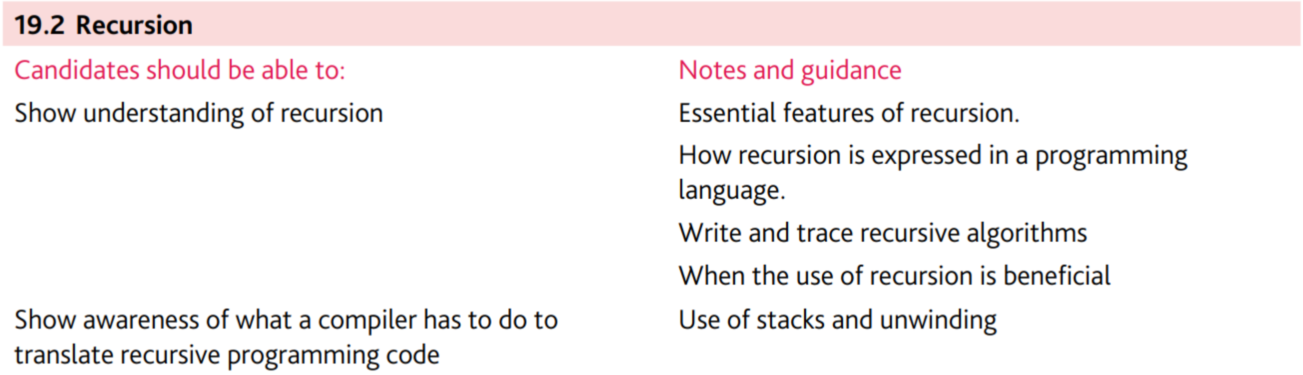
1. Essential features of recursion



19.2 Recursion	
Candidates should be able to:	Notes and guidance
Show understanding of recursion	Essential features of recursion. How recursion is expressed in a programming language. Write and trace recursive algorithms When the use of recursion is beneficial
Show awareness of what a compiler has to do to translate recursive programming code	Use of stacks and unwinding

Essential features of recursion:

- It is defined in terms of itself // it calls itself
- It has a stopping condition
- It has self-contained subroutine
- It can return data to its previous call
- `- -`
- Must have a base case/ stopping condition
- Must have a general case
- … which calls itself recursively // defined in terms of itself
- … which changes its state and moves towards the base case
- Unwinding occurs once the base case is reached

Explain a reason why a stack is a suitable abstract data type(ADT) to implement recursion:

- A stack is a LIFO(last in first out) data structure
- Each recursive call is pushed to the stack
- And is popped when the function ends
- Enable unwinding/back tracking
- … to maintain the required order