Prime Numbers File Writer (10 pts)

Write a Python program that outputs all prime numbers less than 10,000 into a file named `primes.txt` .

- The program should:
  - i. **Reuse the `is_prime(n)` function** you wrote previously.
  - ii. Generate all prime numbers from 2 up to 9,999.
  - iii. Write the primes to `primes.txt` , one number per line.

Example Output File (`primes.txt`):

```
2
3
5
7
11
13
17
19
...
```

Student Contact Lookup (20 pts)

Write a Python program that reads a file containing student contact information and allows the user to look up a student's email by their full name.

- The file will contain one student per line in the format:

  ```
  first name,last name,email
  ```

  Example file content (`students.txt`):

  ```
  Alice,Smith,alice.smith@email.com
  Bob,Johnson,bob.johnson@email.com
  Carol,Wu,carol.wu@email.com
  ```

- Your program should:

    i. Read the file and store the data in a dictionary.

        o Both `first name` and `last name` should consist of alphabetical characters only.

    ii. Ask the user to enter a full name (first and last name separated by a space).

    iii. Print the corresponding email address if the student exists, or an error message if the name is not found.

- The file is named `students.txt` and is in the same folder as your program.

Example Run:

```
Enter full name: Alice Smith
Email: alice.smith@email.com



Enter full name: David Lee
Error: Student not found.
```

Hint:

1. Use `.readlines()` to read all lines from the file.
2. Use `.split(",")` to break each line into `first name`, `last name`, and `email`.

```
first, last, email = line.strip().split(",")
```

3. Similarly, use `.split()` on the user's input to separate the first and last name.

```
name_input = input("Enter full name: ")
first, last = name_input.split(" ")
```

Got it — that makes it more realistic for a sports contest! Here's the refined assignment with **judge names included**:

Contestant Score Aggregator with Judges (30 pts)

Write a Python program that reads a file containing the scores from multiple judges for a single contestant. The program should eliminate the highest and lowest score, then compute the average of the remaining scores. Finally, it should write the results to an output file, including the names of the judges who gave the highest and lowest scores.

- The input file will contain one judge's name and score per line, in the format:

```
JudgeName,Score
```

Example file (`scores.txt`):

```
Alice,8.5
Bob,9.0
Carol,9.3
David,7.8
Emma,8.9
Frank,9.5
```

- Your program should:
    - i. Read all judge–score pairs from the `scores.txt` .
        - ○ `JudgeName` doesn't contain any commas `,` .
    - ii. Compute the average scores in the way mentioned above.
    - iii. Write the results to a new file named `result.txt` , whose format can be found below in **Example Run**.

Example Run (output file `result.txt`):

```
Highest Score: 9.5 (by Frank)
Lowest Score: 7.8 (by David)
Average Score: 8.925
```