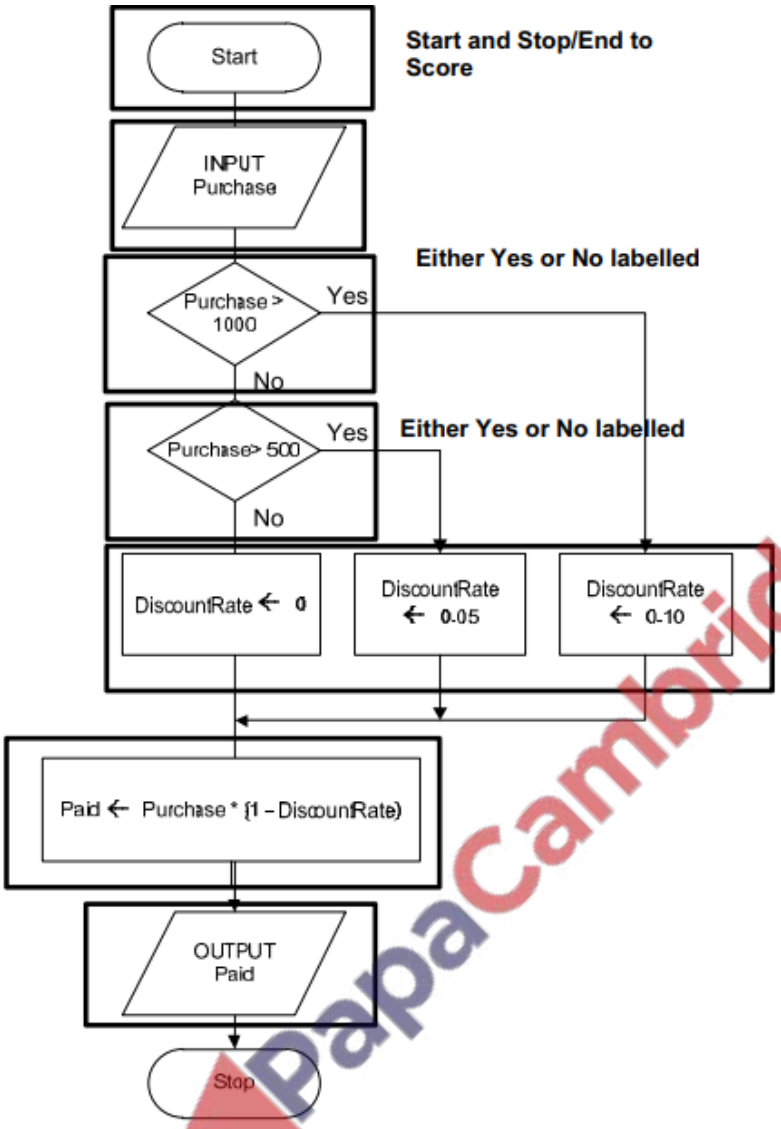
4



Start and Stop/End to Score

Either Yes or No labelled

Either Yes or No labelled

[MAX 6]

[Total: 6]

1 (c)

```
INPUT TicketType
WHILE NOT (TicketType = 'E') OR (TicketType = 'S')
    INPUT TicketType
ENDWHILE
```

One mark for each of:

- `WHILE ... ENDWHILE`
- Correct condition in a loop
- `INPUT` within loop plus one before loop // alternative arrangement leading to correct exit from loop

[3]

Question	Answer	Marks
5 (a) (i)	• So that the data / information is saved after the program is run / when the computer is switched off • So the data / information can be accessed next time the program is run • So the data information can be "permanently stored"	Max 1
(ii)	**Problem:** • When retrieving / searching for / editing (text relating to a particular CD) • Can't tell where the artist name stops and the title begins (or any similar explanation or example) **Solution 1:** • Use of a separator character// or by example • Where the separator character does not occur in the original strings **Solution 2:** • Use a fixed number of characters for each data item • Data items are padded with e.g. <Space> character where needed **Solution 3:** • Convert original data items to CamelCase • …and add a Space separator Mark as follows: Two marks for description of problem Two marks for description of solution	4

4 (c) (i)

```
OPEN "PRODUCTS" FOR READ
i ← 1
WHILE NOT EOF("PRODUCTS")

   READFILE ("PRODUCTS", PCode[i])
   READFILE ("PRODUCTS", PDescription[i])
   READFILE ("PRODUCTS", Temp // PRetailPrice[i])

   PRetailPrice[i] ← TONUM(Temp)

   i ← i + 1
ENDWHILE

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

One mark per bold phrase (three READFILE() counts as a single mark) **[5]**

(ii) Benefit:

- The number of file read operations is reduced (by 2/3rds)
- It may use less storage / space in the file if strings are NOT fixed length
- All the data related to a single product is read at once / in one file operation / grouped together

Drawback:

- The program will need to use the string handling functions to isolate each of the three items of data
- Difficult to isolate data items if the format is not consistent
- More difficult to search

Max one benefit and one drawback **[2]**

2 (a) (i) Displays the menu (choices)
Repeats the prompt and input …
…the input is a number between 1 and 4 // Checks number is between 1 and 4

"within range" is not enough [3]

(ii) …the input number is validated [1]

(b) (i) 3 [1]

(ii) Previous design repeated indefinitely // (new design) limits number of attempts

Penalise "Program terminates/closes" [1]

5 (c) (i) 2 [1]

(ii)

Tick Cross	Explanation (if invalid)
X // ✓	2nd parameter should be `CHAR` // accept just tick
X	Three parameters/should be 2 parameters
✓	

[3]

3 (a)

	Inputs		Output	
Test Case	InA	InB	OutZ	
1	TRUE	TRUE	**FALSE**	[1]
2	TRUE	FALSE	**TRUE**	[1]
3	FALSE	TRUE	**TRUE**	[1]
4	FALSE	FALSE	**TRUE**	[1]

(b)
```
IF InA = TRUE AND InB = TRUE
    THEN
        OutZ ← FALSE
    ELSE
        OutZ ← TRUE
ENDIF
```

Mark as follows

Structure: IF – THEN – ELSE – ENDIF [1]

Condition: InA = TRUE AND InB = TRUE [1]

Logic:
```
OutZ ← FALSE (when condition true)
OutZ ← TRUE (when condition false)
```
[1]

Alternative answer (worth 3 marks):
```
OutZ ← NOT(InA AND InB)
OutZ ← NOT InA OR NOT InB
```

```
TotalSales ← 0
FOR SalesPerson ← 1 TO 6
    FOR MonthNumber ← 1 TO 12
        TotalSales ← TotalSales + Sales[SalesPerson, MonthNumber]
        PersonTotal ← PersonTotal + Sales[SalesPerson, MonthNumber]
    NEXT MonthNumber
    OUTPUT "Person ", SalesPerson, "–", PersonTotal
    PersonTotal ← 0
NEXT SalesPerson
OUYTPUT TotalSales
```

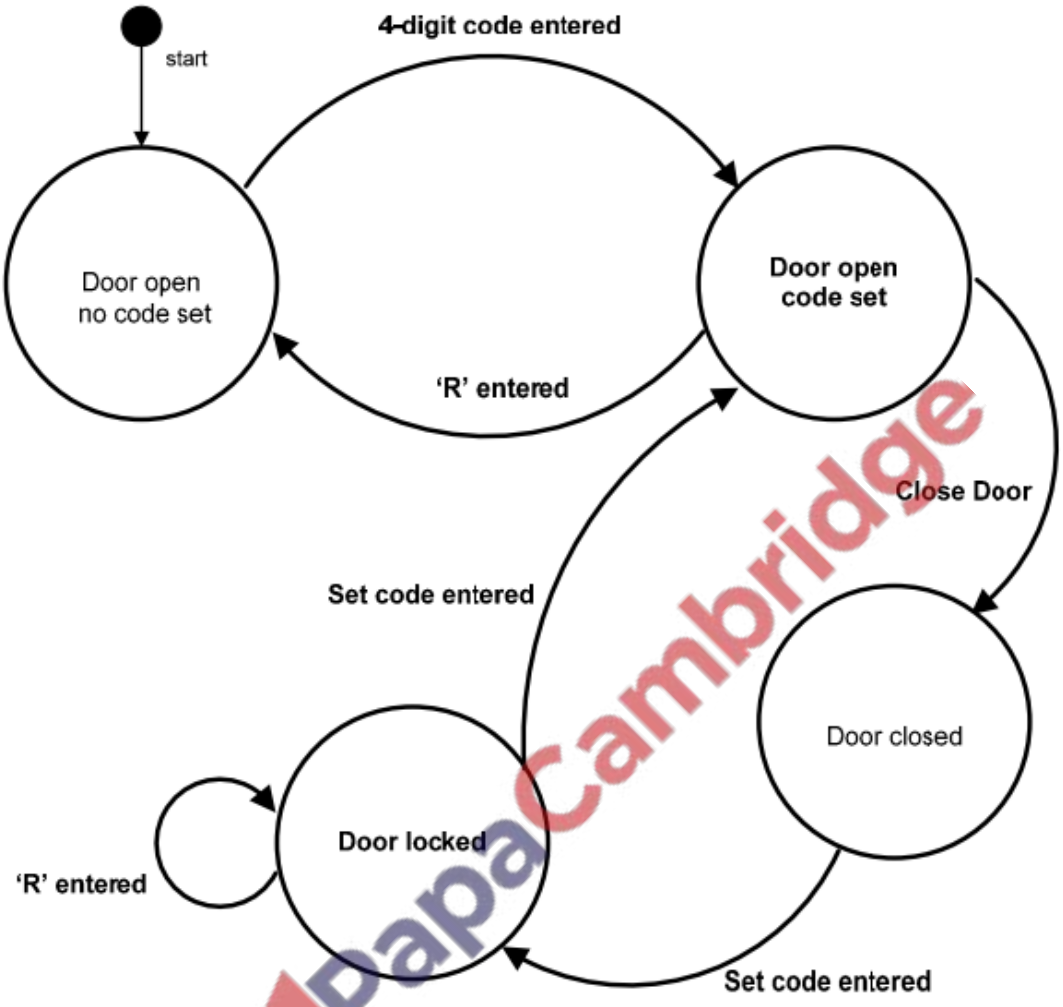
1 (b) Rows 2 to 7 are examples only

TicketType	BaggageWeight	Explanation	Expected output
E	15	Under the allowance	0
E	> 16	Under the allowance	*Charge*
S	<= 20	Under the allowance	0
S	> 20	Under the allowance	*Charge*
E	16	Boundary weight for a type E ticket	0
S	20	Boundary weight for a type S ticket	0
E or S	negative or non-numeric	Invalid weight	Error message

Ticket type	Baggage allowance (kg)	Charge rate per additional kg ($)
'E'	16	3.50
'S'	20	5.75

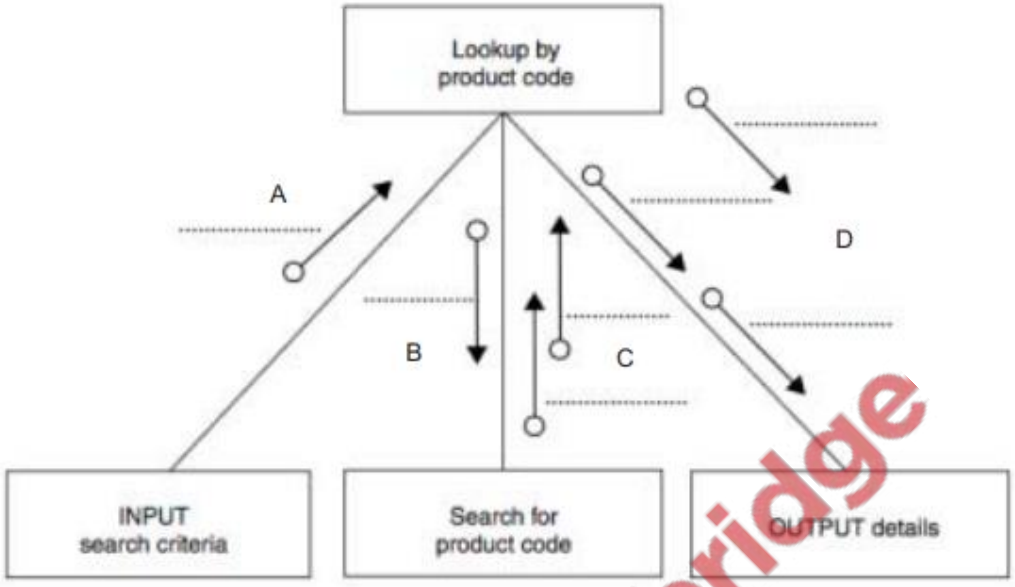
One mark for each different test (examples above) **Max [5]**

1 **(a)** 1 mark for both Set code entered correct. 1 mark for each label.



(b) **(i)** 1 mark per bullet to max 3 **[3]**

4 (d)



One mark per group (one or more names) as follows:

A: SearchCode
B: SearchCode // ThisIndex
C: ThisRetailPrice, ThisDescription
D: SearchCode, ThisDescription, ThisRetailPrice **[4]**