

QUESTION 1.



1	(a) (i)	';' missing	
	(ii)	'2' is not a variable	
	(iii)	'e' is not a valid letter	1
	(b)	$\begin{aligned} \text{<assignment statement>} &::= \\ &\quad \text{<variable> =} \\ &\quad \text{<variable><operator><variable>;} \\ \\ \text{<variable>} &::= \text{<letter> <letter><letter>} \\ &\quad \text{ <letter><letter><letter>} \\ \\ \text{<letter>} &::= \text{a b c d} \\ \\ \text{<operator>} &::= \text{=+ - * ÷} \end{aligned}$	2 2 1 1
	(c)	$\begin{aligned} &\text{<letter> <letter><variable> // <letter> } \\ &\text{<variable><letter>} \end{aligned}$	2
	(d) (i)	debugging is faster / easier // can debug incomplete code // better diagnostics	1
	(ii)	compiler produces executable version – not readable / no need for source code // difficult to reverse-engineer	1
			Total: 13
2	(a)		1

QUESTION 2.



1	(a) (i)	Wrong assignment operator (should be ':= ' not '=')	
	(ii)	0 is not a digit	
	(iii)	'B' is not a number	1
	(b)	$\langle \text{assignmentstatement} \rangle ::=$ $\quad \underline{\langle \text{variable} \rangle} ::= \underline{\langle \text{variable} \rangle \langle \text{operator} \rangle \langle \text{number} \rangle}$ $\langle \text{variable} \rangle ::= \langle \text{letter} \rangle \langle \text{number} \rangle$ $\langle \text{number} \rangle ::= \underline{\langle \text{digit} \rangle} \underline{\langle \text{digit} \rangle \langle \text{number} \rangle}$ $\langle \text{letter} \rangle ::= A B C$ $\langle \text{digit} \rangle ::= 1 2 3 4 5$ $\langle \text{operator} \rangle ::= + - * /$	$1 + 1$ 1 $1 + 1$ 1
	(c) (i)	Use of <u>software</u> ... (idea of using) to implement a <u>hardware</u> set-up (idea of implementing / simulating / emulating)	1 1
	(ii)	e.g. no need to acquire client hardware for testing / reduces set-up time for test system / common development system for all developers	1
	(iii)	e.g. software emulation runs slower than real hardware / not possible to emulate some hardware	1
			Total: 13

QUESTION 3.



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]

Page 3	Mark Scheme	
	Cambridge International A Level – October/November 2015	



(c) (i) Code optimisation

(ii) LDD 234
ADD 235
ADD 236
STO 233

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]
[11]

QUESTION 4.



2 (a)

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

[1]

[1+1]

Page 3	Mark Scheme	
	Cambridge International A Level – October/November 2015	



(b)

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

(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]

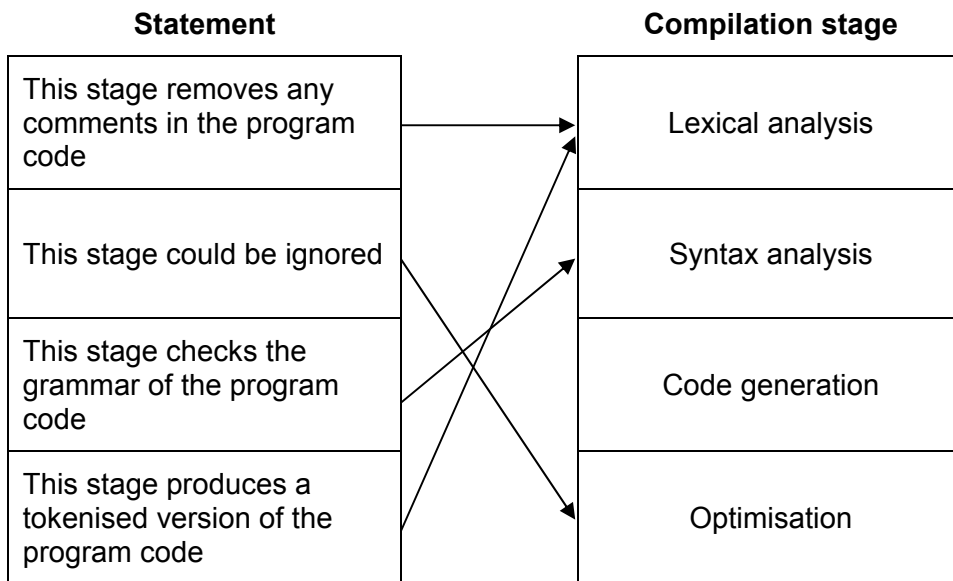
1 for each additional instruction; 0 for copy of original code

QUESTION 5.

Cambridge International A Level – October/November 2016



2 (a)



1 mark
for each
correct
line

[4]

(b) (i) $AB + CD - *$

[1]

[1]

(ii) $A - B / 4 * CD - /$

[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) $x * (w + z - y)$
Order must be correct for both parts

[1]

[1]

(iii) No need for rules of precedence
No need for brackets
In RPN evaluation of operators is always left to right

[1]

[1]

[1]

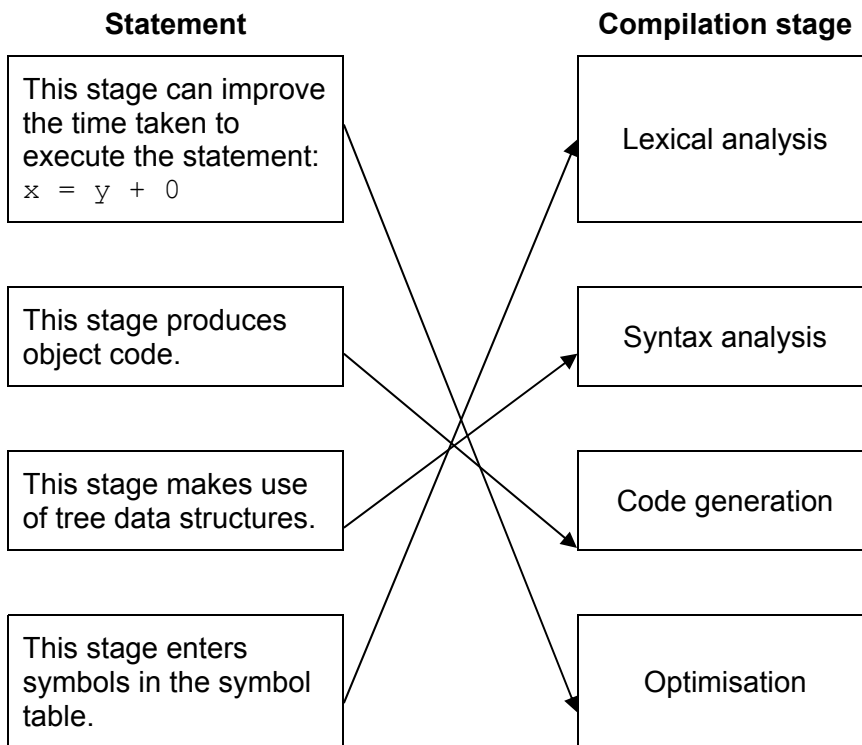
[Max 2]

QUESTION 6.

Cambridge International A Level – October/November 2016



2 (a)



1
mark for
each
correct
line

[4]

(b) $P Q +$
 $R S / -$

[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) $b * a$
 $- (c + d + a)$

[1]

[1]

Order must be correct for both parts

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

[1]

[1]

[1]

[Max 2]

QUESTION 7.



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

QUESTION 8.



Question	Answer	
4(a)(i)	Because a valid unsigned integer can be two digits / one or more digits (1) Both 3 and 2 are digits (1)	
4(a)(ii)	Because a valid unsigned number can be an unsigned integer followed by a decimal point followed by an unsigned integer (1) 32 is an unsigned integer and 5 is an unsigned integer (because it is a digit) and there is a point in between (1) Alternative response for 2 marks, combination of order and validity: 32 is a (valid) unsigned integer, followed by a decimal point, and 5 which is another (valid) unsigned integer Validity mark must refer to 32 and 5	2
4(b)	<code><unsigned number> ::=</code> <code><unsigned_integer> (1)</code> <code><unsigned_integer>.<unsigned_integer> (1)</code> Accept order reversed: <code><unsigned_integer> ::=</code> <code><digit> (1)</code> <code><digit> <unsigned_integer> (1)</code> Accept <code><digit> <unsigned_integer> <digit></code> If order reversed mark as above <code><digit> ::= 1 2 3 4 5 6 7 8 9 0 (1)</code> Accept the list in any order, as long as all 10 digits included	5



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



Question	Answer
4(c)(ii)	<pre><unsigned number> ::= <unsigned_integer > <unsigned integer>.<unsigned_integer> (1) Accept any order <unsigned_integer > <exponent> <unsigned integer>.<unsigned_integer> <exponent> (1) Accept any order <exponent> ::= E <sign> <unsigned_integer> E <unsigned integer> (1) <sign> ::= + - (1)</pre>

QUESTION 9.



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

QUESTION 10.



Question	Answer	Marks																				
6(a)	1 mark for each correct row <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	3
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 <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	2				
60	01	61	51	62	4E	63	01	60	50	64	52	62	02	63	53							

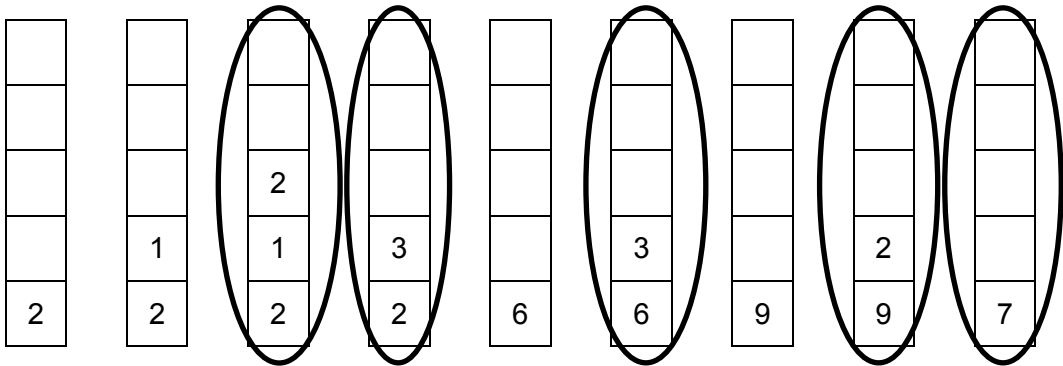


Question	Answer	
6(c)	1 mark per bullet point to max 2: - constructing parse tree // parsing - checking the table of tokens to ensure that the rules/syntax/grammar of the language are/is obeyed - producing an error report	
6(d)(i)	shortens execution time of program// time taken to execute whole program decreases	1
6(d)(ii)	1 mark for each of the following: - LDD 236 ADD 237 STO 512 ADD 238 STO 513 ADD 239 STO 514 - Remove line 4 LDD 236 correct lines 3 and 6 in original code - Remove line 5 ADD 237 correct lines 3 and 6 in original code - Remove line 8 and 9 LDD 236 and ADD 237 correct lines 7 and 11 in original code - Remove line 10 ADD 238 correct lines 7 and 11 in original code	5

QUESTION 11.

9608/32

Question	Answer	Marks																				
4(a)	1 mark per row <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>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> </tbody> </table>	Symbol	Token		Value	Type	Number1	60	Variable	Number2	61	Variable	Answer	62	Variable	10	63	Constant//Literal	0	64	Constant//Literal	3
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	2																				
4(c)(i)	(Code) Optimisation	1																				
4(c)(ii)	1 mark per bullet point: - LDD 236 - ADD 237 - ADD 238 - SUB 239 - STO 235 Copy the instructions - Remove line 4 STO 540 correct lines 3 and 6 in original code - Remove line 5 LDD 540 correct lines 3 and 6 in original code	3																				

Question	Answer	Marks
4(c)(iii)	1 mark per bullet point: • Code has fewer instructions/occupies less space in memory • shortens execution time of program // time taken to execute whole program decreases	2
4(d)	 1 mark ← 1 mark → 1 mark 1 mark no operators on the stack anywhere	4

QUESTION 12.



Question	Answer																												
4(a)	1 mark for 2 correct rows, 2 marks for 3 correct rows, 3 marks for 4 correct rows <table><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> First two cells given in question. 1 mark for next 3 cells <table><tr><td>61</td><td>51</td><td>62</td></tr></table> 1 mark for the remainder <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	2
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)	1 mark per bullet point • Removing the fourth line (LDD 238) ... • Changing operand for second ADD from 236 to 238 ... • ... First three lines and last line unchanged LDD 236 ADD 237 STO 236 ADD 238 STO 238	3																											
4(c)((ii)	1 mark per bullet point (max 2) • Optimisation means that the code will have fewer instructions • Optimised code occupies less space in memory • Fewer instructions reduces the execution time of the program	2																											

QUESTION 13.



Question	Answer	
4(a)	1 mark per bullet point (max 4) - Working from left to right in the expression - If element is a number PUSH that number onto the stack - If element is an operator then POP the first two numbers from stack perform that operation on those numbers - PUSH result back onto stack - End once the last item in the expression has been dealt with	
4(b)	1 mark per ring (not all stacks are shown) Do not allow operators in stacks Accept intermediate correct stack values 	4

Question	Answer	Marks

QUESTION 14.



Question	Answer	
2(a)(i)	35 is not a variable	
2(a)(ii)	:= is not an operator	
2(a)(iii)	9 is not a digit	1
2(b)	1 mark for each bullet point <code><operator> ::=</code> <code>==</code> <code>></code> <code><</code> <code><number> ::=</code> <code><digit><digit></code> <code><variable> ::=</code> <code><letter></code> <code> <letter><variable></code> <code><condition> ::=</code> <code><variable><operator><number></code> <code> <variable><operator><variable></code>	6

Question	Answer	Marks

QUESTION 15.



Question	Answer	
7(a)(i)	1 mark for each bullet point to max 2 Keyword table: • The reserved words used • The operators used • Their matching tokens	
7(a)(ii)	1 mark for each bullet point to max 2 Symbol table: • Identifier name used • ... the (data) type • ... role, e.g. variable, constant, array, procedure / scope • Location (marker) // value of constant	2
7(a)(iii)	1 mark per bullet point to max 2 • Keywords / operators are looked up (in the keyword table) • Keywords / operators are represented by tokens • Identifiers are looked up in (the symbol table) • Identifiers are converted to locations / addresses • Used to create a sequence of tokens (for the program)	2
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 max 2 • Redundant code removed // fewer instructions required • Program requires less memory / storage space • Code reorganised to make it more efficient • Program will complete task in a shorter time	2

Question	Answer	Marks