UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS

GCE Advanced Subsidiary Level and GCE Advanced Level

MARK SCHEME for the May/June 2011 question paper for the guidance of teachers

9691 COMPUTING

9691/22

Paper 2 (Written Paper), maximum raw mark 75

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes must be read in conjunction with the question papers and the report on the examination.

• Cambridge will not enter into discussions or correspondence in connection with these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2011 question papers for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level syllabuses and some Ordinary Level syllabuses.

Page 2	Mark Scheme: Teachers' version	Syllabus	Paper
	GCE AS/A LEVEL – May/June 2011	9691	22

1 (a)

Field Name	Data Type	Size of Field (bytes)
JobID	Integer	4
JobDescription	String / alphanumeric / text	20–50
Price	Currency / integer / real / decimal / float	8
ExpectedCompletionDate	Date / integer	8
Paid	Boolean	1

1 mark per box NOT variant (as a data type)

[10]

- **(b)** Result (e.g. 4+29+8+8+1=50 size of 1 record)
 - Multiplied by 200 (e.g. 10,000)
 - Add (10%) (e.g. 11,000)
 - Divided by 1024 (e.g. 11,000 ÷ 1024)
 - Result between 6.2 and 59.7KB (e.g. 10.7KB)

[5]

Page 3	Mark Scheme: Teachers' version	Syllabus	Paper
	GCE AS/A LEVEL – May/June 2011	9691	22

```
(c) e.g. Pascal
   TYPE JobRecord = RECORD
       JobID: Integer;
       JobDescription: String;
       Price: Currency;
       ExpectedCompletionDate: TDateTime;
       Paid: Boolean
   END;
   e.g. VB6
   Type JobRecord
       DIM JobID AS Integer
       DIM JobDescription AS String
      DIM Price AS Decimal
       DIM ExpectedCompletionDate AS Date
      DIM Paid AS Boolean
   END Type
   e.g. VB 2005
   STRUCTURE JobRecord
       DIM JobID AS Integer
       DIM JobDescription AS String
       DIM Price AS Decimal
       DIM ExpectedCompletionDate AS Date
      DIM Paid AS Boolean
   END STRUCTURE
   e.g. C#
   struct jobRecord
      public int jobID;
      public string jobDescription;
      public decimal price;
      public datetime expectedCompletionDate;
      public bool paid;
   }
   1 mark for heading
   1 mark for structure
   1 mark for all 5 fields correct
                                                                               [3]
(d) (i) -
          to check that data is reasonable / acceptable / follows rules
                                                                               [1]
          to check data is complete
          NOT correctness
   (ii) —
          range check explanation
          length check explanation
          format check explanation
       Max 2 marks
                                                                               [2]
       NOT presence check
```

(e) (JobID > 0) AND (JobID <= 1000)

Page 4	Mark Scheme: Teachers' version	Syllabus	Paper
	GCE AS/A LEVEL – May/June 2011	9691	22

Alternative answers:

(JobID > 0) AND (JobID < 1001) (JobID >= 1) AND (JobID <= 1000) (JobID >= 1) AND (JobID < 1001)

Correct brackets 1 mark; correct operator 1 mark

(Paid=True) OR (Paid=False)

Accept (Paid=yes) OR (Paid=no) (ignore speech marks)

Accept (Paid=1) OR (Paid=0)

Correct brackets 1 mark; correct operator 1 mark

[4]

(f) Any sensible + reason accepted e.g. 500 – valid data – within acceptable range / normal 1 – valid data – lower boundary included / extreme 1000 – valid data – upper boundary included / extreme – 1 – invalid data – below boundary 1001 – invalid data – above boundary

1 mark per data item, 1 mark per matching reason

[8]

Page 5	ge 5 Mark Scheme: Teachers' version		Paper
	GCE AS/A LEVEL – May/June 2011	9691	22

2 (a) (i)

Word	Count	Index	Word(Index)	Word(Index)= 'a'
banana				
	0			
		1		
			b	
				false
		2		
			a	
				true
	1			
		3		
			n	foloo
		4		false
		4	а	
			a	true
	2			
		5		
			n	
				false
		6		
			а	
				true
	3			

¹ mark for each correct column (except Word column)

(ii)

(11)				
Word	Count	Index	Word(Index)	Word(Index)= 'a'
Ant				
	0			
		1		
			Α	
				false
		2		
			n	
				false
		3		
			t	
				false

¹ mark for correct Count column

[3]

[6]

¹ mark for correct sequence

¹ mark for readable presentation

¹ mark for correct Word(Index)='a' column (need false only once after A)

¹ mark for Index column and Word(Index) column correct

raye	U	Wark Scheme. Teachers Version	Syllabus	rapei
		GCE AS/A LEVEL – May/June 2011	9691	22
1 n 1 n // 2	nark f mark f 2 marl 2 marl	d(Index) = 'a') OR (Word(Index) = 'A') for OR (allow lower case or) for separate decisions correct ks for If Uppercase(Word(Index))='A' ks for If Lowercase(Word(Index))='a' flect existing pseudocode style		[2]
(c) (i)	- - - -	meaningful variable names indentation / white space structured English good formatting (lower case, upper case) reserved words are capitalised / in capitals		[2]
(ii)	Ann	otation / comments		[1]
/!!!				
(iii)	_	to make it easier to find / correct errors to make it easier to modify the program / maintenance		[2]
(d) (i)	_	numeric/binary (code where each character has a unic	jue value)	[1]
(ii)	- - -	letter a-z have increasing ASCII codes Each character's ASCII value is compared the character with the smaller value is the first characte the larger value is the second character / (letters are s		er with [3]
(iii)	_	characters are compared in turn from left hand side / start of each word until two characters are different the lower code value determines the first word if 2 words are the same when one ends this is the first word		[4]

Mark Scheme: Teachers' version

Syllabus

Paper

Page 6

Page 7Mark Scheme: Teachers' versionSyllabusPaperGCE AS/A LEVEL – May/June 2011969122

3 (a) 0 (zero) [1]

```
(b) e.g. Pascal
   VAR Letter: ARRAY [1..26] OF Integer;
   FOR I := 1 TO 26
       DO
          Letter[i] := 0;
   Alternative:
   VAR Letter: ARRAY ['a'..'z'] OF Integer;
   FOR 1 := 'a' TO 'z'
       DO
          Letter[1] := 0;
   e.g. VB 2005
   DIM Letter (26) AS Integer
   FOR i = 1 TO 26
       Letter(i) = 0
   NEXT
   e.g. C#
   string[] letter = new string[26]
   for (int i = 1; i \le 26; i++)
       letter[i] = 0
    }
   1 mark for correct declaration range
   1 mark for correct data type
    1 mark for loop to address full range of array
```

1 mark for correct assignment

[4]

Page 8			Mark Scheme: Teachers' version	Syllabus	Paper
			GCE AS/A LEVEL – May/June 2011	9691	22
(c)	e.g.				
	Thi	sLe	tterIndex := ASCII(ThisIetter) - ASCII(Ne/)	1.	
	T.et	ter	ASCII(ThisLetter)-ASCII('a') + [ThisLetterIndex] :=	1;	
	шес	LCI	Letter[ThisLetterIndex] +	1:	
				-,	
			<pre>ve: (if character range used for array index) [ThisLetter] := Letter[ThisLetter] + 1;</pre>		
	A (1	VB ·	2005		
			tterIndex = ASC(ThisLetter)-ASC("a") + 1		
			(ThisLetterIndex) =		
			Letter(ThisLetterIndex) +	1	
		. C#			
			tterIndex = asc(thisLetter) - asc('a') +	⊥;	
	Tet	ter	<pre>[thisLetterIndex] =</pre>	1.	
			recter [thronetterrindex]	± ,	
	1 m	ark f	or finding correct array element		
			or incrementing running total correctly		
	1 m	ark f	or correct overall logic		[1]
4 (-)	/: \	4			F41
4 (a)	(i)	1			[1]
	(ii)	6			[1]
	(,	Ū			[.]
(b)	(i)	_	cannot end		
		_	infinite loop		
		_	produces error message (heap/stack overflow) / 'crash	l´	[2]
	(ii)	_	Before second line extra code needs to be added		
	(11)	_	if n<1 (OR if n<0)		
		_	then error (or equivalent)		[2]
			(
(c)	FUN		ON prod(n)		
		x ←			
			i ← 1 TO n		
		NEX	x ← x * i		
			1		
		_	d ← x CTION // RETURN		
		- 1 011	, , 101 O1W		
	1 m	ark f	or initialisation		
	1 m	ark f	or correct loop from 1 to n		
	1 m	ark f	or multiplying current value by i		
	1 m	ark f	or assigning return value		[4]