Library

The library question is a 2D array question taken from Information-Technology P1 - 2018.

You can download the full paper here : https://www.education.gov.za/Curriculum/NationalSeniorCertificate(NSC)Examinations/2018NSCJunepastpapers.aspx

Included Code

Before we start our programming, take some time to look at the provided code.

Global Varaibles

There are two global variables, take note of the **const** keyword, meaning that these variables values cannot be changed :

```
const
  iMaxRow = 3;
  iMaxCol = 6;
```

Arrays Declared

One 2D array was given :

```
arrPlacements: array [1 .. 3, 1 .. 6] of String =
   (('Nkosi', 'Simon','Anette', 'Bongi', 'Tamzin', 'Trevor'),
    ('Anette', 'Tamzin','Simon', 'Trevor', 'Bongi', 'Nkosi'),
    ('Bongi', 'XXXXX', 'Trevor','Nkosi', 'Nkosi', 'Tamzin'));
```

And then a single dimensional array also :

```
arrStaff: array [1 .. 6] of String = (
   'Trevor',
   'Nkosi',
   'Tamzin',
   'Anette',
   'Bongi',
   'Simon'
 );
```

Procedure Display

They have included a procedure called display:

```
    procedure Display;
```

Which is declared under the **type** keyword:

```
type
  TfrmQ3 = class(TForm)
    redOutputQ3: TRichEdit;
    grpQ3_2: TGroupBox;
    btnQ3_2: TButton;
    grpQ3_1: TGroupBox;
    cmbStaff: TComboBox;
    Panel1: TPanel;
    Image1: TImage;
    procedure btnQ3_2Click(Sender: TObject);
    procedure Display;
    procedure cmbStaffChange(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

And then the definition for the procedure display looks like this :

```
procedure TfrmQ3.Display;
Var
  sLine: String;
  iCnt, iR, iC: integer;
begin
  redOutputQ3.Clear;
  sLine := '' + #9#9;
  for iCnt := 1 to iMaxCol do
  begin
    sLine := sLine + 'Day ' + IntToStr(iCnt) + #9#9;
  end;
  redOutputQ3.Lines.Add(sLine);

  for iR := 1 to iMaxRow do
  begin
    sLine := 'Library ' + IntToStr(iR) + #9;
    for iC := 1 to iMaxCol do
    begin
      sLine := sLine + arrPlacements[iR, iC] + #9#9;
    end;
    redOutputQ3.Lines.Add(sLine);
  end;
end;
```

This piece of code outputs the headings (ie: all the days)

```
sLine := '' + #9#9;
  for iCnt := 1 to iMaxCol do
  begin
    sLine := sLine + 'Day ' + IntToStr(iCnt) + #9#9;
```

And this piece of code outputs the values in the 2D array placements:

```
for iR := 1 to iMaxRow do
  begin
    sLine := 'Library ' + IntToStr(iR) + #9;
    for iC := 1 to iMaxCol do
    begin
      sLine := sLine + arrPlacements[iR, iC] + #9#9;
    end;
    redOutputQ3.Lines.Add(sLine);
  end;
```

Now if you look at its strucutre, it is very similar to what we where doing in our lessons :

```
for row := 1 to 3 do
    begin
        str := '';
        for row := 1 to 4 do
        begin
            str := str + inttostr(arrNums[row,col]);
        end;
        redOutput.Lines.Add(str);
    end;
```

If you are confused about the display procedure, refer back to the 2D-array document.

When the form loads, the display procedure is automatically called :

```
procedure TfrmQ3.FormActivate(Sender: TObject);
Begin
  Display;
end;
```

3.1

Question

3.1 Combo box [3.1 - Select name]

A staff member may request a schedule of his/her duties.

When a name is selected from the combo box **cmbStaff**, the following information must be displayed in the output component provided:

- A heading with the name of the staff member
- The work schedule details of the selected staff member

The format of the work schedule details is as follows:

```
Day <day number>-Library#<library number>
```

Example of output if staff member Tamzin was selected from **cmbStaff**:

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Library 1	Nkosi	Simon	Anette	Bongi	Tamzin	Trevor
Library 2	Anette	Tamzin	Simon	Trevor	Bongi	Nkosi
Library 3	Bongi	XXXXX	Trevor	Nkosi	Nkosi	Tamzin

Tamzin's schedule
Day 2-Library#2
Day 5-Library#1
Day 6-Library#3

How do we approach this question ? Before you continue this document, think about how you would do it, or atleast try.

Lets look at the output required and work from there. A sample output line :

```
Day 2-Library#2
```

The question is where does this information for the output line come from ? On careful examiniaton, you will see that all of the information comes from the 2D array.

- Day 2 : is the column number
- Library 2 : is the row number

Our two loops

Lets get our two loops in place :

```
procedure TfrmQ3.cmbStaffChange(Sender: TObject);
var row,col : integer;
begin
  //Question 3.1
  for row := 1 to iMaxRow do
    begin
      for col := 1to iMaxCol do
      begin
```

```
      end;
    end;
end;
```

Remember with this loop strucutre, row as the outer loop, and col as the inner loop the array gets accessed in the following manner :

```
Nkosi
Nkosi Simon
Nkosi Simon Anette
Nkosi Simon Anette Bongi
Nkosi Simon Anette Bongi Tamzin
Nkosi Simon Anette Bongi Tamzin Trevor
```

Whats happening here is that the loop structure accesses the array row by row.

If you look back at the question, the example shows the output required for **Tamzin**

If we are using this current loop strcutre(row as outer, col as inner) the first occurence of Tamzin would be founda at : Row = 1, Col = 5

And as seen in the question paper, that would represent:

```
Day 5-Library#1
```

But above **Day 5-Library#1** was

```
Day 2-Library#2
```

Which means, we have to change the manner in which we access this 2D array. Therefore, we will access it column-by-column! By changing the loop strucutre:

```
procedure TfrmQ3.cmbStaffChange(Sender: TObject);
var row,col : integer;
str : string;
begin
  //Question 3.1
  for col := 1 to iMaxCol do
    begin
      for row := 1to iMaxRow do
      begin

      end;
```

```
    end;
end;
```

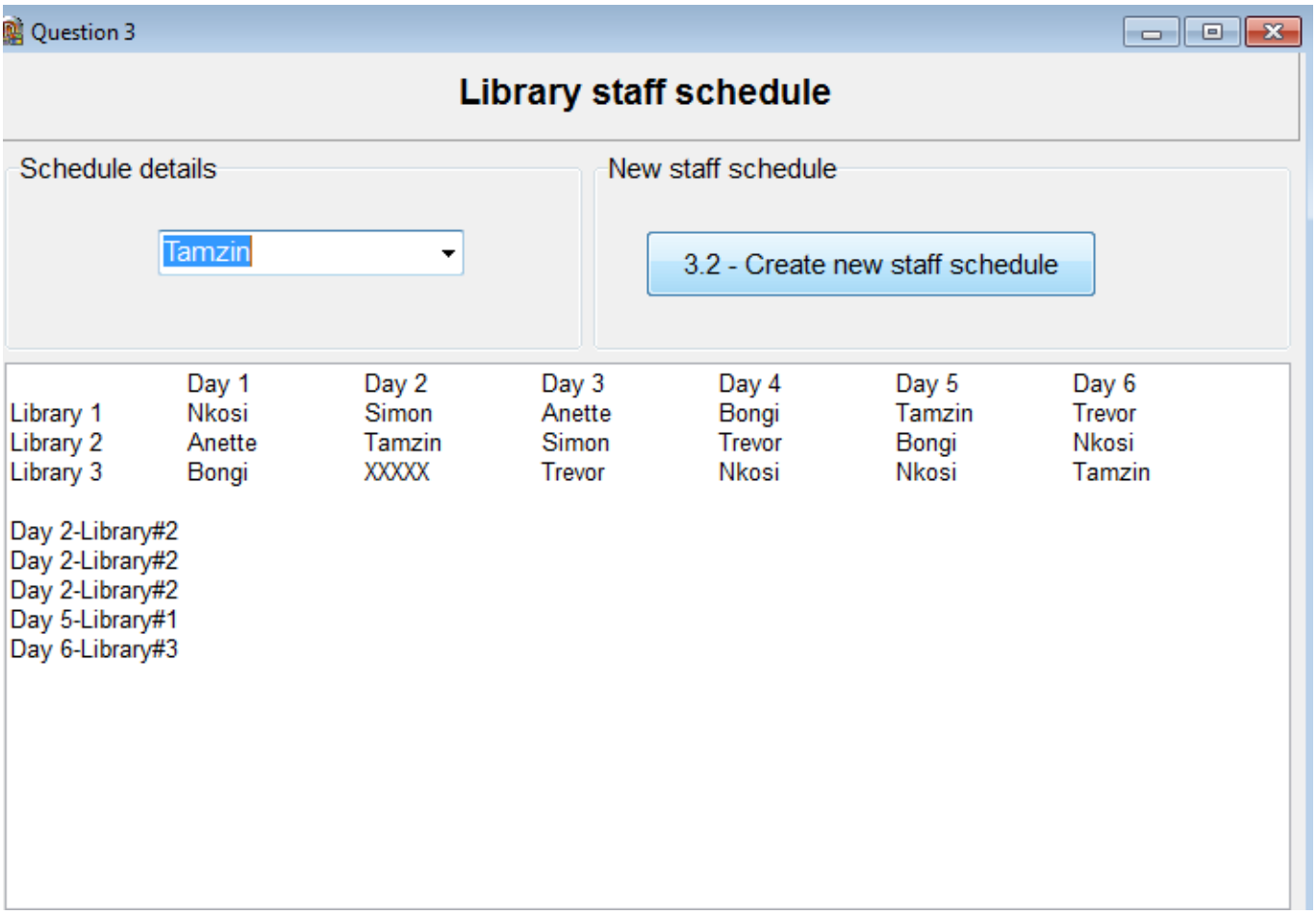
Now we will add in our if statement to locate the staff memember selected in the combo-box:

```
    if arrPlacements[row,col] = cmbStaff.Text then
```

Our code looks like this :

```
procedure TfrmQ3.cmbStaffChange(Sender: TObject);
var row,col : integer;
str : string;
begin
  //Question 3.1
  for col := 1 to iMaxCol do
    begin
      for row := 1to iMaxRow do
      begin
        if arrPlacements[row,col] = cmbStaff.Text then
          str := 'Day ' + inttostr(col) + '-' + 'Library#' + inttostr(row);
      end;
      redOutputQ3.Lines.Add(str);
    end;
end;
```

Unfortunately, our code is not 100% correct as seen below :



First of all we are missing the heading, ie: "Tamzin's schedule" and second we have some code that is repeating.

The heading

To add the heading, you have to include a apostrophe :

```
Tazmin's schedule
```

I used this link, which stated that you can include a apostrophe by adding in four apostrophes:

```
''''
```

https://stackoverflow.com/questions/14259687/how-do-i-put-apostrophes-in-a-string

```
redOutputQ3.Lines.Add(#13 + cmbStaff.Text + ''''+ 's schedule');
```

And recall that #13 adds in a new line.

Removing the duplicates

To remove the duplicates, we where seeing -the line that outputs the richedit has, to be included in the if statement. Our final code now looks like this :

```
procedure TfrmQ3.cmbStaffChange(Sender: TObject);
var row,col : integer;
str : string;
begin
  //Question 3.1
  redOutputQ3.Lines.Add(#13 + cmbStaff.Text + ''''+ 's schedule');
  for col := 1 to iMaxCol do
    begin
      for row := 1 to iMaxRow do
      begin
        if arrPlacements[row,col] = cmbStaff.Text then
        begin
          str := 'Day ' + inttostr(col) + '-' + 'Library#' + inttostr(row);
          redOutputQ3.Lines.Add(str);
        end;
      end;
    end;
end;
```

3.2

3.2 **Button [3.2 - Create new schedule]**

A one-dimensional array called **arrStaff** is provided and must be used to create a new schedule for staff members. The new schedule must be saved in the two-dimensional array **arrPlacements**.

Write code to compile a new placement schedule for staff members by populating the **arrPlacements** array with the names of staff members as follows:

Library 1: Each staff member will be placed on duty according to the order of the appearance of their names in the array **arrStaff**. The first staff member in the array will be assigned to Day 1, the second staff member to Day 2 and so on.

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Library 1	Trevor	Nkosi	Tamzin	Anette	Bongi	Simon

Library 2: Each staff member will be placed on duty in the reverse order of the contents of the array **arrStaff**. The first staff member in the array will be assigned to Day 6, the second staff member to Day 5 and so on:

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Library 1	Trevor	Nkosi	Tamzin	Anette	Bongi	Simon
Library 2	Simon	Bongi	Anette	Tamzin	Nkosi	Trevor

Library 3: Write code to use an input box to prompt the manager to enter a day number (1 to 6) on which Library 3 will be closed. Array **arrPlacements** must show 'XXXXX' for the day the library is closed. The staff will be allocated randomly for the remainder of the days to this library. A test must be done to ensure that the staff member randomly selected for Library 3 is not already allocated to Library 1 or Library 2 for that day.

Example of output if the manager entered Day 3 for Library 3 to be closed:

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Library 1	Trevor	Nkosi	Tamzin	Anette	Bongi	Simon
Library 2	Simon	Bongi	Anette	Tamzin	Nkosi	Trevor
Library 3	Anette	Anette	XXXXX	Nkosi	Simon	Tamzin

(20)

We have to replace the values in arraPlacements with the values in arrStaff.

If you look at the output, the way in the table is printed, is different, there is now a ***new line*** seperating the Library Rows.

In addition to that in our code for 3.2, it was empty, no display function was called.

Library 1

The row for library, must be populated with values from arrStaff

Lets begin by opening up our two loops :

```
procedure TfrmQ3.btnQ3_2Click(Sender: TObject);
var row, col : integer;
begin
  //Question 3.2
  for row := 1 to iMaxRow do
    begin
      for col := 1 to iMaxCol do
        begin

        end;
    end;
end;
```

We have to :

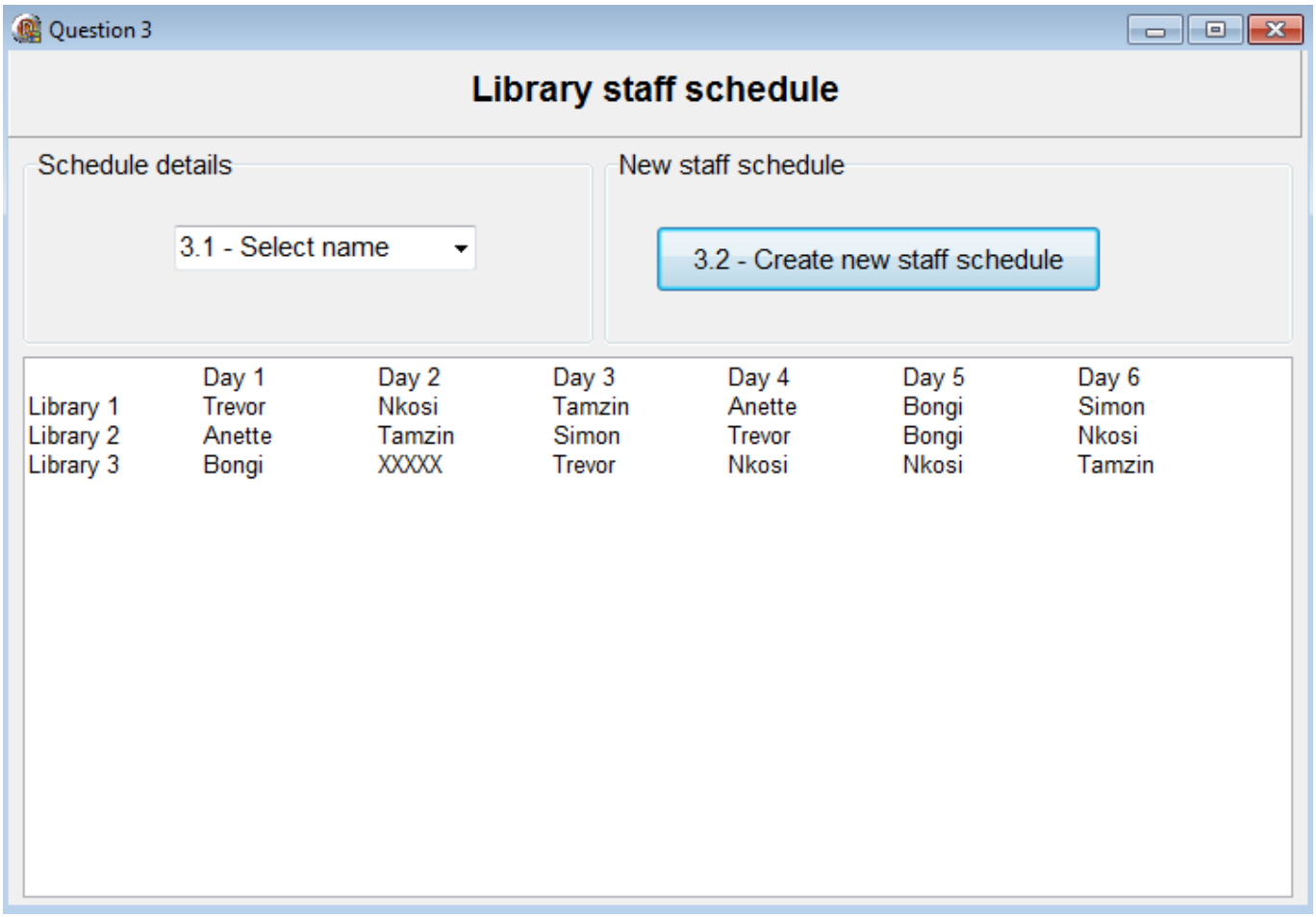
- First reassign arrPlacements with values from arrStaff
- Output arrPlacements

To reassign our values in arrPlacements, we can use a if statement :

```
          if row = 1 then
          begin
            arrPlacements[row,col] := arrStaff;
          end;
```

And we call the Display function at the end of our two loops, two see what our output looks like :

```
procedure TfrmQ3.btnQ3_2Click(Sender: TObject);
var row, col : integer;
begin
  //Question 3.2
  for row := 1 to iMaxRow do
    begin
      for col := 1 to iMaxCol do
        begin
          if row = 1 then
          begin
            arrPlacements[row,col] := arrStaff[col];
          end;
        end;
    end;
    Display;
end;
```



Library 2

Library 2 values, need to be stored in reverse order.

Remember our two loops:

```
for row := 1 to iMaxRow do
    begin
      for col := 1 to iMaxCol do
        begin

        end;
    end;
```

In this case, col starts from 1 and goes to iMaxCol which is 6 so it goes, col = 1, col = 2, col = 3 and so forth.

We took advantage of that order in library 1 when you just accessed the values in arrStaff.

```
 arrPlacements[row,col] := arrStaff[col];
```

Unfortunately, here we cannot do the same thing. In this part of the question, we have to access arrStaff in the order 6,5,4,3,2,1 to represent reverse order. We can accomplish our objective, by stating :

```
 arrPlacements[row,col] := arrStaff[7-col];
```

So on the first itereation of the col inner loop, col = 1 therefore

```
 arrPlacements[row,col] := arrStaff[7-1]; // arrStaff at position 6 which
is the last element
```

And on the second iteration of the col inner loop, col = 2, therefore :

```
 arrPlacements[row,col] := arrStaff[7-2]; // arrStaff at position 5 which
is the second to last element
```

Our code now for Library two looks like this :

```
procedure TfrmQ3.btnQ3_2Click(Sender: TObject);
var row, col : integer;
begin
  //Question 3.2
  for row := 1 to iMaxRow do
    begin
      for col := 1 to iMaxCol do
        begin
          if row = 1 then
          begin
            arrPlacements[row,col] := arrStaff[col];
          end
          else if row = 2 then
          begin
            arrPlacements[row,col] := arrStaff[7-col];
          end;
        end;
    end;
    Display;
end;
```

Library 3

We will break this question down in to parts. First of all lets add our inputBox, to get a user Input value which will represent the day where 'XXXXX' will be placed:

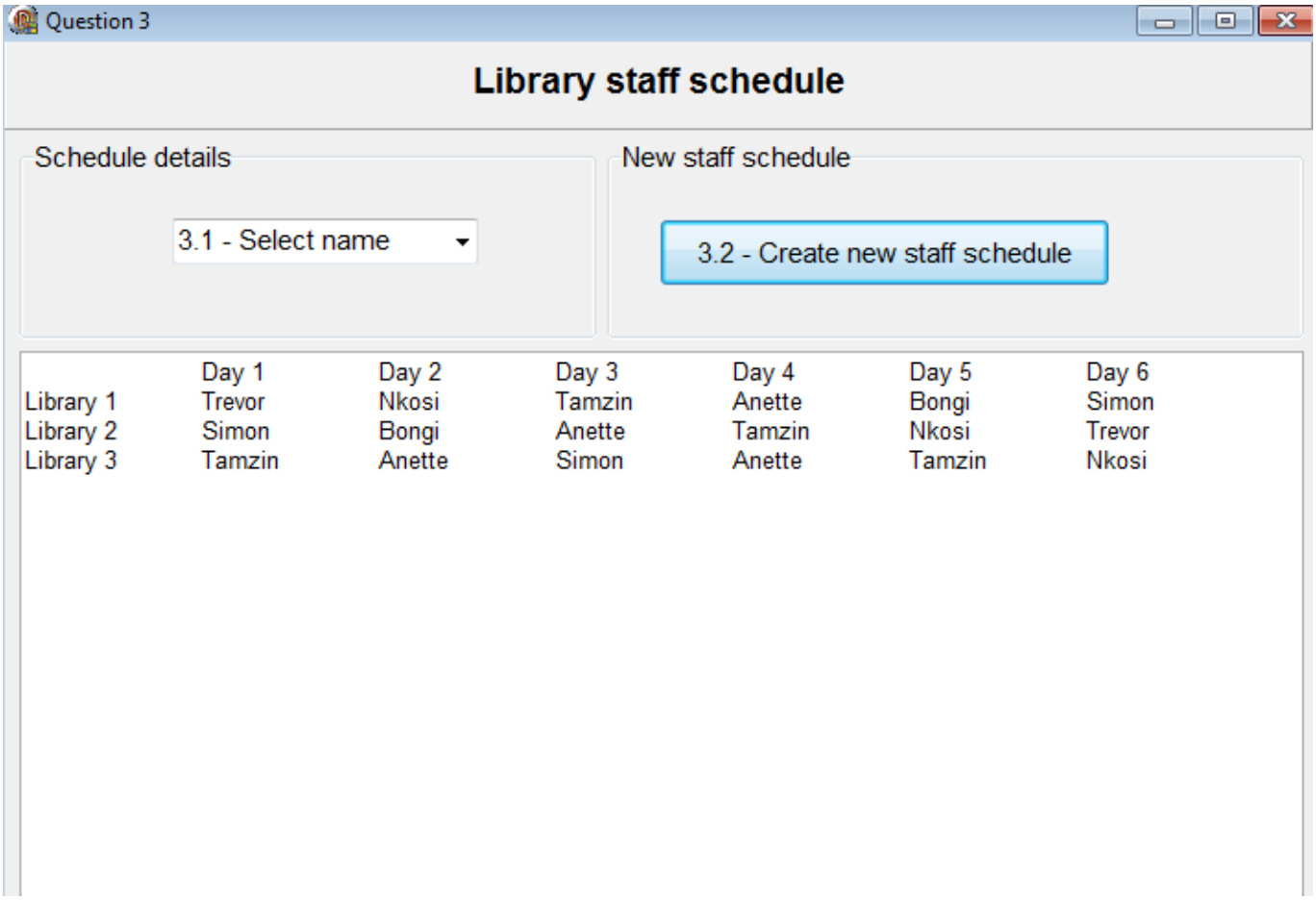
```
  sUserInput := InputBox('Enter in day library 3 will be closed','Library 3
Day Closed','1');
```

We will come back to the inputBox part later - the second part of the question states that the "staff have to allocated randomly". We know that there are 6 staff in the arrStaff array. We will have a random function therefore like this :

```
random(6) + 1
```

Our code now looks like this :

```
procedure TfrmQ3.btnQ3_2Click(Sender: TObject);
var row, col : integer;
sUserInput : string;
begin
  //Question 3.2
  sUserInput := InputBox('Enter in day library 3 will be closed','Library 3
Day Closed','1');
  for row := 1 to iMaxRow do
    begin
      for col := 1 to iMaxCol do
        begin
          if row = 1 then
          begin
            arrPlacements[row,col] := arrStaff[col];
          end
          else if row = 2 then
          begin
            arrPlacements[row,col] := arrStaff[7-col];
          end
          else  // if row = 3
          begin
            arrPlacements[row,col] := arrStaff[random(6)+1]
          end;
        end;
    end;
    Display;
end;
```



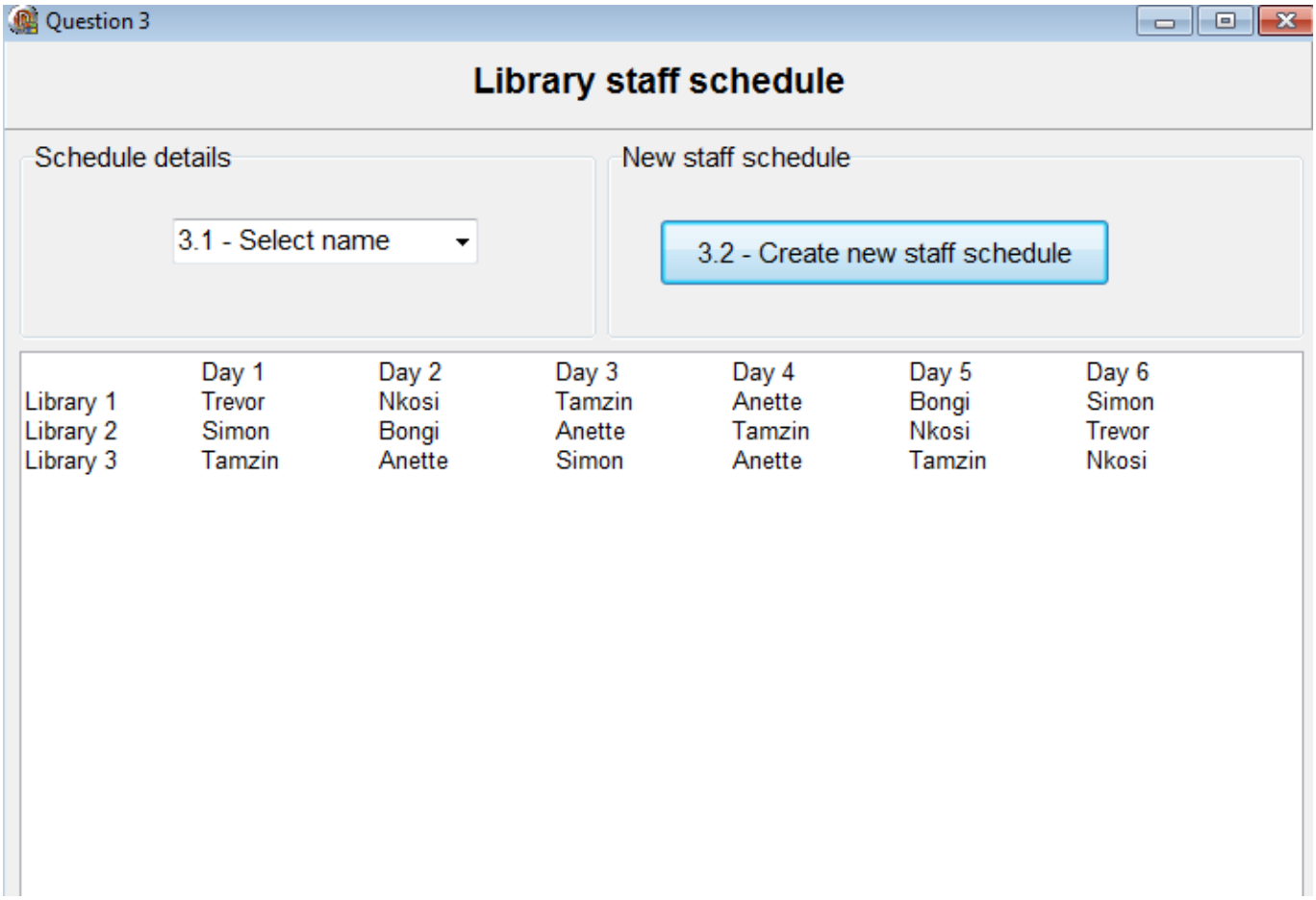
As you can see Annette, was unfortunately selected to work twice per day (which is not allowed), so we will have to add the code to only select a value from arrStaff once. Before we do that, lets finish the part of the question for 'XXXXX' and the inputbox.

So on the day, that was entered in by the user, a 'XXXXX' must inputed.

```
            else  // if row = 3
            begin
              if col = strtoint(sUserInput) then
                arrPlacements[row,col] := 'XXXXX'
              else
                begin
                arrPlacements[row,col] := arrStaff[random(6)+1]
                end;
            end;
```

To make sure our random value does not appear in the other two rows / or more than once in the same column : First we store their values inside two string values :

```
sName1 := arrPlacements[1,col];
sName2 := arrPlacements[2,col];
```



In this case "Trevor" and "Simon" would be stored in sName1 and sName2 respectively. Now that we have the names to compare against, we will:

- assign a random a value

```
arrPlacements[row,col] := arrStaff[random(6)+1];
```

Continously, until it is **not equal*** to sName 1 and sName 2. We use the repeat loop strucutre for this.

Visit this link, if you are unsure about the repeat loop : http://www.delphibasics.co.uk/RTL.asp?Name=Repeat

Our code now looks like this :

```
procedure TfrmQ3.btnQ3_2Click(Sender: TObject);
var row, col : integer;
sUserInput,sName1, sName2 : string;
begin
  //Question 3.2
  sUserInput := InputBox('Enter in day library 3 will be closed','Library 3
Day Closed','1');
  for row := 1 to iMaxRow do
    begin
      for col := 1 to iMaxCol do
        begin
```

```
            if row = 1 then
            begin
              arrPlacements[row,col] := arrStaff[col];
            end
            else if row = 2 then
            begin
              arrPlacements[row,col] := arrStaff[7-col];
            end
            else  // if row = 3
            begin
              if col = strtoint(sUserInput) then
                arrPlacements[row,col] := 'XXXXX'
              else
                begin
                   sName1 := arrPlacements[1,col];
                   sName2 := arrPlacements[2,col];
                   repeat
                    arrPlacements[row,col] := arrStaff[random(6)+1];
                   until (arrPlacements[row,col] <> sName1) and
(arrPlacements[row,col] <> sName2);
                end;
            end;
          end;
      end;
      Display;
end;
```