**2.6 (*Sum the digits in an integer*) Write a program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14. (Hint: Use the % operator to extract digits, and use the // operator to remove the extracted digit. For instance, 932 % 10 = 2 and 932 // 10 = 93.) Here is a sample run:

```
Enter a number between 0 and 1000: 999 Enter
The sum of the digits is 27
```

**2.7 (*Find the number of years and days*) Write a program that prompts the user to enter the minutes (e.g., 1 billion), and displays the number of years and days for the minutes. For simplicity, assume a year has 365 days. Here is a sample run:

```
Enter the number of minutes: 1000000000 Enter
1000000000 minutes is approximately 1902 years and 214 days
```

*2.13 (*Split digits*) Write a program that prompts the user to enter a four-digit integer and displays the number in reverse order. Here is a sample run:

```
Enter an integer: 3125 Enter
5
2
1
3
```

***2.14** (*Geometry: area of a triangle*) Write a program that prompts the user to enter the three points `(x1, y1)`, `(x2, y2)`, and `(x3, y3)` of a triangle and displays its area. The formula for computing the area of a triangle is

$$s = (side1 + side2 + side3) / 2$$

$$area = \sqrt{s(s - side1)(s - side2)(s - side3)}$$

Here is a sample run:

```
Enter three points for a triangle: 1.5, -3.4, 4.6, 5,
 9.5, -3.4 Enter
The area of the triangle is 33.6
```

****2.21** (*Financial application: compound value*) Suppose you save `$100` each month into a savings account with an annual interest rate of 5%. Therefore, the monthly interest rate is 0.05/12 = 0.00417. After the first month, the value in the account becomes

```
100 * (1 + 0.00417) = 100.417
```

After the second month, the value in the account becomes

```
(100 + 100.417) * (1 + 0.00417) = 201.252
```

After the third month, the value in the account becomes

```
(100 + 201.252) * (1 + 0.00417) = 302.507
```

and so on.

Write a program that prompts the user to enter a monthly saving amount and displays the account value after the sixth month. Here is a sample run of the program:

2.22 (*Population projection*) Rewrite Exercise 1.11 to prompt the user to enter the number of years and displays the population after that many years. Here is a sample run of the program:

```
Enter the number of years: 5 ↵Enter
The population in 5 years is 325932970
```

Reference

***1.11** (*Population projection*) The US Census Bureau projects population based on the following assumptions:

One birth every 7 seconds
One death every 13 seconds
One new immigrant every 45 seconds

Write a program to display the population for each of the next five years. Assume the current population is 312032486 and one year has 365 days. Hint: in Python, you can use integer division operator `//` to perform division. The result is an integer. For example, `5 // 4` is `1` (not `1.25`) and `10 // 4` is `2` (not `2.5`).