Aqua.md

Introduction

The Aqua question is an exam question from the Department of Education, for Information Technology Paper 1 - 2016.

Must Do

I would highly advise you, to attempt these questions on your own. Compare them to the answers listed in this document. Use this document as a tutorial to do the exam question, and finally look at the memo and compare answers.

Note The memo has different answers, to the answers listed in this document.

Given Code

The following code is provided in the working files given to you.

Parallel Arrays

In the question paper, you will see these two array referred to as ***parallel*** arrays. It can be understood from looking at the code, that it means these two array arrActivities and arrCodes have the same number of elements.

```
arrCodes: array [1 .. 10] of char = (
    'W',
    'A',
    'S',
    'R',
    'X',
    'D',
    'H',
    'P',
    'T',
    'L'
  );
  arrActivities: array [1 .. 10] of String = (
    'Water park',
    'Aquarium',
    'Sea',
    'Restaurants',
    'Shopping',
    'Diving',
    'Help desk',
    'Penguin park',
    'Shark tank',
    'Dolphin shows'
  );
```

2D Array

The 2D array is presented as :

```
 arrActCodes: array [1 .. 3, 1 .. 4] of String = (('DXWAT', 'HRDST',
'STWLP',
      'RDT'), ('SWA', 'SRXD', 'LWXH', 'SHA'), ('WLSR', 'AT', 'DATX',
'HW'));
```

But if this seems confusing to you, remember there are

- 1..3 : 3 Rows
- 1..4 : 4 Columns

We could have alternatively re-written this array, with some extra formatting:

```
arrActCodes: array [1 .. 3, 1 .. 4] of String =
    (('DXWAT', 'HRDST', 'STWLP','RDT'),
    ('SWA', 'SRXD', 'LWXH', 'SHA'),
    ('WLSR', 'AT', 'DATX', 'HW'));
```

Global Variables

Two integer variables :

```
  iTerminal: integer  = 1;
  iDirection: integer = 1;
```

BitButtons

Clicking on the Bitbuttons, "Terminal 1, Terminal 2, Terminal 3" and "North", "East", "South", "West" on the Gui will assign a value to the Global Variables.

NOTE It is interesting to see how this question made use of inserting an image onto the GUI via the bitButton. As an additional task, you should try recreate this by creating a form, and using a bitButton to insert an image via the "Glyph" property

Finally, note that if no button is selected on the GUI, the global variables are assigned default values (in their initializations)

```
procedure TfrmQuestion3.btnTerminal1Click(Sender: TObject);
begin
  iTerminal := 1;
```

```
end;

procedure TfrmQuestion3.btnTerminal2Click(Sender: TObject);
begin
  iTerminal := 2;
end;

procedure TfrmQuestion3.btnTerminal3Click(Sender: TObject);
begin
  iTerminal := 3;
end;

procedure TfrmQuestion3.btnNorthClick(Sender: TObject);
begin
  iDirection := 1;
end;

procedure TfrmQuestion3.btnSouthClick(Sender: TObject);
begin
  iDirection := 2;
end;

procedure TfrmQuestion3.btnEastClick(Sender: TObject);
begin
  iDirection := 3;
end;


procedure TfrmQuestion3.btnWestClick(Sender: TObject);
begin
  iDirection := 4;
end;
```

Question 3.1

3.1 **Button [3.1 – Activity/Facility codes for all terminals and directions]**

The program must display the content of the two-dimensional array **arrActCodes** neatly in rows and columns. Display the directions as column headings and the terminals as row labels.

Example of output:

	North	South	East	West
Terminal 1	DXWAT	HRDST	STWLP	RDT
Terminal 2	SWA	SRXD	LWXH	SHA
Terminal 3	WLSR	AT	DATX	HW

This question we are printing the 2D array, with some formatting in the richedit.

We will approach this question systematically, doing bit by bit.

Adding Formatting

First, we have to get our formatting correct. Unfortunately, setting the widths is a trial an error experience, which you will have to practice in your own time to get correct.

```
// add formatting to rich edit
  redQ3.Paragraph.TabCount := 4;
  redQ3.Paragraph.Tab[0] := 80;
  redQ3.Paragraph.Tab[1] := 150;
  redQ3.Paragraph.Tab[2] := 200;
  redQ3.Paragraph.Tab[3] := 250;
```

The tabcount property represents the directions North, South, East and West.

Adding the headings

```
//add the upper headings
  redQ3.Lines.Add(#9+'North'+#9+'South'+#9+'East'+#9+'West');
```

Printing the 2D array values

In this part of the question, we used low, and high.

Note You need to experiment with low, and high and to understand what these functions do.

- Low : Returns the lower bound of an array
- High : Returns the upper bound of an array

Before I used the low, and high functions for the two loops required to print the 2D array, I ran some example code to see what the low and high functions returns :

```
  //low = 1 high = 3

redQ3.Lines.Add(inttostr(low(arractcodes))+#9+inttostr(high(arractcodes)));
  //low = 1 high = 4

redQ3.Lines.Add(inttostr(low(arractcodes[1]))+#9+inttostr(high(arractcodes[
1])));
  //low = 1 high = 4

redQ3.Lines.Add(inttostr(low(arractcodes[2]))+#9+inttostr(high(arractcodes[
2])));
```

If the low and high functions are confusing to you, then just constant integers numbers like this :

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

    end;
end;
```

Final Code :

```
for row := low(arrActCodes) to high(arrActCodes) do
  begin
    str := '';
    str := 'Terminal' + inttostr(row) + #9;
    for col := low(arrActCodes[row]) to high(arrActCodes[row]) do
    begin
      str := str + arrActCodes[row,col] + #9;
    end;
    redQ3.Lines.Add(str);
  end;
```

Question 3.2

3.2 **Button [3.2 – Activities/Facilities from a selected terminal and direction]**

The buttons that contain images must be used to select a terminal and direction. Code is provided to assign the selected terminal and direction to variables. The program must then use the supplied arrays to identify all the activities and facilities available on the selected route. Display the selected terminal and direction as a heading and a list of activities and facilities on the route selected.

Example of output if Terminal 2 and South are selected:

In this question, the user will:

- Select a terminal via the GUI
- Select a direction via the GUI

** Note ** The global variables iDirection and iTerminal are used to capture the user input via the the given code. If the user does not select anything, iDirection := 1 and iTerminal := 1 by default.

Once the user selects a terminal and direction, we then have print the corresponding activities from the 2D Array arrActCodes.

Again we will do this question systematically :

Printing the heading

First, we clear the rich edit to make it ready for our output.

```
redQ3.Clear;
```

Next we use a ***case*** statement, to map the integer iDirection to a corresponding string variable :

```
//based on user selection(iDrection) assign direction
  case iDirection of
  1 : sDirection := 'North';
  2 : sDirection := 'South';
  3 : sDirection := 'East';
  4 : sDirection := 'West';
  end;
```

And then, we can output our first line which is the heading :

```
// Output Heading
  redQ3.Lines.Add('Terminal '+inttostr(iTerminal)+', '+sDirection);
```

Getting the User Input from the 2D Array

In the question paper, they give an example of Terminal 2, Direction = South. In Question 3.1 we printed the 2D array.

Based on our 2D array, we can see that Terminal 2 and Direction = South equals

```
SRXD
```

Which is a value in the 2D array arrActCodes.

1. So the user will press Terminal 2, and Direction South on the GUI
2. The global variables iTerminal and iDirection will be assigned values with the given code, based on user input
3. Our task then is to retrieve the corresponding value from the 2D array : ie : SRXD

Remember, when printing values from a 2D array, we need a row and col. Example, to print the first value which is located at arrActCodes[1,1]. In this case, we have two integer variables iTerminal which

corresponds to the row, and iDirection which corresponds to the column. Therefore, we store our value from the 2D array in a new variable called sActCode :

```
// get the srxd corresponding code from arrActCodes
  for col := Low(arrActCodes) to High(arrActCodes)do
  begin
    for row := Low(arrActCodes[col]) to High(arrActCodes[col]) do
    begin
         sActCode := arrActCodes[iTerminal,iDirection];
    end;
  end;
```

Printing the activities

Our final step now, is to print the corresponding activities. Ie: SRXD maps to Sea, Restaurant, Shopping and Diving. These values are stored in the single dimensional array arrActivities.

Outer loop

What we are going to do now, is have a loop that will start at the beginning of our string "SRXD" and then work its ways through to then end:

```
for i := 1 to Length(sActCode)do
    begin

    end;
```

Inner loop

THen we are going to have a inner loop, that checks if each letter in SRXD ie "S", then "R" then "X", then "D" is found in arrCodes. If it is found in arrCodes, then print the value from arrActivities at that position :

```
for i := 1 to Length(sActCode)do
    begin
      // inner j loop will compare specific letter from sActCode[i] against
      // letters in arrCodes  sActCode[i] = arrCodes[j] and if they are
equal
      // then add to rich edit the corresponding item (because both
arrCodes
      // and arrActivities eqal in length) from arrActivities
      for j:= Low(arrCodes) to High(arrCodes) do
        begin
          if sActCode[i] = arrCodes[j] then
            redQ3.Lines.Add(arrActivities[j]);
        end;
    end;
```

```
    // memo presents a more elegant solution*
end;
```

Our complete code for this question now looks like this :

```
procedure TfrmQuestion3.btnQues32Click(Sender: TObject);
var sDirection,sActCode : string;
    row,col,i,j : integer;
begin
  // Quetion 3.2
  redQ3.Clear;

  //based on user selection(iDrection) assign direction
  case iDirection of
  1 : sDirection := 'North';
  2 : sDirection := 'South';
  3 : sDirection := 'East';
  4 : sDirection := 'West';
  end;

  // Output Heading
  redQ3.Lines.Add('Terminal '+inttostr(iTerminal)+', '+sDirection);

  // get the srxd corresponding code from arrActCodes
  for col := Low(arrActCodes) to High(arrActCodes)do
  begin
    for row := Low(arrActCodes[col]) to High(arrActCodes[col]) do
    begin
         sActCode := arrActCodes[iTerminal,iDirection];
    end;
  end;

  // outer i loop will work through selected code
  for i := 1 to Length(sActCode)do
    begin
      // inner j loop will compare specific letter from sActCode[i] against
      // letters in arrCodes  sActCode[i] = arrCodes[j] and if they are
equal
      // then add to rich edit the corresponding item (because both
arrCodes
      // and arrActivities qqual in length) from arrActivities
      for j:= Low(arrCodes) to High(arrCodes) do
        begin
          if sActCode[i] = arrCodes[j] then
            redQ3.Lines.Add(arrActivities[j]);
        end;
    end;
end;
```

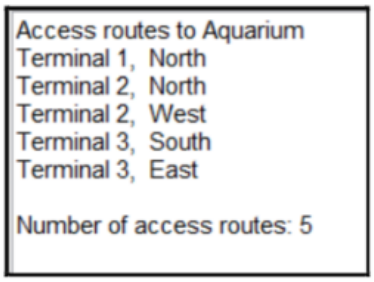
Question 3.3

3.3 **Button [3.3 – Access routes to selected activity/facility]**

Once the user selects a specific activity/facility from the combo box provided, the user must be able to view all access routes to that specific activity/facility. Display the terminal number and the direction for each access route for the visitor to be able to reach the activity/facility selected.

The total number of access routes leading to the activity selected must be determined and displayed.

Example of output if Aquarium was selected from the combo box:



In this question, the user will select a value like "Aquarium" from a comboxbox, and then we need to print our the corresponding Terminal Numbers and Directions from the 2D array arrActCodes. So :

1. User selects a value from comboxbox "Aquarium"
2. We take "Aquarium", and get the corresponding code from arrCodes. In this case Aquarium is A
3. Once we have the "A", we then search in our 2D array for all occurrences of the A, and if that A is found, print the row = Direction and the column = Terminal

Step 1 : Print Our Headings and storing using input

First step is to capture the user input from the combobox.

```
  // get the users selection
  sSelection := cmbQues3.Text;
```

Next we clear our output, and add in our heading :

```
// output heading
  redQ3.Clear;
  redQ3.Lines.Add('Access routes to '+ sSelection);
```

Take note, we will also assign a counter variable which we will use later, which will count for example how many times "A" for "Aquarium" appeared in the 2D array.

```
// counter variable
  counter := 0;
```

Step 2 : Convert Aquarium into A

So we go a simple loop, that will go through arrActivities, and if "Aquarium" is found, then get the corresponding value from arrCodes and store it in sCode.

```
// convert from user input to code ie: Aquarium to 'A'
  // loop through arrActivities array
  for i:= Low(arrActivities) to High(arrActivities)do
    begin
      // if user selection equal one of the items from arrActivities
      if sSelection = arrActivities[i] then
        // store the corresponding code from arrCodes into sCode
        sCode := arrCodes[i];
    end;
```

So sCode will have the value "A" stored in it.

Step 3 : Searching for "A" in the 2D Array

We need to loop through the values in our 2D array. For example, in the first item stored in the 2D Array "SRXD" does this value contain "A" (sCode)?

Our two normal 2D array loops will enable us to access all the values in the 2D Array :

```
for row := Low(arrActCodes) to High(arrActCodes)  do
    begin
      for col := Low(arrActCodes[row]) to High(arrActCodes[row]) do
        begin

        end;
    end;
```

The value of SRXD will retrieved by:

```
arrActCodes[row,col]
```

Once the value from the 2D array is accessed, we then need to work through it ie: Go from "S" to "D" in SRXD and compare each character to our sCode(which is a for Aquarium in this example.) *** Note*** this is the i loop below

```
 for i:= 1 to Length(arrActCodes[row,col]) do
            begin
                if sCode = arrActCodes[row,col][i] then
                begin
                  //based on user selection(iDirection) assign direction
                  case col of
                  1 : sDirection := 'North';
                  2 : sDirection := 'South';
                  3 : sDirection := 'East';
                  4 : sDirection := 'West';
                  end;
                  redQ3.Lines.Add('Terminal' + inttostr(row) + ', ' +
sDirection);
                  inc(Counter);
                end;
            end;
```

Finally, we are going to output our counter variable :

```
// output number of access routes based on counter variable
    redQ3.Lines.Add(#13#10+'Number of access routes:'+inttostr(Counter));
```

Our complete code for this question :

```
procedure TfrmQuestion3.btnQues33Click(Sender: TObject);
var sSelection,sCode,str,sDirection : string;
    row,col,i,counter : integer;
    //iRow,iCol,iCount,iIndex : integer;
begin

  // Question 3.3

  // get the users selection
  sSelection := cmbQues3.Text;

  // output heading
  redQ3.Clear;
  redQ3.Lines.Add('Access routes to '+ sSelection);

  // counter variable
  counter := 0;

  // convert from user input to code ie: Aquarium to 'A'
  // loop through arrActivities array
  for i:= Low(arrActivities) to High(arrActivities)do
    begin
      // if user selection equal one of the items from arrActivities
      if sSelection = arrActivities[i] then
        // store the corresponding code from arrCodes into sCode
```

```
        sCode := arrCodes[i];
    end;

  // loop the arrActCodes array row-by-row (determined by picture in
question paper)
  //
  for row := Low(arrActCodes) to High(arrActCodes)  do
    begin
      for col := Low(arrActCodes[row]) to High(arrActCodes[row]) do
        begin
          // the two outer loops (row, and col) will return items like this
          // arrActCodes[row,col] = DXWAT)
          // our inner i loop then takes that item and works through it
          // comparing each item against sScode which was determined
          // above. If they equal ie : 'A' is found in DXWAT representing
          // aqarium, then output the Terminal + Direction
          // Terminal is determined by row, Direction by col
          for i:= 1 to Length(arrActCodes[row,col]) do
            begin
                if sCode = arrActCodes[row,col][i] then
                begin
                  //based on user selection(iDrection) assign direction
                  case col of
                  1 : sDirection := 'North';
                  2 : sDirection := 'South';
                  3 : sDirection := 'East';
                  4 : sDirection := 'West';
                  end;
                  redQ3.Lines.Add('Terminal' + inttostr(row) + ', ' +
sDirection);
                  inc(Counter);
                end;
            end;
        end;
    end;

    // output number of access routes based on counter variable
    redQ3.Lines.Add(#13#10+'Number of access routes:'+inttostr(Counter));


    // memo answer has a array called arrDirections
    // no! array in our question with that name
    // memo answer almost correct, small changes needed to get it to work
    // you can figure out what needs to be changed to get the correct
answer
    //
    {
    redQ3.Clear;
    iCount := 0;
    iIndex := cmbQues3.ItemIndex;
    redQ3.Lines.Add('Access routes to ' +
cmbQues3.Items[cmbQues3.ItemIndex]);
    for iRow := 1 to 3 do
        for iCol := 1 to 4 do
```

```
            begin
            if pos(arrCodes[iIndex + 1], arrActCodes[iRow,iCol]) > 0 then
            begin
                redQ3.Lines.Add('Terminal '+ inttostr(iRow) + ',
'+arrActivities[iCol]);
                inc(iCount);
            end;
            end;
    redQ3.Lines.Add(#13+ 'Number of access routes: ' + Inttostr(iCount));
    }
end;
```

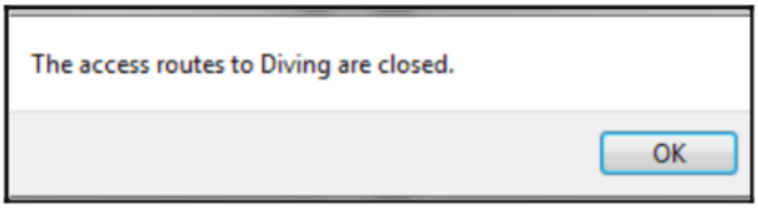
You might want to take note here that the memo, presents a different answer which I have taken the courtesy of typing out for you, since the .pdf has disabled copying text. There is an error, in the memo, which I have left for you to determine. Try fix the memo code.

Question 3.4

3.4 **Button [3.4 – Maintenance at selected activity/facility]**

The area where activities take place or facilities are provided may sometimes be closed due to maintenance. The user must select an activity/facility where maintenance must take place from the combo box provided. The program must remove all references to the selected activity/facility from the two-dimensional array and display a suitable message in a dialog box indicating that the information has been updated. The updated content of the two-dimensional array must be displayed in the output area.

Example of output if Diving was selected from the combo box:



Example of output after removing the letter D from the two-dimensional array **arrActCodes** due to maintenance that must be done on the diving facility:

Updated information:

	North	South	East	West
Terminal 1	XWAT	HRST	STWLP	RT
Terminal 2	SWA	SRX	LWXH	SHA
Terminal 3	WLSR	AT	ATX	HW

In this question, the user will select a value from the comboxbox, our task is to delete that value user input from the 2D array.

1. User selected some value like "Aquarium" from the combobox
2. We map that value to corresponding code from arrCodes ie: Aquarium to A.

3. Once we have the code, we will search the 2D array values ie: "SRXD" for the occurrence of "A", and if it exists in that value delete it

Step 1 : Get the user input

```
  // get user selection
  sSelection := cmbQues3.Text;
```

Step 2 : Convert the user input into a corresponding code

```
// convert from user input to code ie: Aquarium to 'A'
  // loop through arrActivities array
  for i:= Low(arrActivities) to High(arrActivities)do
    begin
      // if user selection equal one of the items from arrActivities
      if sSelection = arrActivities[i] then
        // store the corresponding code from arrCodes into sCode
        sCode := arrCodes[i];
    end;
```

Step 3 : Compare each value from 2D array against code, and delete if exists

Since we are accessing values from the 2D array, immediately open up our two loops :

```
 for row := Low(arrActCodes) to High(arrActCodes)  do
    begin
      for col := Low(arrActCodes[row]) to High(arrActCodes[row]) do
        begin
        end;
    end;
```

Then we will have a third inner loop, that will work its way from the start of the value from arrActCode to the end Ie: Value from arrActCode[row,col]:

```
arrActCode[row,col] // which will return something like SRXD
```

Inner loop to work from start of value to end : Ie: From "S" ... to "D"

```
for j:= 1 to Length(arrActCodes[row,col]) do
```

Then we use a if statement, to compare if each letter is equal to sCode Ie: In SRXD is "S" = "A" for aquarium. No? IF "R" = "A" for aquarium .. and so forth.

```
    if sCode = arrActCodes[row,col][j] then
```

WHen the "A" is found in one of the values from arrActCodes, then delete that occurrence :

```
Delete(arrActCodes[row,col],pos(sCode,arrActCodes[row,col]),1);
```

The delete fucntion takes three parameters

```
Delete(param1,param2,param3)
```

- param1 : string you are deleting from. In our case arrActAcodes[row,col]
- param2 : starting position of what you want to delete. In our case:

```
pos(sCode,arrActCodes[row,col])
```

The pos function takes two parameters

```
pos(param4,param4)
```

- param4 : character you want to locate. In our case sCode
- param5 : string you want to search. In our case arrActCodes[row,col]

And then finally, third parameter for Delete :

- param3 : how many positions. In our case 1

Our last piece of code we need to add in is a ShowMessage :

```
//Show message to confirm delete
  ShowMessage('The access routes to '+ sSelection + ' are closed.');
```

And then call our button 3.1 to re-print the 2D array :

```
// re-output the the 2D Array
  btnques31.Click;
```

Our complete code for this question now looks like :

```
procedure TfrmQuestion3.btnQues34Click(Sender: TObject);
var sSelection,sCode : string;
    row,col,i,j : integer;
begin
  // Question 3.4

  // get user selection
  sSelection := cmbQues3.Text;

  // convert from user input to code ie: Aquarium to 'A'
  // loop through arrActivities array
  for i:= Low(arrActivities) to High(arrActivities)do
    begin
      // if user selection equal one of the items from arrActivities
      if sSelection = arrActivities[i] then
        // store the corresponding code from arrCodes into sCode
        sCode := arrCodes[i];
    end;

  // loop through our 2D Array
  for row := Low(arrActCodes) to High(arrActCodes)  do
    begin
      for col := Low(arrActCodes[row]) to High(arrActCodes[row]) do
        begin
          for j:= 1 to Length(arrActCodes[row,col]) do
            begin
                if sCode = arrActCodes[row,col][j] then
                begin

//Delete(arrActCodes[row,col],pos(sCode,arrActCodes[row,col]),pos(sCode,arr
ActCodes[row,col])+1);

Delete(arrActCodes[row,col],pos(sCode,arrActCodes[row,col]),1);
                end;
            end;
        end;
    end;

  //Show message to confirm delete
  ShowMessage('The access routes to '+ sSelection + ' are closed.');

  // re-output the the 2D Array
  btnques31.Click;

end;
```