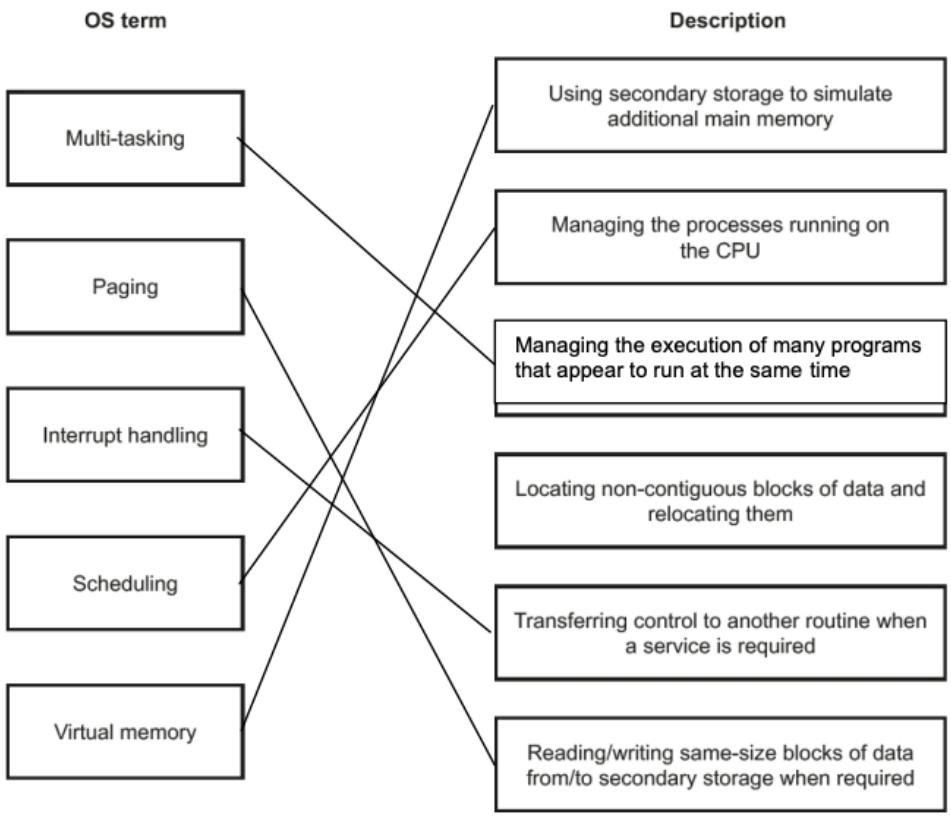
Question	Answer	Marks
3(a)	**One** mark for each correct line from Operating System Term to Description  	**5**
3(b)	**One** mark for each correct statement **(Max 4)** • An interpreter examines source code one statement at a time • Check each statement for errors • …If no error is found the statement is executed • …If an error is found this is reported and the interpreter halts • Interpretation is repeated for every iteration in repeated sections of code/in loops • Interpretation has to be repeated every time the program is run	**4**

Question	Answer	Marks
4(a)(i)	**One** mark for each correct marking point **(Max 2)** • Reverse Polish Notation provides an unambiguous method of representing an expression • … reading from left to right • …without the need to use brackets • …with no need for rules of precedence / BODMAS	**2**

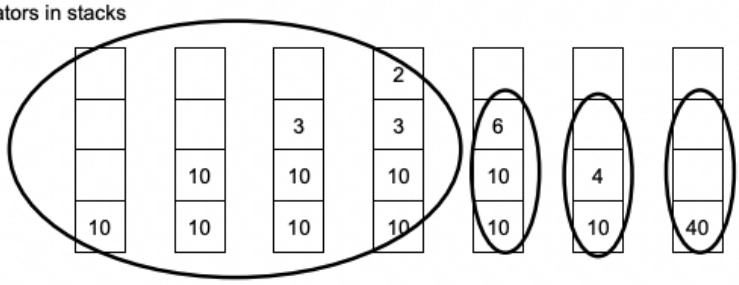
Question	Answer	Marks
4(a)(ii)	**One mark** for identification of the data structure, **One mark** for a sensible reason **Either:** Structure: stack The operands are popped from the stack in the reverse order to how they were pushed **Or:** Structure: Binary tree A (binary) tree allows both infix and postfix to be evaluated (tree traversal)	**2**
4(b)	`a b - a c + * 7 /`	**1**
4(c)	`a / b * 4 - (a + b)`	**1**

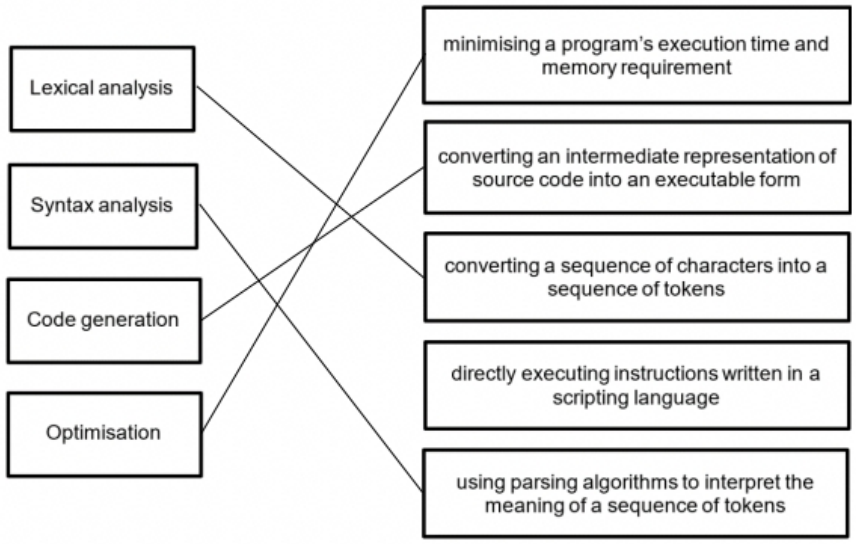
4(d)	**1 mark** for correct structure
	1 mark for correct substitution
	
	`(a + b) / (c / d)`
	`(17 + 3) / (48 / 12)`
		2

Question	Answer	Marks
4(a)	**One** mark for each marking point (**Max 2**) • <character>::= • $|%|&|*|# **Complete answer** <character>::= $|%|&|*|#	**2**
4(b)(i)	For example: $A9E3	**1**
4(b)(ii)	**One** mark for each marking point (**Max 4**) • <password>::=<character> … • … <code> • <code>::= … • … <digit>|<capital_letter> • … |<digit><code>|<capital_letter><code> **Complete answer** <password>::=<character><code> <code>::=<digit>|<capital_letter>|<digit><code>|<capital_letter><code>	**4**

Question	Answer	Marks

5(a)	**One mark** for each in order `jk+jk-/`
	`jk+`
	`jk-/`
		2

5(b)(i)	**1 mark** per ring Do not allow operators in stacks 	**4**
5(b)(ii)	Any **four** from **Max 4** **Max 3** generic answer only • Working from left to right in the expression • PUSH 10/m onto the stack • PUSH the following numbers (10/m, 3/j, 2/k) onto the stack • When the first operator ,*, is reached • … POP the top two numbers, 2/k and 3/j • … apply the operation • PUSH result back onto stack • Continue to the end of the expression	**4**
5(c)	Any **two** from • recursion • implementation of ADTs e.g. linked lists • procedure calls • interrupt handling (storing contents of registers etc)	**2**

Question	Answer	Marks
4	**One** mark for each correct line connecting one stage of compilation to a description **Stage of compilation** / **Description**  	**4**

Question	Answer	Marks
5(a)	`a b * b + d - 15 +`	**1**
5(b)(i)	`(a - b) * (c + d) / a`	**1**
5(b)(ii)	`-39`	**1**