2D Arrays

A 2D array is similar to a ***table*** like one that would appear in Microsoft excel or, even similar to the ones you been seeing in Microsoft access in the sense that there are ***rows and columns***

Declaring a 2D Array

We will declare a 2D array with 3 Rows, and 4 Columns

```
var arrNums : array[1..3,1..4] of integer;
```

- arrNums is ***name*** of the array
- 1..3 are the ***rows***
- 1..4 are the ***columns***

	1	2	3	4
1				
2		23		
3				45

- arrNums[2,2] = 23
- arrNums[3,4] = 45

The rows and columns of arrNums

Note It is unlikely that the exam would request you to print the strucutre, this is only to demonstrate how to do it, if you want to try it yourself on another 2D array.

```
procedure TForm1.btnPrintArrayClick(Sender: TObject);
var row,col : integer;
str : string;
begin
  for row := 1 to 3 do
```

```
  begin
    str := '';
    for col := 1 to 4 do
      begin
        str := str + '['+inttostr(row)+']'+'['+inttostr(col)+']' + ' ';
      end;
      redOutput.Lines.Add(str);
   end;
end;
```

Would output :

```
[1][1] [1][2] [1][3] [1][4]
[2][1] [2][2] [2][3] [2][4]
[3][1] [3][2] [3][3] [3][4]
```

Intialize 2D array with random values

```
procedure TForm1.btnInitializeClick(Sender: TObject);
var row,col : integer;
begin
  for row := 1 to 3 do
    for col := 1 to 4 do
      begin
        arrNums[row,col] := random(9)+1;
      end;
end;
```

Print the elements inside of the 2D Array

```
procedure TForm1.btnPrintElementsClick(Sender: TObject);
var row,col : integer;
str : string;
begin
  for row := 1 to 3 do
  begin
    str := '';
    for col := 1 to 4 do
      begin
        str := str +  inttostr(arrNums[row,col]) + ' ';
      end;
      redOutput.Lines.Add(str);
   end;
end;
```

This code can be explained as follows. First think about how the array looks, refer to the picture above, its 3(rows) x 4 columns

The first for loop says I want you to start at row 1, and go "down" after each increment :

```
for row := 1 to 3 do
    begin

    end;
```

The second for loop says I want you to start at column 1, and go "left" after each increment :

```
for col := 1 to 4 do
    begin

    end;
```

The two loops would run like this :

```
for row := 1 to 3 do
    for col := 1 to 4 do
```

- Row loop = 1, Col loop = 1:
- Row loop = 1, Col loop = 2;
- Row loop = 1, Col loop = 3;
- Row loop =1, Col loop = 4;
- Row loop =2, Col loop = 1;
- ... and so forth

And we end up with this :

```
[1][1] [1][2] [1][3] [1][4]
[2][1] ...
```

So when you think about it, the outer **Row** loop only runs 3 times, and inner **Col** loop runs 4 times for each of those row 3 times, for a total of 12 times.

Add the rows of a 2D Array

```
7 4 8 7
4 1 8 5
8 9 1 5
Sum of row:1 = 26
```

```
Sum of row:2 = 18
Sum of row:3 = 23
```

```
procedure TForm1.btnAddRowClick(Sender: TObject);
var row,col,rowSum : integer;
begin
  for row := 1 to 3 do
  begin
    rowSum  := 0;
    for col := 1 to 4 do
    begin
      rowSum := arrNums[row,col] + rowSum;
    end;
    redOutput.Lines.Add('Sum of row:' + inttostr(row) + ' = ' +
inttostr(rowSum));
  end;
end;
```

Add the columns of a 2D Array

```
3 9 8 1
8 2 8 7
4 5 1 9
Sum of col:1 = 15
Sum of col:2 = 16
Sum of col:3 = 17
Sum of col:4 = 17
```

```
procedure TForm1.btnAddColumnsClick(Sender: TObject);
var row,col,colSum : integer;
begin
  for col := 1 to 4 do
  begin
    colSum := 0;
    for row := 1 to 3 do
    begin
      colSum := arrNums[row,col] + colSum;
    end;
    redOutput.Lines.Add('Sum of col:' + inttostr(col) + ' = ' +
inttostr(colSum));
  end;
end;
```

Note Notice with this function, the outer loop is the Col and the inner loop is Row.

```
for col := 1 to 4 do
    for row := 1 to 3 do
```

- Col loop = 1, Row loop = 1
- Col loop = 1, Row loop = 2
- Col loop = 1, Row loop = 3
- Col loop = 2, Row loop = 1

Would give us:

```
[1,1]
[1,2]
[1,3]
```

And then:

```
[1,1][1,2]
[1,2]
[1,3]
```

Prepping for exams

According to the Government the syllabus for 2D arrays is as follows :

Arrays as a data structure (2-dim)

- Structure
- Step through items-
- Basic operations, e.g. row/column aggregate
- Download the syllabus document here : https://www.education.gov.za/Portals/0/CD/National%20Curriculum%20Statements%20and%20Vocational/CAPS%20FET%20_%20INFORMATION%20TECHNOLOGY%20_%20GR%2010-12%20_%20Web_E677.pdf?ver=2015-01-27-154419-943

If you look at Marina Myburghs(see Marina_Myburg_arrays.pdf) document on arrays, there is a part that covers "stringgrids" - stringgrids work with 2D arrays.

Note Ask your teacher if stringgrids will be tested We have not covered it in this document as to my knowledge it is not included in the syllabus anymore - however I have included a exam quetion in the case that is, with a memo include see (2013-ecexam)

Exam Questions

Max

When you open the question, and go into the **code** section you are presented with two arrays.

First Array :

```
    var
  arrDepartments: array [1..8] of String = (
    'PCs & Notebooks',
    'Tablets & eReaders',
    'Software',
    'Printers, Toners and Ink',
    'Cellphones',
    'Gaming & Drones',
    'Network Equipment',
    'Accessories'
  );
```

There is nothing complicated about this array, its a string array of 8 elements, and they have intialized it for us with values.

Second Array which is the 2D array :

```
arrSales : array[1..8, 1..6] of Real = (
    (935.89,965.99,4056.77,5023.89,3802.66,1146.98),
    (2667.78,2491.78,1989.65,2647.88,1601.56,1921.99),
    (6702.45,4271.56,3424.45,3924.55,3085.45,3359.77),
    (6662.34,6658.45,8075.43,2360.66,2635.44,7365.69),
    (16405.33,9741.37,13381.56,18969.76,8604.55,20207.56),
    (10515.29,7582.66,9856.56,7537.68,9115.67,8401.55),
    (7590.99,9212.65,9070.98,6439.99,7984.88,8767.45),
    (9220.65,8097.99,10067.44,9960.87,10109.56,6571.66));
```

This array does look complicated by its appearance, but its quite simple if you understand it - what you will take from it is :

- Data Type : float
- Row : 1..8
- Col : 1..6

To visualize the strucutre - I copied the code from the section The rows and columns of arrNums and made a few changed and pasted it under the code for the first button :

```
[1][1] [1][2] [1][3] [1][4] [1][5] [1][6]
[2][1] [2][2] [2][3] [2][4] [2][5] [2][6]
[3][1] [3][2] [3][3] [3][4] [3][5] [3][6]
[4][1] [4][2] [4][3] [4][4] [4][5] [4][6]
[5][1] [5][2] [5][3] [5][4] [5][5] [5][6]
[6][1] [6][2] [6][3] [6][4] [6][5] [6][6]
```

```
[7][1] [7][2] [7][3] [7][4] [7][5] [7][6]
[8][1] [8][2] [8][3] [8][4] [8][5] [8][6]
```

The code :

```
//
=============================================================================
// Question 3.1
//
=============================================================================
procedure TfrmQuestion3.btnQ3_1Click(Sender: TObject);
var row, col : integer;
str : string;
begin
  for row := 1 to 8 do
  begin
    str := '';
    for col := 1 to 6 do
      begin
        str := str + '['+inttostr(row)+']'+'['+inttostr(col)+']' + ' ';
      end;
      redQ3.Lines.Add(str);
   end;
end;
```

Question 3.1

3.1 **Button [3.1 – Sales information]**

Display the content of the **arrSales** array with suitable headings in the output component provided. All monetary values must be displayed in currency format with TWO decimal places.

Example of output:

Department	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
PCs & Laptops	R 935.89	R 965.99	R 4 056.77	R 5 023.89	R 3 802.66	R 1 146.98
Tablets & eReaders	R 2 667.78	R 2 491.78	R 1 989.65	R 2 647.88	R 1 601.56	R 1 921.99
Software	R 6 702.45	R 4 271.56	R 3 424.45	R 3 924.55	R 3 085.45	R 3 359.77
Printers, Toners and Ink	R 6 662.34	R 6 658.45	R 8 075.43	R 2 360.66	R 2 635.44	R 7 365.69
Cellphones	R 16 405.33	R 9 741.37	R 13 381.56	R 18 969.76	R 8 604.55	R 20 207.56
Games & Drones	R 10 515.29	R 7 582.66	R 9 856.56	R 7 537.68	R 9 115.67	R 8 401.55
Network Equipment	R 7 590.99	R 9 212.65	R 9 070.98	R 6 439.99	R 7 984.88	R 8 767.45
Accessories	R 9 220.65	R 8 097.99	R 10 067.44	R 9 960.87	R 10 109.56	R 6 571.66

We can see the similarity between the the way our 2D array is structured, and the way the way the question wants the information.

This question is straightforward:

- Create the formatting for the rich edit.(headings,floattostrf)

- two for loops to display the the 2D array

However you should take not the left most column in question ("Department, PCc & laptops..) is contained in the first array now its is not so straightforward because we have to use both arrays.

Lets start by getting our data from the 2D-array onto the richedit.

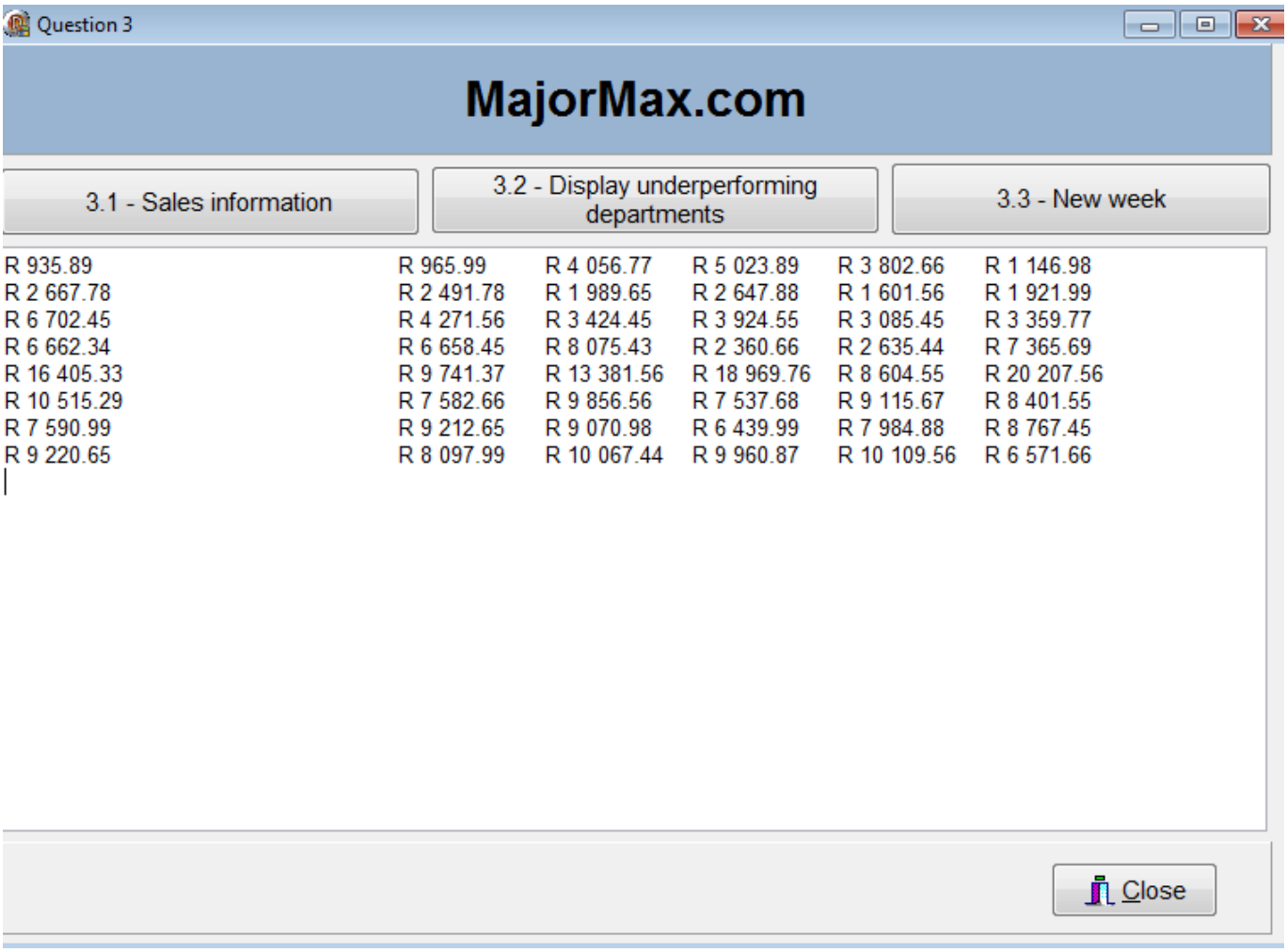
So 8 rows and 6 columns :

```
  for row := 1 to 8 do
    begin
      for col := 1 to 6 do
      begin

      end;

    end;
```

And then adding in a str variable, to output the values to the richedit :

```
procedure TfrmQuestion3.btnQ3_1Click(Sender: TObject);
var row, col : integer;
str : string;
begin
  for row := 1 to 8 do
    begin
      str := '';
      for col := 1 to 6 do
      begin
        str := str + floattostrf(arrSales[row,col],ffcurrency,10,2)+ #9;
      end;
      redQ3.Lines.Add(str);
    end;
end;
```

Unfortunatly, the question did come with **included code**, which affects our output - so we end up with something like this :



From this image, we can see the first column does not look like the rest. The first column is where the information from arrDepartments should be.

Now if you recall, arrDepartments was declared as follows :

```
var
arrDepartments: array [1..8] of String = (
  'PCs & Notebooks',
  'Tablets & eReaders',
  'Software',
  'Printers, Toners and Ink',
  'Cellphones',
  'Gaming & Drones',
  'Network Equipment',
  'Accessories'
);
```

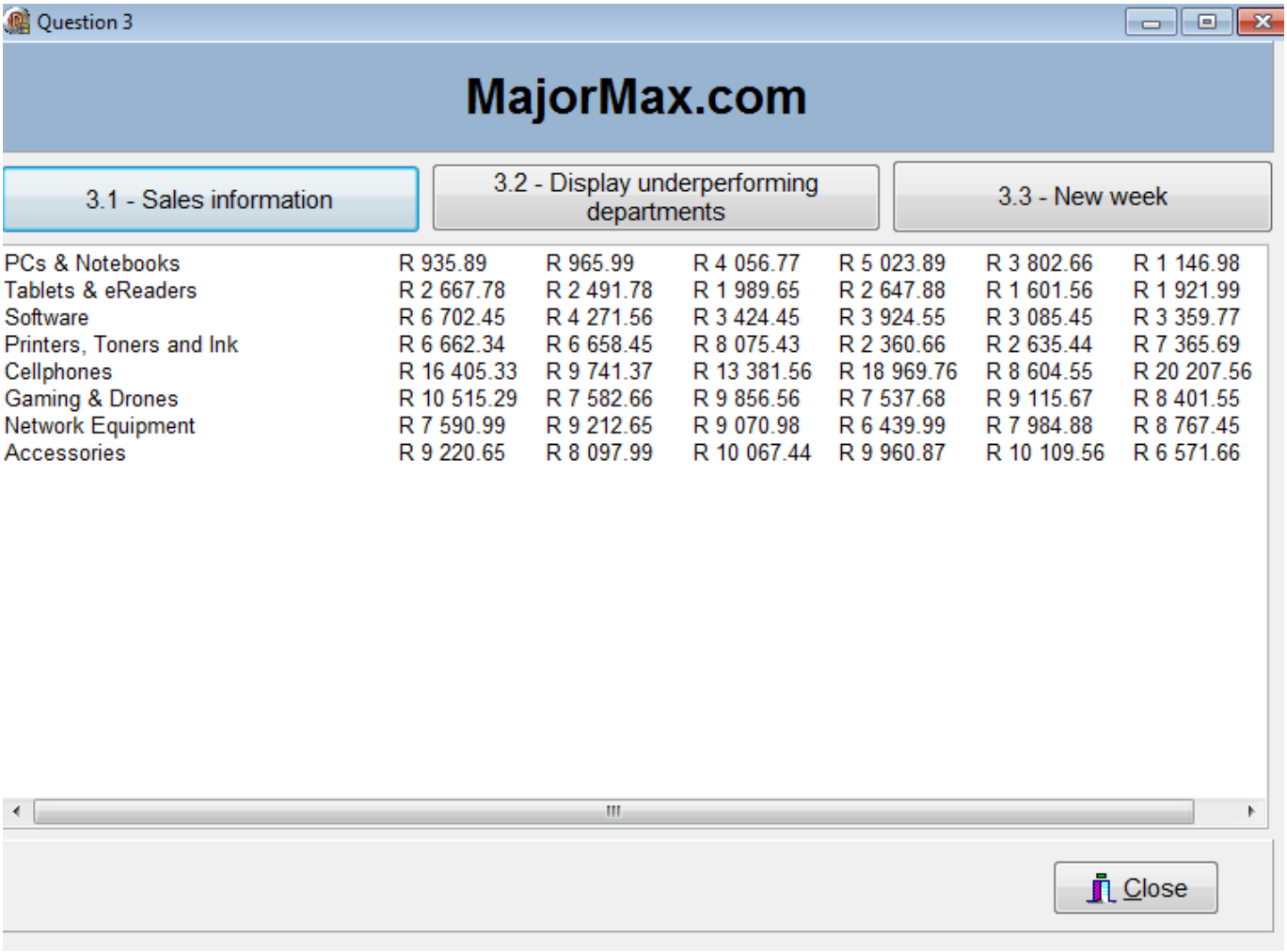
What is intresting is that it also has 8 elements, like the rows in our 2D array. So then, perhaps we could fit into our code ? Remember to output the values from arrDeparments, we would need something like :

```
for i:= 1 to 8 do
    redQ3.Lines.Add(arrDepartments[i]);
```

We change our code by only one line :

```
procedure TfrmQuestion3.btnQ3_1Click(Sender: TObject);
var row, col : integer;
str : string;
begin
  for row := 1 to 8 do
    begin
      str := '';
      for col := 1 to 6 do
      begin
        str := str + floattostrf(arrSales[row,col],ffcurrency,10,2)+ #9;
      end;
      redQ3.Lines.Add(arrDepartments[row] + #9 + str);
    end;
end;
```

Which outputs :



From this, we can see we are almost there - just to add in the upper headings:

Final code for this question :

```
procedure TfrmQuestion3.btnQ3_1Click(Sender: TObject);
var row, col : integer;
str : string;
begin

redQ3.Lines.Add('Department'+#9+'Week1'+#9+'Week2'+#9+'Week3'+#9+'Week4'+#9
'Week5'+#9+'Week6');
  for row := 1 to 8 do
    begin
      str := '';
      for col := 1 to 6 do
      begin
        str := str + floattostrf(arrSales[row,col],ffcurrency,10,2)+ #9;
      end;
      redQ3.Lines.Add(arrDepartments[row] + #9 + str);
    end;
end;
```

Note The memo has a different way of answering the question - you can follow which way is easier.

Question 3.2

3.2 **Button [3.2 – Display underperforming departments]**

A report of all underperforming departments per week is required. A department is underperforming when their sales figure is lower than the average sales for all the departments for that week.

Display the report in the output component provided, with suitable headings. All monetary values must be displayed in currency format with TWO decimal places.

Example of output for the first three weeks using the original data:

Underperforming departments per week:	
Week 1: Average sales figure: R 7 587.59	
PCs & Laptops	R 935.89
Tablets & eReaders	R 2 667.78
Software	R 6 702.45
Printers, Toners and Ink	R 6 662.34
Week 2: Average sales figure: R 6 127.81	
PCs & Laptops	R 965.99
Tablets & eReaders	R 2 491.78
Software	R 4 271.56
Week 3: Average sales figure: R 7 490.35	
PCs & Laptops	R 4 056.77
Tablets & eReaders	R 1 989.65
Software	R 3 424.45

This question wants to make some sort of report. We have to create a report based on when a department is Underperforming.

A depeartment is underperforming when:

- Its sales figure is lower than the average sales for every department per week

This raises the question how do we get the "average sales for every department per week"

If you look at question 3.1 - you will see that we need to add up the colums and divide by 8.

Refer to this section Add the columns of a 2D Array

The first thing we need to do is work out the average per week.

Lets get our our basic loops in place :

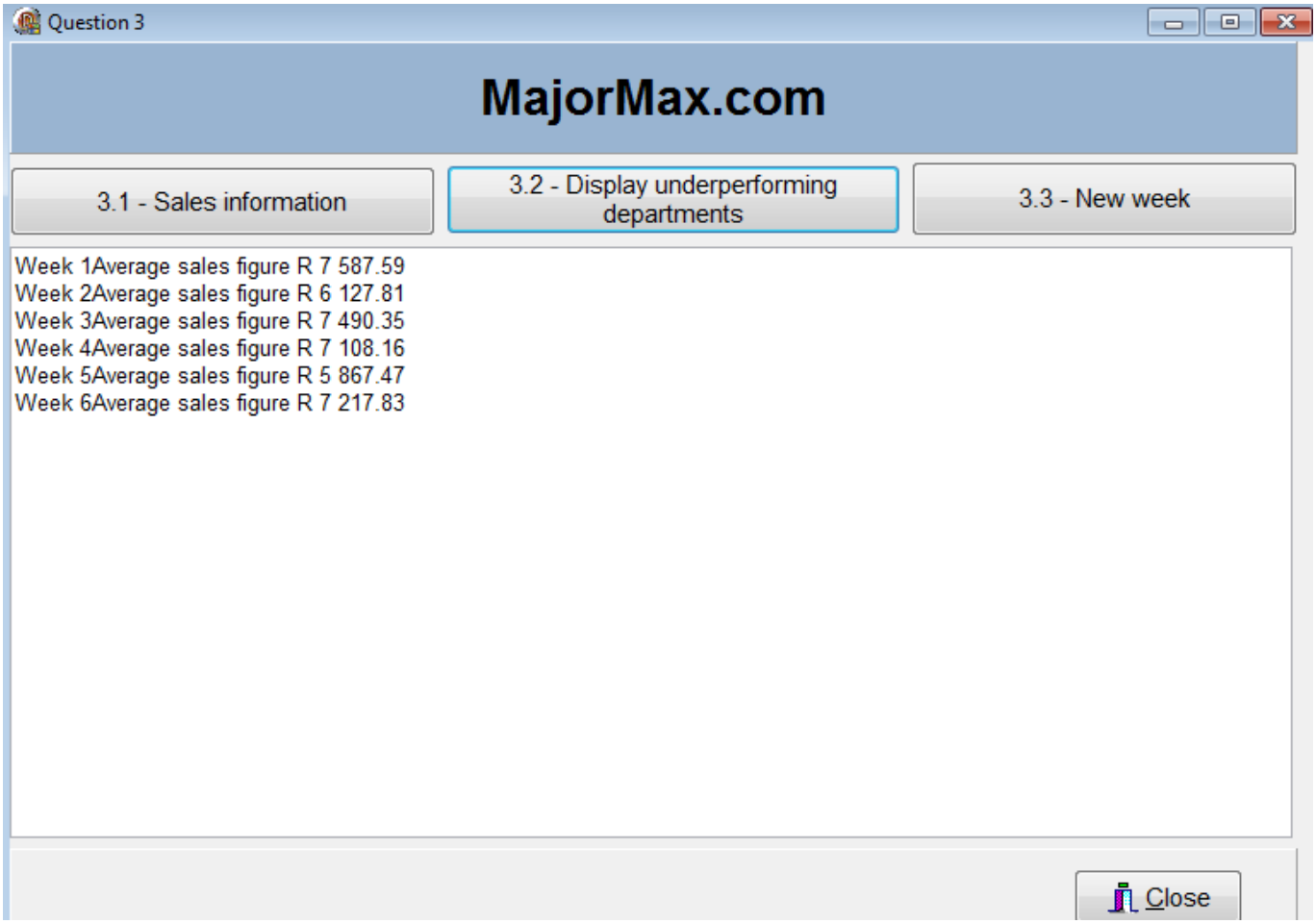
```
procedure TfrmQuestion3.btnQ3_2Click(Sender: TObject);
var row,col : integer;
begin
  //Question 3.2
  redQ3.Clear;
  for col:= 1 to 6 do
    begin
      for row := 1 to 8 do
        begin

        end;
    end;
end;
```

We now calcuate the average per week :

```
procedure TfrmQuestion3.btnQ3_2Click(Sender: TObject);
var row,col : integer;
average : real;
begin
  //Question 3.2
  redQ3.Clear;
  average := 0;
  for col:= 1 to 6 do
    begin
      average := 0;
      for row := 1 to 8 do
        begin
             average := average + arrSales[row,col];
        end;
        average := average/8;
        redQ3.Lines.Add('Week '+ inttostr(col) + 'Average sales figure '+
floattostrf(average,ffcurrency,10,2));
```

```
    end;
end;
```

Which would output :



Now we have to compare each departments weekly sale figure, to the average and if it is below, output it. And if that sale value is less than the average per week, output it to the rich edit.

```
for row := 1 to 8 do
        begin
         if (arrSales[row,col] < average) then
         begin
           redQ3.Lines.Add(arrDepartments[row] + #9 +
floattostrf(arrSales[row,col],ffcurrency,10,2));
         end;
        end;
```

The complete code for this function now looks like this :

```
procedure TfrmQuestion3.btnQ3_2Click(Sender: TObject);
var row,col : integer;
average : real;
begin
```

```
    //Question 3.2
    redQ3.Clear;
    average := 0;
    for col:= 1 to 6 do
      begin
        average := 0;
        for row := 1 to 8 do
          begin
               average := average + arrSales[row,col];
          end;
          average := average/8;
          redQ3.Lines.Add('Week '+ inttostr(col) + ' Average sales figure '+
  floattostrf(average,ffcurrency,10,2));
        for row := 1 to 8 do
          begin
           if (arrSales[row,col] < average) then
           begin
             redQ3.Lines.Add(arrDepartments[row] + #9 +
  floattostrf(arrSales[row,col],ffcurrency,10,2));
           end;
          end;
          redQ3.Lines.Add(#13#10);
      end;
  end;
```