city:
1360771077
Distance between Tokyo and Jakarta:
5777 km
```

Example 2

```
Delivering baked goods to 363 cities.
Enter the ID of the departure city:
1076532519
Enter the ID of the destination city:
```

1152554349
Distance between Sao Paulo and Santiago:
2588 km

Task 8: Bakery Vehicle Travel Time (40%)

Task Description

Write a program that allows the user to input two cities by their ID, and outputs the direct travel time, rounded up to the nearest hour, between these two cities, for each available vehicle. If a vehicle cannot travel directly between the two cities, then the output for this vehicle should be "Infinity".

> Requirement: to represent the vehicles, you must create at least an abstract class in your program, and inherit from this class to represent all vehicles. This holds for the next tasks.

Vehicles

In this task, we are considering three types of vehicles:

1. The Crappy Crêpe Car, which is a vehicle that travels at fixed speed between any two cities.
2. The Diplomacy Donut Dinghy, which:
    1. travels at a given "domestic speed" between two cities that are in the same country,
    2. travels at a given "international speed" between two cities that are "primary" capitals,
    3. otherwise cannot travel between two cities in different countries.
3. The Teleporting Tarte Trolley, which can blink (i.e. teleport) between any two cities within a given range in a fixed time.

Program execution

Starting Output

At the beginning of its execution, the program prints

```
Delivering baked goods to 363 cities.
Bakery vehicles:
 0. Crappy Crepe Car (crappy speed: 250 km/h)
 1. Diplomacy Donut Dinghy (domestic speed: 100 km/h | international speed: 500
km/h)
 2. Diplomacy Donut Dinghy (domestic speed: 50 km/h | international speed: 800
km/h)
 3. Teleporting Tarte Trolley (blink time: 6 h | blink range: 2000 km)
 4. Teleporting Tarte Trolley (blink time: 8 h | blink range: 3000 km)
```

which indicates that 363 cities meet the criteria, and that 5 vehicles are available.

Direct travel time computation

The program reads two city IDs, exactly as in the previous task.

The program then prints

```
Direct travel times from Tokyo to Jakarta:
 0. 24 hours
 1. 12 hours
 2. 8 hours
 3. Infinity
 4. Infinity
```

and terminates. The travel times are rounded up to the nearest integer number of hours. See examples.

> When computing travel times, use the distance as computed by the previous task, i.e. rounded up to the nearest whole kilometre.

Examples

User inputs are in **bold font** below.

Example 1

```
Delivering baked goods to 363 cities.
Bakery vehicles:
 0. Crappy Crepe Car (crappy speed: 250 km/h)
 1. Diplomacy Donut Dinghy (domestic speed: 100 km/h | international speed: 500
km/h)
 2. Diplomacy Donut Dinghy (domestic speed: 50 km/h | international speed: 800
km/h)
 3. Teleporting Tarte Trolley (blink time: 6 h | blink range: 2000 km)
 4. Teleporting Tarte Trolley (blink time: 8 h | blink range: 3000 km)
Enter the ID of the departure city:
1392685764
Enter the ID of the destination city:
1360771077
Direct travel times from Tokyo to Jakarta:
 0. 24 hours
 1. 12 hours
 2. 8 hours
 3. Infinity
 4. Infinity
```

Example 2

```
Delivering baked goods to 363 cities.
Bakery vehicles:
 0. Crappy Crepe Car (crappy speed: 250 km/h)
```

```
 1. Diplomacy Donut Dinghy (domestic speed: 100 km/h | international speed: 500
km/h)
 2. Diplomacy Donut Dinghy (domestic speed: 50 km/h | international speed: 800
km/h)
 3. Teleporting Tarte Trolley (blink time: 6 h | blink range: 2000 km)
 4. Teleporting Tarte Trolley (blink time: 8 h | blink range: 3000 km)
Enter the ID of the departure city:
1076532519
Enter the ID of the destination city:
1152554349
Direct travel times from Sao Paulo to Santiago:
 0. 11 hours
 1. Infinity
 2. Infinity
 3. Infinity
 4. 8 hours
```

Task 9: Bakery Vehicle Path Time (30%)

Task Description

Modify the program you wrote in order to find the travel time from a departure to an arrival city, if layovers (i.e. passing through intermediate cities) are allowed.

In this situation, the Diplomacy Donut Dinghy can transit through primary capital cities in order to make all international travel possible.

Similarly, the Teleporting Tarte Trolley can attempt to blink to nearby cities first if the destination city is not within direct reach.

Program execution

Starting Output

The starting output is the same as for the previous task.

Path travel time computation

The program reads two city IDs, exactly as in the previous task.

The program then prints

```
Travel times from Tokyo to Jakarta:
 0. 24 hours
 1. 12 hours
 2. 8 hours
 3. 24 hours
 4. 16 hours
```

and terminates. These travel times are the duration of a quickest path from the departure to the destination city, rounded up to the nearest integer number of hours. The paths are allowed to go through all 363 cities. See examples.

> When computing path times, use the travel times between two cities as computed by the previous task, i.e. rounded up to the nearest hour.
>
> Use the module `networkx` to compute quickest paths between two cities. Consider checking this background slide on quickest paths.
>
> Remember to use abstract methods and inheritance to avoid writing repeated code.

The time limit on ed is 30 seconds per run. Overall, running the tests should take at most 5 minutes.

Examples

User inputs are in **bold font** below.

Example 1

```
Delivering baked goods to 363 cities.
Bakery vehicles:
 0. Crappy Crepe Car (crappy speed: 250 km/h)
 1. Diplomacy Donut Dinghy (domestic speed: 100 km/h | international speed: 500
km/h)
 2. Diplomacy Donut Dinghy (domestic speed: 50 km/h | international speed: 800
km/h)
 3. Teleporting Tarte Trolley (blink time: 6 h | blink range: 2000 km)
 4. Teleporting Tarte Trolley (blink time: 8 h | blink range: 3000 km)
Enter the ID of the departure city:
1392685764
Enter the ID of the destination city:
1360771077
Travel times from Tokyo to Jakarta:
 0. 24 hours
 1. 12 hours
 2. 8 hours
 3. 24 hours
 4. 16 hours
```

Example 2

```
Delivering baked goods to 363 cities.
Bakery vehicles:
 0. Crappy Crepe Car (crappy speed: 250 km/h)
 1. Diplomacy Donut Dinghy (domestic speed: 100 km/h | international speed: 500
km/h)
 2. Diplomacy Donut Dinghy (domestic speed: 50 km/h | international speed: 800
km/h)
 3. Teleporting Tarte Trolley (blink time: 6 h | blink range: 2000 km)
 4. Teleporting Tarte Trolley (blink time: 8 h | blink range: 3000 km)
Enter the ID of the departure city:
1076532519
Enter the ID of the destination city:
1152554349
Travel times from Sao Paulo to Santiago:
 0. 11 hours
 1. 16 hours
 2. 22 hours
 3. 12 hours
 4. 8 hours
```