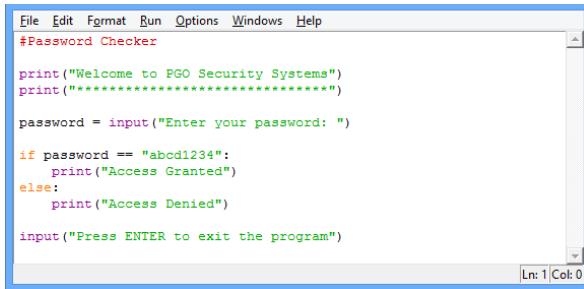


Year 9 Python Knowledge Organiser

Programming with Python



```
File Edit Format Run Options Windows Help
#Password Checker

print("Welcome to PGO Security Systems")
print("*****")

password = input("Enter your password: ")

if password == "abcd1234":
    print("Access Granted")
else:
    print("Access Denied")

input("Press ENTER to exit the program")
```

Python's Development Environment

Called IDLE – Integrated Development Environment

Two Modes:

Interactive Mode lets you see your results as you type them.

Script Mode lets you save your program and run it again later.

Writing error-free code

When writing **programs**, **code** should be as legible and error free as possible. **Debugging** helps keep **code** free of **errors** and documenting helps keep **code** clear enough to read.

Syntax errors

Syntax is the spelling and grammar of a **programming language**. In **programming**, a **syntax error** occurs when:

- there is a **spelling mistake**.
- there is a **grammatical mistake**.

Data Types

String - holds alphanumeric data as text

Integer - holds whole numbers

Float - holds numbers with a decimal point

Boolean - holds either 'True' or 'False'

Defining Variable Data Types

Python automatically assigns a data type to a variable. You can override this to manually define or change the data type using:

str() , **int()** or **float()**

Selection

When designing **programs**, there are often points where a **decision** must be made. This **decision** is known as **selection** and is implemented in **programming** using **IF statements**.

Operator	Meaning	Example	Evaluates to
==	equal to	7==7	True
!=	not equal to	6!=7	True
>	Greater than	7>6	True
<	Less than	5<8	True
>=	Greater than or equal to	6>=8	False
<=	Less than or equal to	7<=7	True

Procedures

A **procedure** is a small section of a **program** that performs a specific task. **Procedures** can be used repeatedly throughout a **program**. **Procedures** can make **code** shorter, simpler, and easier to write. Writing a **procedure** is extremely simple. Every **procedure** needs:

1. A *name*
2. The *program* code to perform the task

Variables

A **variable** is a location in **memory** in which you can temporarily store text or numbers. It is used like an empty box or the Memory function on a calculator. You can choose a name for the box (the "**variable name**") and change its contents in your **program**.

Using a Variable (firstname)

```
print ("What is your name?")
firstname = input()
print ("Hello,",firstname)
```



Functions

Functions are special keywords that do a specific job. **Functions** appear in purple.

print() and **input()** are examples of functions

```
print ("What is your name?")
firstname = input()
print ("Hello,",firstname)
```

Adding Comments

Comments are useful to help understand your **code**. They will not affect the way a **program** runs. **Comments** appear in red and have a preceding **#** symbol.

```
#firstname is a variable
print ("What is your name?")
firstname = input()
print ("Hello,",firstname)
```

Year 9 Python Knowledge Organiser

Iteration

Algorithms consist of steps that are carried out (performed) one after another. Sometimes an **algorithm** needs to **repeat** certain steps until told to stop or until a particular condition has been met. **Iteration is the process of repeating steps.**

Iteration allows us to **simplify** our **algorithm** by stating that we will **repeat** certain **steps** until told otherwise. **Iteration** is implemented in **programming** using **FOR** and **WHILE** statements.

There are **two** ways in which **programs** can **iterate** or '**loop**':

- count-controlled loops
 - o Sometimes it is necessary for **steps** to **iterate** a **specific number** of times.
- condition-controlled loops
 - o **iteration** continues **while**, or **until**, a **condition** is met.

Each type of **loop** works in a slightly different way and produces different results.

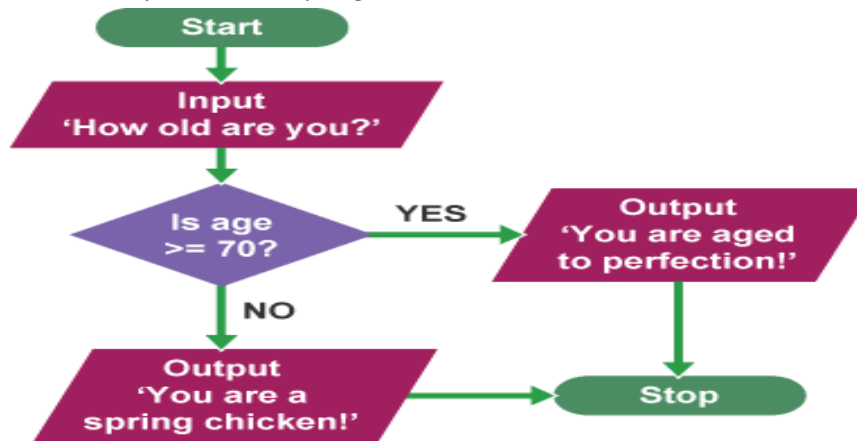
IF Statements

When designing **programs**, there are often points where a **decision** must be made. This **decision** is known as **selection** and is implemented in **programming** using **IF** statements. In **programming**, **selection** is usually represented by the statements **IF** and **ELSE**.

For **selection**, Python uses the statements **if** and **else** (note the lowercase **syntax** that **Python** uses):

Consider the age-related **algorithm** using **Python**. The steps are:

- Ask how old you are
- if you are 70 or older, say "You are aged to perfection!"
- else say "You are a spring chicken!"



The above algorithm would be written in Python (3.x) as:

```

age = int(input("How old are you?"))

if age >= 70:
    print("You are aged to perfection!")
else:
    print("You are a spring chicken!")
  
```

Arrays

An **array** is a series of **memory** locations – or '**boxes**' – each of which holds a single item of **data**, but with each box sharing the same name. All **data** in an **array** must be of the same **data type**.

Arrays are named like **variables**. The number in brackets determines how many **data** items the **array** can hold. The array **score(9)** would allow ten data items to be stored.

scores									
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
3000	2500								

Any **facility** that holds more than one item of **data** is known as a **data structure**. Therefore, an **array** is a **data structure**.

Lists are **data structures** similar to **arrays** that allow **data** of more than one **data type**.

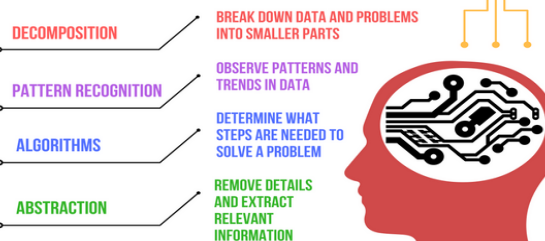
Functions

A **function** is also a small section of a **program** that performs a specific task that can be used repeatedly throughout a **program**, but the task is usually a **calculation**. **Functions** perform the task and return a value to the main **program**.

Every **function** needs:

1. A name
2. The values that it needs to use for calculation
3. The **program** code to perform the task
4. A value to return to the main program

COMPUTATIONAL THINKING



Year 9 Understanding Computers Knowledge Organiser

Binary (Base 2)

The only thing that computers understand is Binary.

8	4	2	1	1	=	ON
0	1	0	1	0	=	OFF

0101 = 5

01011111 = 95

128	64	32	16	8	4	2	1	Odd numbers
0	1	0	1	1	1	1	1	

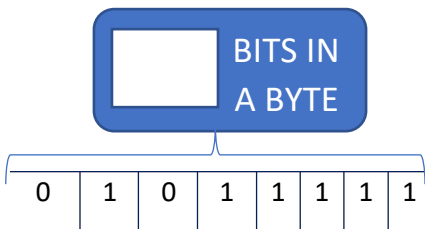
Convert these binary numbers into denary:

1) 1010		6) 1011	
2) 1010		7) 0001	
3) 0110		8) 1011	
4) 0111		9) 1001	
5) 0100		10) 0011	

Convert these denary numbers into binary (4 bits):

11) 14		16) 6	
12) 2		17) 11	
13) 10		18) 15	
14) 4		19) 2	
15) 3		20) 12	

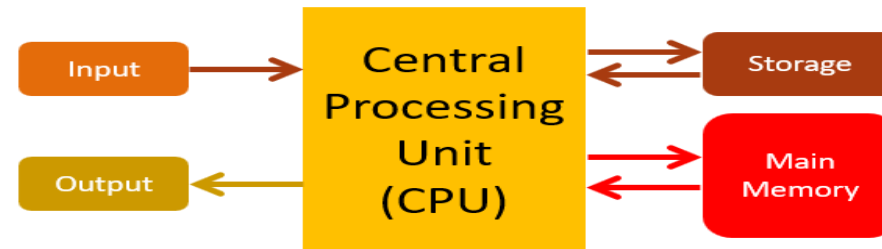
The **ones** and **zeros** in **binary** represent 'bits'. Each '1' or '0' is one 'bit'.



Computer System

A basic, **complete**, and **functional** computer. It will include all the **hardware** and **software** required to make it functional.

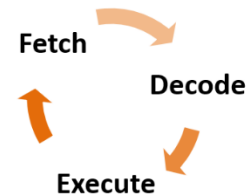
Components of a Computer



Fetch – Decode – Execute Cycle

Computer has a list of instructions in memory to carry out.

- CPU **Fetches** top instruction from the list
- Instructions is passed to **Decoder** to **interpret**
- **Decoder** passes on the instruction
- Instruction is **Executed** or carried out
- CPU **Fetches** top instruction from the list...



Processor Speed

The most common measure of **CPU** speed is the **clock speed**, which is measured in **MHz** or **GHz**. The higher the **clock speed**, the more operations the **CPU** can **execute per second**.

- One cycle per second = **1 Hertz (Hz)** = 1 instruction carried out each second
- **1 Kilohertz (KHz)** = **1,024** cycles per second
- **1 Megahertz (MHz)** = **1,048,576** cycles per second
- **1 Gigahertz (GHz)** = **1,073,741,824** cycles per second (Approximately 1 Billion!)

How fast is your computer's processor?

RAM vs ROM

RAM is alternatively referred to as main memory, **RAM** is **volatile** and allows information to be stored and retrieved on a computer. When opened programs are stored in **RAM**.

ROM is a type of **non-volatile** memory. **ROM** contains **BIOS**, which allows the computer system to start-up.

ASCII

The **ASCII** character set is a **7-bit** set of codes that allows 128 different characters. That is enough for every upper-case letter, lower-case letter, digit, and punctuation mark on most keyboards. **ASCII** is only used for the English language.

Extended ASCII

Extended ASCII code is an **8-bit** character set that represents **256** different characters, making it possible to use characters such as é or ©. Extended ASCII is useful for European languages.

Decimal	Binary	Character
96	01000000	,
97	01000001	a
98	01000010	b

Year 9 Data representation Knowledge

Bitmap graphics

Bitmap graphics made with painting packages consist of many tiny dots called pixels. It is possible to edit each individual pixel.

Since the computer has to store information about every single **pixel** (the colour for example) in the image, the file size of a **bitmap** graphic is often quite large. **Bitmap** graphics lose quality when they are resized.

Representing Bitmaps

Images are made up of **pixels** (Picture Elements). Each **pixel** is set to one colour. Together they look like an image. Individual **pixels** are unidentifiable.

Creating a Bitmap

Each **pixel** is given a binary value. Each value represents a different colour. Using one bit per **pixel** allows only 2 values, 0 and 1.

1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0	1	1
1	1	1	1	1	1	1	1	1	1

1	=	White
0	=	Black

More bits per **pixel** = more colour combinations.

- 1 bit = 2 Colours
- 2 bits = 4 Colours
- 3 bits = 8 Colours
- 4 bits = 16 Colours

How many bits per **pixel** required for 256 colours?

11	11	11	11	11	11	11	11	11	11
11	11	11	00	00	11	11	11	11	11
11	11	11	11	00	10	11	11		
11	11	01	01	01	10	10			
11	01	01	01						
11	01								

01 =	Red	10 =	Green
00 =	Black	11 =	White

Hexadecimals (Hex)

Hexadecimal is a number system using base 16.

It uses numbers 0–9 and letters A–F to represent values:

Decimal	Hex	Binary
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

👉 Why Use Hex?

Easier for humans to read than binary.

Shorter way to write long binary numbers.

Often used in colours (e.g. #FF0000 = red).

Text Representation

- Characters (like A, B, ?, 3) are represented using a **character set**.
- Most common is **ASCII** (American Standard Code for Information Interchange).
 - Each letter/character = 7 or 8 bits of binary.
 - Example: "A" = 01000001

Unicode

- A newer, bigger character set that includes **all languages**, emojis, and symbols.

🎵 Sound Representation

Sound is **analogue** (smooth waves) and must be converted into **digital** signals:

Term	Meaning
Sampling	Taking a number of measurements (samples) per second.
Sample rate	How many samples are taken per second (measured in Hz or kHz).
Bit depth	How many bits are used to store each sample (more bits = better quality).

🖼️ Image Representation (Bonus)

Though not listed, this is often taught alongside sound and text:

- Images are made of **pixels**.
- Each pixel has a **binary value** for colour.

The number of bits used per pixel = **colour depth**

Year 9 Algorithms Knowledge Organiser

What Is an Algorithm?

An **algorithm** is a set of clear, step-by-step instructions to solve a problem or complete a task.

Examples: Making a cup of tea, searching for a number, or sorting a list.

Searching Algorithms

Search Type	How It Works	Best For
Linear Search	Checks each item one by one until it finds the target.	Small or unsorted lists.
Binary Search	Repeatedly divides a sorted list in half to find a target.	Large, sorted lists only.

Reminder: Binary search only works if the list is in **order**!

Sorting Algorithms

Sort Type	How It Works
Bubble Sort	Repeatedly compares and swaps adjacent items until the list is sorted.
Insertion Sort	Builds a sorted list one item at a time, placing each in the correct position.
Merge Sort	Splits the list into halves, sorts them, and merges them back together.

Note:

- Bubble Sort = Simple but slow
- Merge Sort = Fast but uses more memory
- Insertion Sort = Good for small lists

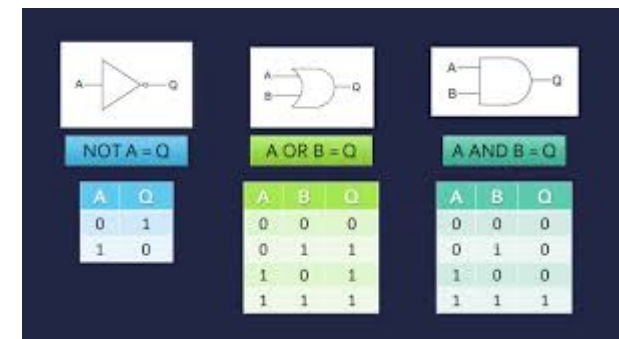
Logic Gates (Used in computer circuits)

Gate	Symbol	What It Does
AND		Both inputs must be 1 (true) to get 1 as output. Otherwise, the output is 0.
OR		At least one input must be 1 to get 1 as output.
NOT		Flips the input: if 1 goes in, 0 comes out; if 0 goes in, 1 comes out.

Example:

A B AND OR NOT A

0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0



Sorting algorithms

selection sort	insertion sort	merge sort	Quicksort
8, 5, 1, 3, 7	8, 5, 1, 3, 7	8, 5, 1, 3, 7	8, 5, 1, 3, 7
1, 5, 8, 3, 7	8, 5, 1, 3, 7	8, 5, 1, 3, 7	1, 8, 5, 3, 7
1, 3, 8, 5, 7	5, 8, 1, 3, 7	8, 5, 1, 3, 7	1, 3, 5, 8, 7
1, 3, 5, 8, 7	1, 5, 8, 3, 7	8, 5, 1, 3, 7	1, 3, 5, 7, 8
1, 3, 5, 7, 8	1, 3, 5, 7, 8	5, 8, 1, 3, 7	
	1, 3, 5, 7, 8	1, 5, 8, 3, 7	
		1, 3, 5, 7, 8	

■ = place ■ = minimum number
 ■ = number to be sorted ■ = sorted number(s)
 ■ = current pivot ■ = previously used pivot(s)