



# **Advanced Higher Computing Science**

## **Software Design and Development: 2-D array workshop materials**

The information in this publication may be reproduced in support of SQA qualifications only on a non-commercial basis. If it is reproduced, SQA must be clearly acknowledged as the source. If it is to be reproduced for any other purpose, written permission must be obtained from [permissions@sqa.org.uk](mailto:permissions@sqa.org.uk).

This edition: December 2024 (version 1.0)

© Scottish Qualifications Authority 2024

# Introduction

This document is for teachers and lecturers and/or Advanced Higher Computing Science candidates.

This document contains 2-D array questions and accompanying marking instructions devised for the workshop at an Understanding Standards event held in 2019.

# Questions

1. A retail store employs 10 salespeople. The store keeps a record of the number of new store cards issued by each salesperson over the first six months of the year.

Sample data is shown in the table below.

	Jan	Feb	Mar	Apr	May	Jun
Adams	12	12	6	8	3	2
Burns	12	17	7	4	5	9
Cook	2	12	0	12	0	3
Davies	4	10	7	4	8	9
East	5	0	0	0	0	0
Faass	6	1	4	6	7	18
Gray	12	19	12	16	17	7
Hill	13	9	7	3	4	5
Iozzi	12	8	4	4	5	4
Jian	14	11	12	4	5	6

The sales data will be stored in a 2-D array called `storeCards`, with each row of the array representing the sales for one salesperson, and each column representing a month.

Two separate 1-D arrays called `person` and `month` will be used to store the name of each salesperson, and the names of the first six months of the year.

The sample data in the table above will be stored in the three arrays as shown below.

storeCards	person	month
<pre> 12 12 6 8 3 2 12 17 7 4 5 9 2 12 0 12 0 3 4 10 7 4 8 9 5 0 0 0 0 0 6 1 4 6 7 18 12 19 12 16 17 7 13 9 7 3 4 5 12 8 4 4 5 4 14 11 12 4 5 6 </pre>	<pre> Adams Burns Cook Davies East Faass Gray Hill Iozzi Jian </pre>	<pre> Jan Feb March April May June </pre>

(a) (i) Use a programming language of your choice to write a declaration statement for the 2-D array `storeCards`. 2

(ii) Write the statement used to assign the value of the April sales for salesperson Gray. 1

- (b) Describe the purpose of the following section of code. 2

```
SEND "Enter a number 1 to 10 to represent one salesperson" TO DISPLAY
RECEIVE salesPersonNumber (INTEGER) FROM KEYBOARD
SET personTotal TO 0
FOR index FROM 0 TO 5 DO
    SET personTotal TO personTotal + storeCards[salesPersonNumber-1,index]
END FOR
SEND "The result is " & personTotal TO DISPLAY
```

- (c) The 1-D array called `monthlyTotals` will store the total number of new cards issued each month. Use a programming language of your choice to write the code for a procedure to work out each monthly total and assign them to the array called `monthlyTotals`. 3

- (d) Any salesperson who issues 16 or more cards in any month will receive a bonus. Details of qualifying salespeople will be stored in a database called `StaffBonus` in a table called `Bonus`.

Structure of the `Bonus` table

field	key	validation
id	PK	auto number
salesPersonID		required
monthID		required
cardsInMonth		required

Use pseudocode to design an algorithm to identify qualifying salespeople and store the relevant details in the database table called `Bonus`. 5

- (e) Write the code for a procedure called `display` that can be activated in the main program. It needs to display the name of the salesperson who issued the highest number of store cards in any one month in the first half of the year. 4

**Note:** teachers and lecturers can extend this question to include a question that requires candidates to use a sort algorithm.

2. A printing company uses a computer program to randomly generate and print bingo tickets.

Each bingo ticket has a grid with three rows and nine columns. Each row contains five numbers and four blank spaces.

4			32	45		68		82
9		26			51	62		88
		24		47	55	65	71	

- (a) Use a programming language of your choice to define a 2-D array called `ticket` to store the numbers selected for each bingo ticket. 2
- (b) To generate the tickets, the program first fills in the columns with random integers, as specified in the table below.

Column	Highest possible random integer	Lowest possible random integer
1	10	1
2	20	11
3	30	21
4	40	31
5	50	41
6	60	51
7	70	61
8	80	71
9	90	81

The numbers selected can only appear once on each bingo ticket.

Use pseudocode to write an algorithm to fill each element of the array called `ticket` with random numbers, according to the rules specified above.

**Note:** there is no need to sort the numbers in each column of the bingo ticket. 5

(c) After filling the array with randomly selected numbers, the program replaces four separate positions on each row with the number 0. The bingo ticket is then printed using the rules:

- ◆ If the value of the array cell is 0, then display a space.
- ◆ Otherwise, display the value stored in the array cell.

Use pseudocode to describe an algorithm that could be used to print the numbers in the array onto a ticket.

4

**Note:** teachers and lecturers can extend this question to include a question that requires candidates to use the count occurrences algorithm.

# Marking instructions

1. (a)(i) DECLARE storeCards AS ARRAY OF ARRAY OF INTEGER < 2-D array with 10 rows and 6 columns, all elements set initially to zero >

**Award 1 mark for correct dimensions.**

**Award 1 mark for correct data type.**

- (a)(ii) SET storeCards [6] [3] TO 16

**Award 1 mark for correct assignment.**

- (b) This section of code allows the user to enter a value 1 to 10 that represents one of the company salespeople and acts as the row index for the 2-D array. The code then totals the values in all six columns of the selected row of the 2-D array.

The total sales for the required salesperson for the first six months of the year are displayed on the screen.

**Award 1 mark for a description that refers to user selection used as row index.**

**Award 1 mark for a description that refers to totalling the sales for the required salesperson.**

- (c) PROCEDURE calculate (ARRAY OF ARRAY OF INTEGER storeCards, ARRAY OF INTEGER monthlyTotals)  
FOR row FROM 0 TO 9 DO  
FOR column FROM 0 TO 5 DO  
SET monthlyTotals[column] TO  
monthlyTotals[column] +  
storeCards[row][column]  
END FOR  
END FOR  
END PROCEDURE

**Award 1 mark for correct use of nested loop.**

**Award 1 mark for correct use of column index to process monthlyTotals array.**

**Award 1 mark for correct use of row and column indices to process storeCards array.**

- (d)
1. open connection with StaffBonus database on the secure database server
  2. start loop for each row from 0 to 9
  3. start loop for each column from 0 to 5
  4. if storeCards[row][column] >= 16 then
  5. create SQL INSERT query to add the salesperson's id, month id and number of store cards issued to the Bonus table
  6. execute SQL INSERT query
  7. end if
  8. end column loop
  9. end row loop
  10. close connection with database server

**Award 1 mark for open and close connection to the database.**

**Award 1 mark for use of nested loop.**

**Award 1 mark for correct use of row and column indices to process storeCards array.**

**Award 1 mark for correct generation of INSERT query to add salesperson's details to the database.**

**Award 1 mark for execution of the INSERT query.**

- (e)
- ```

PROCEDURE display (ARRAY OF ARRAY OF INTEGER storeCards, ARRAY
OF STRING person)
  SET maxIssued TO -1
  SET bestPerson TO -1
  FOR row FROM 0 TO 9 DO
    FOR column FROM 0 TO 5 DO
      IF storeCards[row][column] > maxIssued THEN
        SET maxIssued TO storeCards[row][column]
        SET bestPerson TO row
      END IF
    END FOR
  END FOR
  SEND "The salesperson who issued the most store cards in the
  first half of the year is " & person[bestPerson] TO DISPLAY
END PROCEDURE

```

**Award 1 mark for nested loop.**

**Award 1 mark for correct use of storeCards array using row and column indices.**

**Award 1 mark for correct use of find max algorithm to determine correct salesperson.**

**Award 1 mark for correct use of row index to process person array.**



2. (a) DECLARE ticket AS ARRAY OF ARRAY OF INTEGER < 2D array with 3 rows and 9 columns, all elements set initially to zero >

**Award 1 mark for correct dimensions.**

**Award 1 mark for correct data type.**

- (b)
1. set all 90 elements in chosen array to false
  2. set selected to false
  3. start loop for each row from 0 to 2
  4.     start loop for each column from 0 to 8
  5.         repeat until selected = true
  6.             select random number between column\*10+1 and (column+1)\*10
  7.             if chosen[random number] = false then
  8.                 set chosen[random number] to true
  9.                 set tickets[row][column] to random number
  10.                 set selected to true
  11.             end if
  12.         end repeat
  13.     set selected to false
  14. end column loop
  15. end row loop

**Award 1 mark for correct use of nested loop.**

**Award 1 mark for generation of random numbers.**

**Award 1 mark for checking that number selected has not already been used.**

**Award 1 mark for use of conditional loop to select three unique numbers for each column.**

**Award 1 mark for assignment of random number to 2-D array.**

- (c)
1. start loop for each row from 0 to 2
  2.     set all 9 elements in successfulChoices array to false
  3.     start loop for choices from 1 to 4
  4.         set success to false
  5.         repeat until success = true
  6.             select random column between 0 and 8
  7.             if successfulChoices [random column] = false then
  8.                 set successfulChoices [random column] to true
  9.                 set tickets[row][random column] to 0
  10.                 set success to true
  11.             end if
  12.         end repeat
  13.     end choice loop
  14.     start loop for each column from 0 to 8
  15.         if tickets[row][column] = 0 then
  16.             display space
  17.         else
  18.             display ticket[row][column]
  19.         end if
  20.     end row loop

**Award 1 mark for random selection of four cells in each row of the 2-D array.**

**Award 1 mark for ensuring that four different cells are selected in each row.**

**Award 1 mark for allocation of zero to four cells in each row of the array.**

**Award 1 mark for displaying contents of 2-D array.**