**BCS THE CHARTERED INSTITUTE FOR IT**

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 4 Certificate in IT

**SOFTWARE DEVELOPMENT**

Wednesday 20th April 2011 - Morning
Time: TWO hours

Section A and Section B each carry 50% of the marks. You are advised to spend about 1 hour on Section A (30 minutes per question) and 1 hour on Section B (12 minutes per question).

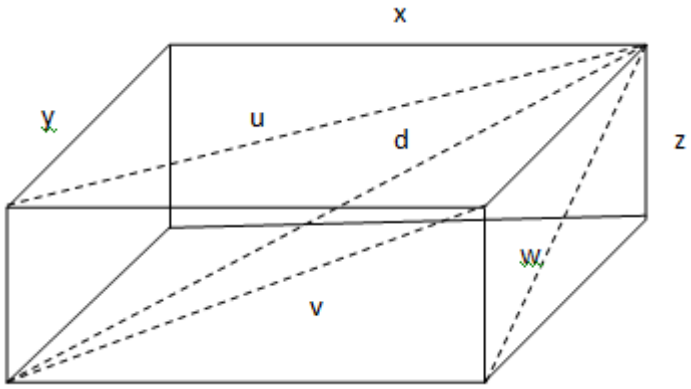
**Answer the Section A questions you attempt in Answer Book A**
**Answer the Section B questions you attempt in Answer Book B**

The marks given in brackets are **indicative** of the weight given to each part of the question.

Calculators are **NOT** allowed in this examination.

**SECTION A**
Answer TWO questions out of FOUR in Answer Book A. Each question carries 30 marks

A1. A regular cuboid has sides (*x, y, z*), face diagonals (*u, v, w*) and body diagonal (*d*) as shown in the diagram below.




No cuboid is known to exist where all the values (*x, y, z, u, v, w, d*) are integers, but some combinations of integer values of *x, y, z* yield values of *u, v, w, d* which are close to integers.

a) Write a real function *side* which takes as parameters two integer values *base* and *height* and which returns the function value *side* calculated by Pythagoras' theorem.

*side* = SQRT($base^2$ + $height^2$)

**(2 marks)**

b) Calculate the face diagonals *u, v, w* and body diagonal *d* using the function *side* and with the relevant base and height variables.

**(4 marks)**

c) Write a real function *diff* which takes one real value *diagonal_value* and returns the difference of this value from its nearest integer; that is,

*diff* = absolute value of (*diagonal_value – nearest integer to diagonal_value)*

**Turn over]**

Assume the values of *diff* are awarded points according to the table given below:

| *diff* is: | Points awarded |
|---|---|
| less than or equal to 0.00001 | 50 |
| greater than 0.00001 and less than or equal to 0.0001 | 20 |
| greater than 0.0001 and less than or equal to 0.001 | 10 |
| greater than 0.001 and less than or equal to 0.01 | 8 |
| greater than 0.01 and less than or equal to 0.5 | 4 |

Write an integer function, *total_pts*, which has four real parameters representing the *diff* of *u, v, w*, and *d* and returns the total points awarded.

**(8 marks)**

d) When the maximum length of any side is *max_length*, and the minimum difference is given by *min_diff*, write the statements which

i) calculate all the diagonals using an appropriate loop structure
ii) print out the values *u, v, w, d* when the *diff* of all of these values is less than *min_diff*.

**(6 marks)**

e) For each *x, y, z* integer combination associated with a cuboid of *max_length*, choose an appropriate data structure to hold *x, y, z, u, v, w, d* and *total_pts*. Write a procedure *mysort()* with appropriate parameters to sort the data structure into descending order of points values.

**(10 marks)**

A2. The following code is an iterative process for calculating the $n^{th}$ member of a Fibonacci series, as shown in the table below:

| Member (n) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Fib(n) | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 |

```
Line                code
  1   WRITELN( “input which member of series”)
  2   READ(n)
  3   p ← 2
  4   prev1 ← 1
  5   prev2 ← 1
  6   WHILE p IS LESS THAN n DO
  7   BEGIN
  8   term ← prev1  +  prev2
  9   prev2 ← prev1
 10   prev1 ← term
 11   p  ← p  +  1
 12   END
 13   WRITELN (“term  =”, term)
```

a) Dry run this code with n input as 4.

**(18 marks)**

b) Write code for a function to implement a recursive method for this process.

**(6 marks)**

c) Do you prefer the iterative style used in part a) or the recursive style used in part b)? Give reasons.

**(6 marks)**

A3. The array **x** has been initialised as follows

| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | 1 | 9 | 5 | 0 | 8 | 6 | 3 | 5 | 1 | 0 | 2 | 9 |

The function ***a*** in the code below is going to be executed with parameter ***b*** set to 1 and parameter ***c*** set to 10. [*You can choose to follow either version of the code*]

a) Trace the call of the function ***a(1,10)*** and show clearly the results of the call.

**(8 marks)**

| | Version 1 (Pascal) | Version 2 (C) |
|---|---|---|
| 1 | float a(int b, int c){ | FUNCTION a(b, c : INTEGER):REAL; |
| 2 | int d,e; | VAR d, e : INTEGER; |
| 3 | /* begin function */ | BEGIN |
| 4 | d = 0; | d := 0; |
| 5 | e = b; | e := b; |
| 6 | while(e <= c) | WHILE e <= c DO |
| 7 | { | BEGIN |
| 8 | d += x[e]; | d := d + x[e]; |
| 9 | e++; | e := e + 1; |
| 10 | } | END; |
| 11 | return( d/(b-c+1) ) | a := d/(b-c+1) |
| 12 | } | END |

b) Write a brief summary of what the function does.

**(6 marks)**

c) Decide on better names for the identifiers (the function name, its parameters and the variables) and rewrite the code [either version 1 or version 2] using your new names and including suitable comments.

**(10 marks)**

d) Rewrite lines 5 to 10 [of either version 1 or version 2] using a for-loop instead of a while-loop.

**(6 marks)**

A4. Base your answers on either program version 1 or version 2 below.

| Line No. | Version 1 | Version 2 |
|---|---|---|
| 1: | /* program compareTest */ | program compareTest; |
| 2: | **int i** ; | **var i : integer** ; |
| 3: | **char** c **;** | c : **char** **;** |
| 4: | char **compare**(int p1,**p2**) | function **compare**( p1, **p2** : integer ) : char ; |
| 5: | { | begin |
| 6: | if( p1 **==** p2 ) | if p1 **=** p2 then |
| 7: | return( **'='** ); | compare := **'='** |
| 8: | else if( **p1 > p2** ) | else if **p1 > p2** then |
| 9: | return('>'); | compare := '>' |
| 10: | **else** | **else** |
| 11: | return('<'); | compare := '<' |
| 12: | } **/* compare */** | end **/* compare */** |
| 13: | void main(){ | begin |
| 14: | **i** **=** **2** ; | **i** **:=** **2** ; |
| 15: | **c** = compare( **i-1** , 1 **+** 1); | **c** := compare( **i-1** , 1 **+** 1); |
| 16: | **i = 3** ; | **i := 3** ; |
| 17: | c = **compare(i, 1+1)** ; | c := **compare(i, 1+1)** ; |
| 18: | } | end. |

| Key: description | | |
|---|---|---|
| A: character constant<br>B: integer constant<br>C: assignment operator<br>D: arithmetic operator<br>E: equality operator<br>F: punctuation character | G: character identifier<br>H: integer identifier<br>I: type identifier<br>J: function identifier<br>K: reserved word<br>L: comment | M: variable declaration<br>N: boolean (logical) expression<br>O: formal parameter<br>P: actual parameter<br>Q: function call<br>R: assignment statement |

a) Copy and complete the following answer table. (You need only copy out either the "Text in version 1" or the "Text in Version 2" column.) Then for each highlighted part in the programs above, decide what key it should have according to the table above.
**(18*1 mark)**

| Line | Text in version 1 | Text in version 2 | Key letter for highlighted code |
|---|---|---|---|
| 2 | int i ; | var : i integer ; | |
| 3 | char | char | |
| 3 | ; | ; | |
| 4 | compare | compare | |
| 4 | p2 | p2 | |
| 6 | == | = | |
| 7 | '=' | '=' | |
| 8 | p1 > p2 | p1 > p2 | |
| 10 | else | else | |
| 12 | /* ... */ | /* ... */ | |
| 14 | i | i | |
| 14 | = | := | |
| 14 | 2 | 2 | |
| 15 | c | c | |
| 15 | i-1 | i-1 | |
| 15 | + | + | |
| 16 | i = 3 | i := 3 | |
| 17 | compare(...) | compare(...) | |

b) Find and copy out one example of each of the following:

i) arithmetic expression
ii) conditional statement
iii) function declaration

**(3 x 4 marks)**

## SECTION B

Answer FIVE questions out of EIGHT in Answer Book B. Each question carries 12 marks.

B5. a) Write a definition for a RECORD (or STRUCTURE) to hold information about a property for sale. The details to be held are: price (up to 6 digits whole number), type (house, flat or bungalow), garage (yes/no), number of bedrooms (small integer), address (3 lines, each of 20 characters).

**(4 marks)**

b) Write a FILE declaration for a file called 'propfile' to hold a sequence of these records.

Write code which requests an upper price limit, and which then searches the file for all properties below this limit with a garage and 3 or more bedrooms. The address of each property satisfying these conditions is to be printed. If no property is found, the message 'no such property available' should be printed.

**(8 marks)**

B6. a) i) Many programming languages contain a statement which allows one action to be selected from a group of actions based on a value held in a variable. Illustrate this with a statement from a language of your choice.

**(2 marks)**

ii) Give its equivalent in terms of IF…THEN…ELSE statements.

**(2 marks)**

b) A Government wishes to implement a Wealth Tax according to the Assets in the following table:

| Assets (pounds) | Assets integer divided by 100,000 | Tax (percentage) |
|---|---|---|
| Less than 100,000 | 0 | 0 |
| 100,000 to 399,999 | 1, 2, 3 | 0.05 |
| 400,000 to 699,999 | 4, 5, 6 | 0.1 |
| 700,000 to 999,999 | 7, 8, 9 | 0.15 |
| Greater than 1,000,000 | 10 | 0.2 |

i) Write an expression relating Assets (pounds) to Tax (percentage)**.**

**(3 marks)**

ii) Incorporate your expression in a program or pseudocode which receives as input a particular value for assets (for example 587,000 pounds) and which outputs the corresponding value of the tax payable in pounds.

You should not write out again your expression for i) but indicate where it belongs in your program.

**(5 marks)**

B7. Conversion of a decimal number to its binary equivalent is carried out as follows:

The decimal number is successively divided by 2 and the remainders are stored. The remainders in reverse order form the binary number; the last remainder being the most significant digit. Thus 13 (decimal) = 1101 (binary)

| Process | Quotient | remainder | |
|---|---|---|---|
| ÷2 | 13 | 1 | (least significant bit) |
| ÷2 | 6 | 0 | |
| ÷2 | 3 | 1 | |
| ÷2 | 1 | 1 | (most significant bit) |
| | 0 | | |

Write pseudocode or a program to input a decimal number and output its binary equivalent. Only integer decimal numbers are to be considered.

**(12 marks)**

B8. a) i) Show diagrammatically a linked list of 4 nodes, where each node consists of one pointer and one integer data item.

**(2 marks)**

ii) Give a declaration of this data structure in a programming language of your choice. State the language you are using.

**(2 marks)**

iii) State, with a reason, an application for which this particular data structure is useful.

**(2 marks)**

b) i) Show diagrammatically a one-dimensional array with one subscript.

**(2 marks)**

ii) Give a declaration of this data structure using the same programming language as in part a).

**(2 marks)**

iii) State, with a reason, an application for which this particular data structure is useful.

**(2 marks)**

B9. One of the standard data structures used in programming is the *stack* with its distinctive operations of *push* (place a new value on the *stack*) and *pop* (remove the top value from the *stack*). Using an array as the basic storage for the *stack*, write code in a language of your choice to implement the operations *push* and *pop*.

**(12 marks)**

B10. Give one reason why someone might prefer:

a) a compiler *rather than* an interpreter
b) a command line operating system *rather than* a WIMP operating system
c) a 4GL programming language *rather than* a 3GL language
d) a phased implementation of a new system *rather than* a parallel implementation
e) a low-level language *rather than* a high-level language
f) an array *rather than* a linked list

**(6 x 2 marks)**

*[Note: It is expected that one well-chosen sentence per part will be sufficient as an answer]*

B11. While working on a prototype of a new system you require a web page for a new user to register. The page should contain:

i) a way for the new user to enter their name
ii) a way to enter the range containing the age of the user (0-9, 10-19, 20-29, etc)
iii) a way to enter the gender of the user
iv) a way for the user to show which by combination of methods (phone, email, letter) they are willing to be contacted
v) a way to announce that the data is ready to be received by the system

a) Draw a diagram to illustrate your idea for the interface. The page should contain a form with appropriate elements to handle the various types on information.

(**5 x 2 marks)**

b) Pick one of the pieces of information that is to be entered, state an alternative method that could have been used to enter that information and state why it would be inferior to the method used in a).

**(2 marks)**

B12. Choose program version 1 or 2 below and then find occurrences of the following 6 errors. For each error, you should give the line number and an explanation of the error. In addition, you should state whether each error will be discovered at compile time or run-time.

**(6 x 2 marks)**

i) identifier not declared
ii) type error - index not allowed
iii) array index out of bounds
iv) syntax error
v) variable required
vi) type error - invalid type

| line | Version 1 (Pascal) | Version 2 (C) |
|---|---|---|
| 1 | program p; | void main(); |
| 2 | var a : array[0..9]of real; | float a[10]; |
| 3 | b, c : integer; | int b, c; |
| 4 | begin | { |
| 5 | b := 1; | b = 1; |
| 6 | 2 := b; | 2 = b; |
| 7 | c := b[1]; | c = b[1]; |
| 8 | d := b+c; | d = i+j; |
| 9 | if b > 1 b := b - 1 ; | if ( b > 1 b = b - 1 ; |
| 10 | a[0] := b; | a[0]=b; |
| 11 | if a[0] then c := 1; | if ( a[0] ) c = 1; |
| 12 | for b:=0 step 1 to 9 do | for( b=0; b<=9; b++ ) |
| 13 | a[b+1]:=b; | a[b+1] = b; |
| 14 | end. | } |

**** END OF EXAM ****