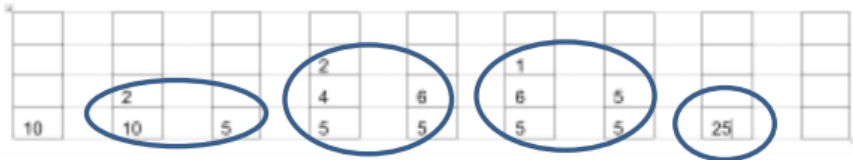


# Answer

## Answer 1

6(a)	$P \ Q \ + \ P \ Q \ - \ *$ <b>One</b> mark for $P \ Q \ +$ <b>One</b> mark for $P \ Q \ - \ *$	<b>2</b>
6(b)(i)	 <p><b>One</b> mark for each correct stack after a calculation</p>	<b>4</b>
6(b)(ii)	$((P + Q) * M) - (R - P)$ <b>One</b> mark for $((P + Q) * M)$ <b>One</b> mark for $-(R - P)$	<b>2</b>
6(c)	Any <b>two</b> from: <ul style="list-style-type: none"> <li>Expressions are always evaluated left to right</li> <li>Each operator uses the two previous values on the stack (except unary minus)</li> <li>Description of pushing and popping on a stack</li> </ul>	<b>2</b>

## Answer 2

4(a)	<b>1 mark</b> for <b>2</b> correct rows, <b>2 marks</b> for <b>3</b> correct rows, <b>3 marks</b> for <b>4</b> correct rows	<b>3</b>																				
<table border="1"> <thead> <tr> <th rowspan="2">Symbol</th><th colspan="2">Token</th></tr> <tr> <th>Value</th><th>Type</th></tr> </thead> <tbody> <tr> <td>Counter</td><td>60</td><td>Variable</td></tr> <tr> <td>0</td><td>61</td><td>Constant</td></tr> <tr> <td>Password</td><td>62</td><td>Variable</td></tr> <tr> <td>"Cambridge"</td><td>63</td><td>Constant</td></tr> <tr> <td>1</td><td>64</td><td>Constant</td></tr> </tbody> </table>			Symbol	Token		Value	Type	Counter	60	Variable	0	61	Constant	Password	62	Variable	"Cambridge"	63	Constant	1	64	Constant
Symbol	Token																					
	Value	Type																				
Counter	60	Variable																				
0	61	Constant																				
Password	62	Variable																				
"Cambridge"	63	Constant																				
1	64	Constant																				

4(b)	<table><tr><td>60</td><td>01</td></tr></table> <p>First two cells given in question.</p> <p><b>1 mark</b> for next 3 cells</p> <table><tr><td>61</td><td>51</td><td>62</td></tr></table> <p><b>1 mark</b> for the remainder</p> <table><tr><td>4E</td><td>4A</td><td>62</td><td>04</td><td>63</td><td>4B</td><td>51</td><td>62</td><td>4C</td><td>60</td><td>....</td></tr></table> <table><tr><td>....</td><td>01</td><td>60</td><td>02</td><td>64</td><td>4F</td><td>62</td><td>03</td><td>63</td><td>52</td><td>60</td></tr></table>	60	01	61	51	62	4E	4A	62	04	63	4B	51	62	4C	60	....	....	01	60	02	64	4F	62	03	63	52	60	<b>2</b>
60	01																												
61	51	62																											
4E	4A	62	04	63	4B	51	62	4C	60	....																			
....	01	60	02	64	4F	62	03	63	52	60																			
4(c)(i)	<p><b>1 mark</b> per bullet point</p> <ul style="list-style-type: none"><li>∞ Removing the fourth line (LDD 238) ...</li><li>∞ Changing operand for second ADD from 236 to 238 ...</li><li>∞ ... First three lines and last line unchanged</li></ul> <p>LDD 236 ADD 237 STO 236 ADD 238 STO 238</p>	<b>3</b>																											
4(c)(ii)	<p><b>1 mark</b> per bullet point (<b>max 2</b>)</p> <ul style="list-style-type: none"><li>∞ Optimisation means that the code will have fewer instructions</li><li>∞ Optimised code occupies less space in memory</li><li>∞ Fewer instructions reduces the execution time of the program</li></ul>	<b>2</b>																											

## Answer 3

4(a)	<p><b>1 mark</b> per bullet point (max 4)</p> <ul style="list-style-type: none"> <li>∞ Working from left to right in the expression</li> <li>∞ If element is a number PUSH that number onto the stack</li> <li>∞ If element is an operator then POP the first two numbers from stack ...</li> <li>∞ ... perform that operation on those numbers</li> <li>∞ PUSH result back onto stack</li> <li>∞ End once the last item in the expression has been dealt with</li> </ul>	<b>4</b>
4(b)	<p><b>1 mark</b> per ring (not all stacks are shown)</p> <p>Do not allow operators in stacks</p> <p>Accept intermediate correct stack values</p>	<b>4</b>

## Answer 4

2(a)(i)	35 is not a variable	<b>1</b>
2(a)(ii)	:= is not an operator	<b>1</b>
2(a)(iii)	9 is not a digit	<b>1</b>
2(b)	<p><b>1 mark</b> for each bullet point</p> <pre> &lt;operator&gt; ::= • ==   &gt;   &lt;  &lt;number&gt; ::= • &lt;digit&gt;&lt;digit&gt;  &lt;variable&gt; ::= • &lt;letter&gt; •  &lt;letter&gt;&lt;variable&gt;  &lt;condition&gt; ::= • &lt;variable&gt;&lt;operator&gt;&lt;number&gt; •  &lt;variable&gt;&lt;operator&gt;&lt;variable&gt; </pre>	<b>6</b>

## Answer 5

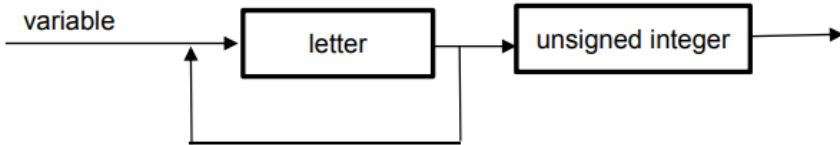
7(a)(i)	<p>1 mark for each bullet point to <b>max 2</b></p> <p>Keyword table:</p> <ul style="list-style-type: none"> <li>• The reserved words used</li> <li>• The operators used</li> <li>• Their matching tokens</li> </ul>	<b>2</b>
7(a)(ii)	<p>1 mark for each bullet point to <b>max 2</b></p> <p>Symbol table:</p> <ul style="list-style-type: none"> <li>• Identifier name used</li> <li>• ... the (data) type</li> <li>• ... role, e.g. variable, constant, array, procedure / scope</li> <li>• Location (marker) // value of constant</li> </ul>	<b>2</b>
7(a)(iii)	<p>1 mark per bullet point to <b>max 2</b></p> <ul style="list-style-type: none"> <li>• Keywords / operators are looked up (in the keyword table)</li> <li>• Keywords / operators are represented by tokens</li> <li>• Identifiers are looked up in (the symbol table)</li> <li>• Identifiers are converted to locations / addresses</li> <li>• Used to create a sequence of tokens (for the program)</li> </ul>	<b>2</b>

7(a)(iv)	The white space removed // redundant characters are removed // removal of comments // identification of errors	1
7(b)	1 mark per bullet point to <b>max 2</b> <ul style="list-style-type: none"> <li>Redundant code removed // fewer instructions required</li> <li>Program requires less memory / storage space</li> <li>Code reorganised to make it more efficient</li> <li>Program will complete task in a shorter time</li> </ul>	2

## Answer 6

5(a)(i)	c4 is not a <u>signed</u> integer	1
5(a)(ii)	10 is not a valid <u>signed</u> integer // 0 is not a valid digit/signed integer // only one digit allowed	1
5(a)(iii)	wrong assignment operator // should be = not := // 6 is not a valid digit/signed integer	1
5(b)	1 mark per bullet assignment $\infty$ <variable>=<variable><operator><signed integer> variable $\infty$ <letter><letter> signed integer $\infty$ +<digit>   -<digit> operator $\infty$ ^   \  <assignment statement> ::= <variable> = <variable><operator><signed integer> <variable> ::= <letter><letter> <signed integer> ::= +<digit>   -<digit> <operator> ::= ^   \	4
5(c)	1 mark per bullet $\infty$ <letter>  $\infty$ <letter><variable>  For example: <letter> <letter><variable> <letter> <variable><letter>	2

## Answer 7

2(a)(i)	5 is not a variable	1
2(a)(ii)	D is not a valid letter	1
2(a)(iii)	There are two operators (only one is allowed) // three variables on the right hand side but only two allowed	1
2(b)	<p>1 mark for each bullet</p> <p>assignment:</p> <ul style="list-style-type: none"> <li>• <code>&lt;variable&gt; = &lt;variable&gt;&lt;operator&gt;&lt;variable&gt;;</code></li> </ul> <p>variable:</p> <ul style="list-style-type: none"> <li>• <code>&lt;letter&gt; </code></li> <li>• <code>&lt;letter&gt;&lt;unsigned integer&gt;</code></li> </ul> <p>unsigned integer:</p> <ul style="list-style-type: none"> <li>• <code>&lt;digit&gt; </code></li> <li>• <code>&lt;digit&gt;&lt;unsigned integer&gt;</code></li> </ul> <p>operator:</p> <ul style="list-style-type: none"> <li>• <code>+   -   *   /</code></li> </ul> <pre> &lt;assignment statement&gt; ::= &lt;variable&gt; = &lt;variable&gt;&lt;operator&gt;&lt;variable&gt;; &lt;variable&gt; ::= &lt;letter&gt;   &lt;letter&gt;&lt;unsigned integer&gt; &lt;unsigned integer&gt; ::= &lt;digit&gt;   &lt;digit&gt;&lt;unsigned integer&gt; &lt;operator&gt; ::= +   -   *   /           </pre>	6
2(c)	<p>1 mark per bullet</p> <ul style="list-style-type: none"> <li>• variable with arrow</li> <li>• followed by repeated letter</li> <li>• followed by unsigned integer and arrow</li> </ul>  <pre> graph LR     var(variable) --&gt; letter[letter]     letter --&gt; ui[unsigned integer]     ui --&gt; out[...]     letter --&gt; var           </pre>	3

## Answer 8

6(a)	1 mark for each correct row	3																				
<table><tr><th rowspan="2">Symbol</th><th colspan="2">Token</th></tr><tr><th>Value</th><th>Type</th></tr><tr><td>Start</td><td>60</td><td>Variable</td></tr><tr><td>1</td><td>61</td><td>Constant</td></tr><tr><td>Number</td><td>62</td><td>Variable</td></tr><tr><td>Counter</td><td>63</td><td>Variable</td></tr><tr><td>12</td><td>64</td><td>Constant</td></tr></table>			Symbol	Token		Value	Type	Start	60	Variable	1	61	Constant	Number	62	Variable	Counter	63	Variable	12	64	Constant
Symbol	Token																					
	Value	Type																				
Start	60	Variable																				
1	61	Constant																				
Number	62	Variable																				
Counter	63	Variable																				
12	64	Constant																				
6(b)	1 mark for each circled section	2																				
<table><tr><td>60</td><td>01</td><td>61</td><td>51</td><td>62</td><td>4E</td><td>63</td><td>01</td><td>60</td><td>50</td><td>64</td><td>52</td><td>62</td><td>02</td><td>63</td><td>53</td></tr></table>			60	01	61	51	62	4E	63	01	60	50	64	52	62	02	63	53				
60	01	61	51	62	4E	63	01	60	50	64	52	62	02	63	53							
6(c)	1 mark per bullet point to max 2:	2																				
<ul style="list-style-type: none"><li>∞ constructing parse tree // parsing</li><li>∞ checking the table of tokens to ensure that the rules/syntax/grammar of the language are/is obeyed</li><li>∞ producing an error report</li></ul>																						
6(d)(i)	shortens execution time of program// time taken to execute whole program decreases	1																				
6(d)(ii)	1 mark for each of the following:	5																				
<ul style="list-style-type: none"><li>∞ LDD 236</li><li>ADD 237</li><li>STO 512</li><li>ADD 238</li><li>STO 513</li><li>ADD 239</li><li>STO 514</li><li>∞ Remove line 4 LDD 236 correct lines 3 and 6 in original code</li><li>∞ Remove line 5 ADD 237 correct lines 3 and 6 in original code</li><li>∞ Remove line 8 and 9 LDD 236 and ADD 237 correct lines 7 and 11 in original code</li><li>∞ Remove line 10 ADD 238 correct lines 7 and 11 in original code</li></ul>																						

## Answer 9

4(a)	1 mark per row																																																												
	<table><tr><th rowspan="2">Symbol</th><th colspan="2">Token</th></tr><tr><th>Value</th><th>Type</th></tr><tr><td>Number1</td><td>60</td><td>Variable</td></tr><tr><td>Number2</td><td>61</td><td>Variable</td></tr><tr><td>Answer</td><td>62</td><td>Variable</td></tr><tr><td>10</td><td>63</td><td>Constant//Literal</td></tr><tr><td>0</td><td>64</td><td>Constant//Literal</td></tr></table>	Symbol	Token		Value	Type	Number1	60	Variable	Number2	61	Variable	Answer	62	Variable	10	63	Constant//Literal	0	64	Constant//Literal																																								
Symbol	Token																																																												
	Value	Type																																																											
Number1	60	Variable																																																											
Number2	61	Variable																																																											
Answer	62	Variable																																																											
10	63	Constant//Literal																																																											
0	64	Constant//Literal																																																											
4(b)	1 mark for each circled section																																																												
	<table><tr><td>51</td><td>60</td><td>51</td><td>61</td><td>51</td><td>62</td><td>4A</td><td>62</td><td>03</td><td>60</td><td>02</td><td>61</td><td>4B</td><td>52</td><td>63</td><td>4D</td><td>52</td><td>64</td><td>4C</td></tr></table>	51	60	51	61	51	62	4A	62	03	60	02	61	4B	52	63	4D	52	64	4C																																									
51	60	51	61	51	62	4A	62	03	60	02	61	4B	52	63	4D	52	64	4C																																											
4(c)(i)	(Code) Optimisation																																																												
4(c)(ii)	1 mark per bullet point: <ul style="list-style-type: none"><li>LDD 236</li><li>ADD 237</li><li>ADD 238</li><li>SUB 239</li><li>STO 235</li></ul> <p>Copy the instructions</p> <ul style="list-style-type: none"><li>Remove line 4 STO 540 correct lines 3 and 6 in original code</li><li>Remove line 5 LDD 540 correct lines 3 and 6 in original code</li></ul>																																																												
4(c)(iii)	1 mark per bullet point: <ul style="list-style-type: none"><li>Code has fewer instructions/occupies less space in memory</li><li>shortens execution time of program // time taken to execute whole program decreases</li></ul>																																																												
4(d)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>3</td><td></td><td>3</td><td></td><td>2</td><td></td><td></td></tr><tr><td>2</td><td>2</td><td>2</td><td>2</td><td>6</td><td>6</td><td>9</td><td>9</td><td>7</td><td></td></tr></table> <p>1 mark ← 1 mark → 1 mark</p> <p>1 mark no operators on the stack anywhere</p>																																	2									1	1	3		3		2			2	2	2	2	6	6	9	9	7	
		2																																																											
	1	1	3		3		2																																																						
2	2	2	2	6	6	9	9	7																																																					

## Answer 10

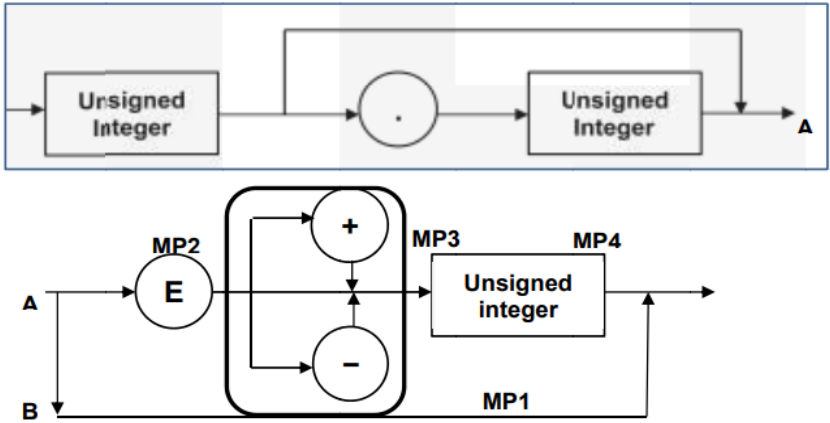
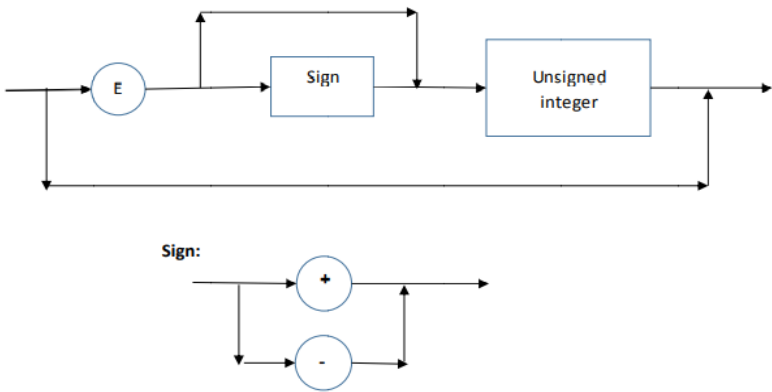
3(a)(i)	There should be a colon before the '=' sign	1
3(a)(ii)	The second operand should be an unsigned integer and not a variable	1
3(a)(iii)	A32 is not a variable, as a variable should be a letter followed by a single digit	1
3(b)	<pre> &lt;assignment_statement&gt; ::= &lt;variable&gt; := &lt;variable&gt; &lt;operator&gt; &lt;unsigned_integer&gt; &lt;variable&gt; ::= &lt;letter&gt; &lt;digit&gt; &lt;unsigned_integer&gt; ::= &lt;digit&gt;   &lt;digit&gt; &lt;unsigned_integer&gt; &lt;letter&gt; ::= A   B   C &lt;operator&gt; ::= +   -   *   ^ </pre>	6
3(c)	<p><b>Variable</b></p> <p>Syntax diagram shows one or two letters</p> <p>Syntax diagram shows zero, one or two digits</p>	2
3(d)	<pre> &lt;assignment_statement&gt; ::= &lt;variable&gt; := &lt;variable&gt; &lt;operator&gt; &lt;real&gt; &lt;real&gt; ::= &lt;unsigned_integer&gt; . &lt;unsigned_integer&gt; </pre>	2

## Answer 11

4(a)(i)	Because a valid unsigned integer can be two digits / one or more digits (1) Both 3 and 2 are digits (1)	2
4(a)(ii)	<p>Because a valid unsigned number can be an unsigned integer followed by a decimal point followed by an unsigned integer (1)</p> <p>32 is an unsigned integer and 5 is an unsigned integer (because it is a digit) and there is a point in between (1)</p> <p>Alternative response for 2 marks, combination of order and validity:</p> <p>32 is a (valid) unsigned integer, followed by a decimal point, and 5 which is another (valid) unsigned integer</p> <p>Validity mark must refer to 32 and 5</p>	2



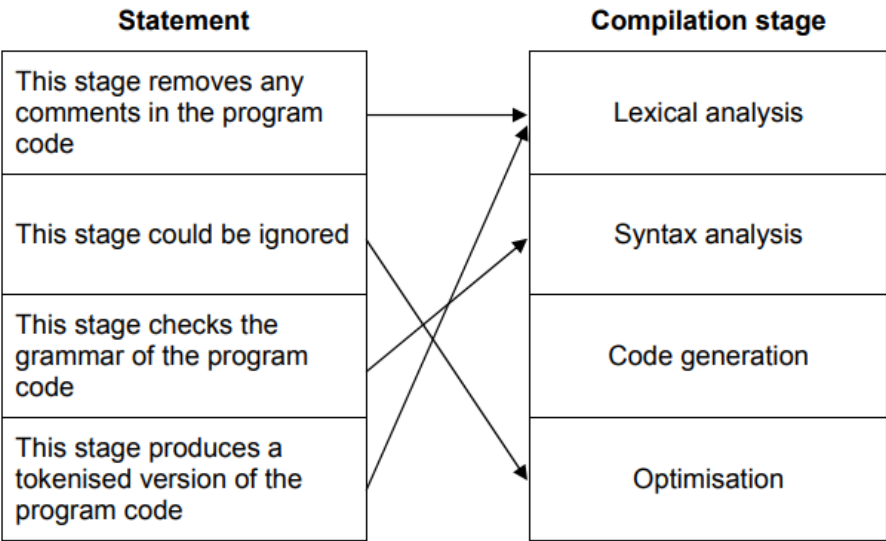
4(b)	<p> <math>\langle \text{unsigned number} \rangle ::=</math>  <math>\langle \text{unsigned\_integer} \rangle \mid (1)</math>  <math>\langle \text{unsigned\_integer} \rangle . \langle \text{unsigned\_integer} \rangle \quad (1)</math> </p> <p>Accept order reversed:</p> <p> <math>\langle \text{unsigned\_integer} \rangle ::=</math>  <math>\langle \text{digit} \rangle \mid (1)</math>  <math>\langle \text{digit} \rangle \langle \text{unsigned\_integer} \rangle \quad (1)</math> </p> <p>Accept <math>\langle \text{digit} \rangle \mid \langle \text{unsigned\_integer} \rangle \langle \text{digit} \rangle</math></p> <p>If order reversed mark as above</p> <p> <math>\langle \text{digit} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 0 \quad (1)</math> </p> <p>Accept the list in any order, as long as all 10 digits included</p>	5
------	--	---

4(c)(i)	 <p> <b>MP1:</b> Line to indicate exponent is optional (B line) (1)  <b>MP2:</b> 'E' present at start of exponent (1)  <b>MP3:</b> Optional '+' or '-' (1)  <b>MP4:</b> Unsigned integer (1) </p> <p>Alternative correct answer:  MP3 needs both the sign 'box' and the sign diagram for the mark</p> 	4
---------	---	---

4(c)(ii)	<pre> &lt;unsigned number&gt; ::=   &lt;unsigned_integer&gt;     &lt;unsigned_integer&gt;.&lt;unsigned_integer&gt;  (1)  Accept any order    &lt;unsigned_integer&gt; &lt;exponent&gt;     &lt;unsigned_integer&gt;.&lt;unsigned_integer&gt; &lt;exponent&gt; (1)  Accept any order  &lt;exponent&gt; ::= E &lt;sign&gt; &lt;unsigned_integer&gt;   E &lt;unsigned_integer&gt;      (1)  &lt;sign&gt; ::= +   -      (1) </pre>	4
----------	---	---

# Answer 12

2 (a)



1 mark  
for each  
correct  
line

[4]

(b) (i) A B +  
C D - \*

[1]  
[1]

(ii) A -  
B / 4 \*  
C D - /

[1]  
[1]  
[1]

(c) (i)

		4		3		
	1	1	5	5	2	
2	2	2	2	2	2	4
		+	-		*	

1  
mark  
per ring

[4]

(ii)  $x^*$

[1]

$(w + z - y)$

[1]

Order must be correct for both parts

(iii) No need for rules of precedence

[1]

No need for brackets

[1]

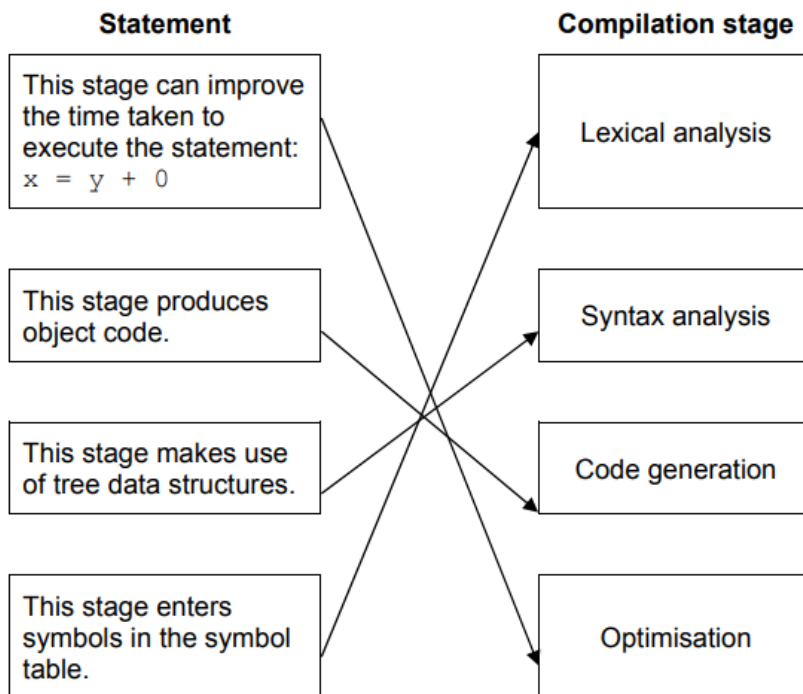
In RPN evaluation of operators is always left to right

[1]

[Max 2]

## Answer 13

2 (a)



1  
mark for  
each  
correct  
line

[4]

(b)  $PQ + RS / -$  [1]  
[1]

(c) (i)

					2			
				3	3	5		
	2		1	1	1	1	6	
2	2	4	4	4	4	4	4	-2
	*				+	+	-	

1  
mark  
per ring

[4]

(ii)  $b * a$  [1]  
 $-(c + d + a)$  [1]  
Order must be correct for both parts

(iii) Rules of precedence means different operators have different priorities // by example [1]  
multiply is done before add  
In RPN evaluation of operators is left to right // operators are used in the sequence in [1]  
which they are read [1]  
No need for brackets // infix may require the use of brackets [1]

[Max 2]

## Answer 14

1 (a) (i)	';' missing	1
(ii)	'2' is not a variable	1
(iii)	'e' is not a valid letter	1
(b)	$\langle \text{assignment statement} \rangle ::=$ $\quad \quad \quad \langle \text{variable} \rangle =$ $\langle \text{variable} \rangle \langle \text{operator} \rangle \langle \text{variable} \rangle ;$ $\langle \text{variable} \rangle ::= \langle \text{letter} \rangle   \langle \text{letter} \rangle \langle \text{letter} \rangle$ $\quad   \langle \text{letter} \rangle \langle \text{letter} \rangle \langle \text{letter} \rangle$ $\langle \text{letter} \rangle ::= a   b   c   d$ $\langle \text{operator} \rangle ::= +   -   *   \div$	2 2 1 1

(c)	<u>&lt;letter&gt;   &lt;letter&gt;&lt;variable&gt;</u> // <u>&lt;letter&gt;  </u> <u>&lt;variable&gt;&lt;letter&gt;</u>	2
(d) (i)	debugging is <u>faster</u> / <u>easier</u> // can debug incomplete code // better diagnostics	1
(ii)	compiler produces executable version – not readable / no need for source code // difficult to reverse-engineer	1

## Answer 15

2 (a)

Symbol	Token	
	Value	Type
Counter	60	variable
1.5	61	constant
Num1	62	variable
5.0	63	constant

[1]

[1+1]

(b)

6	0	6	5	6	4	6	0	6	4	6	0	6	0	6	4
0	1	1	1	2	A	0	3	2	B	2	1	2	2	3	C

[1+1]

(c) (i) Code optimisation

[1]

(ii) LDD 234  
ADD 235  
ADD 236  
STO 233

[1]

[1]

1 mark for first 2 lines, 1 mark for last 2 lines, with no other lines added

(iii) Code has fewer instructions/occupies less space in memory when executed  
minimises execution time of code // code will execute faster

[1]

[1]

## Answer 16

2 (a)

Symbol	Token	
	Value	Type
Start	60	Variable
0.1	61	Constant
Counter	62	Variable
10	63	Constant

[1]

[1+1]

(b)

60	01	61	4E	62	01	60	50	63	52	62	02	60	53
----	----	----	----	----	----	----	----	----	----	----	----	----	----

[1+1]

(c) (i) syntax analysis

[1]

(ii) any **two** points from:

construct parse tree // parsing  
checking syntax/grammar  
produce error report

[max. 2]

(d) (i) Minimise the execution time // code runs faster

[1]

(ii) Compiler could calculate  $2*6$  and replace it with the value 12.

[1]

(iii) LDD 436  
ADD 437  
STO 612  
ADD 438  
STO 613

}  
}  
}  
}  
}

[1]

[1]

[1]

-1 for each additional instruction; 0 for copy of original code