basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

# NATIONAL SENIOR CERTIFICATE

## GRADE 12

## INFORMATION TECHNOLOGY P1

## FEBRUARY/MARCH 2013

## MEMORANDUM

**MARKS: 120**

**This memorandum consists of 29 pages.**

 

## GENERAL INFORMATION

- These marking guidelines are to be used as the basis for the marking session. They were prepared for use by markers, all of whom are required to attend a rigorous standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' scripts.

- It is acknowledged that there may be different views about some matters of emphasis or detail in the guidelines, and different interpretations of the application thereof.

- Note that learners who provide an alternate correct solution to that given in the marking guidelines will be given full credit for the relevant question.

- **ANNEXURES A, B** and **C** (pages 3–6) include the marking grid for each question for using either one of the two programming languages.

- **ANNEXURES D, E** and **F** (pages 7–16) contain the solutions for Delphi for QUESTIONS 1 to 3 in programming code.

- **ANNEXURES G, H, I** and **J** (pages 17–29) contain the solutions for Java for QUESTIONS 1 to 3 in programming code.

- Copies of ANNEXURES A, B and C (pages 3–6) should be made for each learner and completed during the marking session.

 

## ANNEXURE A

## QUESTION 1: MARKING GRID – PROGRAMMING AND DATABASE

| CENTRE NUMBER: | | EXAMINATION NUMBER: | | |
|---|---|---|---|---|
| **QUESTION** | **DESCRIPTION** | | **MAX. MARKS** | **LEARNER'S MARKS** |
| 1.1 | **Query:** | **Correct fields (or *) ✓; correct table ✓; ORDER BY correct fields in correct order✓ Desc✓** | **4** | |
| | SQL: | SELECT * FROM tblTours ORDER BY Destination, StartDate Desc | | |
| 1.2 | **Query:** | **Correct field & table ✓; FirstName starting with C✓; surname starting with C✓; displaying correct gender ✓** | **4** | |
| | SQL(D): | SELECT TourID, FirstName, Surname FROM tblTourists WHERE FirstName LIKE "C%" AND Surname LIKE "C%" AND Gender = "F" | | |
| | SQL (J): | SELECT TourID, FirstName, Surname FROM tblTourists WHERE FirstName LIKE 'C%' AND Surname LIKE 'C%' AND Gender = 'F' | | |
| 1.3 | **Query:** | **Correct fields & table✓; correct WHERE clause displaying if Deposit is paid✓ and build input (sX) into SQL string with LIKE ✓starting with sX✓** | **4** | |
| | SQL (D): | 'SELECT TourID, Surname FROM tblTourists WHERE (Deposit) AND (Country LIKE "' + sX + '%")' | | |
| | SQL (J): | "SELECT TourID, Surname FROM tblTourists WHERE (Deposit) AND (Country LIKE '" + sX + "'%') " | | |
| 1.4 | **Query:** | **Correct fields & table ✓; *NumberOfDays* field correctly calculated and displayed ✓✓; Correct WHERE clause using start date (#2012/06/12#) ✓ and end date (#2012/10/13#)✓; must be away for more than 5 days ✓** | **6** | |
| | SQL: | SELECT Surname, StartDate, (EndDate-StartDate) +1 AS NumberOfDays FROM tblTours WHERE (StartDate >= #2012/06/12#) AND (StartDate <= #2012/10/31#) AND ((EndDate-StartDate) + 1 > 5) | | |
| 1.5 | **Query:** | **DELETE FROM correct table✓; WHERE EndDate✓ testing for Year✓; against correct year✓** | **4** | |
| | SQL: | DELETE FROM tblTours WHERE YEAR(EndDate) = 2011 | | |
| | **NOTE:** | May not use StartDate | | |
| | Alternative: SQL(D) | DELETE FROM tblTours WHERE EndDate LIKE "%2011%" | | |
| | Alternative: SQL(J) | DELETE FROM tblTours WHERE EndDate LIKE '%2011%' | | |

## QUESTION 1: MARKING GRID – PROGRAMMING AND DATABASE – continue

| | | | | |
|---|---|---|---|---|
| 1.6 | **Query:** | **Correct fields & table ✓; SUM on correct field✓ format result as Currency✓ AS IncomePerCountry✓GROUP BY on Country field✓** | **5** | |
| | SQL(D): | SELECT Country, Format(SUM(AmountPaid),"Currency") AS IncomePerCountry FROM tblTourists GROUP BY Country | | |
| | SQL(J): | SELECT Country, Format(SUM(AmountPaid),'Currency') AS IncomePerCountry FROM tblTourists GROUP BY Country | | |
| 1.7 | **Query:** | **Correct fields from both tables✓ including the calculated field✓; correct WHERE clause linking tables with TourID✓; correct GROUP BY clause (note order of fields)✓; correct HAVING clause ✓ count number of tourist✓ less than seats available✓** | **7** | |
| | SQL: | **Using aliases for table names:**<br>SELECT T.Destination, T.StartDate, T.Seats, Count(V.Surname) AS [SeatsBooked] FROM tblTours T, tblTourists V WHERE T.TourID = V.TourID GROUP BY Destination, StartDate, Seats HAVING Count(V.Surname) < T.Seats | | |
| | Alternative: | SELECT Destination, StartDate, tblTours.Seats, Count(tblTourists.Surname) AS [SeatsBooked] FROM tblTours, tblTourists WHERE tblTourists.TourID = tblTours.TourID GROUP BY Destination, StartDate, Seats HAVING Count(tblTourists.Tourist) < tblTours.Seats | | |
| | | **TOTAL:** | **34** | |

## ANNEXURE B

## QUESTION 2: MARKING GRID – OBJECT-ORIENTED PROGRAMMING

| CENTRE NUMBER: | EXAMINATION NUMBER: | | |
|---|---|---|---|
| **QUESTION** | **DESCRIPTION** | **MAX. MARKS** | **LEARNER'S MARKS** |
| 2.1.1 | **setTariff METHOD:**<br>Test for Apr OR✓ Sept OR Dec all three months✓<br>Assign 1250✓ to tariff attribute✓<br>Repeat for Jan, Feb, Aug, Oct, Nov ; Assign 1000✓<br>Repeat for March, May, June, July ; Assign 900✓ | **6** | |
| 2.1.2 | **PARAMETERISED CONSTRUCTOR:**<br>Correct order and data type of parameters✓✓;<br>Assign five parameters✓; call set method for tariff per day✓ | **4** | |
| 2.1.3 | **shortenString METHOD:**<br>First character of month✓; Loop✓ from second char✓<br>build result string✓ if character not a vowel ✓ | **5** | |
| 2.1.4 | **findLuckyChar METHOD:**<br>Loop until not a space ✓; random number ✓✓ based on length of destination ✓ result ✓ | **5** | |
| 2.1.5 | **toString METHOD:**<br>Use shortenString method ✓; <eoln or #13> character ✓<br>Labels ✓ display numerical data correct ✓ other attributes display correctly ✓ | **5** | |
| 2.2.1 | Declares object array size 50✓ and array counter ✓ | **2** | |
| 2.2.2 | **INITIALISATION OF ARRAY:**<br>Test if file exist ✓<br>*File doesn't exists*:<br>Display message and terminate/exit program/event✓<br>*File does exists*:<br>{*Delphi: AssignFile, Reset and CloseFile*<br>*Java: Create object to read from file*} ✓;<br>Init array counter and change array counter ✓;<br>Loop through file ✓; Read two lines from text file✓ ;<br>Separate first line using (&)✓;<br>Separate second line using (" for ")✓ and (" days#") ✓<br>Instantiate object using parameterized constructor ✓✓✓ | **12** | |
| 2.2.3 | **MENU OPTION A:**<br>Input month ✓; display headings ✓; Loop through array ✓<br>IF month ✓; Display info (counter correct) ✓ | **5** | |
| | Prompt and input tour no ✓<br>Display tour information using toString() method ✓ | **2** | |
| | Prompt for character and display destination ✓<br>Input guess char✓; use findLuckyChar method ✓<br>IF guess char = lucky char ✓<br>THEN: calculate discount ✓ and display ✓<br>ELSE: display lucky character and message ✓<br>Both messages must display original price ✓ | **8** | |
| | **TOTAL:** | **54** | |

## ANNEXURE C

## QUESTION 3: MARKING GRID - PROBLEM-SOLVING PROGRAMMING

| CENTRE NUMBER: | EXAMINATION NUMBER: |
|---|---|

| QUESTION | DESCRIPTION | MAX. MARKS | LEARNER'S MARKS |
|---|---|---|---|
| 3.1 | **MENU OPTION A:**<br>Init Total ✓;<br>Loop through array ✓<br>IF Country = France OR Germany OR Spain ✓✓<br>Extract the number of euro's and add to total ✓✓<br>Calculate rand value ✓<br>Display both values with formatting ✓ | **8** | |
| 3.2 | **MENU OPTION B:**<br>Loop though array ✓<br>IF Robben Island tour ✓✓<br>Find index of #RO# ✓<br>IF tourist from ENGLAND OR✓ CANADA ✓<br>Replace #RO# ✓ with #ROEnglish# ✓<br>Display name of tourist ✓✓<br>ELSE: Replace #RO# with #ROOther# ✓ | **11** | |
| 3.3 | **MENU OPTION C:**<br>Step with loop through destination array ✓<br>initialise visit counter ✓<br>copy destination code ✓<br>Step with loop through tourist array ✓<br>determine IF tourist is visiting destination ✓✓<br>increase visit count ✓<br>calculate number of stars ✓<br>build output string: destination ✓ and<br>Step with loop to add number of stars ✓✓ and<br>(number of visits) ✓<br>Display string ✓ | **13** | |
| | **TOTAL:** | **32** | |

## SUMMARY OF LEARNER'S MARKS:

| | QUESTION 1 | QUESTION 2 | QUESTION 3 | GRAND TOTAL |
|---|---|---|---|---|
| **MAX. MARKS** | **34** | **54** | **32** | **120** |
| **LEARNER'S MARKS** | | | | |

 

## ANNEXURE D: SOLUTION – QUESTION 1: DELPHI

```
unit Question1U_MEMO;
   //Solution for Question 1
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, Grids, DBGrids, ExtCtrls, Buttons, Menus;

type
  TfrmRec = class(TForm)
    qryRec: TADOQuery;
    dsrQry: TDataSource;
    grdRec: TDBGrid;
    mnuMain: TMainMenu;
    mnuOptionA: TMenuItem;
    mnuOptionB: TMenuItem;
    mnuOptionC: TMenuItem;
    mnuOptionD: TMenuItem;
    mnuOptionE: TMenuItem;
    mnuOptionF: TMenuItem;
    mnuOptionG: TMenuItem;
    mnuQuit: TMenuItem;
    procedure mnuOptionAClick(Sender: TObject);
    procedure mnuOptionBClick(Sender: TObject);
    procedure mnuOptionCClick(Sender: TObject);
    procedure mnuOptionDClick(Sender: TObject);
    procedure mnuOptionEClick(Sender: TObject);
    procedure mnuOptionFClick(Sender: TObject);
    procedure mnuOptionGClick(Sender: TObject);
    procedure mnuQuitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmRec: TfrmRec;

implementation

{$R *.dfm}
//=============================================================================
procedure TfrmRec.mnuOptionAClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT * FROM tblTours ORDER BY Destination, StartDate
Desc';
  qryRec.Open;
end;
//=============================================================================
procedure TfrmRec.mnuOptionBClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT TourID,  FirstName, Surname ' +
                      'FROM tblTourists ' +
                      'WHERE Surname LIKE "C%" AND FirstName LIKE "C%" AND ' +
                      'Gender = "F"';
  qryRec.Open;
end;
//=============================================================================
```

```
procedure TfrmRec.mnuOptionCClick(Sender: TObject);
var
  sX : String;
begin
  sX := InputBox('Question 1', 'Country of origin?', 'Spain');
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT TourID, Surname  FROM tblTourists '+
                      'WHERE (Deposit = TRUE) AND ' +
                      '(Country = "' + sX + '")';
  qryRec.Open;
end;
//=============================================================================
procedure TfrmRec.mnuOptionDClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT Surname, StartDate, EndDate, ' +
                      '(EndDate-StartDate)+1 AS NumberOfDays ' +
                      'FROM tblTours '+
                      'WHERE (Startdate >= #2012/06/12#) AND ' +
                      '      (StartDate <= #2012/10/31#) AND ' +
                      '      ((EndDate-StartDate) + 1 > 5) ';
  qryRec.Open;
end;
//=============================================================================
procedure TfrmRec.mnuOptionEClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'DELETE FROM tblTours WHERE Year(EndDate) = 2011';
  qryRec.ExecSQL;
  MessageDlg('Records Processed Successfully',mtInformation,[mbOk],0);
end;

procedure TfrmRec.mnuOptionFClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT  Country, ' +
                      'Format(Sum(AmountPaid),"Currency")AS IncomePerCountry' +
                      'FROM tblTourists ' +
                      'GROUP BY Country';
  qryRec.Open;
end;
//=============================================================================
procedure TfrmRec.mnuOptionGClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT T.Destination, T.StartDate, T.Seats, ' +
                      ' Count(V.Surname) AS [SeatsBooked]' +
                      'FROM tblTours T, tblTourists V ' +
                      'WHERE T.TourID = V.TourID ' +
                      'GROUP BY Destination, StartDate, Seats ' +
                      'HAVING Count(V.Surname) < T.Seats';
  qryRec.Open;
end;
//=============================================================================

procedure TfrmRec.mnuQuitClick(Sender: TObject);
begin
   Application.Terminate;
end;

end.
```

## ANNEXURE E: SOLUTION – QUESTION 2: DELPHI

## CLASS UNIT:

```
unit uQuest2_MEMO;
//solution for Question 2 - class
interface

TYPE
   TData = class(TObject)
     private
        fGName       : String;
        fDName       : String;
        fMName       : String;
        fNumD        : Integer;
        fNumT        : Integer;
        fTariff      : Real;
     public
         function getGName : String;
         function getDName : String;
         function getMName : String;
         function getNumD  : Integer;
         function getNumT  : Integer;
         function getTariff: Real;

         constructor Create(sGuide, sDestination, sMonth: String; iDays, iNum :
Integer);
         procedure setTariff;
         function shortenString : String;
         function findLuckyChar : Char;
         function toString : String;
     end;

implementation

uses SysUtils;

{ TData }
//============================================================================
constructor TData.Create(sGuide, sDestination, sMonth: String; iDays, iNum:
Integer);
begin
   fGName       := sGuide;
   fDName       := sDestination;
   fMName       := sMonth;
   fNumD        := iDays;
   fNumT        := iNum;
   setTariff;
end;
//============================================================================
procedure TData.setTariff;
var
  sMonth : String;
begin
  sMonth := Uppercase(fMName);
  //accept solution without the use of uppercase.
  iF (sMonth = 'APRIL') OR (sMonth = 'SEPTEMBER') OR (sMonth = 'DECEMBER')
    then
       fTariff := 1250.00
    else
     iF (sMonth = 'MARCH') OR (sMonth = 'MAY') OR (sMonth = 'JUNE') OR (sMonth
= 'JULY')
     then
```

```
        fTariff := 900.00
     else
        fTariff := 1000.00;
end;
//============================================================================
function TData.shortenString: String;
var
   a : Integer;
   sTemp  : String;
begin
   sTemp := fMName[1];
   for a := 2 to length(fMName) do
    if NOT(Upcase(fMName[a]) in ['A','E','I','O','U'])
     then sTemp := sTemp + fMName[a];
   Result := sTemp;
   //accept solution without upcase where small letters are added to set
end;
//============================================================================
function TData.findLuckyChar: Char;
var
 iMax, iLuckyNum : Integer;
begin
  iMax := length(fDName);
  Repeat
    iLuckyNum := random(iMax)+1;
  Until fDName[iLuckyNum] <> #32;

  Result := fDName[iLuckyNum];
end;
//============================================================================
function TData.toString: String;
begin
   Result := 'Month: ' + shortenString + #13 +
             'Destination: ' + fDName + ' with ' + fGName + ' as the tour
guide' + #13 +
             'Price: ' + FloatToStrF(fTariff, ffCurrency, 9,2) + ' per day for
a period of '+ IntToStr(fNumD) + ' days' +
#13+InttoStr(fNumT) + ' tourists are taking this tour.';
end;
//============================================================================
function TData.getGName: String;
begin
   Result := fGName;
end;

function TData.getDName: String;
begin
   Result := fDName;
end;

function TData.getMName: String;
begin
   Result := fMName;
end;

function TData.getNumD: Integer;
begin
   Result := fNumD;
end;

function TData.getNumT: Integer;
begin
   Result := fNumT;
end;
```

```
function TData.getTariff: Real;
begin
   Result := fTariff;
end;

end.
```

**FORM UNIT:**

```
unit Question2U_MEMO;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Menus,
  uQuest2_MEMO;
type
  TfrmQ2 = class(TForm)
    mnuMain: TMainMenu;
    mnuOptionA: TMenuItem;
    mnuQuit: TMenuItem;
    redQ2: TRichEdit;
    procedure mnuQuitClick(Sender: TObject);
    procedure mnuOptionAClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQ2: TfrmQ2;

  arrTours : Array[1..50] of TData;
  iCount   : Integer = 0;

implementation

{$R *.dfm}
{$R+}
//============================================================================
procedure TfrmQ2.FormCreate(Sender: TObject);
var
   TF                  : TextFile;
   sLineA, sLineB      : String;
   sGName, sDName, sMnth, sDays, sNumT : String;
begin
   {Code for onCreate event of form}
   randomize;
   IF NOT FileExists('DataQ2.txt')
    then
     begin
       MessageDlg('ERROR: File not found.', mtError, [mbOk], 0);
       mnuOptionA.Enabled := False;
       Exit;
     end;

   AssignFile(TF, 'DataQ2.txt');
   Reset(TF);
```

```
   While NOT EOF(TF) DO
    Begin
      Readln(TF, sLineA);
      Readln(TF, sLineB);

      sGName := copy(sLineA, 1, pos('&', sLineA)-1);
      sDName := copy(sLineA, pos('&', SlineA)+1, length(sLineA));

      sMnth  := copy(sLineB, 1, pos(' for ', sLineB) -1);
      Delete(sLineB, 1, pos('for', sLineB)+3); //delete space after for also
      sDays  := copy(sLineB, 1, pos(' ', sLineB)-1);
      Delete(sLineB, 1, pos('#', sLineB));
      sNumT  := sLineB;

      Inc(iCount, 1);
      arrTours[iCount] := TData.Create(sGName, sDName, sMnth, StrtoInt(sDays),
StrToInt(sNumt));
    End; //while
   CloseFile(TF);
end;
//============================================================================
procedure TfrmQ2.mnuOptionAClick(Sender: TObject);
var
   sMnth, sLucy                          : String;
   a, iTNum                              : Integer;
   rNTariff                              : Real;
   cLChar                                : Char;
begin
  {Code Option A}
  sMnth := InputBox('Question 2', 'Enter the month of tour (e.g. February)?',
'');
  redQ2.Lines.Clear;
  redQ2.Paragraph.TabCount := 1;
  redQ2.Paragraph.Tab[0]   := 100;
  redQ2.Lines.Add('Tours for the month of ' + sMnth);
  redQ2.Lines.Add('==============================');
  redQ2.Lines.Add('Number' + #9 + 'Destination');
  for a := 1 to iCount do
   begin
     if Uppercase(arrTours[a].getMName) = Uppercase(sMnth)
      then redQ2.Lines.Add(intToStr(a) + #9 + arrTours[a].getDName);
   end;

  iTNum := StrtoInt(InputBox('Question 2', 'Enter the number of a tour from the
list', '35'));
  redQ2.Lines.Add('');
  redQ2.Lines.Add(arrTours[iTNum].toString);

  redQ2.Lines.Add(' ');
  sLucy := InputBox('Question 2', 'Enter any character from '+
            arrTours[iTNum].getDName +' .', 'a');
  cLChar := arrTours[iTNum].findLuckyChar;
  IF Upcase(cLChar) = UpCase(sLucy[1])
   then
    begin
      rNTariff := arrTours[iTNum].getTariff * 0.75; //25% discount
      redQ2.Lines.Add('Congratulations! You have received 25% discount '+
                      'on the daily tariff!' + #13 +
                      'The tariff was ' +
          FloatToStrF(arrTours[iTNum].getTariff, ffCurrency, 8,2) +
                      ' per day. It has been reduced to ' +
          FloattoStrF(rNTariff, ffCurrency, 8, 2) + ' per day');
    end
```

```
   else
    begin
      redQ2.Lines.Add('The lucky character was the letter '+ cLChar +'.'+#13 +
                      'No discount. The tariff is still ' +
          FloatToStrF(arrTours[iTNum].getTariff, ffCurrency, 8,2) + ' per
day');
    end;
end;
//============================================================================
procedure TfrmQ2.mnuQuitClick(Sender: TObject);
begin
   Application.Terminate;
end;

end.
```

## ANNEXURE F: SOLUTION – QUESTION 3: DELPHI

```
unit Question3U_MEMO;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Menus;

type
  TfrmQ3 = class(TForm)
    mnuMain: TMainMenu;
    mnuOptionA: TMenuItem;
    mnuOptionB: TMenuItem;
    mnuQuit: TMenuItem;
    redQ3: TRichEdit;
    mnuOptionC: TMenuItem;
    procedure mnuQuitClick(Sender: TObject);
    procedure mnuOptionAClick(Sender: TObject);
    procedure mnuOptionBClick(Sender: TObject);
    procedure mnuOptionCClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQ3: TfrmQ3;

  arrData : Array[1..40] of String =
  ('Rachel Delarosa@Canada#SH#11861','Corradino Grande@Spain#RO#5788',
'Lucas Herder@Germany#KR#7709', 'Estotz Lizarazu@France#GA#12349',
'Chynna Taylor@England#GA#8551', 'Renata Di@Spain#RO#4906',
'Ugs Boulot-Tolle@France#CA#7300', 'Lena Bucholtz@Germany#GA#10344',
'Maria Heimpel@Germany#SH#9438', 'Julian Amstadter@Germany#RO#8840',
'Sofie Mosbauer@Germany#GA#5894', 'Fiona Green@England#CA#9094',
'Sara Escobedo@Canada#KR#4381', 'Nataly Mahan@Canada#RO#12642',
'Wyatt Parham@Canada#SH#4799', 'Noah Donovan@Canada#SH#3888',
'Joseph Scott@England#SH#7928', 'Emily Smith@England#KR#3110',
'Adriana Mancuso@Spain#RO#3724', 'Cassandra Wilder@Canada#KR#12583',
'Tomasino Camporese@Spain#KR#6777', 'Stacy Anderson@England#RO#3686',
'Guiraud Bluteau@France#RO#11592', 'Damian Friedman@Canada#RO#9012',
'Anne Loef@Germany#KR#13035', 'Terence Brown@England#SH#8180',
'Lion Ghislieri@Spain#RO#14343', 'Giraudetz Girardin@France#CA#11644',
'Guglielmo Capriati@Spain#SH#5408', 'David Geiberger@Germany#RO#9854',
'Irisa Cooper@England#KR#11456', 'Hayden Mcdonough@Canada#KR#7840',
'Jonas Hipp@Germany#RO#3137', 'Emily Kohler@Germany#GA#6509',
'Emily Thul@Germany#RO#8551', 'Gino Lazzaretti@Spain#CA#2329',
'Alex Hofstater@Germany#GA#6751', 'Peers Scott@England#RO#9470',
'Liliana Horne@Canada#RO#14689', 'Leon Kleinpaul@Germany#RO#15194');
implementation

VAR
  //array used for Option 3.
  arrDest  : array[1..6] of string =
           ('Cape Winelands','Garden Route','Kruger National Park',
            'Robben Island (English tour)', 'Robben Island (Other tour)',
            'Shakaland');

{$R *.dfm}
```

 

```
{$R+}
//============================================================================
procedure TfrmQ3.mnuOptionAClick(Sender: TObject);
var
   a               : Integer;
   rRand, rTotal   : Real;
   sTemp           : String;
begin
  {Code Option A}
  redQ3.Lines.Clear;
  rTotal := 0;
  for A := 1 to 40 do
   begin
      IF ((pos('France', arrData[a]) > 0) OR (pos('Germany', arrData[a]) > 0)
           OR (pos('Spain', arrData[a]) > 0))
       then
        begin
          sTemp := arrData[a];
          Delete(sTemp, 1, pos('#', sTemp));
          Delete(sTemp, 1, pos('#', sTemp));
          rTotal := rTotal + StrToFloat(sTemp);
        end;
   end;
   rRand := rTotal * 10.75;
   redQ3.Lines.Add('Total amount in euro: '+ FloatToStr(rTotal));
   redQ3.Lines.Add('Total amount in South African rand: ' +
                          FloatToStrF(rRand, ffCurrency, 8, 2));
end;
//============================================================================
procedure TfrmQ3.mnuOptionBClick(Sender: TObject);
var
   a, Index               : Integer;
begin
  {Code Option B}
  redQ3.Lines.Clear;
  redQ3.Lines.Add('List of English-speaking tourists to Robben Island:');
  redQ3.Lines.Add('================================================');
  for a := 1 to 40 do
   begin
     if pos('#RO#', arrData[a]) > 0
      then
       begin
         Index := pos('#RO#', arrData[a]);
         IF (pos('Canada', arrData[a]) > 0) OR
            (pos('England', arrData[a]) > 0)
            then
             begin
                Delete(arrData[a], Index, 4);
                Insert('#ROEnglish#', arrData[a], Index);
                //Insert('English', arrData[a], Index+4); //alternative
                redQ3.Lines.Add(Copy(arrData[a], 1, pos('@', arrData[a])-1));
             end //Canada & England
            else
             begin
                Delete(arrData[a], Index, 4);
                Insert('#ROOther#', arrData[a], Index);
                //Insert('Other', arrData[a], Index+4); //alternative
             end;
       end; //Robben Island
   end;
end;
//============================================================================
```

```
procedure TfrmQ3.mnuOptionCClick(Sender: TObject);

var
  arrCount :array[1..6] of integer;
  a, b,iRating  :integer;
  sDest :string;
  sRating      :string;
begin
  redQ3.Lines.Clear;
  redQ3.Paragraph.TabCount := 2;
  redQ3.Paragraph.Tab[0]   := 150;
  redQ3.Paragraph.Tab[1]   := 200;
  redQ3.Lines.Add('Star rating of tours');
  redQ3.Lines.Add('====================================');
  redQ3.Lines.Add('Destination' + #9 + 'Rating' + #9 + 'Number of tourists');
  redQ3.Lines.Add('====================================');
  for a := 1 to 6 do
    arrCount[a] := 0;

  for a := 1 to 40 do
    begin
      sDest := Uppercase(copy(arrData[a], pos('#',arrData[a])+1,3));
      case sDest[1] of
        'C' : inc(arrCount[1],1);//Cape Winelands
        'G' : inc(arrCount[2],1);//Garden Route
        'K' : inc(arrCount[3],1);//Kruger National Park
        'R' : case sDest[3] of //Robben Island
               'E' : inc(arrCount[4],1);//English
               'O' : inc(arrCount[5],1);//Other
              end;
        'S' : inc(arrCount[6],1);//Shakaland
      end;
    end;//for
 // Output
  For a := 1 to 6 do
    begin
      sRating := '';
      iRating := arrCount[a] div 3;
      for b := 1 to iRating do
         sRating := sRating + '*';
      redQ3.Lines.Add(arrDest[a] + #9 + sRating + #9 + '(' +
                                  IntToStr(arrCount[a]) + ')');
   end;/
end;
//============================================================================
procedure TfrmQ3.mnuQuitClick(Sender: TObject);
begin
   Application.Terminate;
end;

end.
```

 

## ANNEXURE G: SOLUTION – QUESTION 1: JAVA

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.sql.*;

public class TestQuestion1_MEMO
{
      public static void main (String[] args) throws SQLException,IOException
      {

            BufferedReader inKb = new BufferedReader (new InputStreamReader
(System.in));

            Tourism DB = new Tourism();
            System.out.println();

            char choice = ' ';
            do
            {
            System.out.println("\n\n      MENU");
            System.out.println();
            System.out.println("    Option A");
            System.out.println("    Option B");
            System.out.println("    Option C");
            System.out.println("    Option D");
            System.out.println("    Option E");
            System.out.println("    Option F");
            System.out.println("    Option G");
            System.out.println();
            System.out.println("    Q - QUIT");
            System.out.println(" ");
            System.out.print("    Your choice? ");
            choice = inKb.readLine().toUpperCase().charAt(0);
            System.out.println(" ");
            String sql = "";
            switch(choice)
            {
            case 'A':   // Question 1.1
            {
                  sql = "SELECT * FROM tblTours ORDER BY Destination, StartDate
Desc";
                  DB.query(sql);
                  break;
            }
//=============================================================================
            case 'B':   // Question 1.2
            {
                  sql = "SELECT  TourID,  FirstName, Surname FROM tblTourists
WHERE  FirstName LIKE 'C%'AND Surname LIKE 'C%' AND Gender = 'F'";
                  DB.query(sql);
                  break;
            }
//=============================================================================
            case 'C':   // Question 1.3
            {
                  System.out.println("Option C\nCountry of origin (e.g. Spain)?
");
                  String sX = inKb.readLine();
                  sql = "SELECT  TourID, Surname FROM tblTourists WHERE Deposit
AND Country LIKE '" + sX + "%'";
                  DB.query(sql);
```

```
                  break;
            }
//=============================================================================
            case 'D':   // Question 1.4
            {
                  sql = "SELECT Surname, StartDate, EndDate, (Enddate-
StartDate)+1 AS NumberOfDays FROM tblTours WHERE (Startdate >= #2012/06/12#)
AND (StartDate <= #2012/10/31#) AND (endDate-StartDate + 1 > 5)";
                  DB.query(sql);
                  break;
            }
//=============================================================================
            case 'E':   // Question 1.5
            {
                  sql = "DELETE FROM tblTours WHERE YEAR(EndDate) = 2011";
                  DB.query(sql);
                  break;
            }
//=============================================================================
            case 'F':   // Question 1.6
            {
                  sql = "SELECT Country, Format(SUM(AmountPaid),'Currency') AS
IncomePerCountry FROM tblTourists GROUP BY Country";
                  DB.query(sql);
                  break;
            }
//=============================================================================
            case 'G':   // Question 1.7
            {
               sql = "SELECT Destination, StartDate, Seats,
Count(tblTourists.Surname) AS [SeatsBooked]  FROM tblTours, tblTourists WHERE
tblTourists.TourID = tblTours.TourID GROUP BY Destination, StartDate, Seats
HAVING Count(tblTourists.Surname) < tblTours.Seats";
               DB.query(sql);
               break;
            }
            }
            }while (choice != 'Q');

            DB.disconnect();
            System.out.println("Done");
      }
}
```

## ANNEXURE H: SOLUTION – QUESTION 2: JAVA

## OBJECT CLASS:

```
import java.text.DecimalFormat;

/**
 *
 * Memo Question 2
 */
public class Quest2_MEMO {
    private String gName;
    private String dName;
    private String mName;
    private int numD;
    private int numT;
    private double tariff;

    public Quest2_MEMO(String gName, String dName, String mName, int numD, int
numT) {
        this.gName = gName;
        this.dName = dName;
        this.mName = mName;
        this.numD = numD;
        this.numT = numT;
        setTariff();
    }

//============================================================================
  private void setTariff()
    {
        String sName = mName.toUpperCase();
        //accept solution that doesn't use toUpperCase()
        if (sName.equals("DECEMBER") || sName.equals("APRIL") ||
sName.equals("SEPTEMBER"))
           tariff = 1250;
        else
          if (sName.equals("MAY") || sName.equals("MARCH") ||
sName.equals("JUNE") || sName.equals("JULY"))
           tariff = 900;
          else
            tariff = 1000;

    }
//============================================================================
    private String shortenString()
    {
        String shortname = mName.substring(0,1);
        String vowels = "AEIOU";
        //accept solution using lower case letters as well
        for (int cnt = 1; cnt < mName.length();cnt++)
        {
            char letter = mName.charAt(cnt);
            if(vowels.indexOf(mName.toUpperCase().charAt(cnt))<0)
            {
                shortname =shortname + letter;
            }
        }
        return  shortname;
    }
//============================================================================
    public char findLuckyChar()
    {
```

```
        int last = dName.length()-1;
        int first = 1;
        int position = 0;
        boolean repeat = true;
        char charac = ' ';
        do
        {
            position = (int)(Math.random() * (last-first+1) + first);
            charac = dName.charAt(position);
            if (charac != ' ')
            {
                repeat = false;
                charac = dName.charAt(position);
            }

        }while (repeat == true);
        return charac;
    }
//=========================================================================
    public String toString()
    {
        DecimalFormat df = new DecimalFormat("R 0.00");
        return "Month: " + shortenString() + "\nDestination: " + dName + " with
" +  gName + " as the tour guide\nPrice: " + df.format(getTariff())+ " per day
for a period of " + numD + " days\n" + numT+ " tourists are taking this
tour\n";
    }
//=========================================================================
    public String getGName() {
        return gName;
    }

    public void setGName(String gName) {
        this.gName = gName;
    }

    public String getDName() {
        return dName;
    }

    public void setDName(String dName) {
        this.dName = dName;
    }

    public String getMName() {
        return mName;
    }

    public int getNumD() {
        return numD;
    }

    public void setNumD(int numD) {
        this.numD = numD;
    }

    public int getNumT() {
        return numT;
    }

    public void setNumT(int numT) {
        this.numT = numT;
    }
```

```
    public double getTariff() {
        return tariff;
    }
}
```

## APPLICATION/DRIVER CLASS:

```
   import java.io.BufferedReader;
   import java.io.InputStreamReader;
   import java.io.FileReader;
   import java.io.FileNotFoundException;

    public class Question2_MEMO {

    public static void main(String[] args) throws Exception
    {
     Quest2_MEMO[] tourArray = new Quest2_MEMO[50];
     int counter = 0;
     BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));
     // read from file
     try {
     BufferedReader bf = new BufferedReader(new FileReader("DataQ2.txt"));
     while (bf.readLine() != null)
     {
       counter++;
     }
     counter = counter/2;

     bf = new BufferedReader(new FileReader("DataQ2.txt"));
     for (int cnt = 0; cnt < counter; cnt++)

     {
      String line1 = bf.readLine();
      String line2 = bf.readLine();
      String[] temp1 = line1.split("&");
      String[] temp2 = line2.split(" for ");
      String[] temp3 = temp2[1].split(" days#");
      tourArray[cnt] = new Quest2_MEMO(temp1[0], temp1[1], temp2[0],
Integer.parseInt(temp3[0]), Integer.parseInt(temp3[1]));

      }
   }
   catch (FileNotFoundException e) {
     System.out.println(e);
     System.exit(0);
  }
   catch (Exception f) {
   System.out.println(f);
  }
 char choice = ' ';
 do {
   System.out.println("   MENU\n");
   System.out.println("Option A");
   System.out.println("");
   System.out.println("Q - QUIT");
   System.out.println("\nYour choice?  ");

   choice = kb.readLine().toUpperCase().charAt(0);
   switch (choice) {
   case 'A':
     System.out.print("Enter the month of tour(e.g. February): ");
```

```
     String mnth = kb.readLine();

     System.out.println("\n\nTours for the month of " + mnth);
     System.out.println("==============================\n");
     System.out.println("Number          Destination\n");

     for (int cnt = 0; cnt < counter - 1; cnt++) {
     if (tourArray[cnt].getMName().equalsIgnoreCase(mnth)) {
     System.out.println((cnt + 1)+ "\t\t"+ tourArray[cnt].getDName());
    }
 }
     System.out.print("\nEnter the number of a tour from the list e.g. 35:");
     int num = Integer.parseInt(kb.readLine());
     System.out.println("\n" + tourArray[num - 1]);
     System.out.println("\nEnter any character from: " + tourArray[num -
1].getDName());

     char luckyC = kb.readLine().charAt(0);
     char genChar = tourArray[num - 1].findLuckyChar();
     if (luckyC == genChar) {
     System.out.println("Congratulations! You have received 25% discount on the
daily tariff! \nThe tariff was R " + (tourArray[num - 1].getTariff()) + " per
day.  It has been reduced to R" + (tourArray[num - 1].getTariff() * 0.75) + "
per day\n\n");
     }
    else {
     System.out.println("The lucky character was the letter " + genChar + ".
\nNo discount. The tariff is still R " + tourArray[num - 1].getTariff() + " per
day\n\n");
    }
    break;
  case 'Q':
     System.out.println("Quit");
   }
  } while (choice != 'Q');
 }
}
```

 

## ANNEXURE I: SOLUTION WITH OOP – QUESTION 3: JAVA

```
  import java.io.BufferedReader;
  import java.io.InputStreamReader;



// Object class to describe a tourist object

public class Tourist
{
  private String name;
  private String country;
  private String destination;
  private double money;
//============================================================================
  public Tourist(String touristStr)
  {
    int atpos = touristStr.indexOf("@");
     int hashpos = touristStr.indexOf("#");
    name = touristStr.substring(0, atpos);

    country = touristStr.substring(atpos + 1,hashpos);
    String []  temp = touristStr.split("#");

    destination = temp[1];
    money = Double.parseDouble(temp[2]);
 }
//============================================================================

 public void setDestination(String dest)
 {
   destination = dest;
 }

 public String getName()
 {
   return name;
   }

public String getCountry()
 {
   return country;
   }

public String getDestination()
 {
   return destination;
   }

public double getMoney()
 {
   return money;
   }

}

// Class for the menu
   import java.io.BufferedReader;
   import java.io.InputStreamReader;
```

```
public class Question3_MEMO {

String[] arrData = {"Rachel Delarosa@Canada#SH#11861","Corradino
Grande@Spain#RO#5788",
"Lucas Herder@Germany#KR#7709", "Estotz Lizarazu@France#GA#12349",
"Chynna Taylor@England#GA#8551", "Renata Di@Spain#RO#4906",
"Ugs Boulot-Tolle@France#CA#7300", "Lena Bucholtz@Germany#GA#10344",
"Maria Heimpel@Germany#SH#9438", "Julian Amstadter@Germany#RO#8840",
"Sofie Mosbauer@Germany#GA#5894", "Fiona Green@England#CA#9094",
"Sara Escobedo@Canada#KR#4381", "Nataly Mahan@Canada#RO#12642",
"Wyatt Parham@Canada#SH#4799", "Noah Donovan@Canada#SH#3888",
"Joseph Scott@England#SH#7928", "Emily Smith@England#KR#3110",
"Adriana Mancuso@Spain#RO#3724", "Cassandra Wilder@Canada#KR#12583",
"Tomasino Camporese@Spain#KR#6777", "Stacy Anderson@England#RO#3686",
"Guiraud Bluteau@France#RO#11592", "Damian Friedman@Canada#RO#9012",
"Anne Loef@Germany#KR#13035", "Terence Brown@England#SH#8180",
"Lion Ghislieri@Spain#RO#14343", "Giraudetz Girardin@France#CA#11644",
"Guglielmo Capriati@Spain#SH#5408", "David Geiberger@Germany#RO#9854",
"Irisa Cooper@England#KR#11456", "Hayden Mcdonough@Canada#KR#7840",
"Jonas Hipp@Germany#RO#3137", "Emily Kohler@Germany#GA#6509",
"Emily Thul@Germany#RO#8551", "Gino Lazzaretti@Spain#CA#2329",
"Alex Hofstater@Germany#GA#6751", "Peers Scott@England#RO#9470",
"Liliana Horne@Canada#RO#14689", "Leon Kleinpaul@Germany#RO#15194"};

String []arrDestinations = {"Cape Winelands","Garden Route", "Kruger National
Park", "Robben Island (English tour)","Robben Island (Other tour)",
"Shakaland"};

      BufferedReader kb;

//============================================================================
       public void runMenu() throws Exception {

        kb = new BufferedReader(new InputStreamReader(System.in));
         char choice =' ';
         do {
            System.out.println("MENU");
            System.out.println();
            System.out.println("   Option A");
            System.out.println("   Option B");
            System.out.println("   Option C");

            System.out.println();
            System.out.println("Q - QUIT");
            System.out.println();
            System.out.println("Your choice?");
            choice = kb.readLine().toUpperCase().charAt(0);
            switch (choice) {
               case 'A': convertEuros();
                         break;
               case 'B': divideGroup();
                         break;
               case 'C': determinePopularity();
                         break;
               case 'Q':
                  System.out.println("QUIT");
            }
                 } while (choice != 'Q');
```

```
      }

//============================================================================
//Option A
      public void convertEuros()
      {
       double value = 0;
       for (int c = 0; c < arrData.length; c++)
       {
        Tourist tourist = new Tourist(arrData[c]);
        String country = tourist.getCountry();
        if (country.equalsIgnoreCase("France")
||country.equalsIgnoreCase("Spain")||country.equalsIgnoreCase("Germany"))
         {
           value = value + tourist.getMoney();
         }
      }  //for
    System.out.printf("%s%-8.0f\n","Total amount in euro: ", value);
    double rand = value*10.75;
    System.out.printf("%sR%10.2f\n\n","Total amount in South African rand: ",
      rand);
  }

//============================================================================
// Option B
      public void divideGroup()
      {
       System.out.println("List of English-speaking tourists to Robben
Island");

System.out.println("================================================");
       for (int c = 0; c < arrData.length; c++)
       {
         Tourist tourist = new Tourist(arrData[c]);
         String dest = tourist.getDestination();

         if (dest.equals("RO"))
          {
           String country = tourist.getCountry();
           if (country.equalsIgnoreCase("England") ||
country.equalsIgnoreCase("Canada"))
            {
             System.out.println(tourist.getName());
             arrData[c] = arrData[c].replace("#RO#","#ROEnglish#");
             tourist.setDestination("ROEnglish");
            }
            else
            {
              arrData[c] = arrData[c].replace("#RO#","#ROOther#");
              tourist.setDestination("ROOther");
            } // else
          } // if
       }// for

       System.out.println("\n\n");
   }


//============================================================================
// Option C
   public void determinePopularity()
   {
```

```
          int[] arrCount = new int[6];
    System.out.println("Star rating of tours");
    System.out.println("=================================================");
    System.out.println("Destination                    Rating    Number of
tourists");
    System.out.println("=================================================");
    for (int c = 0; c < 6; c++)
      {
       arrCount[c]=0;
      }
    for (int c = 0; c < arrData.length; c++)
     {
        Tourist tourist = new Tourist(arrData[c]);
        String destcode = tourist.getDestination();
        switch (destcode.toUpperCase().charAt(0))
        {
            case 'C' : arrCount[0]++; break;
            case 'K' : arrCount[2]++; break;
            case 'G' : arrCount[1]++; break;
            case 'R' : if (destcode.toUpperCase().charAt(2) == 'E')
                        arrCount[3]++;
                       else arrCount[4]++;break;
            case 'S' : arrCount[5]++;break;
        }
      }// for
      // output
     for (int i = 0; i < 6; i++)
     {
      String starString = "";
      int numStars = arrCount[i]/3;
      for (int s = 0; s < numStars; s++)
       {
         starString = starString + "*";
       }
       String outString = String.format("%-35s%-
            10s(%d)",arrDestinations[i],starString,arrCount[i]);
            System.out.println(outString);
       }

       System.out.println("\n\n");
      }
      }
//=============================================================================
// Test class creating an object of the menu class

import java.io.IOException;

public class TestQuestion3_Memo
{
   public static void main(String[] args) throws Exception {

         Question3_MEMO Q3 = new Question3_MEMO();
         Q3.runMenu();
      }
}
```

## ANNEXURE J: SOLUTION WITHOUT OOP – QUESTION 3: JAVA

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Question3_MEMO {

String[] arrData = {"Rachel Delarosa@Canada#SH#11861","Corradino
Grande@Spain#RO#5788",
"Lucas Herder@Germany#KR#7709", "Estotz Lizarazu@France#GA#12349",
"Chynna Taylor@England#GA#8551", "Renata Di@Spain#RO#4906",
"Ugs Boulot-Tolle@France#CA#7300", "Lena Bucholtz@Germany#GA#10344",
"Maria Heimpel@Germany#SH#9438", "Julian Amstadter@Germany#RO#8840",
"Sofie Mosbauer@Germany#GA#5894", "Fiona Green@England#CA#9094",
"Sara Escobedo@Canada#KR#4381", "Nataly Mahan@Canada#RO#12642",
"Wyatt Parham@Canada#SH#4799", "Noah Donovan@Canada#SH#3888",
"Joseph Scott@England#SH#7928", "Emily Smith@England#KR#3110",
"Adriana Mancuso@Spain#RO#3724", "Cassandra Wilder@Canada#KR#12583",
"Tomasino Camporese@Spain#KR#6777", "Stacy Anderson@England#RO#3686",
"Guiraud Bluteau@France#RO#11592", "Damian Friedman@Canada#RO#9012",
"Anne Loef@Germany#KR#13035", "Terence Brown@England#SH#8180",
"Lion Ghislieri@Spain#RO#14343", "Giraudetz Girardin@France#CA#11644",
"Guglielmo Capriati@Spain#SH#5408", "David Geiberger@Germany#RO#9854",
"Irisa Cooper@England#KR#11456", "Hayden Mcdonough@Canada#KR#7840",
"Jonas Hipp@Germany#RO#3137", "Emily Kohler@Germany#GA#6509",
"Emily Thul@Germany#RO#8551", "Gino Lazzaretti@Spain#CA#2329",
"Alex Hofstater@Germany#GA#6751", "Peers Scott@England#RO#9470",
"Liliana Horne@Canada#RO#14689", "Leon Kleinpaul@Germany#RO#15194"};

String []arrDestinations = {"Cape Winelands","Garden Route", "Kruger National
Park", "Robben Island (English tour)","Robben Island (Other tour)",
"Shakaland"};

  BufferedReader kb;
//=============================================================================
//Option A
   public void convertEuros()
   {
    double value = 0;
    for (int cnt = 0; cnt < arrData.length; cnt++)
    {
     if (arrData[cnt].indexOf("France")>=0
||arrData[cnt].indexOf("Spain")>=0||arrData[cnt].indexOf("Germany")>=0)
      {
       String[] temp = arrData[cnt].split("#");
       value = value + Double.parseDouble(temp[2]);
      }
    }
   System.out.printf("%s%-8.0f\n","Total amount in euro: ", value);
   double rand = value*10.75;
   System.out.printf("%sR%10.2f\n\n","Total amount in South African rand: ",
rand);
 }
//=============================================================================
//Option B
public void divideGroup()
{
 System.out.println("List of English-speaking tourists to Robben Island");



 System.out.println("================================================");
 for (int cnt = 0; cnt < arrData.length; cnt++)
```

```
 {
  if (arrData[cnt].indexOf("#RO#") >= 0)
  {
  if (arrData[cnt].indexOf("England")>=0 ||arrData[cnt].indexOf("Canada") >= 0)
  {
   String[] temp = arrData[cnt].split("@");
   System.out.println(temp[0]);
   arrData[cnt] = arrData[cnt].replace("#RO#","#ROEnglish#");
  }
 else
  {
    arrData[cnt] = arrData[cnt].replace("#RO#","#ROOther#");
  }
 }
}
System.out.println("\n\n");
}

//============================================================================
      // Option C
   public void determinePopularity()
   {
    int[] arrCount = new int[6];
    System.out.println("Star rating of tours");

System.out.println("=========================================================");
System.out.println("Destination                          Rating    Number of
tourists");

System.out.println("=========================================================");
for (int c = 0; c < 6; c++)
{
 arrCount[c]=0;
}

for (int c = 0; c < arrData.length; c++)
 {
      int endpos = arrData[c].indexOf("#");
      String destcode = arrData[c].substring(endpos + 1,endpos + 4);
      switch (destcode.toUpperCase().charAt(0))
        {
            case 'C' : arrCount[0]++; break;
            case 'K' : arrCount[2]++; break;
            case 'G' : arrCount[1]++; break;
            case 'R' : if (destcode.toUpperCase().charAt(2) == 'E')
                        arrCount[3]++;
                       else arrCount[4]++;break;
            case 'S' : arrCount[5]++;break;
        }
  }// for
    // output
    for (int index = 0; index < 6; index++)
    {
      String starString = "";
      int numStars = arrCount[index]/3;
      for (int stars = 0; stars < numStars; stars++)
      {
        starString = starString + "*";
      }
     String outString = String.format("%-35s%-
10s(%d)",arrDestinations[index],starString,arrCount[index]);
      System.out.println(outString);
```

```
    }
   System.out.println("\n\n");
 }
 public Question3_MEMO() throws Exception {
 kb = new BufferedReader(new InputStreamReader(System.in));
 char choice =' ';


 do {
      System.out.println("MENU");
      System.out.println();
      System.out.println("   Option A");


      System.out.println("   Option B");
      System.out.println("   Option C");
      System.out.println();
      System.out.println("Q - QUIT");
      System.out.println();
      System.out.println("Your choice?");
      choice = kb.readLine().toUpperCase().charAt(0);
      switch (choice) {
      case 'A': convertEuros();
                break;
      case 'B':divideGroup();
                break;
      case 'C':determinePopularity();
                break;
      case 'Q':
               System.out.println("QUIT");
     }
     } while (choice != 'Q');
    }

    public static void main(String[] args) throws Exception {
         new Question3_MEMO();
    }

   }
```