basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

EXEMPLAR 2014

MARKS: 150

TIME: 3 hours

This question paper consists of 18 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper contains THREE questions.

2. Answer ALL the questions.

3. The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.

4. This question paper is set in programming terms that are not specific to any particular programming language (Delphi/Java (making use of the Netbeans IDE)).

5. Make sure that you answer the questions according to the specifications that are given in each question. Marks will only be awarded based on the set requirements.

6. Only answer what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.

7. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.

8. Make sure that you develop routines, such as search, sort and selection, from first principles and not use the built-in features of a programming language for any of these routines.

9. You, as the programmer, must define all data structures. You may not use components provided within the user interface to store and later retrieve data.

10. Save your work regularly on the disk (CD/flash disk/DVD, et cetera) that you have been given, or on the disk space allocated to you for this examination.

11. Make sure that your examination number appears as a comment in the first line of code you did to answer a question. Also include the question number as part of the comment.

12. If printouts are required make printouts of the code of all the programs/classes/units that you did and NOT of the code that is generated automatically.

13. Printing must be done after the examination within the timeframe provided for printing.

14. At the end of this examination session, you must hand in the disk/CD with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you. Ensure that all files can be read.

15. You have been supplied with either a disk or disk space containing files you need to complete this question paper.

These files are password protected. To access the data files, use the password: **SecureData2014@%2#**

List of files provided:

Delphi Files

Question1:
Question1_P.dpr
Question1_P.res
Question1_U.dfm
Question1_U.pas

Question2:
Question2_P.dpr
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question3:
Question3_P.dpr
Question3_P.res
Question3_U.dfm
Question3_U.pas
Results.txt

Netbeans Files

Question1:
Question1.form
Question1.java

Question2:
Question2.form
Question2.java

Question3:
Question3.form
Question3.java
Results.txt

SCENARIO:

High school learners annually participate in the PC Athletics Championships in three different age groups in all the events. The age groups are u/15, u/17 and u/19. The description of these age groups are as follows:

DESCRIPTION	AGE GROUP
Learners turning 14 or 15 years of age	u/15
Learners turning 16 or 17 years of age	u/17
Learners turning 18 or 19 years of age	u/19

Follow the instructions provided for each question to complete the question paper.

SECTION A

QUESTION 1: GENERAL PROGRAMMING SKILLS

INSTRUCTIONS:

DELPHI	JAVA
The project **Question1** is provided to you in the Delphi folder: • Main form unit file named **Question1_U.pas** • Open the incomplete project file **Question1_P.dpr** in the **Question1** folder. • Insert your examination number as a comment in the first line of the unit file **Question1_U.pas**.	The project **Question1** is provided to you in the Netbeans folder: • Open the incomplete class named **Question1.java** contained in the **Source Packages**, **Question1Package**. • Insert your examination number as a comment in the first line of the class **Question1.java**.

Do the following:

- Compile and execute the program. The interface displays four different sections named QUESTION 1.1 to QUESTION 1.4. The program currently has no functionality.

- Complete the code for each section of QUESTION 1 as described in QUESTION 1.1 to QUESTION 1.4 below.

1.1 **Button [QUESTION 1.1]**

Write code to change the caption of the heading label to display the following text:

PC Athletics Championships (2)

1.2 Button [QUESTION 1.2]

Every athlete needs a name tag for the championships. Allow the user to enter the name, surname and ID of an athlete in the text fields provided on the user interface.

Create a name tag that contains the following information about the athlete:

Line 1: Surname and initials in capital letters followed by a full stop

Line 2: Gender (Male or Female)

HINT: The four digits in position 7 to 10 of an identity number (ID number) indicate a person's gender. The following applies:

>=5000 : Male
< 5000 : Female

Compile the string in the following format:

Display the string in the output area provided.

```
Athlete:  <SURNAME><space><INITIALS><full stop>
          <GENDER>
```

Example: For an athlete with the surname Mbali, full names Tandi Nkandla and ID number 9904123456081, the following name tag will be compiled:

```
Athlete: MBALI TN.
         Female
```

(12)

1.3 Button [QUESTION 1.3]

The first six digits of an ID number represent a person's date of birth in the format <YYMMDD>, where YY refers to the year, MM to the month and DD to the day. Validation of the date of birth is not required.

Use the athlete's ID number and determine the age group the athlete may participate in for the championships. The championships only allow for three age groups, that is u/15, u/17 and u/19. (See table on page 4.)

NOTE: Assume the championships take place during the year 2014 when calculating an athlete's age.

Add a string containing the athlete's age group to the name tag that was created in QUESTION 1.2. Construct the string in the following format:

```
Age group: u/<Age group>
```

Learners who turn 20 years of age or older are not eligible to participate. Create the string 'Not eligible to participate' for these athletes as part of the string for the name tag.

The output must be displayed in the output area provided.

Examples of name tags:

Example 1: The name tag for athlete Tandi Nkandla Mbali with ID number 9904123456081:

```
Athlete: MBALI TN.
         Female
Age group: u/15
```

Example 2: The name tag for athlete John Smith with ID number 9308156786082:

```
Athlete: SMITH J.
         Male
Age group: Not eligible to participate
```

(13)

1.4 **Button [QUESTION 1.4]**

Refreshments will be sold at the championships. Discount is allowed based on the number of items a customer buys. The following applies:

A discount of 20% is granted for every two items purchased.

Examples:

- If one item is purchased, no discount is granted.
- If two items are purchased, a 20% discount is granted on the total amount.
- If three items are purchased, a 20% discount is granted on the total amount to be paid for the first two items. The full price is paid for the third item.
- If four items are purchased, a 20% discount is granted on the total amount.
- If five items are purchased, a 20% discount is granted on the total amount to be paid for the first four items. The full price is paid for the fifth item, et cetera.

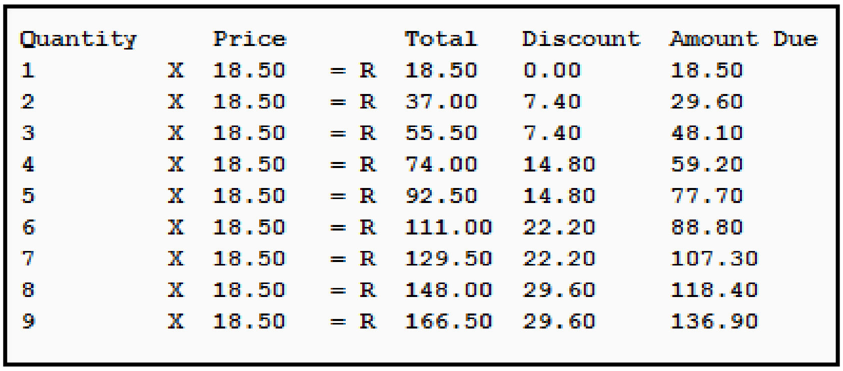
The user has to enter the price of one item and the quantity the customer wants to buy.

When the user clicks on the button, a list must be displayed that shows the quantity, price, total amount due, discount and final amount due. The list must display the information on quantities that range from one to the number of items the customer wants to buy.

Make use of a control loop structure to generate and display the required list.

Example:

If the price of the item is R18,50 and the customer buys 9 of these items, the following list must be displayed:



```
Quantity      Price          Total    Discount   Amount Due
1          X  18.50     = R  18.50    0.00       18.50
2          X  18.50     = R  37.00    7.40       29.60
3          X  18.50     = R  55.50    7.40       48.10
4          X  18.50     = R  74.00    14.80      59.20
5          X  18.50     = R  92.50    14.80      77.70
6          X  18.50     = R  111.00   22.20      88.80
7          X  18.50     = R  129.50   22.20      107.30
8          X  18.50     = R  148.00   29.60      118.40
9          X  18.50     = R  166.50   29.60      136.90
```

Do the following:

- Allow the user to enter the price of a single item and the quantity that the customer wants to purchase from the keyboard using a Message dialog box. No validation of the input values is required.

- Calculate the discount and the amount due and display it in the output area.

  amount due = total – discount

- Allow the user to enter the amount the customer pays and do the following:

  - Calculate and display the change in the text area.

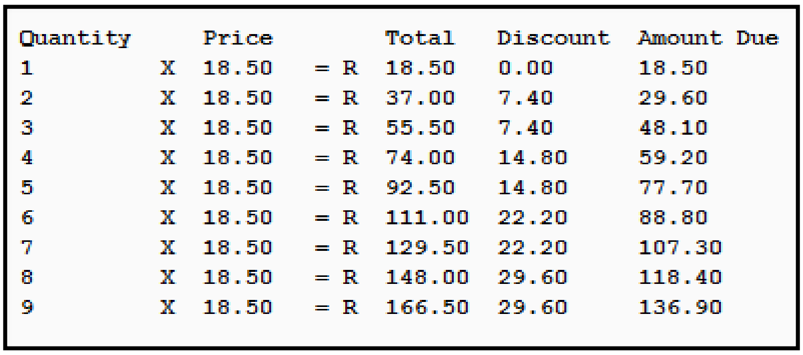
    change = amount customer pays – amount due

  - Write code to calculate and display the change the customer has to receive in terms of the number of one rand, 50c, 20c, 10c coins, in the text area provided. For change between 1c and 9c, an extra 10c should be given as change.

  - Display the amount the customer paid and the correct change below the previously displayed list.

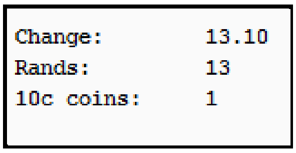
Example 1:

If the price of the item is R18,50 and the customer buys 9 of these items, the following list will be displayed:



```
Quantity      Price         Total    Discount   Amount Due
1          X  18.50    = R  18.50    0.00       18.50
2          X  18.50    = R  37.00    7.40       29.60
3          X  18.50    = R  55.50    7.40       48.10
4          X  18.50    = R  74.00    14.80      59.20
5          X  18.50    = R  92.50    14.80      77.70
6          X  18.50    = R  111.00   22.20      88.80
7          X  18.50    = R  129.50   22.20      107.30
8          X  18.50    = R  148.00   29.60      118.40
9          X  18.50    = R  166.50   29.60      136.90
```

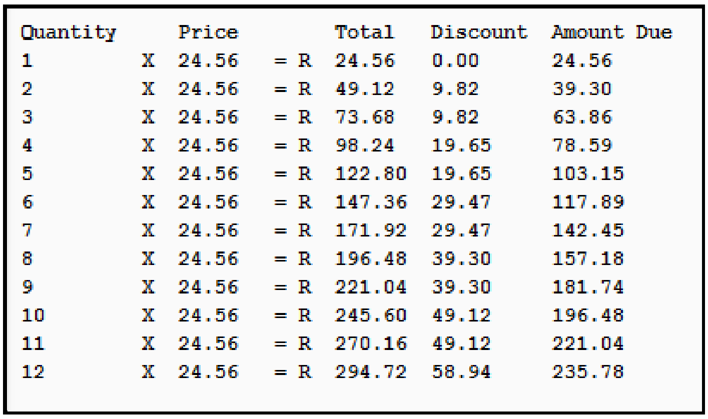
If the customer pays an amount of R150,00, the following information will be displayed:



```
Change:         13.10
Rands:          13
10c coins:      1
```

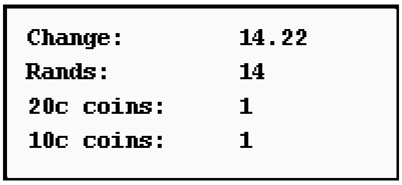
Example 2:

If the price of the item is R24,56 and the customer buys 12 of these items, the following list will be displayed:



```
Quantity      Price         Total    Discount   Amount Due
1          X  24.56    = R  24.56    0.00       24.56
2          X  24.56    = R  49.12    9.82       39.30
3          X  24.56    = R  73.68    9.82       63.86
4          X  24.56    = R  98.24    19.65      78.59
5          X  24.56    = R  122.80   19.65      103.15
6          X  24.56    = R  147.36   29.47      117.89
7          X  24.56    = R  171.92   29.47      142.45
8          X  24.56    = R  196.48   39.30      157.18
9          X  24.56    = R  221.04   39.30      181.74
10         X  24.56    = R  245.60   49.12      196.48
11         X  24.56    = R  270.16   49.12      221.04
12         X  24.56    = R  294.72   58.94      235.78
```

If the customer pays an amount of R250,00, the following information will be displayed below the list:



```
Change:         14.22
Rands:          14
20c coins:      1
10c coins:      1
```

(23)

- Enter your examination number as a comment in the first line of the program file.
- Save the program.
- You may have to print out the code.

[50]

QUESTION 2: OBJECT-ORIENTATED PROGRAMMING

The highlight of the championships is the 4 x 100 metres (4x100m) relay event for Boys u/19. The current record for the Boys u/19 4x100m relay item was set in 2009 by the team of Bristol House with a winning time of 41,13 seconds.

INSTRUCTIONS:

DELPHI	JAVA
The project **Question2** is provided to you in the Delphi folder: • Main form unit file named **Question2_U.pas** • Open the incomplete project file **Question2_P.dpr** in the **Question2** folder. • Insert your examination number as a comment in the first line of the unit file **Question2_U.pas**.	The project **Question2** is provided to you in the Netbeans folder: • Open the incomplete class named **Question2.java** contained in the **Source Packages**, **Question2Package**. • Insert your examination number as a comment in the first line of the class **Question2.java**.

Do the following:

- Compile and execute the program. The program currently has no functionality.
- Complete the code for each section of QUESTION 2 as described in QUESTION 2.1 to QUESTION 2.2 below.

2.1 Create a new object class called **RelayEvent** based on the class diagram below and a description of the task each method has to perform in QUESTIONS 2.1.1 to 2.1.4.

The class diagram below lists the attributes and methods that are required.

```
RelayEvent
Attributes:
   -  event
   -  team
   -  year
   -  recordTime
Methods:
   +  Constructor (event, team, year, recordTime)
   +  getEvent()
   +  getTeam()
   +  getYear()
   +  getRecordTime()
   +  checkForRecord(newTeam, newTime)
   +  toString()
```

Descriptions of the various attributes:

Attribute	**Description**
event	The title of the event
team	The name of the team currently holding the record. If more than one team holds the record, the names of all the teams holding the record must be listed, separated by semicolons (;).
year	The year during which the current record time was recorded. If more than one team holds the record, the years in which the record was equalled must be listed, separated by semicolons (;).
recordTime	The time recorded as the record time, in seconds

(4)

2.1.1 Write code for a **constructor** that receives parameters to initialise all attributes of the class. (2)

2.1.2 Create **accessor** (get) methods to return the attribute values of the class. (5)

2.1.3 Write a **checkForRecord** method that receives the name of the team and the recorded time. The method needs to determine whether the recorded time is a new record for this event.

If the time received is:

- A new record, the relevant attributes must be updated with the new values

- Equal to the current record time, the name of the team received by this method must be added to the **team** attribute and the current year must be added to the **year** attribute of the class separated by semicolons (;). Use the system's date function to determine the current year.

Example of the current record for the Boys u/19 4x100m relay event set by the team of Bristol House in the year 2009:

```
Team: Bristol House
Year: 2009
Time: 41.13
```

Example of the previous record for the Boys u/19 4x100m relay event:

```
Team: Kendal High; Fenham College
Year: 2006; 2008
Time: 44.23
```

(13)

2.1.4 Write a **toString** method to return a string comprising the title of the event, the name of the team, the year the record was set and the record time for the event.

The string must be formatted as follows:

```
Current record for <the title of the event>

Team: <name of team>
Year: <year record was achieved>
Time: <current time for record> seconds
```

Example of output of current record:

```
Current record for Boys u/19 4x100m relay

Team: Bristol House
Year: 2009
Time: 41.13 seconds
```

(4)

2.2 Do the following to code the tasks that must be performed when each of the buttons are activated:

2.2.1 **Button [Current Record Holder]**

Write code to declare a **Boys19Relay** object based on the **RelayEvent** class.

Instantiate the object using the following specific data: (No input from the user is required.)

```
Event: Boys u/19 4x100m relay
Team:  Bristol House
Year:  2009
Time:  41.13
```

Make use of the **toString** method to display the information about the current **Boys19Relay** object. (5)

2.2.2 **Button [Validate Time]**

Allow the user to enter the name of the school the winning team is from and the recorded winning time in seconds, for the Boys u/19 4x100m relay event.

Validate the recorded winning time that was entered to make sure a numerical value is entered. If an invalid time is entered, a suitable error message must be displayed and the user must be allowed to re-enter the time.

The program can only continue once a valid time has been entered.

If a valid time is entered, a button must be dynamically instantiated with the following specifications and functionality:

Specifications:

DELPHI: GroupBoxQ22	
Left:	72
Top:	158
Height:	55
Width:	235
Caption:	Test Record
OnClick:	btnCheckStatus
JAVA:	
Name of the Panel:	pnlQ2_2
Name of the Button:	btnTestRecord
Text on the Button:	Test Record
Bounds:	35, 235, 205, 55

Functionality:

Make use of the **checkForRecord** method of the **Boys19Relay** object to check whether the recorded winning time that was entered is a new record.

Make use of the **toString** method to display the information on the current record for the Boys u/19 4x100m relay event.

Examples of output:

Example 1:

Output if the team of Griffiths House completed the event in 53,23 seconds:

```
Current record for Boys u/19 4x100m relay

Team: Bristol House
Year: 2009
Time: 41.13 seconds
```

Example 2:

Output if the team of Fenham College completed the event in 41,13 seconds:

```
Current record for Boys u/19 4x100m relay

Team: Bristol House; Fenham College
Year: 2009; 2014
Time: 41.13 seconds
```

Example 3:

Output if the team of Edenburgh High School completed the event in 40,45 seconds:

```
Current record for Boys u/19 4x100m relay

Team: Edenburgh High School
Year: 2014
Time: 40.45 seconds
```

(21)

2.2.3 **Button [Record Status]**

If the current record has been set in 2013 or 2014, display a message 'Recent record', otherwise display 'Old record'.

Example 1:

Output if the team of Fenham College completed the event in 41,13 seconds in 2014:

```
Current record for Boys u/19 4x100m relay

Team: Bristol House; Fenham College
Year: 2009; 2014
Time: 41.13 seconds

Recent record
```

Example 2:

Output if the team of Broadlands Technical High completed the event in 49,20 seconds in 2014:

```
Current record for Boys u/19 4x100m relay

Team: Bristol House
Year: 2009
Time: 41.13 seconds

Old record
```

(6)

- Enter your examination number as a comment in the first line of the class and the form.
- Save the program.
- You may have to print out the code (in which case you have to print the code contained in both the class and the main form).

[60]

QUESTION 3: PROBLEM-SOLVING

Schools are interested in a report containing place achievements. A maximum of eight athletes take part in the final round of each event. The place achievement report lists the number of athletes per school who achieved a first, second, et cetera, up to an eighth place at the championships.

The following schools participated:

School	Abbreviation
Bedworthpark High School	BPK
Bristol House	BSL
Broadlands Technical High	BRT
Griffiths House	GFH
Fenham College	FNH
Edenburgh High School	EDB
Rethanda College	RTN
Sheffield High School	SFD

Description of the contents of the **Results.txt** text file:

The places obtained by the athletes of the different schools for each event have been recorded in the **Results.txt** text file in the following format:

```
<Entry ID>;<abbreviation of participating school>#<item>-
<gender><space><age group>;<place achieved>
```

Example of data in the text file:

```
71;BSL#Javelin-Boys u/15;7
154;FNH#Shot Put-Girls u/17;1
40;GFH#100m-Girls u/17;2
:
```

The first two lines of data can be interpreted as follows:

- **`71;BSL#Javelin-Boys u/15;7:`**

  Entry number 71, Bristol House (BSL) achieved a 7th place in the Javelin Boys u/15 event.

- **`154;FNH#Shot put-Girls u/17;1:`**

  Entry number 154, Fenham College (FNH) achieved a 1st place in the Shot Put Girls u/17 event.

INSTRUCTIONS:

DELPHI	JAVA
The project **Question3** is provided to you in the Delphi folder: • Main form unit file named **Question3_U.pas** • Open the incomplete project file **Question3_P.dpr** in the **Question3** folder. • Insert your examination number as a comment in the first line of the unit file **Question3_U.pas**.	The project **Question3** is provided to you in the Netbeans folder: • Open the incomplete class named **Question3.java** contained in the **Source Packages**, **Question3Package**. • Insert your examination number as a comment in the first line of the class **Question3.java**.

Do the following:

- Compile and execute the program. The program currently has no functionality.
- Complete the code for each section of QUESTION 3 as described in QUESTION 3.1 to QUESTION 3.3 below.

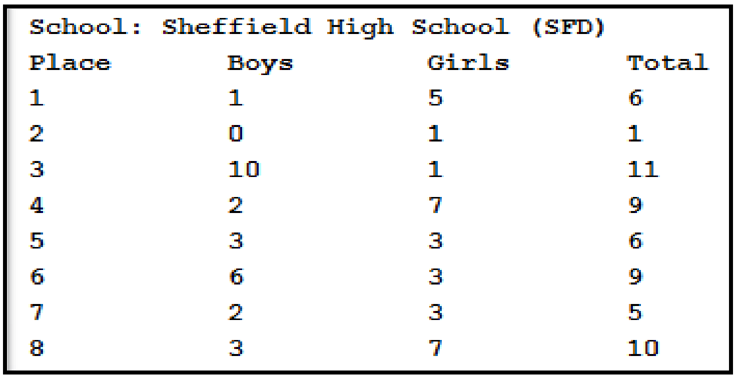
3.1 **Button [Display Report]**

Make use of the data captured in the **Results.txt** text file, the array of school names and the array of school abbreviations supplied to compile the achievement report for a school, for example Sheffield High School.

Select the abbreviated school name from the list supplied in the combo box. If, for example, SFD was selected for Sheffield High School, a report must be displayed that meets the following requirements:

- The report must be provided with a suitable heading and subheadings.
- There must be eight places for each event.
- The number of boys and girls for each place must be displayed separately.
- The total number of athletes for the first place up to the eighth place must be calculated and displayed.

Example of the report for Sheffield High School (SFD):



```
School:  Sheffield High  School  (SFD)
Place        Boys        Girls       Total
1            1           5           6
2            0           1           1
3            10          1           11
4            2           7           9
5            3           3           6
6            6           3           9
7            2           3           5
8            3           7           10
```

(19)

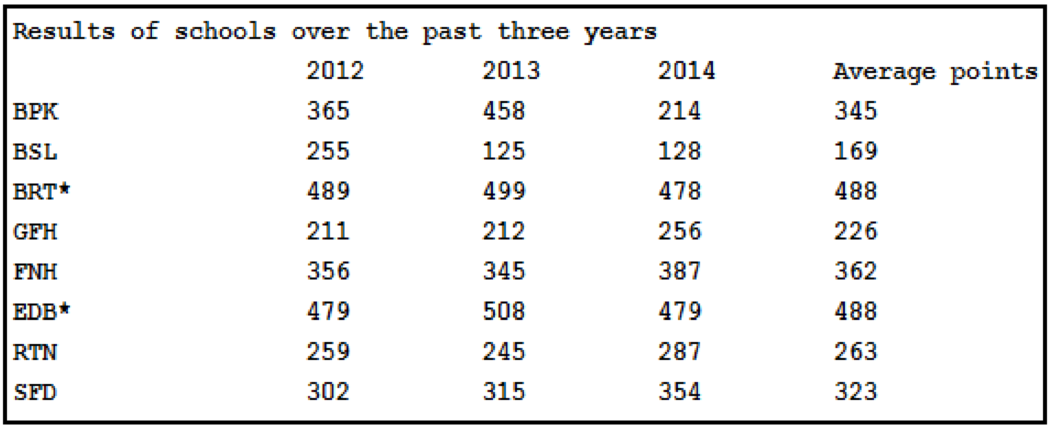
3.2 Button [Average Results]

A two-dimensional array named **arrSchoolResults** has been supplied consisting of the total points achieved by the schools over the last three years.

A report is required to indicate the average points of the eight schools over the last three years.

Write code to do the following:

- Calculate the average points for each school.
- Determine the name(s) of the school(s) that achieved the highest average points.
- A star symbol (*) must be used to indicate the name(s) of the school(s) that achieved the highest average points.



```
Results of schools over the past three years
                  2012       2013       2014       Average points
BPK               365        458        214        345
BSL               255        125        128        169
BRT*              489        499        478        488
GFH               211        212        256        226
FNH               356        345        387        362
EDB*              479        508        479        488
RTN               259        245        287        263
SFD               302        315        354        323
```

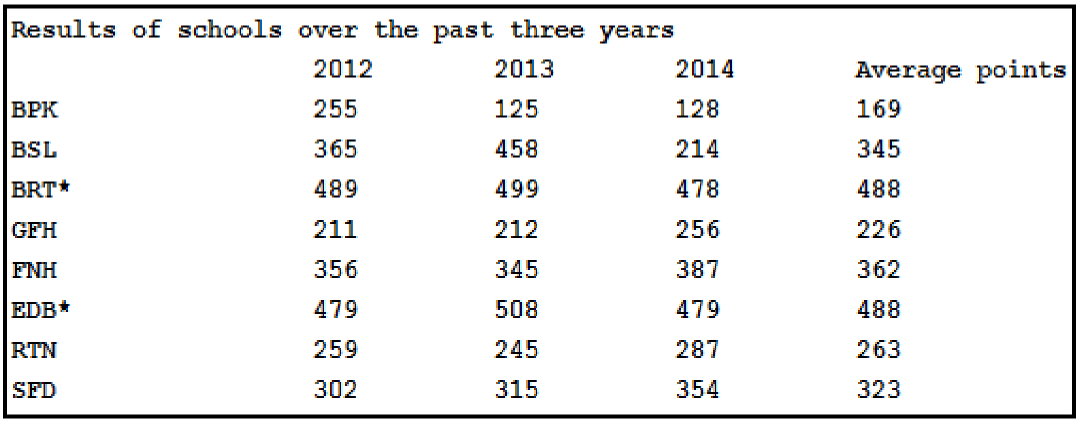
(17)

3.3 Button [Swap Points]

The point statistics for the first two schools have been captured incorrectly. Write code to swap the points for the first two schools 'BPK' and 'BSL'.

Display the amended statistics by calling the [Average Results] button.

Example of output:



```
Results of schools over the past three years
                  2012       2013       2014       Average points
BPK               255        125        128        169
BSL               365        458        214        345
BRT*              489        499        478        488
GFH               211        212        256        226
FNH               356        345        387        362
EDB*              479        508        479        488
RTN               259        245        287        263
SFD               302        315        354        323
```

(4)

- Enter your examination number as a comment in the first line of the program file.
- Save the program.
- A printout of the code may be required.

[40]

TOTAL: 150