Chapter 15 Software development: Answers to coursebook questions and tasks

Syllabus sections covered: 2.1 (2.1.3, 2.1.4) and 2.4

Task 15.01

Students use facilities available in the editor available to them (depending on college software).

Task 15.02

```
FUNCTION ConvertFromHex(HexString : STRING) RETURNS INTEGER
   DECLARE ValueSoFar, HexValue, HexLength, i : INTEGER
   DECLARE HexDigit : CHAR
   ValueSoFar ← 0
   HexLength ← Length(HexString)
   FOR i ← 1 TO HexLength
      begin
         HexDigit ← HexString[i]
         CASE OF HexDigit
            'A': HexValue ← 10
            'B': HexValue ← 11
            'C': HexValue ← 12
            'D': HexValue ← 13
            'E': HexValue ← 14
            'F': HexValue ← 15
            OTHERWISE HexValue ←
StringToInt(HexDigit)//convert char to integer
         ENDCASE
         ValueSoFar ← ValueSoFar * 16 + HexValue
      ENDFOR
   RETURN ValueSoFar
ENDFUNCTION
```

HexString	HexLength	i	HexDigit	HexValue	ValueSoFar
'A5'	**2**				**0**
		1	'A'	**10**	**10**
		2	'5'	**5**	**165**

Task 15.03

```
INPUT BinaryString
StringLength ← Length(BinaryString)
FOR i ← 1 TO StringLength
Bit ← BinaryString[i]
```

```
BitValue ← IntegerValue(Bit) // convert string to integer
DenaryValue ← DenaryValue + 2 + BitValue
ENDFOR
```

BinaryString	StringLength	i	Bit	BitValue	DenaryValue
'101'	3	1	'1'	1	3
		2	'0'	0	5
		3	'1'	1	8

Errors:

- DenaryValue should be initialised to 0 before the loop
- The + highlighted in the code above should be a *

Task 15.04

```
01 CALL InitialiseBoard
02 CALL SetUpGame
03 CALL OutputBoard
04 WHILE GameFinished = FALSE
05    CALL ThisPlayerMakesMove
06    CALL OutputBoard
07    CALL CheckIfThisPlayerHasWon
08    IF GameFinished = FALSE
09       THEN
10          CALL SwapThisPlayer
11    ENDIF
12 ENDWHILE
```

The procedure ThisPlayerMakesMove needs to be changed to:

```
PROCEDURE ThisPlayerMakesMove
   IF ThisPlayer = 'X'
      THEN
         ValidColumn ← ComputerChoosesColumn
      ELSE
         ValidColumn ← ThisPlayerChoosesColumn
   ENDIF
   ValidRow ← FindNextFreePositionInColumn
   Board[ValidRow, ValidColumn] ← ThisPlayer
ENDPROCEDURE

FUNCTION ComputerChoosesColumn // returns a valid column
number
   REPEAT
      ColumnNumber ← Random(1,7) // a random number between 1
and 7
```

```
   UNTIL ColumnNumberValid = TRUE // check whether the column
number is valid
   RETURN ColumnNumber
ENDFUNCTION
```

Exam style questions

1

```
FUNCTION Binary(Number : INTEGER) : STRING
DECLARE BinaryString : STRING
DECLARE PlaceValue : INTEGER
   BinaryString ← ''  // empty string
   PlaceValue ← 8
   REPEAT
      IF Number >= PlaceValue
         THEN
            BinaryString ← BinaryString & '1' // &concatenates
two strings
            Number ← Number - PlaceValue
         ELSE
            BinaryString ← BinaryString & '0'
      ENDIF
      PlaceValue ← PlaceValue DIV 2
   UNTIL Number = 0
   RETURN BinaryString
ENDFUNCTION
```

a Dry-run the function call <code>Binary(11)</code> by completing the given trace table.

Number	BinaryString	PlaceValue	Number>=PlaceValue
	''	8	
11	'1'	4	TRUE
3	'10'	2	FALSE
1	'101'	1	TRUE
0	'1011'	0	TRUE

The return value is 1011

b i Dry-run the function call <code>Binary(10)</code> by completing the given trace table.

Number	BinaryString	PlaceValue	Number>=PlaceValue
	''	8	
10	'1'	4	TRUE
2	'10'	2	FALSE
0	'101'	1	TRUE

The return value is 101.

ii Number should be PlaceValue in code above.