Week 8

Text processing
(Reading and writing files)

Week 8
Today's Tasks

Tasks for Today!

- Tasks
  - Merge a set of files into a new file
  - Handle coordinate information from a file
  - Handle temperature information from a file

Task 1 | Merge files

- Merge a set of input files into a single output file
  - All contents in the input files are put into the output file

Alan Mathison Turing was a British pioneering computer scientist, mathematician, logician, cryptanalyst, philosopher, mathematical biologist, and marathon and ultra distance runner.

Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Alan Mathison Turing was a British pioneering computer scientist, mathematician, logician, cryptanalyst, philosopher, mathematical biologist, and marathon and ultra distance runner. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Task 1 | Merge files

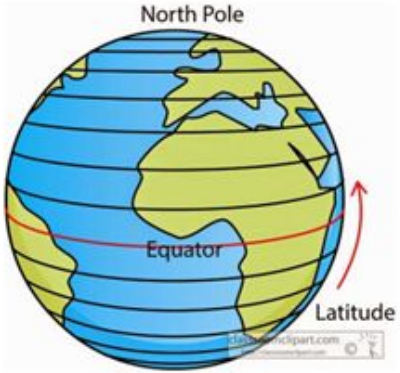
- Define a function ‘merge’ with two parameters
  - 1st parameter : names of input files (**list of string**)
  - 2nd parameters : a name of a string output file (**string**)
  - **No output** in this function

- Merge input files by calling the function ‘merge’
  - merge ([“kaist1.txt”, “kaist2.txt”, “kaist3.txt”], “result.txt”)
  - merge ([“kaist1.txt”, “kaist2.txt”], “result.txt”)

- Use ***elice_utils.send_file()*** function
  - Then, you can see a link to download the file

Task 2 | Handle coordinate information

1. Read and print data from the file '*average-latitude-longitude-countries.csv*'
   - Each line (except for the head) contains country code, country name, latitude and longitude, which are comma-separated
     - E.g. “KR”, “Korea, Republic of”, 37, 127.5
     - -90.0° ≤ latitude ≤ 90.0°, -180.0° ≤ longitude ≤ 180.0°
     - Caution: several country names contain comma.
   - Make two lists and print them
     - List of tuples ➜ (country code, country name)
     - List of tuples ➜ (country code, (latitude, longitude))
     - Data type
       - country code, country name ➜ ***string***
       - latitude, longitude ➜ ***float***



Task 2 | Handle coordinate information

2. Print the names of all country whose location lies in the south of the equator.

3. Let the user enter a country code, and then print the full name of the corresponding country.

Tip. You can **reuse the lists of tuples in task 2-1.**
Then, you don't need to read the file again.

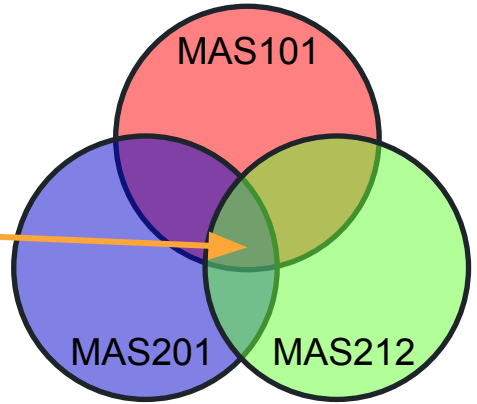
Task 3 | Course Enrollments

- The file (“class_enrollment.csv”) contains a list of student names for each course.

1. math_addicts()

Goal: Return a list of students who are taking all three math courses.

Hints: 1. Names of math course starts with MAS
2. Use Intersection of three student sets



Task 3 | Course Enrollments

2. only_statistics()

Goal: Return a list of students who are taking only cc511.

Hint: Use difference between sets

questions?

Optional Task | Handle temperature information

- The file (“tpmon.txt”) includes England’s monlty temperatures information for the years 1723～1970

1. Read the data and print average winter and summer temperatures for each year
   - Winter average = (January’s + February’s) / 2
   - Summer average = (July’s + August’s) / 2
   - Print every year and winter/summer averages in a **nicely formatted table**

```
...
1737:      5.2 / 15.6
1738:      4.6 / 16.2
1739:      5.4 / 15.4
1740:     -2.2 / 15.0
1741:      3.1 / 16.1
...
```

Optional Task| Handle temperature information

2. Write the contents in another format
   - To the file ‘tpmon.csv’
   - Each year and its 12 monthly temperatures in one line
   - Write in CSV format (i.e. comma-separate the data)
   - Try to open the file in Excel

```
1723,1.1,4.4,7.5,8.9,11.7,15.0,15.3,15.6,13.3,11.1,7.5,5.8
1724,5.6,4.2,4.7,7.2,11.4,15.3,15.0,16.2,14.4,8.6,5.3,3.3
1725,4.4,3.3,5.0,8.1,10.8,12.2,13.8,13.3,12.8,9.4,6.9,3.9
1726,1.1,4.2,4.2,8.4,13.4,16.4,16.0,15.6,14.7,10.2,6.1,1.8
1727,4.2,5.0,5.1,9.2,13.6,14.9,16.9,16.9,14.4,10.8,4.7,3.6
1728,3.9,2.4,7.1,8.3,12.5,16.4,16.9,16.0,12.8,9.1,7.2,1.6
1729,1.2,2.3,2.8,7.1,10.3,15.1,16.8,15.7,16.6,10.1,8.1,5.0
1730,4.1,4.7,6.2,8.7,12.4,14.0,15.3,16.3,15.3,10.9,9.2,3.4
…
```