basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIOR CERTIFICATE EXAMINATIONS

INFORMATION TECHNOLOGY P1

2018

MARKS: 150

TIME: 3 hours

This question paper consists of 18 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper is divided into THREE sections. Candidates must answer ALL the questions in each of the THREE sections.

2. 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.

3. This question paper is set with programming terms that are specific to the Delphi programming language.

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

5. Answer only 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.

6. 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.

7. Routines, such as search, sort and selection must be developed from first principles. You may NOT use the built-in features of Delphi for any of these routines.

8. All data structures must be defined by you, the programmer, unless the data structures are supplied.

9. You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.

10. Make sure that your examination number appears as a comment in every program that you code, as well as on every event indicated.

11. If required, print the programming code of all the programs/classes that you completed. You will be given half an hour printing time after the examination session.

12. At the end of this examination session you must hand in a disk/CD/DVD/ flash disk 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 for this examination session. Ensure that all files can be read.

13. The files that you need to complete this question paper have been given to you on the disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

NOTE:

Candidates must use the file **DataENGJune2018.exe**.

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **Bks18@Lib*#?**

Once extracted, the following list of files will be available in the folder **DataENGJune2018**:

SUPPLIED FILES

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

Question2:
Book_U.pas
BooksData.txt
BooksDataBackup.txt
Question2_P.dpr
Question2_P.dproj
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question3:
Question3_P.dpr
Question3_P.dproj
Question3_P.res
Question3_U.dfm
Question3_U.pas

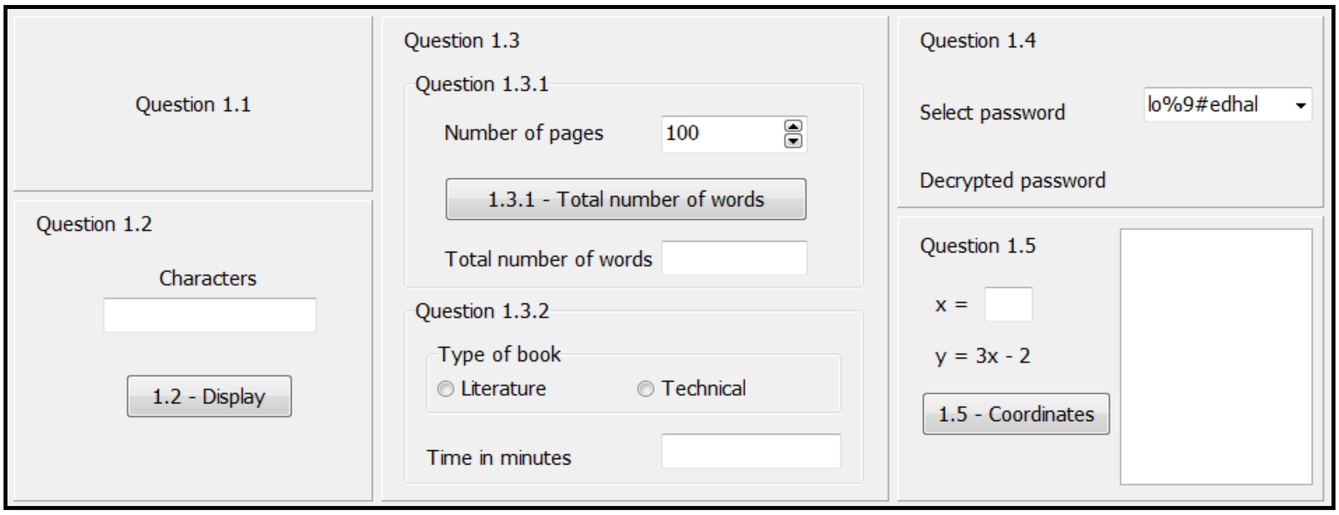
SECTION A

QUESTION 1: GENERAL PROGRAMMING SKILLS

Do the following:

- Open the incomplete program in the **Question1** folder.
- Enter your examination number as a comment in the first line of the **Question1_U.pas** file.
- Compile and execute the program. The GUI displays FIVE sections labelled QUESTION 1.1 to QUESTION 1.5. Currently the program has no functionality.

  The following user interface is displayed:



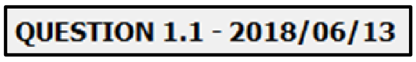
- Follow the instructions below to complete the code for EACH section of QUESTION 1, as described in QUESTION 1.1 to QUESTION 1.5.

1.1 **OnCreate Event [Question 1.1]**

Write code in the 'OnCreate' event handler of the Form to change the text displayed on the **pnlQ1_1** panel as follows when the program is executed:

- Assign the existing text from the panel to a variable.
- Add a dash ('-') and use the relevant function to also add the system date to the text.
- Change the text to upper case and display the text in bold.

Example of output:



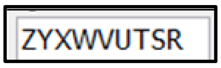
```
QUESTION 1.1 - 2018/06/13
```

NOTE: The format of the system date may differ from the example. (6)

1.2 **Button [1.2 - Display]**

Write code using the loop counter variable **cCounter** and a loop to compile a string containing the letters 'Z' to 'R' in the variable **sOutput**. Assign the variable to the edit box to display.

Example of output:



```
ZYXWVUTSR
```

(6)

1.3 The time it takes a student to read a textbook depends on the total number of words the textbook contains and whether the content is of a technical nature or not.

1.3.1 **Button [1.3.1 - Total number of words]**

The average number of words per page in a textbook of standard size has been declared as 300 in a constant variable called **WordsPerPage**.

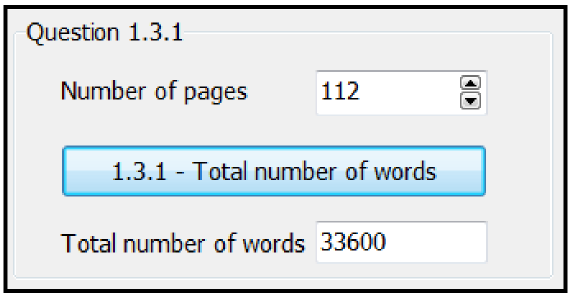
The user needs to select the number of pages the book contains from the spin edit component.

Complete the provided code to do the following:

- Declare an integer variable for the number of pages.
- Obtain the value of number of pages from the spin edit component.
- Use the constant variable and the number of pages to calculate the total number of words.
- Display the total number of words in the edit box provided.

NOTE: The variable **iTotalNumWords** has been declared as a global variable to be used again in QUESTION 1.3.2.

Example of output:



Number of pages 112

Total number of words 33600

(5)

1.3.2 **Radio group [1.3.2 – Type of book]**

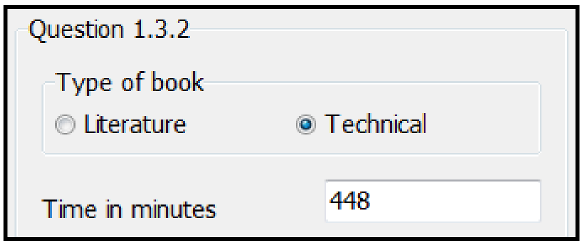
Research shows that a student can read a literature book at the speed of 250 words per minute and a technical book at the speed of 75 words per minute. These values have been declared as constant values using the variables **WordsLit** and **WordsTech** respectively. The user needs to select the type of book from the radio buttons provided.

Write code to do the following to determine the time it should take a student to read the book:

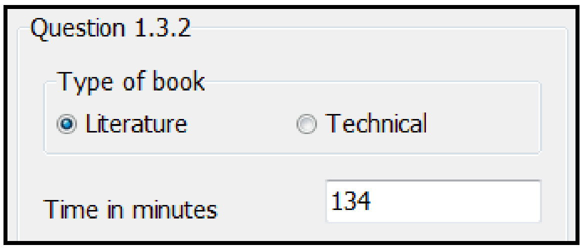
- Determine the type of book that has been selected.
- Use the global variable **iTotalNumWords** and the relevant constant variable to calculate the time (in minutes) it would take a student to read the book.
- Display the time in edit box **edtTimeInMinutes** provided.

NOTE: If you were not able to calculate the total number of pages in QUESTION 1.3.1, assign a value of 33 600 to the global variable **iTotalNumWords** to be able to do QUESTION 1.3.2.

Example of output if the book is a technical book and has 112 pages, which results in 33 600 words, as calculated in QUESTION 1.3.1:


Question 1.3.2
Type of book
○ Literature [✓ ● Technical]
Time in minutes 448

Example of output if the book is a literature book and has 112 pages, which results in 33 600 words, as calculated in QUESTION 1.3.1:


Question 1.3.2
Type of book
[✓ ● Literature] ○ Technical
Time in minutes 134

(7)

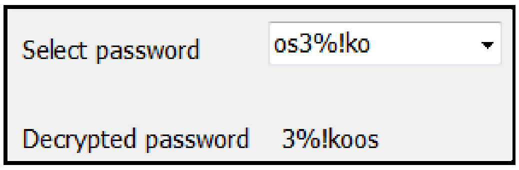
1.4 **Combo box [cmbQ1_4]**

The user needs to select a password from the combo box **cmbQ1_4** which contains a number of encrypted passwords.

Write code to decrypt the selected password according to the instructions below.

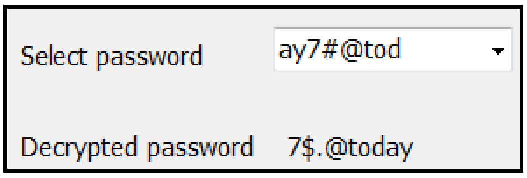
- The first two characters of the password must be moved to the end of the password.
- If the password contains a hash character (#), replace it with a dollar character ($), followed by a full stop (.).

Example of output if the password **os3%!ko** has been selected:



```
Select password      os3%!ko

Decrypted password   3%!koos
```

Example of output if the password **ay7#@tod** has been selected:



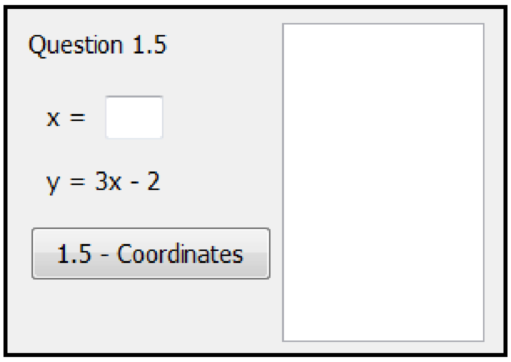
```
Select password      ay7#@tod

Decrypted password   7$.@today
```

(13)

1.5 **Button [1.5 - Coordinates]**

The formula for a straight line is $y = mx + c$. The formula must be used to calculate the value of the y-coordinate of the straight line if the value of x is entered by the user.

The values of 3 and -2 will be used as the values for m and c in the formula and has been assigned to the variables **iM** and **iC** respectively in the code provided.



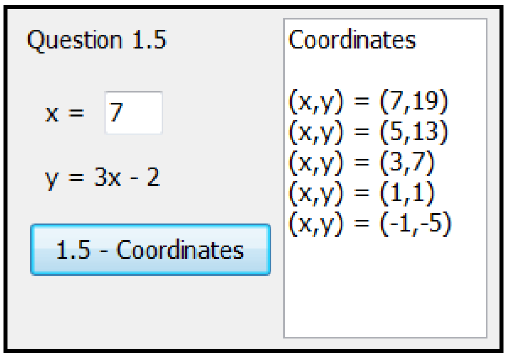
Write code to display the (x,y) coordinates of a straight line in the output area using the following instructions:

- Clear the output area.
- Display the heading 'Coordinates' followed by a blank line.
- Obtain the value of x that was entered from edit box **edtQ1_5**.
- Calculate the value of y using the formula $y = mx + c$.

  Example of the calculation of y if the value 7 is entered for x:

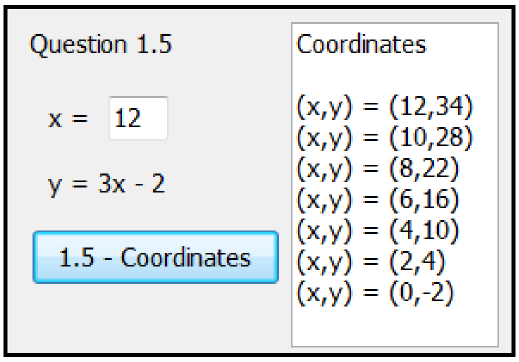
  $$\begin{aligned} y &= mx + c \\ &= 3(7) + (-2) \\ &= 19 \end{aligned}$$
- Display the values of coordinates x and y in the output area in the format shown in the example below, for example (x,y) = (7,19) for the first coordinate.
- Decrease the value of x by 2 and repeat the previous two steps (calculate the value of y and display the coordinates) until the value of y is equal to zero (0) or is a negative value.

Example of output if the value of 7 was entered for the value of x:



```
Coordinates

(x,y) = (7,19)
(x,y) = (5,13)
(x,y) = (3,7)
(x,y) = (1,1)
(x,y) = (-1,-5)
```

Example of output if the value of 12 was entered for the value of x:



```
Coordinates

(x,y) = (12,34)
(x,y) = (10,28)
(x,y) = (8,22)
(x,y) = (6,16)
(x,y) = (4,10)
(x,y) = (2,4)
(x,y) = (0,-2)
```

(13)

- Ensure that your examination number has been entered as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

TOTAL SECTION A: 50

SECTION B

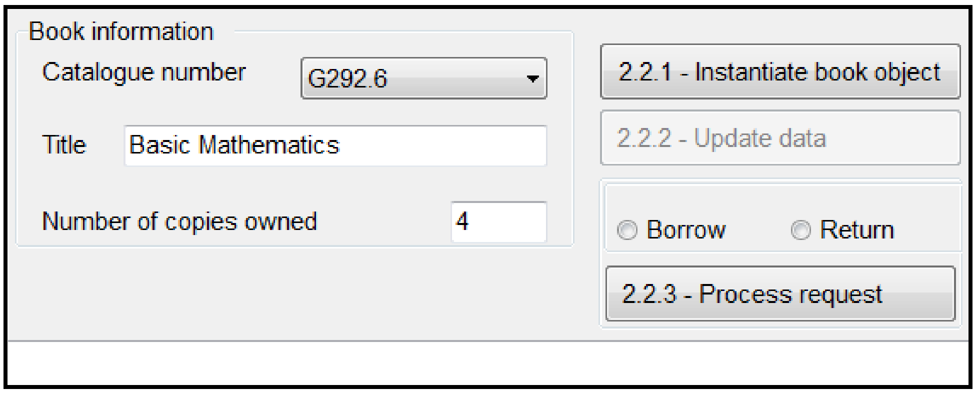
QUESTION 2: OBJECT-ORIENTATED PROGRAMMING

Students are allowed to borrow and return books in the research wing of the library during the week that they are working on their research projects. Data has been captured in a text file on the books that were borrowed and returned during a research week. The librarian needs to use the captured data to see how many copies of the books are available currently to allow books to be borrowed and returned during the next research week.

Do the following:

- Open the incomplete project file called **Question2_P.dpr** in the **Question2** folder.
- Add your examination number as a comment in the first line of both the **Question2_U.pas** and **Book_U.pas** files.
- Compile and execute the program. Currently the program has limited functionality.

The following GUI is displayed when the program is executed:



- Complete the code as described in QUESTION 2.1 and QUESTION 2.2 below.

2.1 Complete the code in the **TBook** object class provided, as described in QUESTION 2.1.1 to QUESTION 2.1.7 that follow.

2.1.1 Declare the attributes of the class using the information in the table below.

NAMES OF ATTRIBUTES	DESCRIPTION
fTitle	The title of the book
fCatNum	A unique catalogue number assigned to each book in the library, e.g. D324.2
fNumBooks	An integer number that indicates the number of copies of this book that the library owns, e.g. 12
fNumBooksOut	An integer number that indicates the number of copies of this book that is currently borrowed, e.g. 5

(4)

2.1.2 Write code to create a **constructor** which receives THREE parameter values (the title, the catalogue number of the book and the number of copies of the book that the library owns) and initialises the relevant attributes using the parameter values.

The attribute for the number of copies of the book out (currently borrowed) must be initialised to zero. (5)

2.1.3 Write an accessor method **getCatalogueNumber** for the catalogue number attribute of the book. (2)

2.1.4 Write an accessor method **getNumBooksOut** for the number of books out attribute. (2)

2.1.5 Write code for a mutator method **setNumBooksOut** to receive an integer as a parameter and set the **fNumBooksOut** attribute to the parameter value. (3)

2.1.6 Write code for a method called **isBookAvailable** that returns a Boolean value. This method must use the **fNumBooks** and **fNumBooksOut** attributes to determine whether a copy of the book is currently available in the library or not.

The return value must be **true** if a book is available or **false** if not available. (3)

2.1.7 Write code to create a **toString** method to return a string formatted as follows:

The library owns < **`number of copies owned`**> copies of the book titled <**`book title`**> [<**`catalogue number`**>].

The number of copies currently out is <**`number of copies of the book currently borrowed`**>. (5)

Do the following to complete the code for Buttons 2.2.1, 2.2.2 and 2.2.3 in the main form unit, as described below.

2.2.1 **Button [2.2.1 - Instantiate book object]**

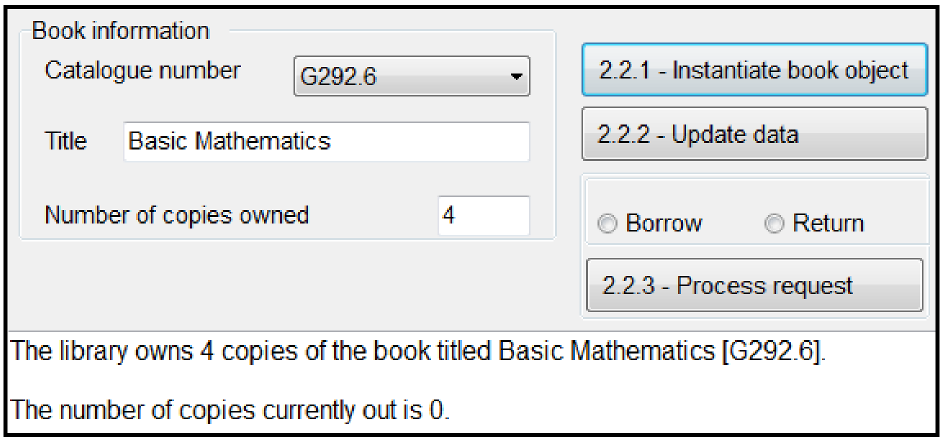
The user needs to select the catalogue number of the book from the combo box. Code has been provided to show the title and the number of copies of the selected book that the university owns in the relevant edit boxes.

Write code to do the following:

- Extract the required data items from the relevant components and instantiate a new **objBook** object.

- Call the **toString** method to display the details of the object in the output area provided.
- Activate the **Update data** button.

Example of output for the book if catalogue number G292.6 has been selected and the object has been instantiated:



```
The library owns 4 copies of the book titled Basic Mathematics [G292.6].

The number of copies currently out is 0.
```

(8)

2.2.2 **Button [2.2.2 - Update data]**

A text file **BooksData.txt** contains the records of books that were borrowed and returned during a research week.

The data in the text file is saved in the following format:

```
<catalogue number of book>#<a character indicating whether the book was borrowed (B) or returned (R)>
```

Example of the first six lines of data in the **BooksData.txt** text file:

```
G292.6#B
E202.4#B
C284.3#B
D237.9#B
D237.9#B
G292.6#R
```

The text file must be used to determine the number of copies of the selected book that were borrowed and returned during the research week.

Write code to do the following:

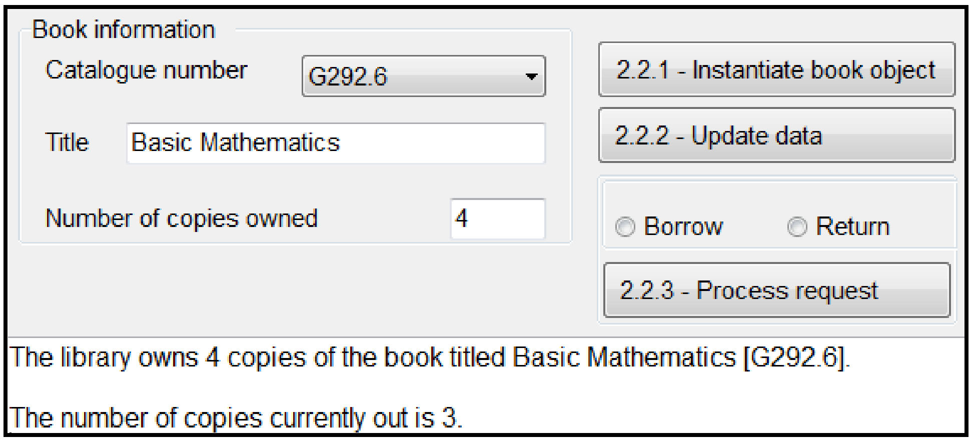
- Test whether the text file **BooksData.txt** exists or not. Display a suitable message if the text file does not exist and exit the event handler.

- Do the following if the text file exists:

    - Use a loop and search the text file for all instances of the catalogue number of the book object.
    - Determine how many times the book has been borrowed ("B") and returned ("R") and use these values to determine the number of copies currently out of the library using the formula:

      `BooksOut = Number borrowed – Number returned`
    - Call the **setNumBooksOut** method and use the **BooksOut** value as the argument.
    - Call the **toString** method to display the updated details of the book object in the output area provided.
    - Activate the **Process request** button.

Example of output for the **Update data** button if catalogue number G292.6 has been selected by the user and an object of this book has been instantiated:



```
The library owns 4 copies of the book titled Basic Mathematics [G292.6].

The number of copies currently out is 3.
```

(20)

2.2.3 **Button [2.2.3 - Process request]**

Special requests to borrow books from the research wing can be granted if copies of the book are available. Students who did not return books that they borrowed during the research week are also allowed to return the books.

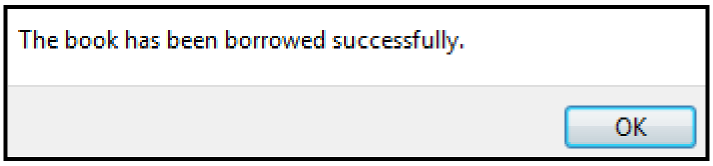
The radio group component must be used to select to either borrow or return the selected book. If the request to borrow or return the selected book has been granted, the text file **BooksData** and the object details of the book must be updated.

Write code for the following algorithm to process the borrow or return request for the selected book:

```
Set the provided sCode variable to an empty string
If Borrow has been selected from the radio group
    Test whether the book is available using the relevant method
        Display a suitable message
        Set sCode to 'B'
    else
        Display a message to indicate there are no copies available
If Return has been selected from the radio group
    If number of books out is 0
        Display 'Invalid request'
    else
        Display a message indicating that the book has been returned
        Set sCode to 'R'
If sCode contains data
    Write the catalogue number of the book and the value of sCode
    to the BooksData.txt text file in the correct format, e.g. G292.6#R
    Clear the selection from the radio group
    Update the object's data by executing the code in the btnQ2_2_2
    button.
```

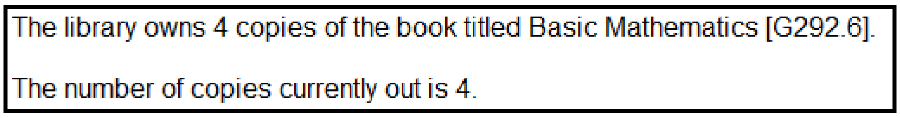
NOTE: A **Restore text file** button has been provided that can be used to restore the content of the **BooksData** text file if the content of the **BooksData** text file becomes corrupted while testing your code. This will copy the data in the **BooksDataBackup** text file to the **BooksData** text file.

Example of output if a student requested to borrow the book titled *Basic Mathematics* if copies of this book are available to be borrowed:



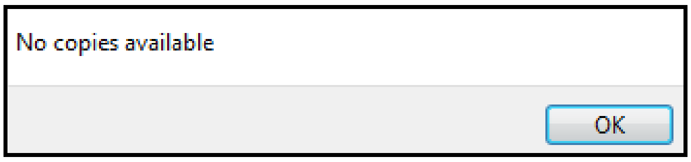
```
The book has been borrowed successfully.
```

The following updated data should be displayed in the output area:



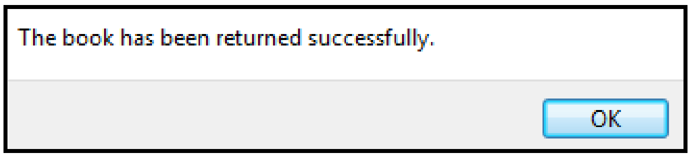
```
The library owns 4 copies of the book titled Basic Mathematics [G292.6].

The number of copies currently out is 4.
```

Example of output if a student requested to borrow the book titled *Basic Mathematics* if no copy of this book is available to be borrowed:



```
No copies available
```

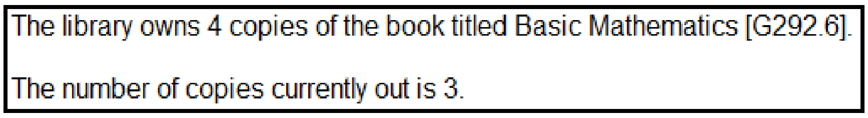
Example of output if a student requested to return the book titled *Basic Mathematics*:



```
The book has been returned successfully.
```

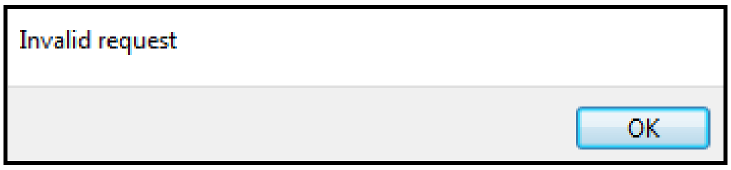
OK

The following updated data should be displayed in the output area:



```
The library owns 4 copies of the book titled Basic Mathematics [G292.6].

The number of copies currently out is 3.
```

Example of output if there is a request to return the book titled *Coding in Objects* with catalogue number C284.3. All the copies of this book have been returned. Therefore the request is invalid.



```
Invalid request
```

OK

(18)

- Ensure that your examination number has been entered as a comment in the first line of the class file and the form.
- Save your program.
- Print the code contained in both the class file and form, if required.

TOTAL SECTION B: 70

SECTION C

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

SCENARIO

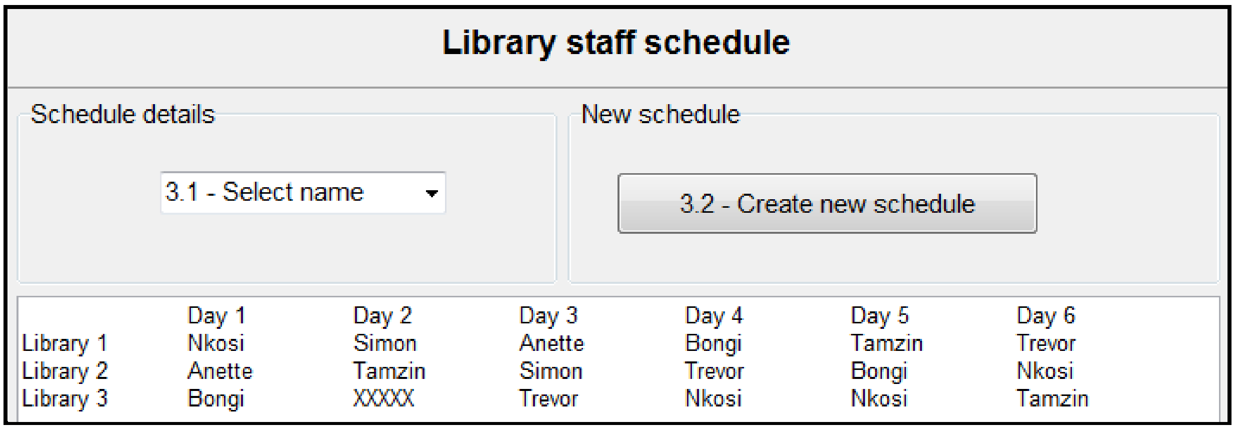
The university has three libraries. The first two libraries are open for six days of the week and the third library is open for five days of the week. There are six staff members who will be on duty in the libraries on the days the libraries are open. The manager of the library requires a program to create and maintain a placement schedule for staff members.

Do the following:

- Compile and execute the program in the **Question3** folder. Currently the program has no functionality.
- Complete the code for each section of the question as described in QUESTION 3.1 to QUESTION 3.3.

Supplied GUI:

The GUI below represents an interface used by the manager of the library to allocate duties shifts to staff members.



```
          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
```

The following code has been provided:

- An array called **arrStaff** which contains the names of the six staff members.

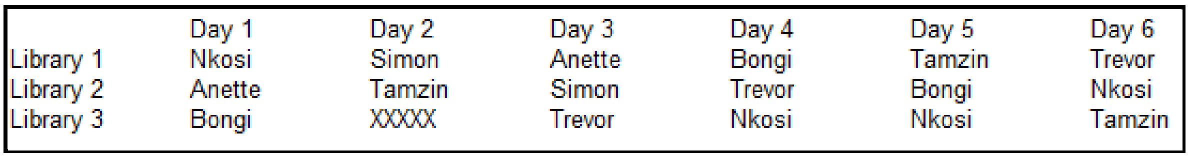
```
arrStaff: array [1 .. 6] of String =
  ('Trevor','Nkosi','Tamzin','Anette','Bongi','Simon');
```

- The declaration of a two-dimensional array called **arrPlacements** that will be used to store the names of staff members according to the days they will be on duty (columns 1 to 6) in each library (rows 1 to 3):

```
  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'));
```

- A completed **Display** procedure that will display the schedule of staff placements (content of **arrPlacements**).

  Example of the output of a schedule that will be displayed when the display procedure is called to display the content of **arrPlacements**. Library 3 can be closed on any day, as decided by the manager. For this schedule, the manager has decided to close Library 3 on Day 2, which is represented by 'XXXXX' in array **arrPlacements**.



```
           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
```

  The row and column headings are not part of the content of the two-dimensional array provided.

NOTE:

- You are NOT allowed to modify supplied data manually. Code must be written to manipulate the supplied data according to the requirements.
- The use of good programming techniques and modular design must be applied in the design and coding of your solution.
- NO marks will be awarded for the use of hardcoding to populate array **arrPlacements**.

3.1 **Combo box [3.1 - Select name]**

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

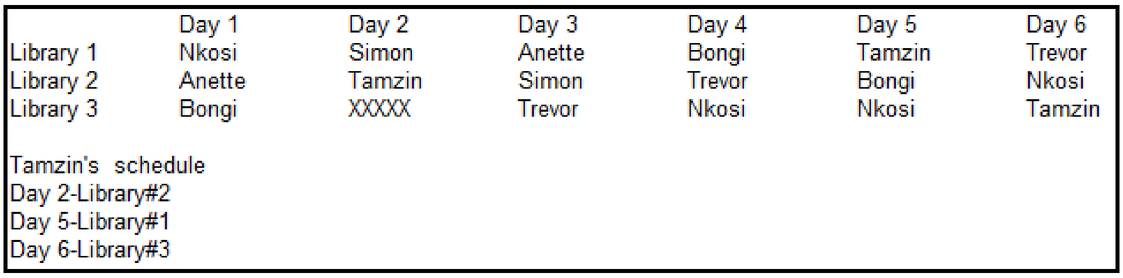
When a name is selected from the combo box **cmbStaff**, the following information must be displayed in the output component provided:

- 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:

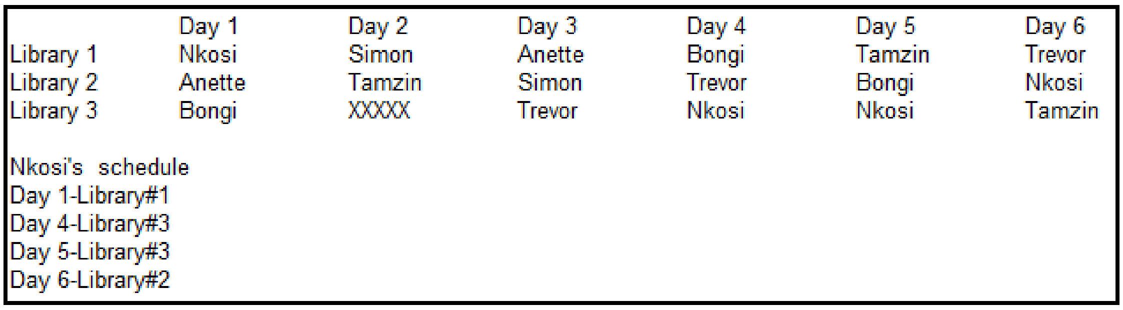
```
Day <day number>-Library#<library number>
```

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
```

Example of output if staff member Nkosi 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

Nkosi's  schedule
Day 1-Library#1
Day 4-Library#3
Day 5-Library#3
Day 6-Library#2
```

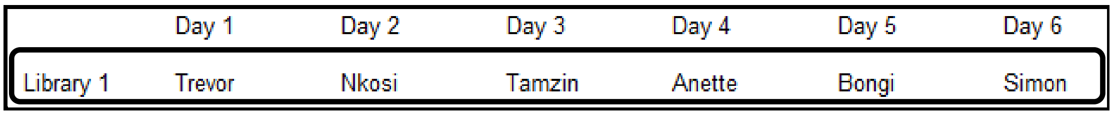
(10)

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**.

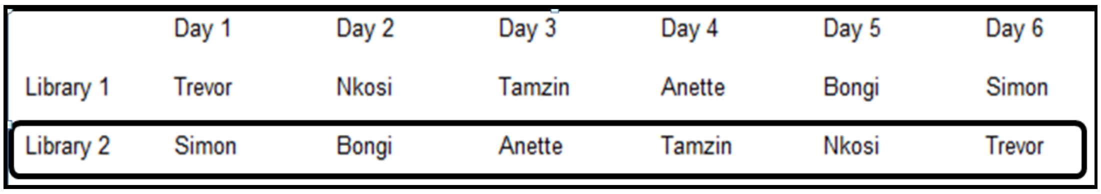
Write code to compile a new placement schedule for staff members by populating the **arrPlacements** array with the names of staff members as follows:

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)

- Ensure that your examination number has been entered as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

TOTAL SECTION C: 30
GRAND TOTAL: 150