



Province of the
EASTERN CAPE
EDUCATION

**NATIONAL
SENIOR CERTIFICATE**

GRADE 11

NOVEMBER 2018

INFORMATION TECHNOLOGY P1

MARKS: 150

TIME: 3 hours

This question paper consists of 13 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper consists of FOUR questions. Candidates must answer ALL four questions.
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. 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.
4. 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 this question paper.
5. Routines such as search, sort and selection must be developed from first principles. You may NOT use the built-in features of a programming language for any of these routines.
6. Save your work regularly.
7. The files that you need to complete this question paper have been given to you. The files are provided in the form of password-protected executable files.

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **18Work_Gr11**

Once extracted, the following list of files will be available in the folder **DataNov2018**:

Question 1:

Question1_u.pas
Question1_u.dfm
Question1_p.dpr
Question1_p.res

Question 2:

Question2_u.pas
Question2_u.dfm
Question2_p.dpr
Question2_p.res
Waiters.mdb

Question 3:

Question3_u.pas
Question3_u.dfm
Question3_p.dpr
Question3_p.res
stocklist.txt

Question 4:

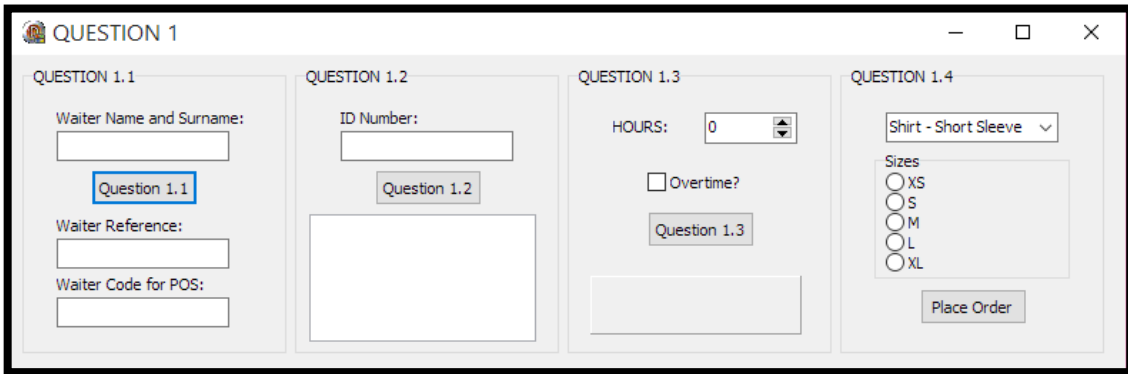
Question4_u.pas
Question4_u.dfm
Question4_p.dpr
Question4_p.res

QUESTION 1: GENERAL PROGRAMMING SKILLS

Most of the Grade Elevens are involved in some form of casual work. Some are cashiers, others are waiters or waitresses. When businesses employ casual workers they will need to complete some administrative duties for payslips, staff clothing, etc. Complete the program to help with some of the tasks.

Open the project, **Question1_p** in the **Question 1** folder.

The program has no functionality and the GUI looks as follows:



1.1 Button [Question 1.1]

Each waiter will receive a special reference which will be used on their payslips and official documents. For the Point Of Sales system, a code must be entered.

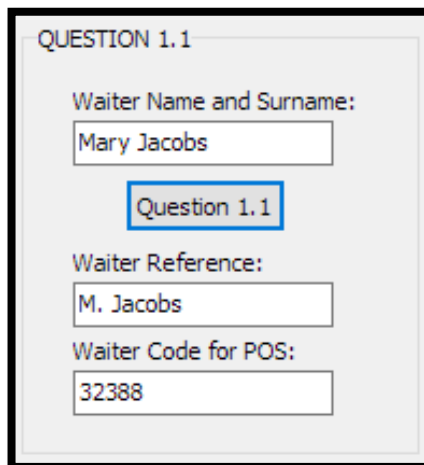
The waiter's name and surname must be entered in the text area and the reference and POS code will then be generated.

The waiter reference will consist of the first letter of the waiter's name, followed by a full stop and then the surname.

The POS code will be a randomly generated five-digit number between 10000 and 99999.

Display the reference and code in the respective text areas.

Example of output:



1.2 Button [Question 1.2]

The age of an employee determines whether they can be a full-time or part-time employee. If the employee is still school-going age, then they can only work one weekday and weekends. If they are younger than 16, they are too young to work. Any person that has finished school or older than 18, can work all weekdays and one day of the weekend.

Complete the code for this button which requires the ID number of the employee to be entered, to determine the days that the employee is allowed to work, based on their age.

The ID number must consist of 13 numerical characters. Calculate the age in years only. (Assume that no one is older than 100 years.)

Example of output:

(16)

1.3 Button [Question 1.3]

Write code to simulate a wage calculator. The number of hours (normal working hours) will be entered. If the overtime checkbox is selected, then an Inputbox component must appear, to ask how many overtime hours were worked.

Normal hours = R18.25 per hour

Overtime hours = one and a half times the normal hours rate

Calculate and display the total wage that the employee will receive, formatted to currency and two decimal places. Change the colour of the panel to SkyBlue.

Example of output if 5 hours were worked overtime:

(10)

1.4 Button [Question 1.4]

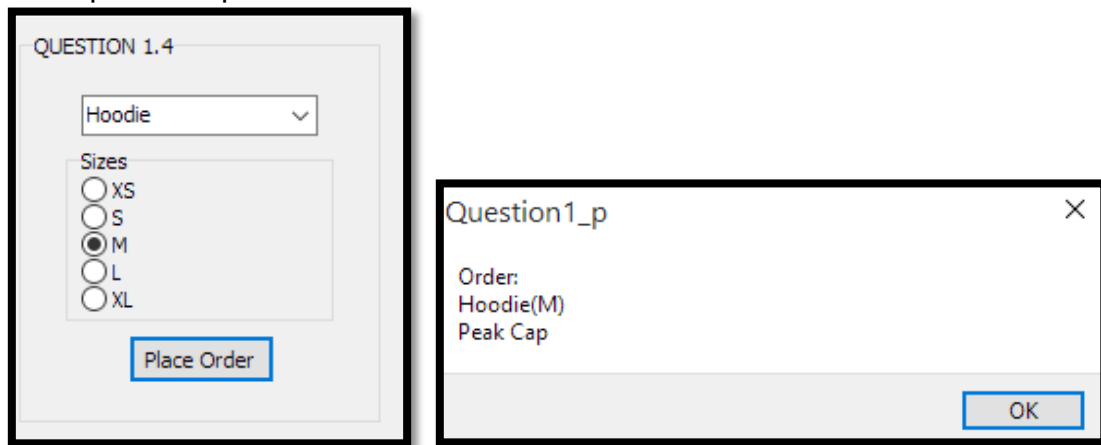
Staff clothing must be ordered on a regular basis. Each employee will receive a complementary peak cap with their order. They can only order one item at a time. The item of clothing can be a short sleeve shirt, long sleeve shirt or a hoodie.

The user must choose an item and select the size they would like to order.

Display the order, using a ShowMessage component, in the following layout:

Order:
<Item>(<Size>)
Peak Cap

Example of output:



(8)

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

[44]

QUESTION 2: DATABASE PROGRAMMING

A record of each shift is kept. The number of guests at a table and the amount they paid will be stored, as well as who their waiter was.

This information is stored in a database called **Waiters.mdb**. It contains a table called **tblTables** which has the following fields:

Field Name	Data Type
TableID	AutoNumber
WaiterID	Number
TableGuests	Number
TableAmountPaid	Number

The following is an example of the records in **tblTables**:

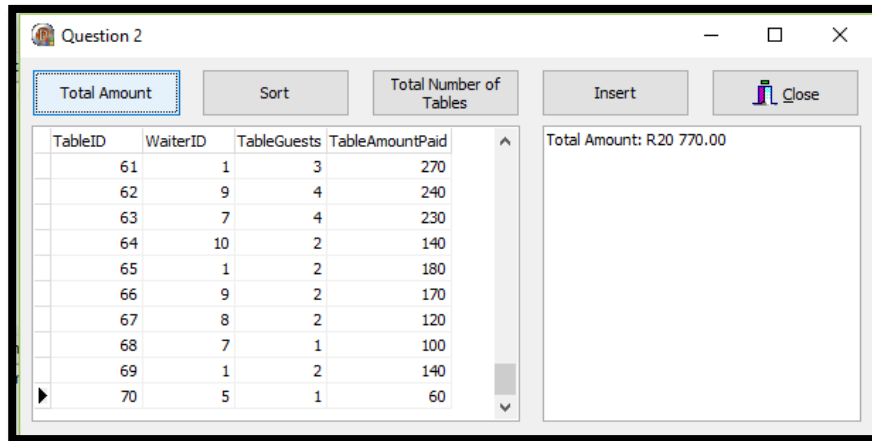
TableID	WaiterID	TableGuests	TableAmountPaid
1	5	7	R450.00
2	10	2	R250.00
3	3	6	R480.00
4	7	7	R680.00
5	9	7	R710.00
6	5	10	R620.00
7	2	7	R860.00
8	1	11	R880.00
9	3	12	R750.00
10	6	1	R90.00
11	7	6	R350.00
13	9	6	R420.00
14	11	5	R470.00
15	6	8	R430.00
16	2	8	R510.00
17	9	2	R250.00
18	14	9	R840.00
19	5	5	R350.00
20	11	8	R520.00
21	2	5	R520.00

DO NOT use filters when answering this question.

2.1 Button [Total Amount]

Calculate the total amount of money that was paid by the customers during this shift. Display the total amount in currency format and rounded off to two decimal places.

Example of output:

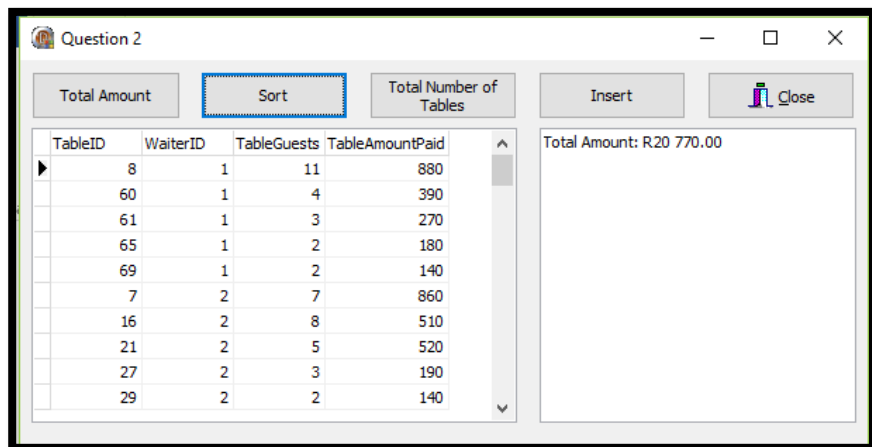


(6)

2.2 Button [Sort]

Sort the information according to the WaiterID, in ascending order.

Example of output:



(1)

2.3 Button [Total Number of Tables]

The manager would like to check how many tables each waiter served during the shift. An inputbox component must be used to enter the WaiterID of the respective waiter.

The waiter's waiterID must be displayed as shown in the example. A list of all the table numbers must be shown as well as the total number of tables that the waiter served.

Example of output:

TableID	WaiterID	TableGuests	TableAmountPaid
31	12	6	350
46	12	2	150
50	12	1	110
26	13	5	280
33	13	4	240
18	14	9	840
32	14	5	360
39	14	3	260
48	14	1	80
49	14	2	270

WaiterID: 2

7
16
21
27
29
52

Total Number of Tables: 6

(10)

2.4 Button [Insert]

A new group of 5 people have just been seated at a table (TableID 71). The waiter's waiterID that was allocated to serve them, is 10. Add the information to the table `tblTables`. Show a message that the record has been added.

Take note: `TableAmountPaid` does not have any information assigned to it.

(5)

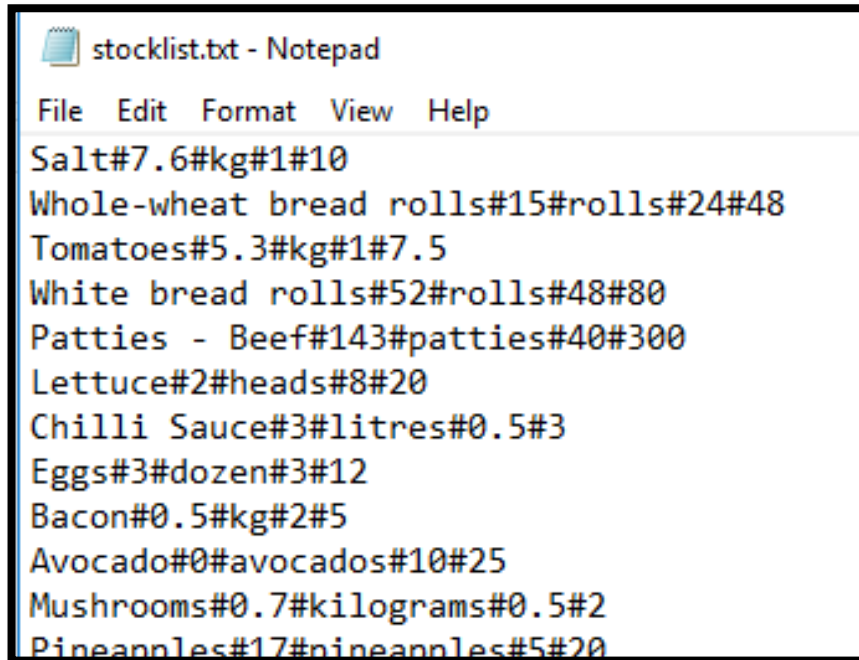
- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[22]

QUESTION 3: TEXT FILE MANIPULATION

A shopping list must be generated based on the current supplies. The stock list is stored in a text file, **stocklist.txt**. The layout of the text file is as follows:

item#current stock#unit of measure#minimum quantity#maximum quantity

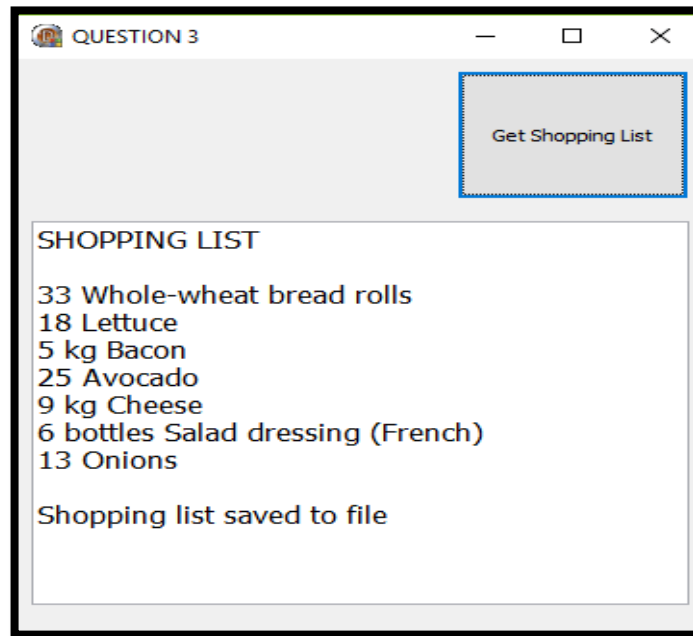


```
stocklist.txt - Notepad
File Edit Format View Help
Salt#7.6#kg#1#10
Whole-wheat bread rolls#15#rolls#24#48
Tomatoes#5.3#kg#1#7.5
White bread rolls#52#rolls#48#80
Patties - Beef#143#patties#40#300
Lettuce#2#heads#8#20
Chilli Sauce#3#litres#0.5#3
Eggs#3#dozen#3#12
Bacon#0.5#kg#2#5
Avocado#0#avocados#10#25
Mushrooms#0.7#kilograms#0.5#2
Pineapples#17#pineapples#5#20
```

Based on the current stock levels a shopping list must be generated. An item must be purchased if the current supply is less than the minimum amount required. All the amounts must be rounded to the nearest whole number as one can't buy a portion of an item.

If the unit is 'kg' or 'bottles' then the unit must be displayed, or else no unit must be displayed. The shopping list must be displayed, including a heading. This list must also be stored to a text file, **ShoppingList.txt**. Display a message indicating that the file was written.

Example of output:



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

[30]

QUESTION 4: ARRAYS AND SUBPROGRAMS

The menu is stored electronically. This makes the searching of a menu item much easier. When prices have to be increased or decreased it can be done all at once.

TAKE NOTE:

The menu items are stored in the array, **arritem**. Each item has either an F or D at the end to indicate whether it is a FOOD or a DRINK. The cost price, excluding VAT, of each item is stored in the constant array, **arrcost**.

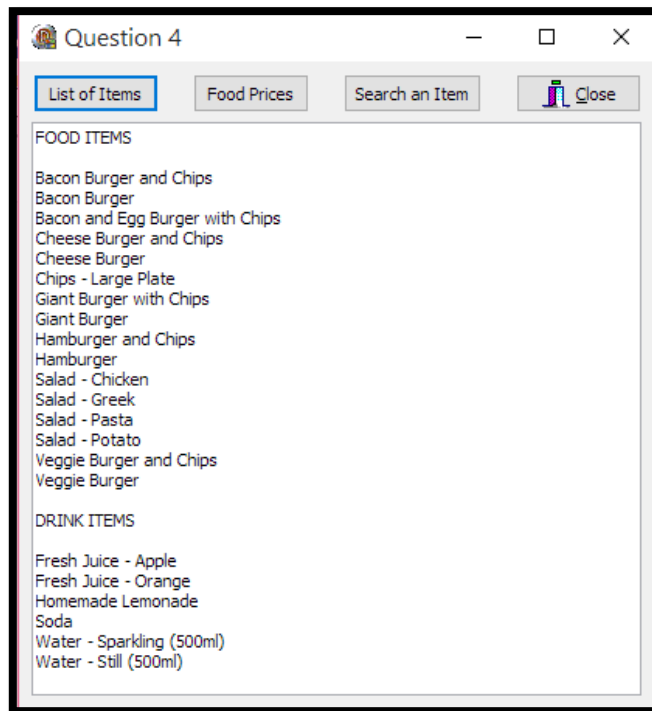
4.1 Write a **Sort** method which will sort **arritem** in alphabetical order. The correct price must still correspond with the correct item. (11)

4.2 **Button [List of Items]**

Separate the items into the two different categories, food and drink. Display an alphabetical list of all the food items and an alphabetical list of all the drinks.

TAKE NOTE: The food items and drink items will be used later in the question.

Example of output:



(20)

4.3 Button [Food Prices]

- Clear the output component.
- The information must be displayed in columns. Set a tab to 150.

When this button is selected, a list of all the food items and their menu price must be displayed.

The menu price will include 15% VAT as well as a mark-up of 45%. The prices that are displayed on the menu will be rounded up, so that only whole numbers are displayed on the menu.

Example of output:

FOOD ITEM	PRICE
Bacon Burger and Chips	R47
Bacon Burger	R41
Bacon and Egg Burger with Chips	R54
Cheese Burger and Chips	R40
Cheese Burger	R33
Chips - Large Plate	R9
Giant Burger with Chips	R60
Giant Burger	R55
Hamburger and Chips	R38
Hamburger	R32
Salad - Chicken	R28
Salad - Greek	R23
Salad - Pasta	R16
Salad - Potato	R20
Veggie Burger and Chips	R36
Veggie Burger	R30

(10)

- 4.4.1 Write a method called **Found** which will receive an item as a parameter and return whether the item was found (true) or not (false). (7)

4.4.2 Button [Search an Item]

The user must enter the item they are looking for in an Input box component. The method **Found** must be called and an appropriate message must be displayed whether the item is on the menu or not. The user must be allowed to enter part of the item name, for example Chips instead of Chips – Large Plate.

Example of output if **Chips** is entered:



(6)

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

[54]

TOTAL: 150