

The rule on answers. Where pseudo-code is accepted you may answer in any suitable format, as long as it is clear, consistent and unambiguous. Questions use this style and index arrays and strings from 0 unless they say otherwise.

Comments

```
# comment
# comment and so on
```

Multi-line comments repeat # on every line.

Assignment

```
a ← 3
b ← a + 1
c ← 'Hello'
```

Constants

```
CONSTANT PI ← 3.141
CONSTANT CLASS_SIZE ← 23
```

Names of constants are always written in capitals.

Input

```
a ← USERINPUT
```

Output

```
OUTPUT a
OUTPUT a, g
```

Integer division and remainder

```
9 DIV 5 # evaluates to 1
9 MOD 5 # evaluates to 4
```

Comparison

```
= ≠ < ≤ > ≥
```

Joining strings

```
'computer' + 'science'
```

Evaluates to 'computerscience'.

IF... ELSE

```
IF (a MOD 2) = 0 THEN
    OUTPUT 'even'
ELSE
    OUTPUT 'odd'
ENDIF
```

Else-if chains

```
IF (a MOD 4) = 0 THEN
    OUTPUT 'multiple of 4'
ELSE IF (a MOD 4) = 1 THEN
    OUTPUT 'leaves a remainder of 1'
ELSE
    OUTPUT 'leaves a remainder of 3'
ENDIF
```

Count-controlled loop

```
FOR a ← 1 TO 3
    OUTPUT a
ENDFOR
# will output 1, 2, 3
```

The loop variable does not exist after the loop has finished.

Pre-condition loop (WHILE)

```
WHILE a < 4
    OUTPUT a
    a ← a + 1
ENDWHILE
```

Post-condition loop

```
REPEAT
    OUTPUT a
    a ← a + 1
UNTIL a = 4
```

Declaring a 1D array

```
primes ← [2, 3, 5, 7, 11, 13]
```

Exam questions index from 0 unless they say otherwise.

2D arrays

```
table ← [[1, 2],[2, 4],[3, 6],[4, 8]]
table[3][1] # evaluates to 8
table[3][1] ← 16
```

Records

```
RECORD Car
    make : String
    model : String
    reg : String
    price : Real
    noOfDoors : Integer
ENDRECORD

myCar ← Car('Ford', 'Focus', 'DX17
GYT', 1399.99, 5)
myCar.model ← 'Fiesta'
```

Defining a procedure

```
SUBROUTINE showAdd(a, b)
    result ← a + b
    OUTPUT result
ENDSUBROUTINE
```

A subroutine with no RETURN is a procedure.

Calling a procedure

```
showAdd(2, 3)
```

Defining a function

```
SUBROUTINE add(a, b)
    result ← a + b
    RETURN result
ENDSUBROUTINE
```

Calling a function

```
answer ← add(2, 3) * 6
```

String length

```
LEN('computer science') # evaluates
to 16
```

Substrings

```
SUBSTRING(2, 9, 'computer science')
# evaluates to 'mputer s'
```

Start position, end position (both included), then the string. Indexing from 0.

Converting between types

```
STRING_TO_INT('16')
STRING_TO_REAL('16.3')
INT_TO_STRING(16)
REAL_TO_STRING(16.3)
```

Not defined in this guide

Declaring a variable and its type; CASE / switch; By value and by reference; Opening and reading a text file; Writing to a text file; Class and constructor.

Examiners have penalised

- Pseudo-code keywords inside a program-code answer
- Spaces inside identifiers
- / where the algorithm says DIV
- Quotes in an exact-output answer
- Replacing a record instead of updating a field
- Loops that run once too often