



A7.1 Number sets, bases and units

Data representation · A level · OCR H446 1.4.1, AQA 7517 4.5.1.1, Eduqas
A500QS 2.3 · about 25 min

Name _____

Class _____

Date _____

What this lesson is about

Natural, integer, rational, irrational, real and ordinal numbers; any number base; bits, bytes, kilo and kibi.

- Number sets
- Counting and measurement
- Number bases
- Bits, bytes and units

Questions

6 marks in all

- Which set of numbers is the square root of 2 in, but not in the rational numbers? [1 mark]
☐ A Irrational numbers
☐ B Natural numbers
☐ C Integers
☐ D Ordinal numbers
- Which of these are rational numbers? [1 mark]
Tick every answer that is true.
☐ A 0.75
☐ B -3
☐ C $22/7$
☐ D The square root of 2
☐ E Pi
- What are ordinal numbers used for? [1 mark]
☐ A To describe a position in an ordered list, such as 1st, 2nd, 3rd
☐ B To count how many things there are
☐ C To measure a quantity such as a length
☐ D To store negative values
- Write the denary number 255 in base 8. [1 mark]

- How many bytes are in 3 KiB? [1 mark]

6. What does this program print?

[1 mark]

```
n = 47
digits = ""
while n > 0:
    digits = str(n % 5) + digits
    n = n // 5
print(digits)
```

The task: any base

Write `to_base(n, base)`. The parameter `n` is a whole number, 0 or more; `base` is a whole number from 2 to 16. It returns the digits of `n` in that base as a string, using 0 to 9 then A to F, and returns "0" when `n` is 0. Use repeated division with `//` and `%`: you may not use `bin`, `hex`, `oct` or `format`. The loop at the bottom prints lines such as 173 in base 16 = AD.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

DIGITS = "0123456789ABCDEF"

def to_base(n, base):
    # repeated division goes here
    return ""

for n, base in [(173, 2), (173, 8), (173, 16), (0, 2), (2024, 16)]:
    print(n, "in base", base, "=", to_base(n, base))
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/a7-1-numbers-bases-and-units/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Extend `to_base` to refuse a base outside 2 to 16 by raising a `ValueError`.
2. Write `from_base` without `DIGITS.index`, using character codes instead.

3. Which set does each belong to: the number of robots on the mat, the robot's heading in degrees, $22/7$, the square root of 16?