Still More on Python Programming

J.K. Denny

Department of Mathematics, Mercer University

Working with Lists

Lists are a very useful data type in Python. Here are some key commands when using lists. Assume L is a list.

- L.append(item) – adds ‘item‘ to the end of the list
- L.reverse() – reverses the order of the items in L
- L.sort() – sorts the list alphabetically and numerically
- L.pop(i) – removes the item in the i^{th} position of L
- L[i] – accesses the i^{th} item of L (remember that indexing starts with i=0)
- map(function,L) – applies ‘function‘ to all items in L
- list(object) – converts ‘object‘ to a list

List examples

Try this:

- Create list with 6 elements.
- Append your age, name, and hometown to the list.
- Reverse the order of the list and print it on the screen.
- Sort the list and print it on the screen.
- Pop off the 4th item.
- Print the 2nd item.
- Create a list of 4 numbers.
- Create a function that squares a number, map it to your new list, and convert the result to a list.

Base 10 to Binary

Now, we want to write a function to convert from base 10 to base 2.

Assume $d_0, d_1, \ldots, d_n \in \{0, 1\}$ are digits.

$$\begin{aligned} m &= d_0 + d_1 \cdot 2 + d_2 \cdot 2^2 + \cdots d_n \cdot 2^n \\ &= d_0 + 2 \cdot (d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1}) \end{aligned}$$

What happens when we divide by 2?

The remainder is d_0 and dividend is $d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1}$.

Base 10 to Binary

Now, we want to write a function to convert from base 10 to base 2.

Assume $d_0, d_1, \ldots, d_n \in \{0, 1\}$ are digits.

$$\begin{aligned} m &= d_0 + d_1 \cdot 2 + d_2 \cdot 2^2 + \cdots d_n \cdot 2^n \\ &= d_0 + 2 \cdot (d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1}) \end{aligned}$$

What happens when we divide by 2?

The remainder is d_0 and dividend is $d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1}$.

Base 10 to Binary

Now, we want to write a function to convert from base 10 to base 2.

Assume $d_0, d_1, \ldots, d_n \in \{0, 1\}$ are digits.

$$\begin{aligned} m &= d_0 + d_1 \cdot 2 + d_2 \cdot 2^2 + \cdots d_n \cdot 2^n \\ &= d_0 + 2 \cdot (d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1}) \end{aligned}$$

What happens when we divide by 2?

The remainder is d_0 and dividend is $d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1}$.

Base 10 to Binary

Now, consider the dividend.

$$d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1} = d_1 + 2 \cdot (d_2 + d_3 \cdot 2 + \cdots d_n \cdot 2^{n-2})$$

What happens when we divide by 2?

The remainder is d_1 and dividend is $d_2 + d_3 \cdot 2 + \cdots d_n \cdot 2^{n-2}$.

Base 10 to Binary

Now, consider the dividend.

$$d_1 + d_2 \cdot 2 + \cdots d_n \cdot 2^{n-1} = d_1 + 2 \cdot (d_2 + d_3 \cdot 2 + \cdots d_n \cdot 2^{n-2})$$

What happens when we divide by 2?

The remainder is d_1 and dividend is $d_2 + d_3 \cdot 2 + \cdots d_n \cdot 2^{n-2}$.

Base 10 to Binary

We continue to divide by 2 and take the remainder. In the end, we have generated the binary digits $d_0, d_1, \ldots, d_n$ for m. Note that we need to reverse the order of the digits to get the usual binary representation, $d_n d_{n-1} \ldots d_2 d_1 d_0$.

Example: Use repeated division by 2 to find the binary representation of 35.

Base 10 to Binary

We continue to divide by 2 and take the remainder. In the end, we have generated the binary digits $d_0, d_1, \ldots, d_n$ for m. Note that we need to reverse the order of the digits to get the usual binary representation, $d_n d_{n-1} \ldots d_2 d_1 d_0$.

Example: Use repeated division by 2 to find the binary representation of 35.

Integer division in Python

How do you do this in Python?

- Dividend – "div" – `m // n` gives the dividend when m is divided by n
- Remainder – "mod" – `m % n` gives the remainder when m is divided by n

Examples . . .

Integer division in Python

How do you do this in Python?

- Dividend – “div” – `m // n` gives the dividend when m is divided by n
- Remainder – “mod” – `m % n` gives the remainder when m is divided by n

Examples . . .

Base 10 to Binary

Try it: Write a function to convert an integer from base 10 to binary. The function should take an integer as its input and return a list of the binary digits in the order $d_n, d_{n-1}, \ldots, d_2, d_1, d_0$.

One more command you need . . .

The `while` command is another way to run a loop. For example:

```
x = 100
while (x>5):
    x = x - 9
    print x
```