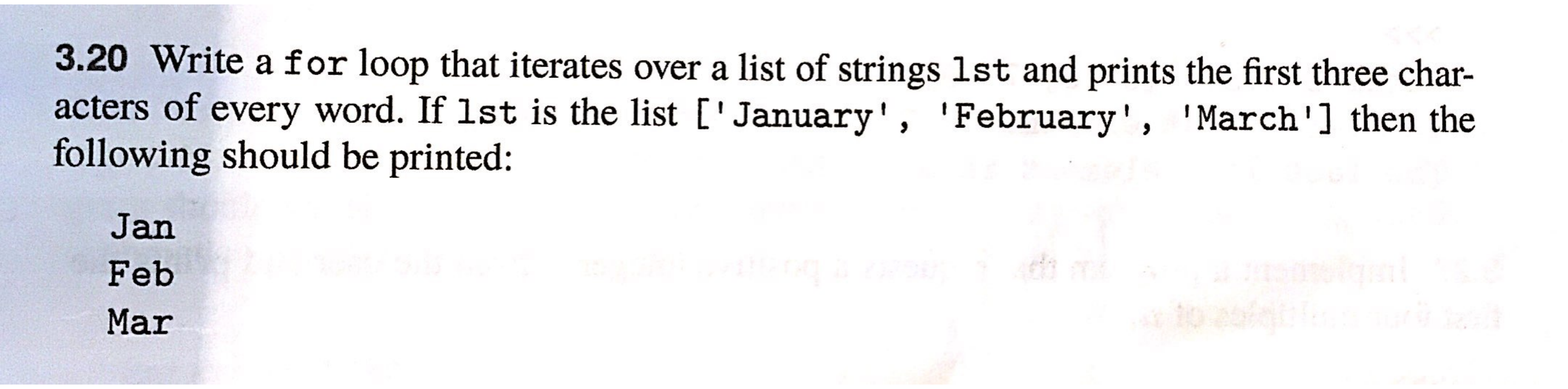

**3.20** Write a `for` loop that iterates over a list of strings `lst` and prints the first three characters of every word. If `lst` is the list `['January', 'February', 'March']` then the following should be printed:

```
Jan
Feb
Mar
```

**3.25** Implement a program that requests a list of student names from the user and prints those names that start with letters A through M.

```
>>>
Enter list: ['Ellie', 'Steve', 'Sam', 'Owen', 'Gavin']
Ellie
Gavin
```

**3.41** Write function `lastF()` that takes as input two strings of the form `'FirstName'` and `'LastName'`, respectively, and returns a string of the form `'LastName, F.'`. (Only the initial should be output for the first name.)

```
>>> lastF('Albert', 'Camus')
'Camus, A.'
```

**3.42** Implement function `avg()` that takes as input a list that contains lists of numbers. Each number list represents the grades a particular student received for a course. For example, here is an input list for a class of four students:

```
[[95, 92, 86, 87], [66, 54], [89, 72, 100], [33, 0, 0]]
```

The function avg should *print*, one per line, every student's average grade. You may assume that every list of grades is nonempty, but you may *not* assume that every student has the same number of grades.

```
>>> avg([[95, 92, 86, 87], [66, 54], [89, 72, 100], [33, 0, 0]])
90.0
60.0
87.0
11.0
```

**3.43** The computer game function `hit()` takes five numbers as input: the $x$ and $y$ coordinates of the center and the radius of a circle $C$, and the $x$ and $y$ coordinates of a point $P$. The function should return `True` if point $P$ is inside or on circle $C$ and `False` otherwise.

```
>>> hit(0, 0, 3, 3, 0)
True
>>> hit(0, 0, 3, 4, 0)
False
```

**3.44** Write a function `distance()` that takes as input a number: the time elapsed (in seconds) between the flash and the sound of thunder. Your function should return the distance to the lightning strike in kilometers. The speed of sound is approximately 340.29 meters per second; there are 1000 meters in one kilometer.

```
>>> distance(3)
1.0208700000000002
```