Paper 2 Topic: Structure Charts

TASK 1 – Structure charts

Describe a processing activity that can be represented by one main task with two or more sub-tasks. The activity can relate to any scenario, but should include aspects of selection and iteration.

Activity examples may be taken from different areas, such as:

- school or college
- factory or workplace
- clubs or hobbies.

TASK 1.1

Consider how a problem is decomposed by splitting it into smaller parts.

Discuss the advantages of this approach.

TASK 1.2

Design a modular program to implement the activity described in TASK 1.

Produce a structure chart to represent the modular structure of the solution.

The structure chart should address:

- the sequence of module execution
- any module selection or iteration
- the parameters that are passed between the modules.

TASK 1.3

For each module, decide whether the solution should be implemented as a procedure or a function.

Justify your choices.

Produce **pseudocode** headers for each module.

S18_qp21_Q4

4 **(a)** A structure chart is used in modular program design.

Iteration and selection are two features of an algorithm that may be shown on a structure chart.

Give **three** other features.

Feature 1 .....

.....

Feature 2 .....

.....

Feature 3 .....

.....

[3]

(b) Pseudocode for a function is shown.

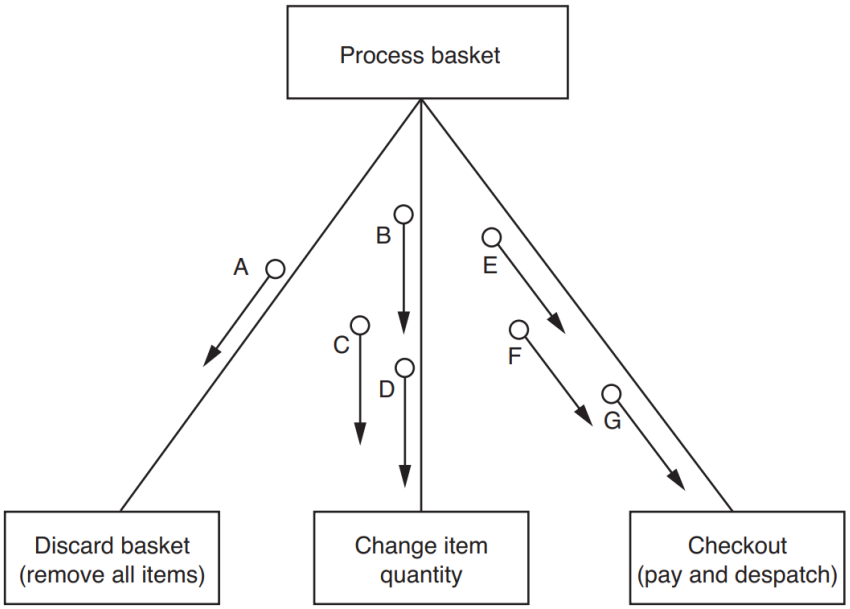
```
FUNCTION ItemProcess (AddItem, InString : STRING) RETURNS BOOLEAN
  DECLARE RetFlag : BOOLEAN
  RetFlag ← FALSE

  IF AddItem = "Yes"
     THEN
        RetFlag ← AddToList(InString)
     ELSE
        CALL RemoveFromList(InString)
  ENDIF

  RETURN RetFlag

ENDFUNCTION
```

Draw a structure chart on the next page to represent this pseudocode.

[6]

S18_qp22_Q4

4 The structure chart shows part of the design of a program for an online shopping system.



(a) (i) Draw on the chart to show the following facts.

- Each of the modules at the lower level returns a Boolean parameter, X.
- Process basket will call only one of the modules shown at the lower level.

[2]

(ii) The parameters A to G shown on the chart will be used to pass the following information.

```
PaymentDetails
Quantity
BasketID
DeliveryAddress
ItemID
```

Complete the following table to show the parameter and the information it represents.

Parameter	Information
A	
B	
C	
D	
E	
F	
G	

[3]

3 A modular program design consists of four modules:

`Module1` has three sub-tasks. Each sub-task is implemented by a single subroutine (either a function or a procedure).

The subroutine headings are defined as follows:

```
FUNCTION  Module2 (Weight : REAL) RETURNS BOOLEAN
PROCEDURE Module3 (Weight : REAL, Customer : STRING, Purchased : DATE)
FUNCTION  Module4 (Purchased : DATE, Account : INTEGER) RETURNS INTEGER
```

(a) State the term given to values passed between modules.

.....[1]

(b) Draw a structure chart to represent the program design.

Use the letters in the table to label the values passed between modules.

Value	Label
Boolean return value	A
Integer return value	B
`Account`	C
`Customer`	D
`Purchased`	E
`Weight`	F

Structure chart

[4]

S17-qp-21-Q4

4 **(a)** A structure chart is a tool used in modular program design.

State **three** pieces of information that a structure chart can convey about a program design.

1 .....

.....

2 .....

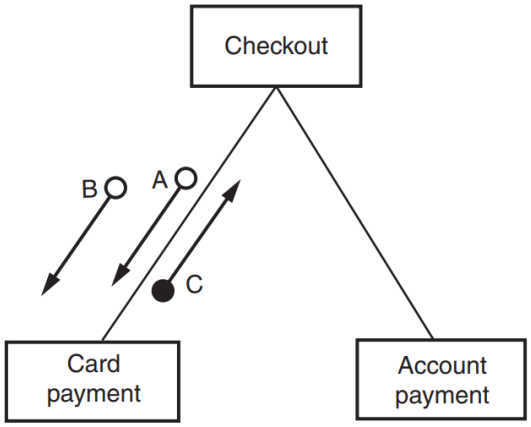
.....

3 .....

.....

[3]

(b) The following diagram shows part of a structure chart.



Examples of the data items that correspond to the arrows are given in this table:

Arrow	Data item
A	`234.56`
B	`"Mr Robert Zimmerman"`
C	`True`

Use **pseudocode** to write the function header for the **Card payment** module.

.....

.....[3]

4 **(a)** Structured programming involves the breaking down of a problem into modules.

Give **two** reasons why this is done.

1 .....

.....

2 .....

.....

[2]

(b) A team needs to write a program to implement an online shopping system. Customers will access the program via a website.

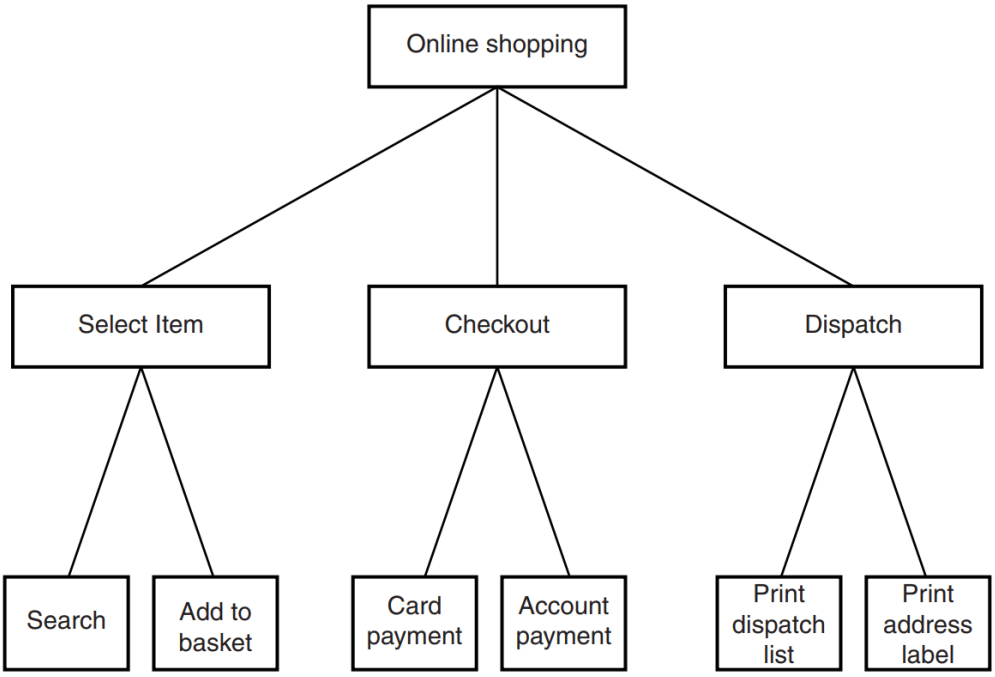
Customers can search for items before adding them to a virtual shopping basket. When they have finished shopping, they pay for the items. The program provides output for the dispatch of the items.

Some of the key features of the system are as follows:

- a customer can add many items to the shopping basket
- payment may be either by credit or debit card, or by adding to a customer account
- the shop may dispatch the items in one or more packages

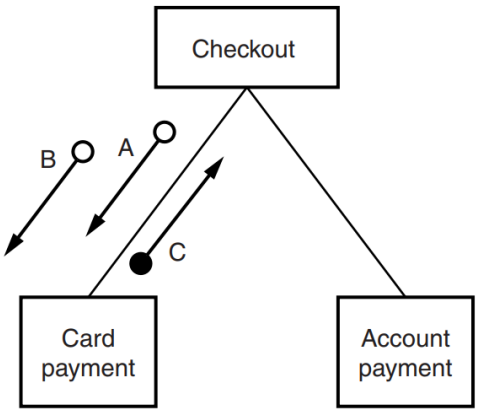
The structure chart below shows the program modules only.

(i) Draw on the chart, the symbols that represent the key features listed in **part (b)** above.



[3]

(ii) A section of the chart in **part (b)(i)** is shown below. It is to show the parameters passed between the Checkout and Card payment modules.



Name the three data items corresponding to the arrows.

Arrow	Data item
A	
B	
C	

[3]

4 **(a)** Name **two** features of your chosen high-level programming language that support the implementation of a modular design.

1 .....

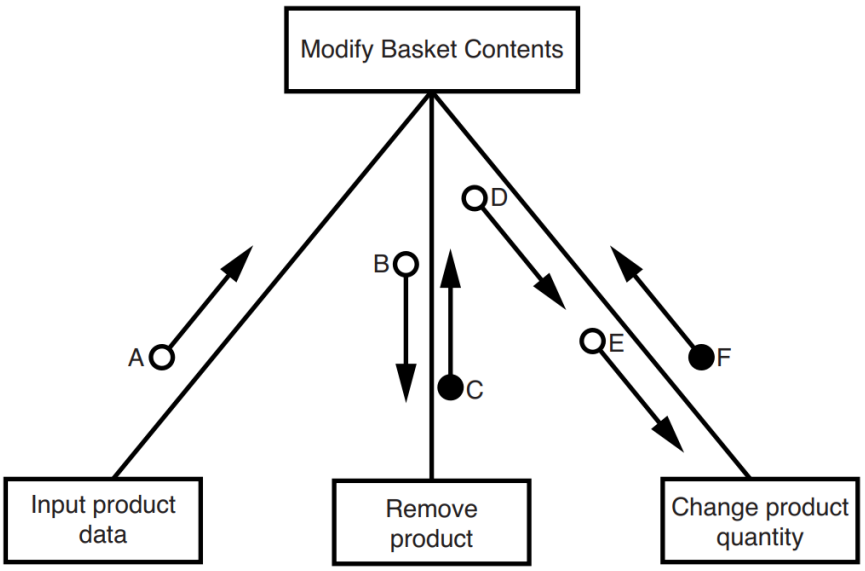
2 .....

[2]

(b) **(i)** The structure chart shows part of the design of a program for an online shopping system.

The user has already added a number of products to their virtual basket.

Draw on the chart, the symbol to show that the process of modifying the basket contents may be iterated (repeated).



[1]

(ii) Each arrow in the structure chart above represents a parameter.

The table below shows the three data items that the six parameters pass between modules.

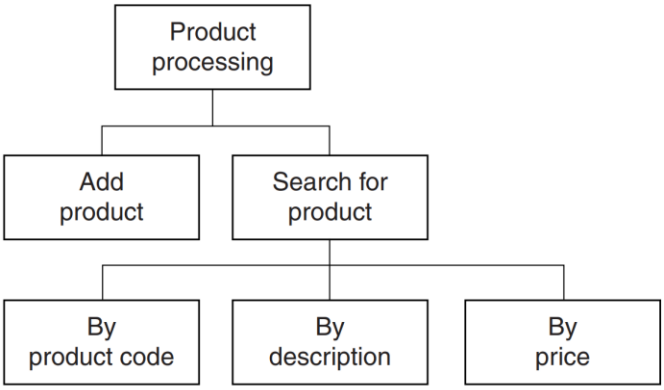
Tick (✓) to match each parameter to the correct data item.

Data item	Parameter					
	A	**B**	**C**	**D**	**E**	**F**
Product ID						
Quantity						
Flag Value – indicating operation success or fail						

[4]

W16-qp-21-Q4c

(c) The company maintains a file of product data. Ahmed is to write a program to add a new product and search for a product based on the structure diagram shown:



The program records the following data for each product:

- product code
- product description
- product retail price

The text file `PRODUCTS` stores each data item on a separate line, as shown below:

File **PRODUCTS**

```
0198
Plums(10kg)
11.50
0202
Onions(20kg)
10.00
   ~
0376
Mango chutney(1kg)
02.99
   ~
0014
Mango(10kg)
12.75
```

The program uses the variables shown in the identifier table.

Identifier	Data type	Description
PRODUCTS	TEXT FILE	Storing the code, description and retail price for all current products
PCode	ARRAY[1:1000] OF STRING	Array storing the product codes
PDescription	ARRAY[1:1000] OF STRING	Array storing the product descriptions
PRetailPrice	ARRAY[1:1000] OF REAL	Array storing the product retail prices
i	INTEGER	Array index used by all three arrays

(i) The first operation of the program is to read all the product data held in file PRODUCTS and write them into the three 1D arrays.

Complete the pseudocode below.

```
OPEN .....

i ← 1

WHILE .....

  READFILE ("PRODUCTS", .....)

  READFILE ("PRODUCTS", .....)

  READFILE ("PRODUCTS", .....)

  .....

  .....

ENDWHILE

CLOSE "PRODUCTS"

OUTPUT "Product file contents written to arrays"
```

[5]

When Ahmed designed the PRODUCTS file, he considered the alternative file structure shown opposite.

It stores one product per line in the text file.

File **PRODUCTS**

0198 Plums(10kg)	11.50
0202 Onions(20kg)	10.00
~	
0376 Mango chutney(1kg)	02.99
~	
0014 Mango(10kg)	12.75

(ii) State **one** benefit and **one** drawback of this file design.

Benefit .....

.....

Drawback .....

.....[2]

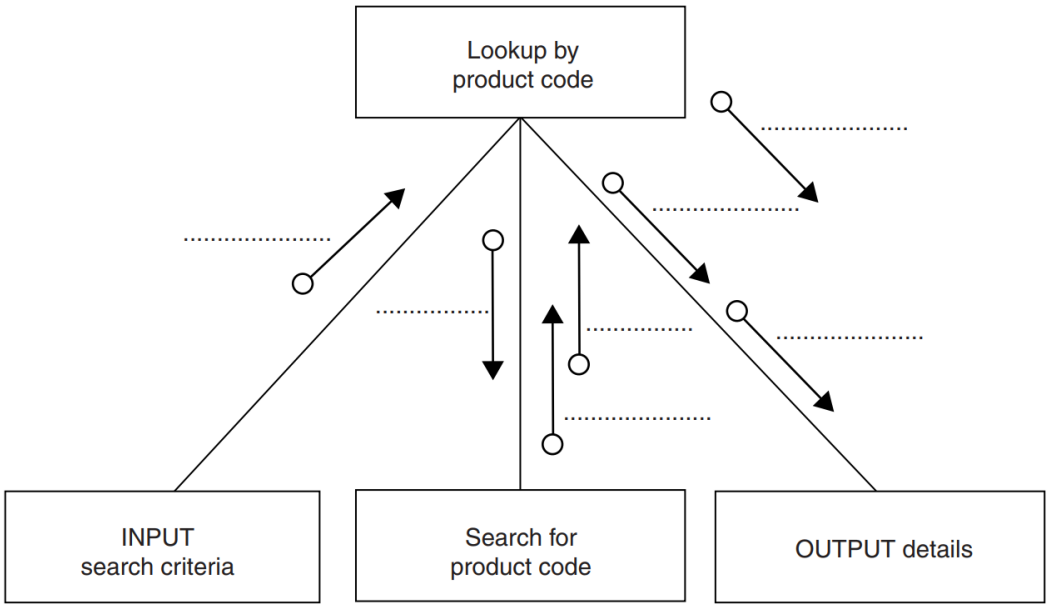
(d) To code the 'Search by product code' procedure, Ahmed draws a structure chart showing the different stages.

The procedure uses the variables shown in the identifier table.

Identifier	Data type	Description
`SearchCode`	`STRING`	Product code input by the user
`ThisIndex`	`INTEGER`	Array index position for the corresponding product
`ThisDescription`	`STRING`	Product description found
`ThisRetailPrice`	`REAL`	Product retail price found

You can assume that before the procedure is run, all the product data is read from file `PRODUCTS` and then stored in three 1D arrays as described in **part (c)(i)**.

Label the structure chart to show the input(s) and output(s).



[4]

3 When the guarantee on a computer runs out, the owner can take out insurance to cover breakdown and repairs.

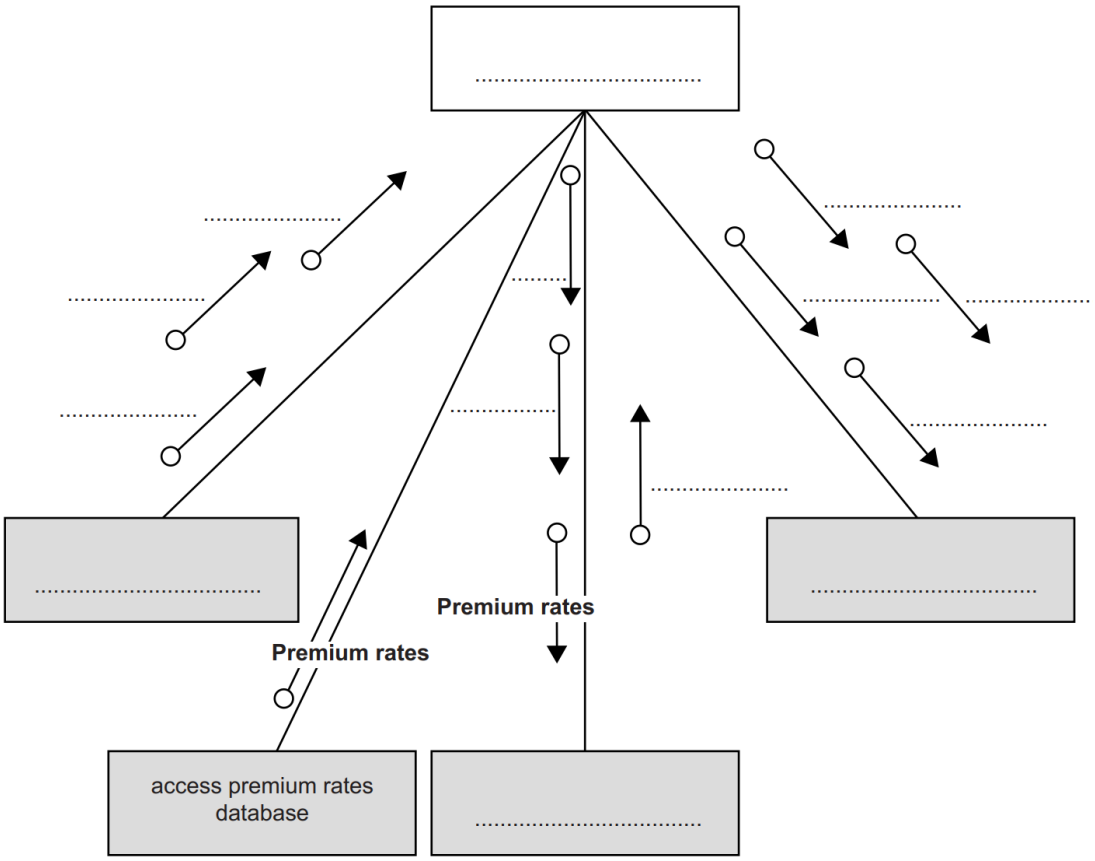
The price of the insurance is calculated from:

- the model of the computer
- the age of the computer
- the current insurance rates

Following an enquiry to the insurance company, the customer receives a quotation letter with the price of the insurance.

A program is to be produced.

The structure chart below shows the modular design for this process:



(a) Using the letters **A** to **D**, add the labelling to the chart boxes on the opposite page.

Modules	
A	Send quotation letter
B	Calculate price
C	Produce insurance quotation
D	Input computer details

[2]

(b) Using the letters **E** to **J**, complete the labelling on the chart opposite.

Some of these letters will be used more than once.

Data items	
E	`CustomerName`
F	`CustomerEmail`
G	`Model`
H	`Age`
I	`PolicyCharge`
J	`PolicyNumber`

[4]

S15-qp-23-Q3

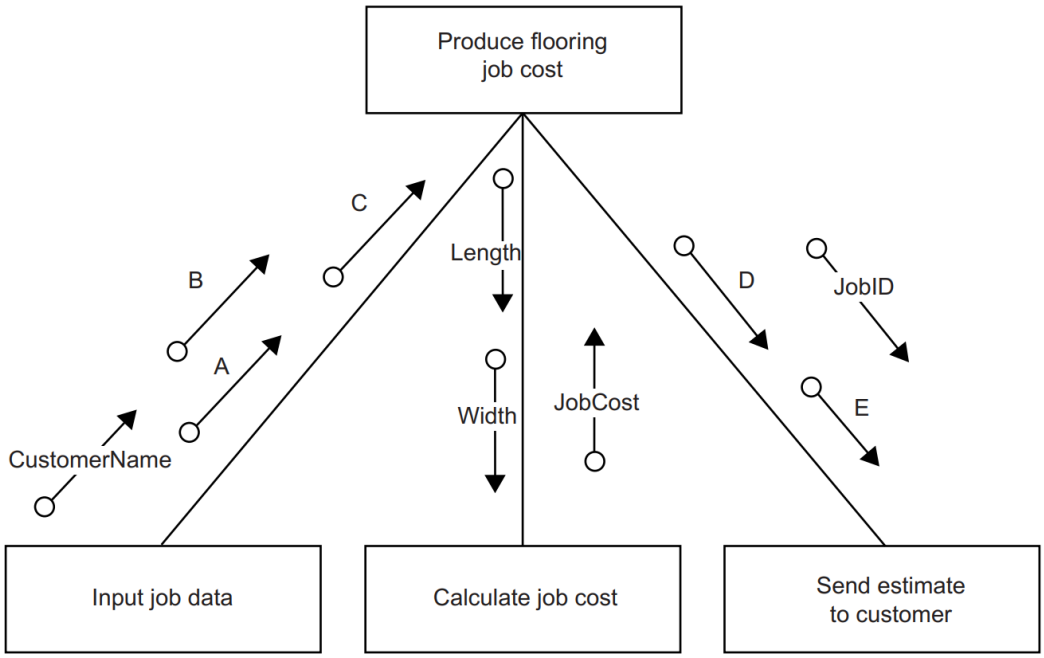
3 A flooring company provides for each customer an estimated price for a new job. Each job is given a Job ID.

The job cost is calculated from the length (nearest metre) and width (nearest metre) of the room.

The process for calculating the price is as follows:

- the floor area is calculated with 18% added to allow for wastage
- the job cost is calculated at $50 per square metre

The structure chart shows the modular design for a program to produce a new job cost.



(i) Give the data items corresponding to the labels A to E in the structure chart.

A .....

B .....

C .....

D .....

E .....

[5]

(ii) The procedure below is one of the modules shown on the structure chart.

Parameters can be passed 'by value' or 'by reference'.

Complete the procedure header below showing for each parameter:

- its parameter passing mechanism
- its identifier
- its data type

```
PROCEDURE CalculateJobCost( ..........................................................
                            ..........................................................
                            ..........................................................
                            .......................................................... )
  JobCost ← (Length * Width * 1.18) * 50
ENDPROCEDURE
```

[5]