9.2 Algorithms

Name: _____ Class: _____ Date: _____ **Total: 62 marks**

KEY WORDS algorithm • data type • identifier table • pseudocode • sequence

Objective

Start designing **algorithms**. On this first sheet of 9.2 you learn what an algorithm is, how to name and list its data in an **identifier table**, and how to write a first algorithm that has **input, process and output**. You need no loops and no `IF` yet. Sheet 9.2.1 adds decisions, sheet 9.2.2 flowcharts, and sheet 9.2.3 structured English and stepwise refinement.

You must be able to:

- state what an algorithm is
- choose suitable **identifier** names and data types, and complete an identifier table
- say whether a statement is input, process or output
- write **pseudocode** that contains input, process and output, as a **sequence** of steps
- name the three basic constructs: sequence, **selection** and **iteration**

1 Key knowledge and worked examples

Study these first. Each one shows a **method** that a question later on this sheet needs.

What an algorithm is

An **algorithm** is a **solution to a problem, expressed as a sequence of defined steps** (a recipe is one). For a computer each step must be **exact**, and the steps must be in the **right order**.

Almost every algorithm has the same shape:

part	what it does	in pseudocode
input	gets data from outside: keyboard, file, sensor	`INPUT RoomLength`
process	works on the data: calculates, stores, compares	`Area ← RoomLength * Width`
output	gives the result out: screen, file, printer	`OUTPUT Area`

An **assignment** such as `Total ← 0` is a **process**: nothing comes in and nothing goes out.

Identifiers and the identifier table

Every item of data in an algorithm needs a name, called an **identifier**. Before you write any pseudocode, list the identifiers in an **identifier table**.

list every piece of data first - name, type, description

name	data type	description
`ItemCost`	`REAL`	cost of the item
`Category`	`STRING`	the product category
`SaleDate`	`DATE`	when the item was sold
`InStock`	`BOOLEAN`	TRUE if in stock

use **descriptive** names (`ItemCost`, not `x`) - the table forces you to plan the data

A good identifier:

- is **meaningful**: it says what the data is (`ItemCost`, not `x` or `Data1`);
- has **no spaces**, starts with a **letter**, and is not a keyword such as `INPUT`;
- uses capital letters to show the words: `ItemCost`, `NumberOfStudents`.

Choose the data type from the values the item can take:

the value is ...	example	data type
a whole number	`16`	`INTEGER`
a number with a fraction part	`72.5`	`REAL`
text	`"Ahmed"`	`STRING`
one character	`'A'`	`CHAR`
true or false	`TRUE`	`BOOLEAN`
a date	`20/02/2025`	`DATE`

A telephone number is a `STRING`: it can start with 0, and nobody adds telephone numbers together.

In the exam the table often gives an example value and an explanation, and you add the name and the type:

Example value	Explanation	Variable name	Data type
`"Fruit"`	a category of stock	`Category`	`STRING`
`12.67`	the cost of one item	`ItemCost`	`REAL`
`TRUE`	whether the item is in stock	`InStock`	`BOOLEAN`

■ Writing a first algorithm: a sequence

Method: (1) list the identifiers; (2) write the inputs; (3) write the process; (4) write the outputs. Declare every variable first with `DECLARE`.

Worked example: input the length and the width of a room in metres, and output its area.

identifier	data type	description
`RoomLength`	`REAL`	the length of the room in metres
`Width`	`REAL`	the width of the room in metres
`Area`	`REAL`	the area of the room in square metres

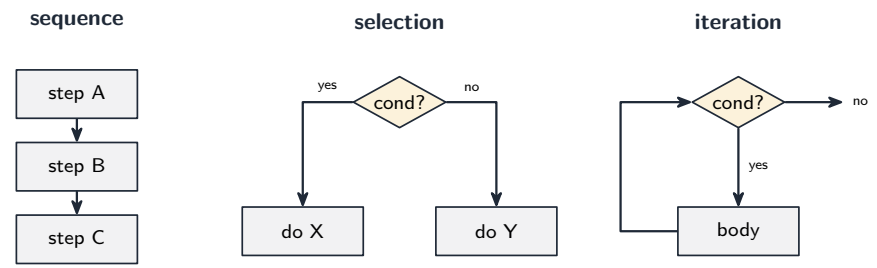
```
DECLARE RoomLength : REAL
DECLARE Width : REAL
DECLARE Area : REAL
OUTPUT "Enter the length and the width"
INPUT RoomLength
INPUT Width
Area ← RoomLength * Width
OUTPUT "The area is ", Area
```

- The arrow ← means "**becomes**": the value on the right is worked out and stored in the variable on the left.
- The steps run **in order**, one after the other. This is a **sequence**. If `Area ← Room Length * Width` came before the two inputs, the algorithm would not work.

■ The three basic constructs

Every algorithm is built from only three **constructs** :

- **sequence**: steps that run one after the other, in order;
- **selection**: a **choice** between paths, which depends on a condition (`IF`, `CASE`);
- **iteration**: steps that are **repeated** (`FOR`, `WHILE`, `REPEAT`). Another word is a **loop** .



On this sheet you only need to **recognise** selection and iteration. You learn to write them on sheets 11.2 to 11.2.3.

■ Key knowledge: evaluating built-in functions

Function	Meaning and example
`INT(x)`	integer part of a real: `INT(3.8)` → 3
`MID(s,start,n)`	n characters from position start, counting from 1: `MID("ALGORITHM",3,4)` → `"GORI"`
`LENGTH(s)`	number of characters: `LENGTH("A B")` → 3
`CHR(n)` / `ASC(c)`	code to character / character to code; use the supplied character-code table
`TO_UPPER(s)` / `TO_LOWER(s)`	change letter case: `TO_UPPER("As2")` → `"AS2"`
`STR_TO_NUM(s)` / `NUM_TO_STR(x)`	numeric text to a number / number to text
`&`	join strings: `"AS" & "2"` → `"AS2"`

Evaluate inner calls first. Check each argument's type; write `ERROR` when the question requests it for an invalid call, such as `LENGTH(27)`. Use `CALL DisplayResult(Total)` for a procedure; a function returns a value used in an expression, such as `Total ← CalculateTotal(Count)`.

Worked example. `TO_UPPER(MID("physics",2,3))` first produces `"hys"`, then `"HYS"`. `STR_TO_NUM("12") + 3` produces the number 15, while `"12" & "3"` produces the string `"123"`.

2 Practice

Now use the methods above. The first questions need one idea each.

2.1 Give the technical term for a solution to a problem that is expressed as a sequence of defined steps. [1]

1- 2- 3-

2.2 Tick (✓) **one** box in each row to show whether the statement is input, process or output. [5]

Statement	input	process	output
`INPUT Age`			
`Total ← Price * Quantity`			
`OUTPUT "Hello"`			
`Count ← 0`			
`OUTPUT Total`			

2.3 A program stores the number of goals a football team has scored. A student suggests these identifiers: `g`, `Goals Scored`, `GoalsScored`, `2ndGoal`.

(a) State which identifier is the most suitable. [1]

(b) Give a reason why **each** of the other three identifiers is not suitable. [3]

2.4 State the most suitable data type for each value. [5]

(a) `42`

(b) `"Lin Wei"`

(c) `3.75`

(d) `FALSE`

(e) `'M'`

(a) 42
(b) "Lin Wei"
(c) 3.75
(d) FALSE
(e) 'M'

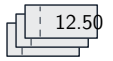
2.5 Complete the identifier table for a program that stores one student's record. [4]

Ahmed, 16, 72.5, passed

Example value	Explanation	Variable name	Data type
`"Ahmed"`	the student's name		
`16`	the student's age in years		
`72.5`	the student's average mark		
`TRUE`	whether the student passed		

2.6 These pseudocode statements find the total cost of some tickets. They are in the wrong order.

- **A** `OUTPUT TotalCost`
- **B** `INPUT NumberOfTickets`
- **C** `TotalCost ← NumberOfTickets * 12.50`
- **D** `DECLARE NumberOfTickets : INTEGER` and `DECLARE TotalCost : REAL`

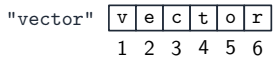


Write the letters in the correct order. [2]

2.7 Write pseudocode that inputs a distance in kilometres and outputs the same distance in metres. Declare your variables. *(Hint: copy the shape of the worked example: declare, input, process, output.)* [4]

2.8 Evaluate each expression; write `ERROR` for a type mismatch. Assume character code 65 represents `A`.

(a) `INT(8.9)` (b) `TO_UPPER(MID("vector",2,3))` (c) `LENGTH("A B")` (d) `CHR(65)` (e) `STR_TO_NUM("15") + 2` (f) `"15" & "2"` (g) `LENGTH(15)` [7]



3 Exam-style questions

3.1 A program is being developed for a bicycle-hire shop. An identifier table is used during the design.

(a) Complete the identifier table. [4]

Example value	Explanation	Variable name	Data type
`"HR004821"`	the code of a hire		
`3`	the number of hours of the hire		
`4.50`	the cost of one hour		
`TRUE`	whether the bicycle has been returned		

(b) An identifier table includes the names of the identifiers. State **two** other pieces of information that an identifier table should contain. [2]

(c) State **two** reasons why an identifier table is written **before** the pseudocode. [2]

3.2 A student's algorithm uses the variables `a`, `b` and `temp2`.

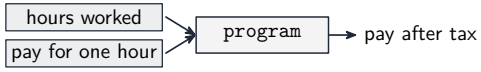
Explain why these names are **not** good practice, and state what the student should do. [2]

`a` `b` `temp2`

3.3 An algorithm is described in structured English:

1. Input the number of hours worked.
2. Input the pay for one hour.
3. Multiply the hours by the pay for one hour to give the pay before tax.
4. Take 20% of this pay as tax.
5. Output the pay after tax.

Write pseudocode for this algorithm. Declare your variables. [5]



3.4 The table contains extracts of pseudocode. Tick (✓) **one or more** boxes in each row to show which constructs the extract contains. [4]

Pseudocode extract	assignment	selection	iteration
`Total ← Total + 1`			
`IF Mark > 50 THEN`			
`FOR Index ← 1 TO 10`			
`WHILE Count < 5`			

3.5 A petrol station wants a program for its pumps. The program inputs the customer's name, the number of litres of fuel taken and the price of one litre. It outputs the cost of the fuel, and then a message that contains the customer's name.

(a) Complete the identifier table for this program. [4]

identifier	data type	description
		the customer's name
		the number of litres of fuel taken
		the price of one litre
		the cost of the fuel

(b) Write pseudocode for the program. Declare your variables. [5]

(c) State which **one** of the three basic constructs your pseudocode uses, and explain how you can tell. [2]