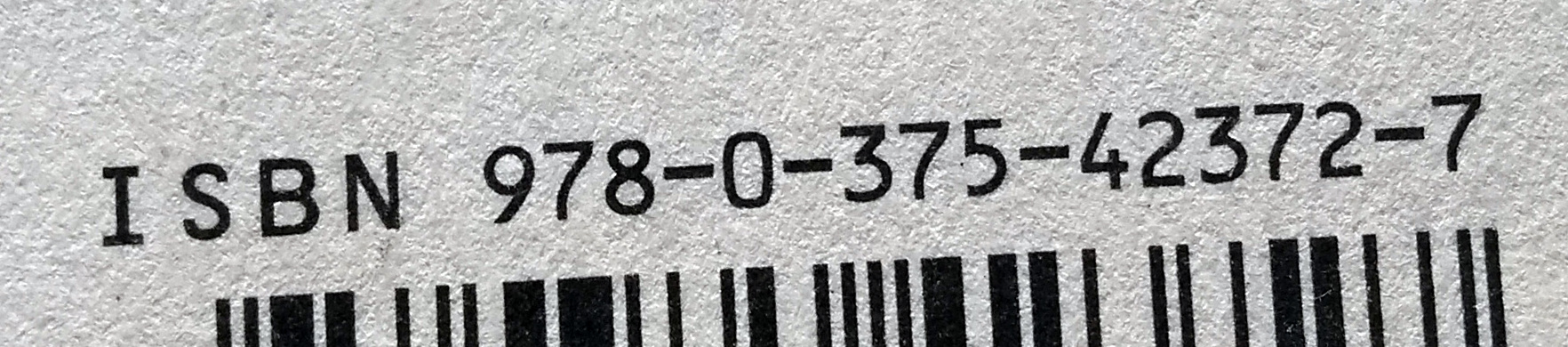
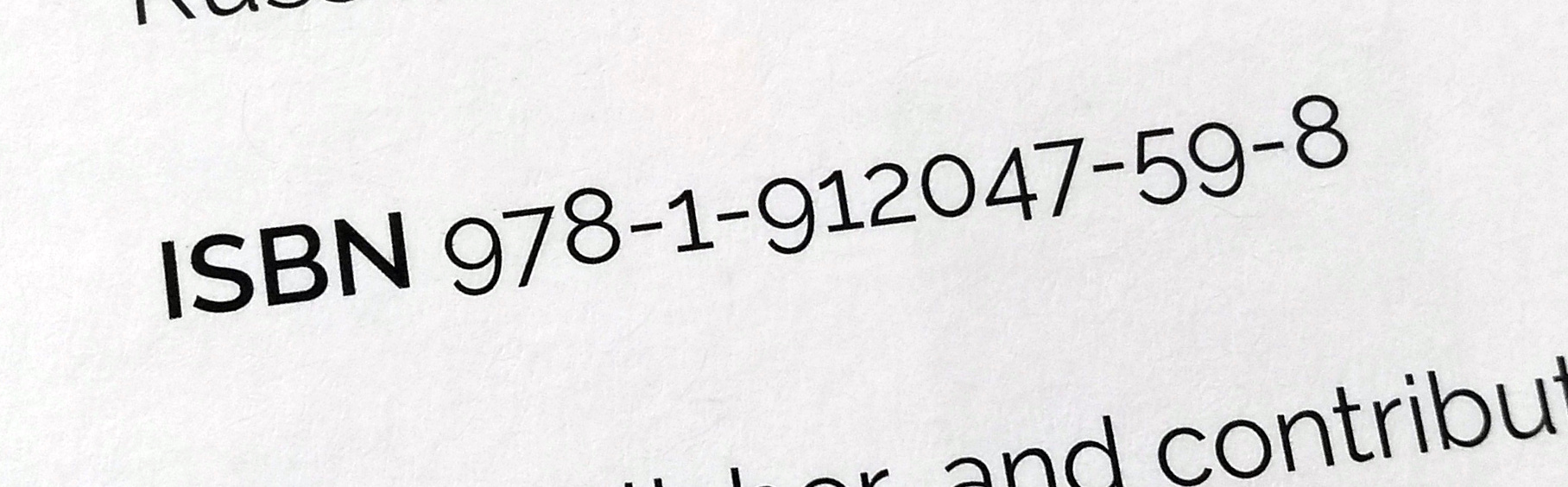
**ISBN**

Every book published after 2007 is assigned a unique 13-digit code called ISBN (International Standard Book Number). Here are a few examples:







In order for an ISBN to be **valid**, the **weighted sum** of the 13 digits (where each digit is multiplied by a weight alternating **between 1 and 3**) must be a **multiple of 10**.

**Example**:

*The Information: A History, A Theory, A Flood*

by James Gleick (2011)

ISBN: 9780375423727

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| digits | 9 |  | 7 |  | 8 |  | 0 |  | 3 |  | 7 |  | 5 |  | 4 |  | 2 |  | 3 |  | 7 |  | 2 |  | 7 |  |  |
|  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  |  |
| weights | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  |  |
|  | 9 | + | 21 | + | 8 | + | 0 | + | 3 | + | 21 | + | 5 | + | 12 | + | 2 | + | 9 | + | 7 | + | 6 | + | 7 | = | **110** |

**Example**:

*Adventures in Raspberry Pi*

by Carrie Anne Philbin (2013)

ISBN: 9781118751251

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| digits | 9 |  | 7 |  | 8 |  | 1 |  | 1 |  | 1 |  | 8 |  | 7 |  | 5 |  | 1 |  | 2 |  | 5 |  | 1 |  |  |
|  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  | ⨉ |  |  |
| weights | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  | 3 |  | 1 |  |  |
|  | 9 | + | 21 | + | 8 | + | 3 | + | 1 | + | 3 | + | 8 | + | 21 | + | 5 | + | 3 | + | 2 | + | 15 | + | 1 | = | **100** |

**If you get stuck with your Python try looking at the last page for some clues.**

Task 1. Compute the weighted sum

Create a Python program that prompts the user to enter an ISBN (as a single piece of text) and then computes and displays the weighted sum of its digits.

|  |  |
| --- | --- |
| **Example** | |
| Note: Use this example to check your program. This is the output your program should produce for the given input. | |
| The program displays a prompt and waits for keyboard input | Enter an ISBN: |
| The user types a reply | 9780375423727 |
| The program displays the weighted sum | Weighted sum: 110 |

**Note:** For the purposes of this task, you can **assume** that the user input will be 13 characters long and will only contain numerical digits. If other characters are included, your program will terminate with a ValueError but you don’t need to worry about that.

**Checklist**: Tick (✔) the corresponding box if your program:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Prompts the user for an ISBN and reads it as a piece of text (a string). |
| |  | | --- | |  | | Computes and displays the weighted sum of its digits. |

Task 2 . Check validity

**Extend** your program to check the weighted sum and determine if the ISBN is valid.

|  |  |
| --- | --- |
| **Example** | |
| Note: Use this example to check your program. This is the output your program should produce for the given input. | |
| The program displays a prompt and waits for keyboard input | Enter an ISBN: |
| The user types a reply | 9780375423727 |
| The program displays the weighted sum and whether or not the ISBN is valid | Weighted sum: 110  This is a valid ISBN |

|  |  |
| --- | --- |
| **Example** | |
| Note: Use this example to check your program. This is the output your program should produce for the given input. | |
| The program displays a prompt and waits for keyboard input | Enter an ISBN: |
| The user types a reply | 9780375423728 |
| The program displays the weighted sum and whether or not the ISBN is valid | Weighted sum: 111  This is not a valid ISBN |

**Checklist**: Tick (✔) the corresponding box if your program:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Checks if the weighted sum is a multiple of 10 and displays a message that informs the user whether or not the ISBN is valid. |

**Take a screenshot of your code that does both tasks and upload to Bourne To Learn to the earn your platinum badge.**

Clues . Look here if you need help

**What are the variables I will need?**

Think about the quantities you will need to refer to in your program, the values that your program will need to keep track of.

You will probably need:

* The isbn entered by the user (as a piece of text)
* Each individual character in the ISBN
* Each individual digit in the ISBN
* The weight associated with each digit
* The weighted sum

**How do I iterate over the digits in the ISBN?**

Read the isbn as a piece of text, i.e. don’t convert it to a number directly, so that it is easier to iterate over individual characters.

Use a for-loop to iterate over each character in the ISBN. Use the int function to convert each character into a digit. The pseudocode below illustrates this idea:

|  |  |
| --- | --- |
|  | for character in isbn:  digit = int(character)  process the digit |

**How do I compute the weighted sum of the ISBN digits?**

A sum can be accumulated by adding each term iteratively to the sum, using an assignment like this:

|  |  |
| --- | --- |
|  | sum = sum + *term* |

In this case, the *term* will be the product of a digit by its associated weight. Don’t forget to initialise the sum.

**How do I alternate the weight between 1 and 3?**

You can check if it is equal to 3 and then set it to one and vice versa.

There’s also another way: remember that the sum of two consecutive weights is always 4. So if you know the current value of the weight, it’s straightforward to compute the next value as well.

**How do I check if the weighted sum is a multiple of 10?**

If you divide a number by 10 and there is no remainder (i.e. the remainder equals zero), then this number is a multiple of 10.

Use the % operator to compute the remainder of any integer division. For example, this expression computes the remainder of dividing sum by 10:

|  |  |
| --- | --- |
|  | sum % 10 |

This resource is available online at [ncce.io/prg5-5-a3-wi](http://ncce.io/prg5-5-a3-wi). Resources are updated regularly — please check that you are using the latest version.

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](http://ncce.io/ogl).