

Revision Sheet

A-Level Computer Science · Unit 11 Programming

9 vocabulary items · 41 marks of exam questions (4 questions)

Part 1 · Vocabulary review (9 items). The terms this unit is examined on most. Mark it yourself from the answer key.

Section A · Chinese to English

Write the English word.

1.

[1]
2.

[1]
3.

[1]
4.

[1]
5.

[1]
6.

[1]
7.

[1]
8.

[1]
9.

[1]

Part 2 · Exam questions (41 marks, 4 questions) begins on the next page. They are ordered easy to difficult and none of them has been set before.

10 A string function `Compare()` will compare two strings.

The function will:

- take **four** parameters:
 - two strings, `String1` and `String2`
 - a character, `Position`, to indicate whether `String1` will be compared with the start ('s'), or the end ('e') of `String2`
 - a Boolean, `CaseMatters`, to indicate whether an upper case character and lower case character (for example, 'A' and 'a') are regarded as different (`TRUE`), or as the same (`FALSE`)
- return `FALSE` if `String2` has fewer characters than `String1`
- return `TRUE` if the comparison is true, otherwise return `FALSE`

For example:

Parameter				Return value
String1	String2	CaseMatters	Position	
"Cat"	"Catalogue"	TRUE	's'	TRUE
"CAT"	"Catalogue"	TRUE	's'	FALSE
"CAT"	"Catalogue"	FALSE	's'	TRUE
"Cat"	"Catalogue"	TRUE	'e'	FALSE
"GUE"	"Catalogue"	FALSE	'e'	TRUE
"Catalogue"	"Cat"	TRUE	's'	FALSE

Write pseudocode for the function `Compare()`

This image shows a full page of a document template. It consists of approximately 28 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting or typing. The background is plain white.

12 A record structure is declared to hold data relating to components being produced in a factory:

```
TYPE Component
  DECLARE Item_ID : STRING
  DECLARE Reject : BOOLEAN
  DECLARE Weight : REAL
ENDTYPE
```

The factory normally produces a batch (or set) of 1000 components at a time. A global array is declared to store 1000 records for a batch:

```
DECLARE Batch : ARRAY [1:1000] OF Component
```

Two global variables contain the minimum and maximum acceptable weight for each component. The values represent an inclusive range and are declared as:

```
DECLARE Min, Max : REAL
```

(a) (i) A program uses a variable `ThisIndex` as the array index to access a record.

Write a pseudocode clause to check whether or **not** the weight of an individual component is within the acceptable range.

.....
.....
..... [3]

(ii) When batches of less than 1000 components are processed, it is necessary to indicate that certain elements in the array are unused.

Suggest how an unused array element could be indicated.

.....
..... [1]

(b) A module `InRange()` will:

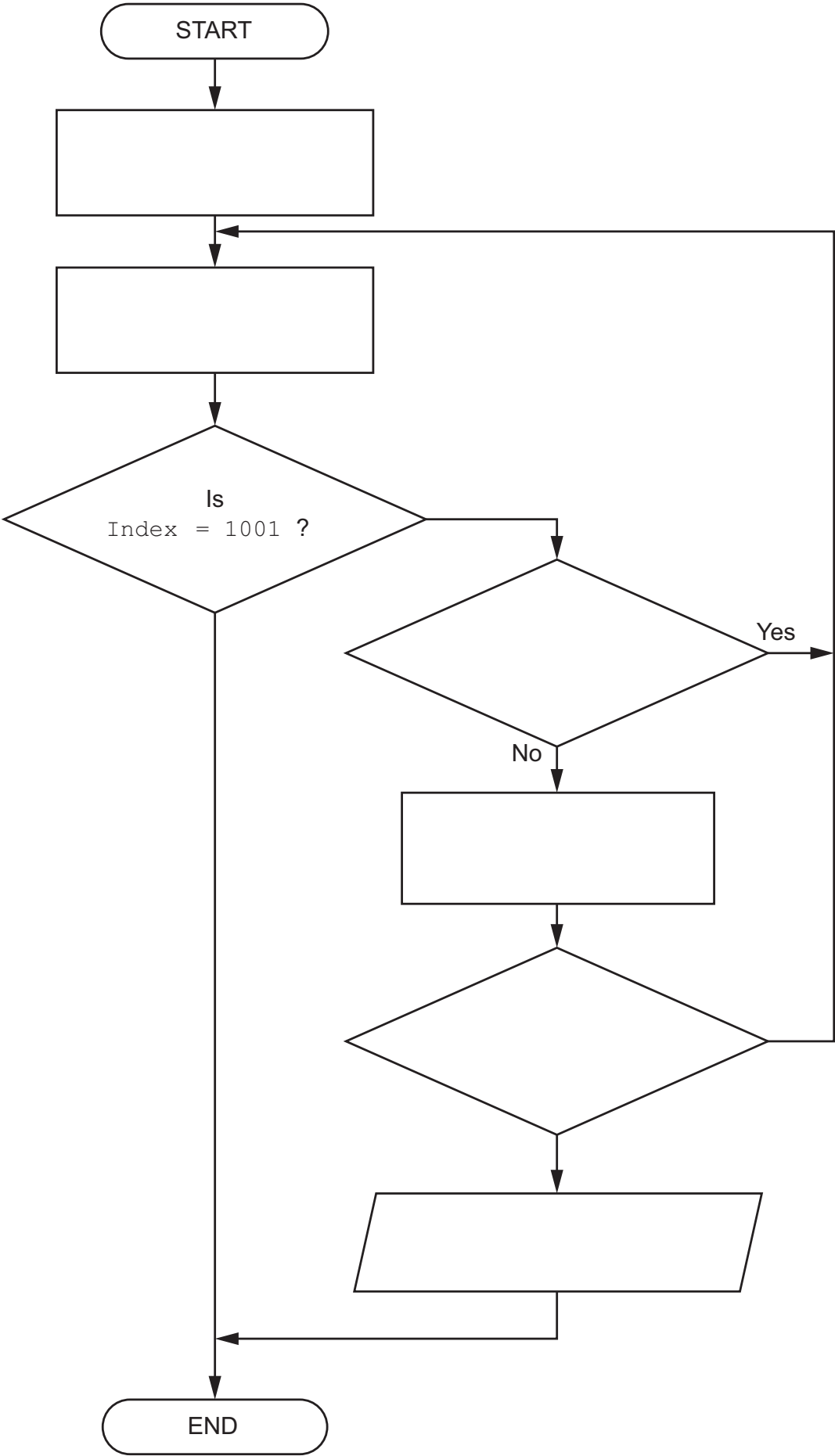
- be called with an integer parameter representing an index value of a record in the `Batch` array
- check if the weight of the indexed component is within the acceptable range
- return `TRUE` if the weight is in the range and `FALSE` if it is **not**.

A module `BatchCheck()` will:

- iterate through a batch of 1000 component records
- call module `InRange()` to check each individual component record
- keep a count of the number of components that fail
- output a suitable warning message and immediately stop if the number of failed components exceeds 5.



Complete the program flowchart to represent the algorithm for module `BatchCheck()`.



13 (a) Program variables have values as follows:

Variable	Value
Today	"Tuesday"
WeekNumber	37
Revision	'C'
MaxWeight	60.5
LastBatch	TRUE

(i) Give an appropriate data type for each variable.

Variable	Data type
Today	
WeekNumber	
Revision	
MaxWeight	
LastBatch	

[5]

(ii) Evaluate each expression in the following table.
If an expression is invalid then write ERROR.

Refer to the **Insert** for the list of pseudocode functions and operators.

Expression	Evaluates to
MID(Today, 3, 2) & Revision & "ape"	
INT(MaxWeight + 4.2)	
LENGTH(MaxWeight)	
MOD(WeekNumber, 12)	
(Revision <= 'D') AND (NOT LastBatch)	

[5]

(b) Simple algorithms usually consist of input, process and output.

Complete the table to show if each statement is an example of input, process or output. Place one or more ticks (✓) for each statement.

Item	Statement	Input	Process	Output
1	SomeChars ← "Hello World"			
2	OUTPUT RIGHT(SomeChars, 5)			
3	READFILE MyFile, MyChars			
4	WRITEFILE MyFile, "Data is " & MyChars			

[4]

(c) Write in pseudocode a **post-condition loop** to output all the odd numbers between 100 and 200.

[4]