

UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS General Certificate of Education Advanced Subsidiary Level and Advanced Level

COMPUTING 9691/02

Paper 2 Practical Programming Techniques SPECIMEN MARK SCHEME

For Examination from 2011

2 hours

MAXIMUM MARK: 75



(a)	-Company logo -Areas shown for answering each question -Sensible first question e.g. 'How many bedrooms?' -Sensible second question e.g. 'What price range?' -Next button -Back button -No unreasonable blank areas (1 per -, max 6)			[6]
(b)	-String -Integer -Date/integer -Boolean (2 per -, max 8)	10-20 1/2 2/4/6/8 1		[8]
(c)	-Size of record = 14 to 31 (allow follow through) -Result multiplied by 100 -Divided by 1024 -10% overheads added -Answer 1.5KB to 3.33KB (1 per -, max 5)			[5]
(d)	-Input new record -Compare with record 1 -Repeat -If same name then -replace record/end -else read next record -until record = 100 -Read new record to new copy of file -Read records 1 to 99 to positions 2 to 100 in new file (1 per -, max 5)			[5]
(a)	-Small number makes input easy -Expected results are easy to work out -Information can be assumed to be representative of larger volumes -Large volume of data does not alter processing tasks (1 per -, max 2)			[2]
(b)	e.g. 10, 20, 40, 50 10, 10, 10, 10 10, 20, 41, 50 (3 per different test	Can it handle normal data Do repeated values cause a problem Can it handle real answers , max 9)	50, 10, 30 10, 10, 10 50, 10, 30.25	[9]

© UCLES 2008 9691/02/SM/11

3 (a) (i) -Algorithm assumes that the start values are 0/May contain values from previous processing (ii) FOR i = 1 TO 3 CANDIDATE TOTALS(i) = 0 Mark points: -Use of FOR loop with correct condition -Correct array name/subscript and 0 [2] (iii) -Algorithm reads each vote from the CANDIDATE TOTALS array -Decides whether the vote is for A,B,C -Keeps a running total of the votes for A,B,C -Outputs 0,0,0/meaningless output (1 per -, max 4) [4] (iv) -Line 2: operator is relational/comparative -returns a value true or false -Line 3: operator is arithmetic -changes the value in the stated variable [3] (1 per -, max 3) (v) OUTPUT CANDIDATE_TOTALS(1), CANDIDATE_TOTALS(2), CANDIDATE TOTALS(3) Mark points: -Use of array CANDIDATE TOTALS -Use of correct subscripts, in correct order [2] **(b) (i)** -1,7 -8,10 -11,12,13 C [4] (ii) -Check if all three are equal -output suitable message -Check if two are equal and one is different -IF the one different is the smallest -then output a message that there is a tie -else declare the winner

[4]

-Repeat three times (1 per -, max 4)

(a) (i) -Set of program instructions -Performs a specific task -Not a full program -Must be incorporated into a program to be used -Implies a machine code subroutine (1 per -, max 3) [3] (ii) -A function returns a single value to the calling program -A procedure can make the results of processing available to the main program [2] (iii) Mark points: -Initialise total -Diagram demonstrates order from left to right -Read data from file -extract hours -calculate half hours -extract minutes -calculate half hours (credit once at lower level) [4] (1 per -, max 4) **(b) (i)** -Modularisation -to make parts of the code shorter and easier to understand -Indentation -to show lines of the code that go together -Comments -to explain the logic of the code -Sensible variable names -so that the reader does not have to keep cross referencing with a table of names (1 per -, 2 pairs, max 4) [4] (ii) Note: Any language is acceptable, we are examining the logic and the use of the syntax and semantics of the language. -The function extracts a string from the file as input -The number of hours is correctly extracted from the string -The number of minutes is correctly extracted from the string -The number of hours and minutes are converted to a numeric format/integer -The number of half hours is correctly calculated -The number of half hours is returned as an integer

indentation/correct use of end statements or equivalents

-Appropriate sensible variable names used/code is annotated/shows evidence of

[7]

© UCLES 2008 9691/02/SM/11

(additional mark point, max 7)

(1 per -, max 6)