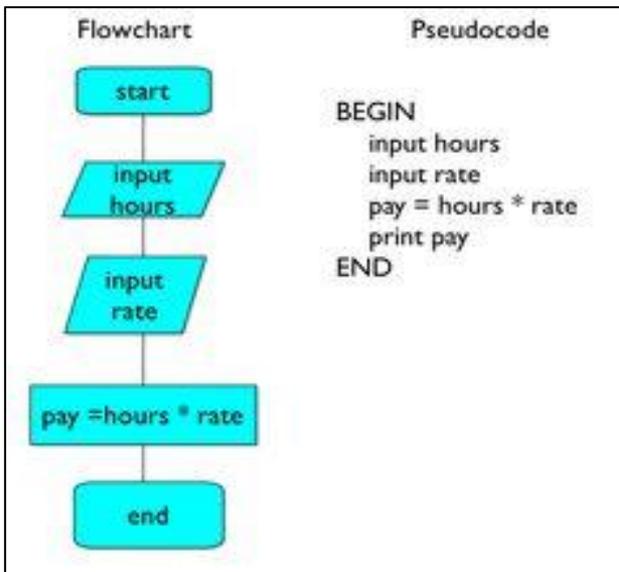


# GCSE | Fundamentals of Algorithms | Required knowledge

## Key terms

Algorithm	A sequence of instructions to be followed (usually by a computer,) to carry out a task.	
Computational thinking	A method of solving a problem logically, using abstraction, decomposition, pattern recognition and algorithms.	
Abstraction		Focusing on what is important, ignoring the unnecessary.
Decomposition		Breaking something into smaller parts, in order to solve the smaller parts first.
Pseudocode	A notation resembling a simplified programming language, used to create a program design.	
Flow Chart	A visual representation of an algorithm, using shapes and arrows to show a clear sequence of instructions.	



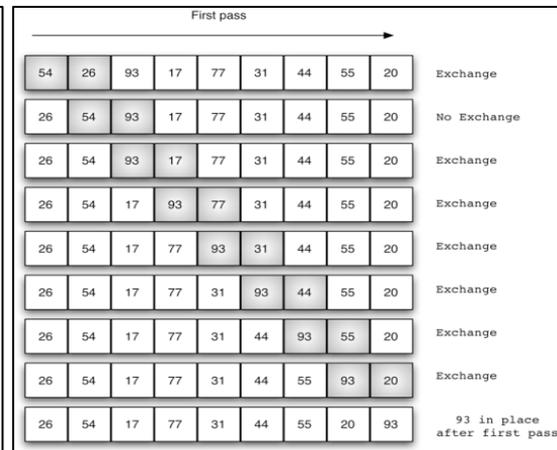
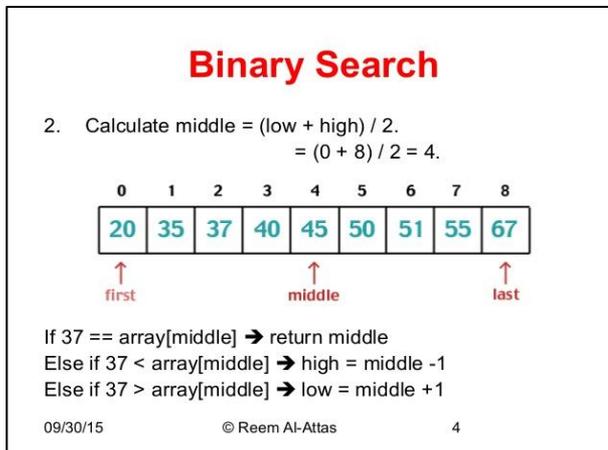
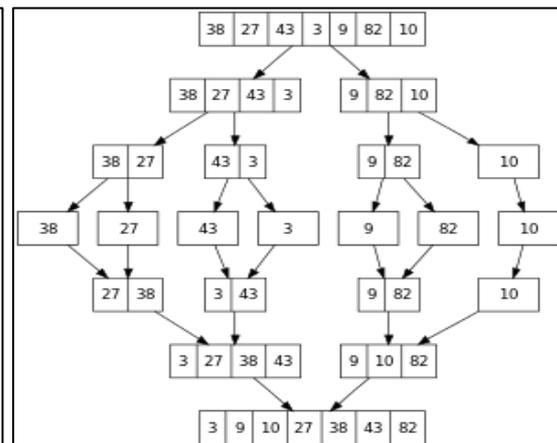
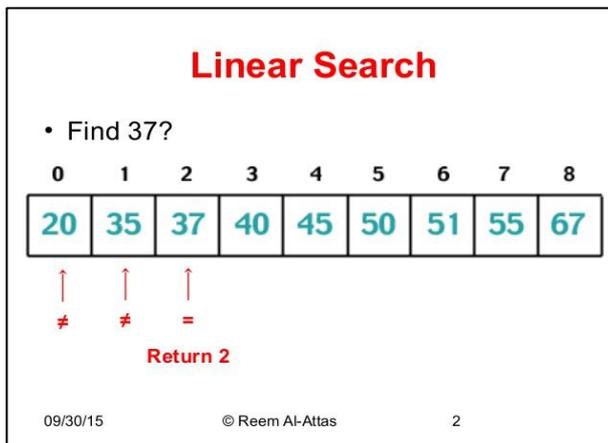
## Types of algorithms

### Searching algorithms

- Linear search
- Binary search
- Compare and contrast both types

### Sorting algorithms

- Merge sort
- Bubble sort
- Compare and contrast both types



## Efficiency - Time/Space. How long? How much memory?

# GCSE | Fundamentals of Algorithms | Required knowledge

Self-Quiz Questions	Self-Quiz Questions	Challenge Self-Quiz Quizzing
1. What is an algorithm?	11. What is computational thinking?	21. What is pattern recognition?
2. Why are algorithms used to design programs?	12. How does a binary search work?	22. Compare the use of flowcharts and pseudocode.
3. What technique is used to break problems down into smaller parts?	13. How does the bubble sort work?	23. How is pseudocode similar to python code?
4. What is abstraction?	14. How does the merge sort work?	24. How is pseudocode different to python code?
5. What shapes are used in a flowchart?	15. What does the word 'efficient' mean?	25. Why does creating algorithms make programming quicker?
6. What do each of these shapes do?	16. What 2 factors do you need to consider when comparing the efficiency of algorithms?	26. Why does creating algorithms allows us to create efficient programs?
7. What do the arrows show in a flowchart?	17. Which is more efficient? Linear or binary search?	27. What is a subroutine?
8. What is the purpose of a search algorithm?	18. Explain your decision.	28. How is a subroutine represented in a flowchart?
9. What is the purpose of a sort algorithm?	19. Which is more efficient? Bubble or merge sort?	29. How is a subroutine represented in pseudocode?
10. How does a linear search work?	20. Explain your decision.	30. What is the purpose of a trace table?

# GCSE | Programming | Required knowledge

## Data types

- Integer e.g. 23
- Real e.g. 23.7
- Character e.g. A or 5
- String e.g. A546TH
- Boolean e.g. TRUE or FALSE

## Programming Concepts

- Sequence
- Selection
  - IF... ELSE... /CASE
- Iteration
  - For
  - Do While/ Repeat Until

## Variables & constants

- Named storage space reserved in memory.
- Both **declared**. (←)
- Variable – Data that changes (score)
- Constant – Data that remains the same. (PI - 3.1415)

## Random number generation

## Arithmetic Operations

- ADD +
- SUBTRACT –
- DIVIDE /
- MULTIPLY \*
- MOD
- DIV

## Boolean operations

- NOT,
- AND,
- OR

## Relational operations

Operations	Meaning
= or ==	Is equal to
>	Is greater than
<	Is less than
<> or !=	Is not equal to
>=	Greater than or equal to
<=	Less than or equal to

## Data structures

- Arrays list = [1,1,1,1]
- 2d Arrays list = [[1,1,1,1], [2,2,2,2]]
  - Definition
  - Example
- Records

## Input/output and file handling

- User input/output from keyboard
- Read/write to text file

## Robust and secure programming

- Data validation
- Authentication routines
- Test data – normal, boundary, erroneous

## String Handling

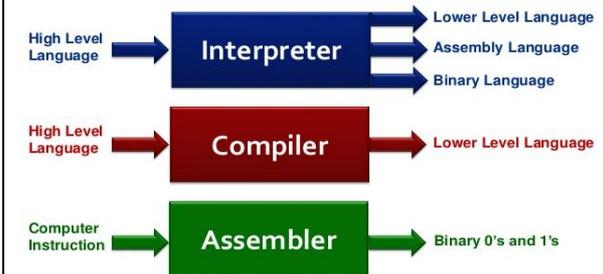
- Length
- Position
- Substring
- Concatenation
- Convert character to character code
- Convert character code to character
- String conversion operations

## Subroutines and functions/structured programming

- Concept
- Advantages
- Passing parameters
- Return values
- Local variables/global variables
- Modular programming

## Programming languages

- **Low level languages:**
  - Machine language
  - Assembly language
  - Mnemonics
- **High level languages:**
  - Source code/machine code
  - Advantages of high & low level
- **Program translators:**



# GCSE | Programming | Quiz questions

Self-Quiz Questions	Self-Quiz Questions	Challenge Self-Quiz Quizzing
1. Name the 5 data types.	11. What data type is.... likesChocolate = True	21. Define a data structure.
2. What does selection mean?	12. What data type is.... height = 1.52	22. What is a subroutine?
3. What does iteration mean?	13. What data type is.... age = 24	23. What are the parameters used for?
4. What is the purpose of a variable?	14. Why is it important to have a meaningful identifier name?	24. When might a 2d array be used?
5. What is a constant?	15. What is the job of the compiler?	25. Explain the difference between reading and writing a text file.
6. What is machine code?	16. What is the job of the assembler?	26. Discuss the 3 types of test data?
7. What is assembly code?	17. What is the job of the interpreter?	27. How do you know what code will be repeated in a loop?
8. Give some examples of low level languages.	18. What might we mean by 'robust programming'?	28. What is nested iteration/selection?
9. Give some examples of high level languages.	19. What is concatenation?	29. What is the purpose of a validation routine?
10. Why are low level languages needed?	20. What is the purpose of the relational operators?	30. What is the purpose of an authentication routine?