Chapter 3  Problems  91

3.33  The probability of getting $n$ heads in a row when tossing a fair coin $n$ times is $2^n$. Implement function `prob()` that takes a nonnegative integer $n$ as input and returns the probability of $n$ heads in a row when tossing a fair coin $n$ times.

```python
>>> prob(1)
0.5
>>> prob(2)
0.25
```

3.34  Implement function `reverse_int()` that takes a three-digit integer as input and returns the integer obtained by reversing its digits. For example, if the input is 123, your function should return 321. You are not allowed to use the string data type operations to do this task. Your program should simply read the input as an integer and process it as an integer using operators such as `/` and `%`. You may assume that the input integer does not end with the 0 digit.

```python
>>> reverse_int(123)
321
>>> reverse_int(908)
809
```

3.35  Implement function `points()` that takes as input four numbers $x_1$, $y_1$, $x_2$, $y_2$ that are the coordinates of two points $(x_1,y_1)$ and $(x_2,y_2)$ in the plane. Your function should compute:

- The slope of the line going through the points, unless the line is vertical
- The distance between the two points

Your function should print the computed slope and distance in the following format. If the line is vertical, the value of the slope should be string `'infinity'`. *Note:* Make sure you convert the slope and distance values to a string before printing them.

```python
>>> points(0, 0, 1, 1)
The slope is 1.0 and the distance is 1.41421356237
>>> points(0, 0, 0, 1)
The slope is infinity and the distance is 1.0
```

3.36  Implement function `abbreviation()` that takes a day of the week as input and returns its two-letter abbreviation.

```python
>>> abbreviation('Tuesday')
'Tu'
```

3.37  The computer game function `collision()` checks whether two circular objects collide; it returns `True` if they do and `False` otherwise. Each circular object will be given by its radius and the $(x,y)$ coordinates of its center. Thus the function will take six numbers as input: the coordinates $x_1$ and $y_1$ of the center and the radius $r_1$ of the first circle, and the coordinates $x_2$ and $y_2$ of the center and the radius $r_2$ of the second circle.

```python
>>> collision(0, 0, 3, 0, 5, 3)
True
>>> collision(0, 0, 1.4, 2, 2, 1.4)
False
```