# Worksheet 4 Arrays Answers

**Task 1**

1. Write a program to read 6 numbers into an array **numbers[0]** to **numbers[5**], ouput them in reverse order and then output the total and average.

numbers = [0,0,0,0,0,0]

total 🡨 0

FOR index 🡨 0 TO 5

 numbers[index] 🡨 USERINPUT

 total 🡨 numbers[index] + total

ENDFOR

FOR index 🡨 5 TO 0 STEP -1

 OUTPUT numbers[index]

ENDFOR

OUTPUT total

average 🡨 total/6

OUTPUT average

2. A teacher uses a program that stores pupil names in an array. The array is indexed from 0, so the first element in the array is **name[0]**. Occasionally the teacher needs to search for a name to find the student’s record number, which is **index + 1**.

 Write a pseudocode algorithm that will search an array **name** containing **max** elements, to find a name and output record number if it exists. If the name does not exist the user should be told the name was not found. Use appropriate prompts for input and output in your solution.

max 🡨 6

name 🡨 [“Annie”,”Bob”,”Charles”,”Dan”,”Erica”,Faisal”]

OUTPUT “Enter search name”

searchName 🡨 USERINPUT

found 🡨 False

FOR index 🡨 0 TO max - 1

 IF name[index] = searchName THEN

 OUTPUT “Record number: “, index + 1

 found 🡨 True

 ENDIF

ENDFOR

IF found = False THEN

 OUTPUT “Search name not found”

ENDIF

3. Sales quantities of a certain item, calculated to the nearest thousand, for Jan-March, April-June, July-Sep and Oct-Dec are held in separate arrays for each of 3 outlets. The sales figures for each quarter are to be totalled and output in the format

 Total for quarter 1 xxxx

 Total for quarter 2 xxxx

 Total for quarter 3 xxxx

 Total for quarter 4 xxxx

 Write a pseudocode algorithm for this program. Initialise the array with the following test data:

 Outlet1 = [10, 12, 15, 10]

 Outlet2 = [5, 8, 3, 6]

 Outlet3 = [10, 12, 15, 10]

 outlet1Sales 🡨 [10, 12,15,10]

 outlet2Sales 🡨 [5, 8, 3, 6]

 outlet3Sales 🡨 [10, 12, 15, 10]

totalSales 🡨 [0, 0, 0, 0]

FOR quarter 🡨 0 TO 3

 totalSales[quarter] 🡨 totalSales[quarter] + outlet1Sales[quarter] +outlet2Sales[quarter] + outlet3Sales[quarter]

 OUTPUT "Total for quarter ",quarter+1, totalSales[quarter]

END FOR

**Task 2**

4. (a) Now suppose, in question 3, there were 50 outlets. Assuming the array **outletSales** holds the sales values for each quarter for each outlet, complete the following program to output the total sales figures for each quarter.

 Fill array outletSales with sales values

 Initialise each element of array total[4] to zero

 FOR quarter 🡨 0 TO 3

 FOR outlet 🡨 0 TO 49

 total[quarter] 🡨 total[quarter] + outletSales[quarter][outlet]

 ENDFOR

 OUTPUT "Total for quarter ",quarter+1, total[quarter]

 ENDFOR

5. A grid game draws a 6 by 4 grid with each square denoted by “x”. A character “O” can move by entering a row coordinate from 1 to 6 and column co-ordinate from 1 to 4. The character starts at array position grid[0][0] (Figure 1) and will move, for example, to row 0 column 1 (Figure 2) if the user enters 1, 2 for the row and column coordinates. **Remember that the indices of the array both start at 0**.

 Write a pseudocode algorithm that initialises a 2-D array called grid, drawn as shown in Figure 1.

 Prompt the user to enter a row and column value. Update the character position and draw the new grid.

O x x x x O x x

x x x x x x x x

x x x x x x x x

x x x x x x x x

x x x x x x x x

x x x x x x x x

Figure 1 Figure 2

FOR row 🡨 0 to 5

 FOR column 🡨 0 to 3

 grid[row][column] 🡨 “x”

 ENDFOR

ENDFOR

grid[0][0] 🡨 ”O”

OUTPUT “Enter row to move to”

row 🡨 USERINPUT – 1 # adjust for rows 1 to 6, not 0 to 5

OUTPUT “Enter column to move to”

column 🡨 USERINPUT - 1 # adjust for columns 1 to 4, not 0 to 3

grid[row,column] 🡨 ”O”

FOR row 🡨 0 to 5

 FOR column 🡨 0 to 3

 OUTPUT grid[row][column] ,end = ” “ # print without moving to a new line

 ENDFOR

 OUTPUT “ “ #This is to provide a newline

ENDFOR

6. A company runs a private car park near an airport. The car park has 10 rows numbered 1-10 and each row has spaces (referred to as columns) numbered 1-6 for 6 cars. Customers leave their cars with keys at the car park office, and a driver parks it in a free space and then records where it is parked.

 The space is referenced by its grid coordinates row and column. E.g. a car parked in the 3rd row, 5th space would have the grid reference grid[2][3]. (Indices start at 0)

 The driver enters the car registration into the computer. A car with registration AVH 61 HU parked at grid reference [3][5] would assign “AVH 61 HU” to **park[3][5]**. Empty spaces are denoted, for example, by **park[3][5] = “empty”**

 Write pseudocode for a program which :

 Initialises the grid, with each element holding “empty”.

 “Parks a car”. This option asks the user to enter the registration number of a car and the grid reference (row and column number) where it has been parked.

 Validates the user entry row between 1 and 10, column between 1 and 6 and asks user to re-enter until entry is valid.

 Checks that this is an empty space, and if it is, puts the registration number in the appropriate element of the array. If it is not, displays “That space is taken” and asks the user to re-enter the grid reference.

 Displays the grid.

Note that the way the 2-D array is initialised will vary in different programming languages – the pseudocode does not go into details. Indices start at 0. Program solutions in Python and VB are provided.

Initialise the car park grid to “empty”

(For test purposes, set carpark[0][0] to “TAKEN”)

Display grid

prompt for and enter car registration

prompt for and enter row and column where car is parked

emptySpace 🡨 False

WHILE emptySpace = False

 WHILE row < 1 OR row > 10

 OUTPUT "Row must be between 1 and 10 - please re-enter: "

 row 🡨 USERINPUT

 ENDWHILE

 row 🡨 row - 1

 rowValid 🡨 True

 WHILE column < 1 or column > 6

 OUTPUT "Column must be between 1 and 6 - please re-enter: "

 column 🡨 USERINPUT

 ENDWHILE

 column 🡨 column - 1

 columnValid 🡨 True

 IF rowValid AND columnValid and carPark[row][column]== "empty" THEN

 emptySpace 🡨 True

 ELSE

 rowValid 🡨 False

 columnValid 🡨 False

 OUTPUT("That space is taken”)

 ENDIF

ENDWHILE

carPark[row][column] = regNo

#print the grid

FOR row = 0 TO 9

 FOR column = 0 TO 5

 symbol = carPark[row][column]

 OUTPUT (symbol,” “, end = “”) #print without moving to new line

 ENDFOR

 OUTPUT() #move to new line

ENDFOR