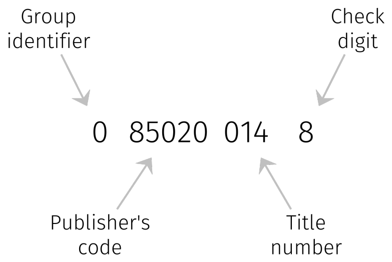
ISBN Numbers

Many codes have been designed for use with new technology. These include bar codes, ISBN numbers, ASCII codes, post codes, bank account codes. Many of these modern codes rely on a checking system, often referred to as a check digit.

An example of this is the ISBN (International Standard Book Number) numbers, used universally on all books published since 1972.

When ordering a book you usually give the author and the title. Sometimes it's more convenient to give its ISBN number, to make sure you get the exact edition you are looking for.

An ISBN is preceded by letters ISBN and has ten digits made up from components, as illutrated opposite. The digits are arranged according to the number of digits in each component part, with a space or a hyphen between each part. The check digit is designed so that any one error in the previous nine digits is spotted.



It is calculated so that any *single* error in the previous nine digits is spotted, in the following way :

Multiply the first nine numbers by 10, 9, 8, … , 2 respectively and find the sum of the resulting numbers. The check digit is the smallest number that needs to be added to this total so that it is exactly divisible by 11.

For the example above, we have $0\times10+8\times9+5\times8+0\times7+2\times6+0\times5+0\times4+1\times3+4\times2=135$ so the check digit must be 8, since 143 is divisible by 11. Note that if the number 10 is needed for the check digit, the symbol X is used.

1. Determine the check digit a for the following ISBN number 186993100 a. Explain your answer.

2. There is exactly one error in the following ISBN number : 1869932238. Can you correct this error ? Explain your answer.

3. Apply the following Python program to the ISBN number : 1834721164 .

Python

```python
# create a list with nine 0 in it
d = [0] * 9

# repeat for i in 0, ..., 8
for i in range(9):

    # ask the user for a digit and put it in the list
    d[i] = int(input("Enter a digit : "))

# calculate the total
total = 10 * d[0] + 9 * d[1] + 8 * d[2] + 7 * d[3] + 6 * d[4]
        + 5 * d[5] + 4 * d[6] + 3 * d[7] + 2 * d[8]

check_digit = 0

# while total is not divisible by 11
while total % 11 != 0:
    check_digit = check_digit + 1

print(check_digit)
```

4. Change this program so that it will detect a mistake in a given ISBN number : it should display a message which says whether there is an error in the check digit (and therefore in the ISBN number) or not.

5. Why can this method detect only single errors ? Give an example of two valid ISBN numbers, the latter derived from the former by changing two digits.

110	121	132	143	154	165	176	187
198	209	220	231	242	253	264	275
286	297	308	319	330	341	352	363
374	385	396	407	418	429	440	451
462	473	484	495	506	517	528	539
550	561	572	583	594	605	616	627
638	649	660	671	682	693	704	715

A fragment of the 11 times table.