

MARK SCHEME for the May/June 2014 series

9691 COMPUTING

9691/23

Paper 21 (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 should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2014 series for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level components and some Ordinary Level components.

1 (a) (i)

Identifier	Data type	Description
Ounces	INTEGER	Variable used as control variable in FOR loop
Grams	REAL/Float/single/decimal/double	Variable used for storing result of conversion calculation

1 mark for each data type

[2]

```
(ii) OUTPUT " Conversion Table"
OUTPUT "Ounces          Grams"
FOR Ounces ← 1 TO 16
  Grams ← Ounces * 28.35
  Grams ← ROUND(Grams)
  OUTPUT Ounces, "          ", Grams
ENDFOR // NEXT Ounces
```

1 mark for each correct line

[4]

```
(b) (i) 20 DIV 6 = 3
20 MOD 6 = 2
```

[2]

```
(ii) FUNCTION CalculateNumberOfBoxes (NumberOfEggs : INTEGER)
      RETURNS INTEGER } 1 mark
  DECLARE NumberOfBoxes : INTEGER } 1 mark
  NumberOfBoxes ← NumberOfEggs DIV 6 } 1 mark
  IF NumberOfEggs MOD 6 > 0 } // 6 >= 1 } 1 mark
  THEN
    NumberOfBoxes ← NumberOfBoxes + 1 } 1 mark
  ENDIF
  RETURN NumberOfBoxes
ENDFUNCTION
```

[5]

(c) A function always returns a value. A procedure may or may not return one or more values [1]

- (d) – indentation
 – meaningful identifier names
 – annotation
 – parameters
 – local variables

[max 4]

Page 3	Mark Scheme	Syllabus
	GCE AS/A LEVEL – May/June 2014	9691

- 2 (a) (i) mark as follows:
 1 mark for labels/explanations, heading, customer name and telephone number
 1 mark for date choice using calendar or showing required format
 1 mark for type of cake as drop-down list or similar
 1 mark for delivery required as radio buttons or similar [4]
- (ii) suitable explanation of a feature (drop-down box, radio button, etc.) [1]

(b) (i)

Field Name	Data Type	Field size (bytes)	
CustomerName	String	24 (approx.)	} 1 mark
TelephoneNumber	String	13 (approx.)	
DateReady	Date/string/real	8	1 mark
CakeType	Char	1	2 marks
Price	Real/float/single(4)/double(8)/currency(8)/decimal(16)		1 mark
ToBeDelivered	Boolean	1	2 marks

- (ii) 1 mark for record header [8]
 1 mark for record end
 1 mark for every three fields correct [4]

Pascal

```

TYPE CakeOrder = RECORD
    CustomerName: String[24];
    TelephoneNumber: String[13];
    DateReady: TDateTime;
    CakeType: Char;
    Price: Currency;
    ToBeDelivered: Boolean;
END;
```

VB6

```

TYPE CakeOrder
    CustomerName AS String
    TelephoneNumber AS String
    DateReady AS Date
    CakeType AS Char
    Price AS Currency
    ToBeDelivered AS Boolean
END TYPE
```

Page 4	Mark Scheme	Syllabus
	GCE AS/A LEVEL – May/June 2014	9691

VB.NET

```
STRUCTURE CakeOrder
    DIM CustomerName AS String
    DIM TelephoneNumber AS String
    DIM DateReady AS Date
    DIM CakeType AS Char
    DIM Price AS Currency
    DIM ToBeDelivered AS Boolean
END STRUCTURE
```

Python

```
class CakeOrder :
    def __init__(self) :
        customerName = ""
        telephoneNumber = ""
        dateReady = "" // Datetime.datetime.now()
        cakeType = ""
        price = 0.0
        toBeDelivered = FALSE
```

Page 5	Mark Scheme	Syllabus
	GCE AS/A LEVEL – May/June 2014	9691

- 3 1 mark for each flowchart box correctly “translated”
 1 mark for REPEAT
 1 mark for each ELSE& matching ENDIF

[max 10]

Pascal

```

Randomize; 1
MyNumber := RANDOM(100) + 1; 1
EndGame := FALSE; 1
NumberOfGuesses := 0; 1
WriteLn('Guess my number. Type 0 (zero) to end game'); 1
REPEAT 1
  ReadLn(Guess); 1
  IF Guess = 0 1
    THEN 1
      BEGIN 1
        WriteLn('You gave up after ', NumberOfGuesses); 1
        EndGame := TRUE; 1
      END 1
    ELSE 1
      BEGIN 1
        NumberOfGuesses := NumberOfGuesses + 1; 1
        IF Guess = MyNumber 1
          THEN 1
            BEGIN 1
              WriteLn('Correct - you took ', NumberOfGuesses, 1
                'to guess my number'); 1
              EndGame := TRUE; 1
            END 1
          ELSE 1
            IF Guess > MyNumber 1
              THEN 1
                WriteLn('Too high - try again') 1
              ELSE 1
                WriteLn('Too low - try again') 1
            END 1
          END; 1
        UNTIL EndGame = TRUE; 1
  
```

Page 6	Mark Scheme	Syllabus
	GCE AS/A LEVEL – May/June 2014	9691

www.PapaCambridge.com

QBASIC

```

MyNumber = RND(100)
EndGame = FALSE
NumberOfGuesses = 0
PRINT("Guess my number. Type 0 (zero) to end game")
REPEAT
    INPUT Guess
    IF Guess = 0 THEN
        PRINT("You gave up after ", NumberOfGuesses)
        EndGame = TRUE
    ELSE
        NumberOfGuesses = NumberOfGuesses + 1
        IF Guess = MyNumber THEN
            PRINT("Correct - you took ", NumberOfGuesses,
                "to guess my number")
            EndGame = TRUE
        ELSE
            IF Guess > MyNumber THEN
                PRINT("Too high - try again")
            ELSE
                PRINT("Too low - try again")
            ENDIF
        ENDIF
    ENDIF
UNTIL EndGame = TRUE

```

VB6

```

Randomize
MyNumber = INT(RND * 100 + 1)
EndGame = FALSE
NumberOfGuesses = 0
MsgBox("Guess my number. Type 0 (zero) to end game")
DO WHILE NOT EndGame = TRUE
    Guess = INPUTBOX("")
    IF Guess = 0 THEN
        MsgBox("You gave up after ", NumberOfGuesses)
        EndGame = TRUE
    ELSE
        NumberOfGuesses = NumberOfGuesses + 1
        IF Guess = MyNumber THEN
            MsgBox("Correct - you took ", NumberOfGuesses,
                "to guess my number")
            EndGame = TRUE
        ELSE
            IF Guess > MyNumber THEN
                MsgBox("Too high - try again")
            ELSE
                MsgBox("Too low - try again")
            ENDIF
        ENDIF
    ENDIF
LOOP

```

Page 7	Mark Scheme	Syllabus
	GCE AS/A LEVEL – May/June 2014	9691

www.PapaCambridge.com

VB.NET

Alternative to get a random number between 1 and 100:

```
DIM Random AS NEW Random()
```

```
MyNumber = Random.Next(1,100)
```

```
MyNumber = CINT(INT((100 * RND()) + 1))           1
EndGame = FALSE                                  1
NumberOfGuesses = 0                              1
Console.WriteLine("Guess my number. Type 0 (zero) to end game") 1
DO                                                1
    Guess = Console.ReadLine()                   1
    IF Guess = 0 THEN                             1
        Console.WriteLine("You gave up after ", NumberOfGuesses) 1
        EndGame = TRUE                           1
    ELSE                                          1
        NumberOfGuesses = NumberOfGuesses + 1    1
        IF Guess = MyNumber THEN                 1
            Console.WriteLine("Correct - you took ", 1
                NumberOfGuesses, "to guess my number")
            EndGame = TRUE                       1
        ELSE                                     1
            IF Guess > MyNumber THEN              1
                Console.WriteLine("Too high - try again") 1
            ELSE                                  1
                Console.WriteLine("Too low - try again") 1
            ENDFIF
        ENDFIF
    ENDFIF
LOOP UNTIL EndGame = TRUE                        1
```

Python

```
Random.seed()                                   1
MyNumber = random.randint(1, 100)              1
EndGame = FALSE                                1
NumberOfGuesses = 0                            1
print("Guess my number. Type 0 (zero) to end game") 1
WHILE EndGame != TRUE :                        1
    Guess = int(input())                        1
    IF Guess == 0 :                             1
        print("You gave up after ", NumberOfGuesses) 1
        EndGame = TRUE                         1
    ELSE:                                       1
        NumberOfGuesses = NumberOfGuesses + 1    1
        IF Guess == MyNumber :                 1
            print("Correct - you took ", NumberOfGuesses, 1
                "to guess my number")
            EndGame = TRUE                     1
        ELSE:                                   1
            IF Guess > MyNumber :              1
                print("Too high - try again") 1
            ELSE:                               1
                print("Too low - try again") 1
```

- 4 (a) – GUI
 – touch screen / WIMP / etc.
 – virtual keyboard / pop-up menu
- (b) (i) – array
 – 2-dimensional // 9×9 [2]
- (ii) `Puzzle[1,3] ← 5` [2]
1 mark for correct index, 1 mark for remainder correct
- (c) `(Entry>='1') AND (Entry<='9')` [2]
1 mark for first bracket & AND, 1 mark for second bracket.
- (d) store the x-y co-ordinates of each entry in a serial file / (linked list) / on a stack so they can be accessed in reverse order [max 4]

5 (i)

x	ThisValue	y	List[y]	(List[y] > ThisValue) AND (y > 0)	List			
					[1]	[2]	[3]	[4]
–	–	–	–	–	56	23	67	12
2	23	1	56	TRUE		56		
		0		FALSE	23			
3	67	2	56	FALSE			(67)	
4	12	3	67	TRUE				67
		2	56	TRUE			56	
		1	23	TRUE		23		
		0		FALSE	12			

- 1 mark for each column correct [9]
- (ii) (insertion) sort // ascending order [1]