Centre Number	Candidate Number	Candidate Name

NAMIBIA SENIOR SECONDARY CERTIFICATE

COMPUTER SCIENCE ADVANCED SUBSIDIARY LEVEL 8231/2

PAPER 2 2 hours 30 minutes

Marks 75 **2022**

Additional Material: Non-programmable calculator

INSTRUCTIONS AND INFORMATION TO CANDIDATES

- Candidates answer on the Question Paper in the spaces provided.
- Write your Centre Number, Candidate Number and Name in the spaces at the top of this page.
- Write in dark blue or black pen.
- You may use a soft pencil for any rough work, diagrams or graphs.
- Do not use correction fluid.
- Do not write in the margin *For Examiner's Use*.

- Answer **all** questions.
- Non-programmable calculators may be used.

- The number of marks is given in brackets [ ] at the end of each question or part question.
- The businesses mentioned in this question paper are entirely fictitious.
- No marks will be awarded for using brand names of software packages or hardware.

For Examiner's Use

Marker	
Checker	

This document consists of **18** printed pages and **2** blank pages.

Republic of Namibia

MINISTRY OF EDUCATION, ARTS AND CULTURE

1 The average mark for each pupil in a class has been calculated. Willem writes an algorithm in pseudocode to count and output the number of pupils who have an average mark of 70 or more.

```
01 SET List ← [85.3, 63.6, 71.9, 92.5]
02 Amount ← 0
03 FOR Counter FROM 0 TO 2 DO
04 IF List[Counter] >= 70 THEN
05    Amount ← 1 + 1
06 ENDIF
07 ENDFOR
08 PRINT Amount
```

When Willem tests the program, it outputs the wrong number of A passes.

(a) (i) State the output from the pseudocode above.

..... [1]

(ii) There is an error, state the name of this type of error.

..... [1]

(iii) Identify the line number of the error. Write the correct pseudocode statement.

Line .....

Correction: .....

.....

..... [2]

(b) Indentation has been used to make the pseudocode above more readable or understandable.
Name **two** other techniques that can be used to make algorithms more readable or understandable.
Provide an example using the pseudocode above to illustrate each technique.

1. Name .....

Example .....

.....

.....

2. Name .....

Example .....

.....

.....

[4]

2 Translators are used to convert high-level languages into machine code.

(a) Identify each type of translator. The answers must be different.

	Type of Translator
This translator program reports errors at the end of translation.	.....
This translator needs to be present in memory each time the program is executed.	.....

[2]

(b) Two computer programs that calculate a discount in a retail shop are shown in **Fig. 1**. One is written in a low-level language and the other is written in a high-level language.

Explain why computer programs are most commonly written in high-level languages instead of low-level languages?

Fig. 1

.....

.....

.....

.....

[2]

(c) The low-level program shown in **Fig. 1** is written in machine code.

Suggest **two** reasons why it would have been better for the programmer to use assembly language instead of machine code.

1 .....

.....

2 .....

.....

[2]

3 **(a)** Complete the truth table for the XNOR logic gate:

S	T	U
0	0	
0	1	
1	0	
1	1	

[1]

(b) Complete the truth table for the logic circuit shown in **Fig. 2** below:

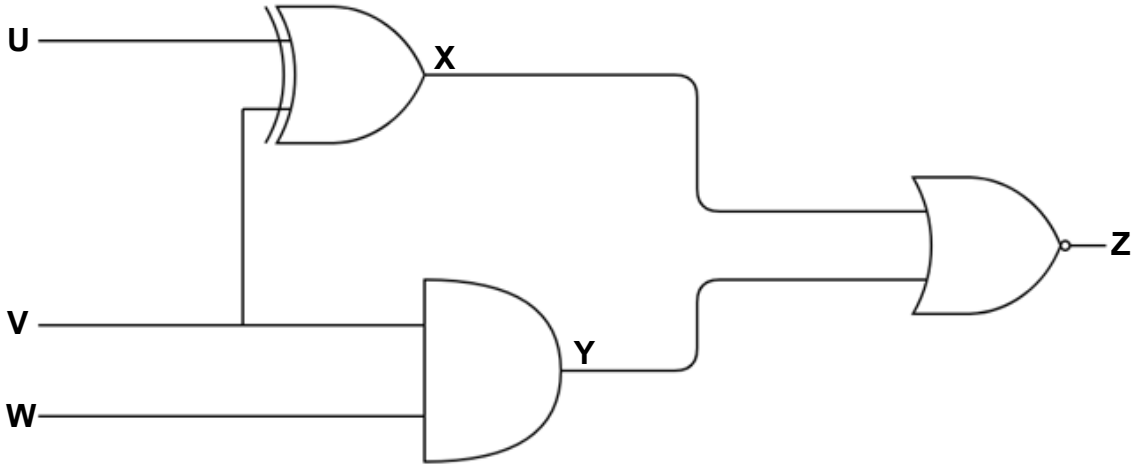


Fig. 2

U	V	W	X	Y	Z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

[4]

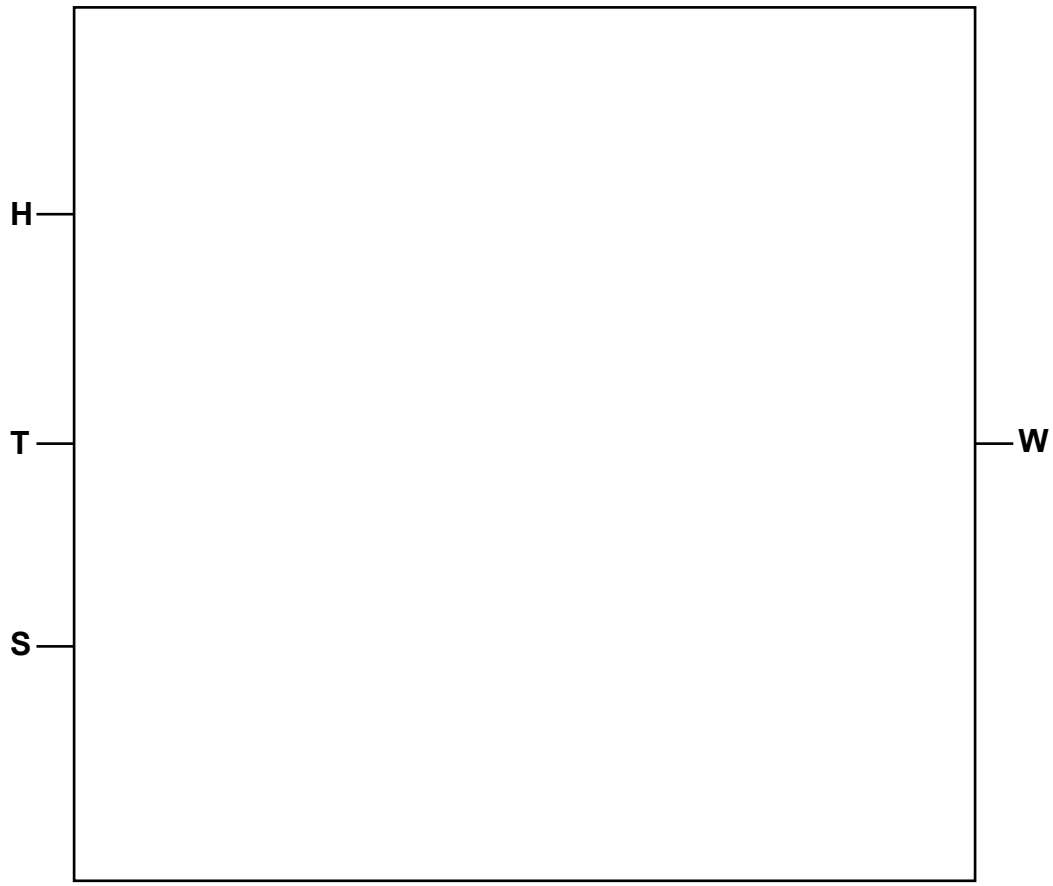
For Examiner's Use

(c) A logic circuit is being developed for an automatic watering system in an orchard. The automatic watering system has two sensors.

Sensor **H** attached to contacts buried in the soil and turns ON when these contacts are too dry.

- Sensor **T** is attached to a timer that is ON when it is night-time.
- The output relay **W**, closes and starts the pump only when these two conditions are met.
- A switch **S** can be switched ON or OFF by the gardener during the day and the water pump works regardless.

Draw the logic circuit for this automatic watering system.



[4]

BLANK PAGE

Question 4 is printed on page 8.

4 The algorithm in Fig. 3 is a sorting algorithm.

- Array indexing starts at 0.
- Line numbers are included but are not part of the algorithm.

```
01 SortArray ← [5, 2, 8]
02 Sorted ← FALSE
03 WHILE NOT Sorted
04     Sorted ← TRUE
05     Count ← 0
06     WHILE Count < 2
07     IF SortArray[Count+1] < SortArray[Count] THEN
08     Temp ← SortArray[Count]
09     SortArray[Count] ← SortArray[Count+1]
10     SortArray[Count+1] ← Temp
11     Sorted ← FALSE
12     ENDIF
13     Count ← Count + 1
14     ENDWHILE
15 ENDWHILE
```

Fig. 3

(a) State the data type of the variable `sorted` in the algorithm shown in Fig. 3.

..... [1]

(b) The identifier `sorted` is used in the algorithm shown in **Fig. 3**. Explain why this is a better choice than using the identifier **S**.

.....

.....

.....

..... [1]

(c) Complete the trace table for the algorithm shown in **Fig. 3**. Some values have already been entered.

SortArray			Sorted	Count	Temp
[0]	[1]	[2]			
5	2	8	false		

[6]

For Examiner's Use

5 Write an algorithm using either pseudo-code or a flowchart that allows an e-bicycle (electronic bicycle) company to calculate how much the e-bicycle rental fare should be.

The algorithm should:

- prompt the user to enter the journey distance in kilometers
  - o check that the distance entered must be greater than zero
  - o the user must be made to re-enter the distance until the distance entered is greater than zero
- prompt the user to enter the number of e-bicycles (no validation is required)
- calculate the e-bicycle rental fare by
  - o charging N$10 for every e-bicycle rented regardless of the distance
  - o charging a further N$2.50 for every kilometre regardless of how many e-bicycles rented out
- output the final e-bicycle rental fare.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

For Examiner's Use

6 A software engineer is planning to use a structure chart when developing a new system for a small guest house.

Describe a structure chart.

.....

.....

.....

.....

.....

..... [2]

For Examiner's Use

7 A queue Abstract Data Type (ADT) has these associated operations:

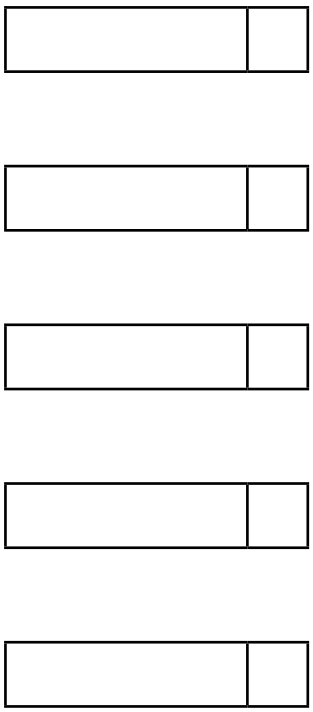
- create queue
- add item to queue
- remove item from queue

The queue ADT is to be implemented as a linked list of nodes. Each node consists of data and a pointer to the next node.

(a) The following operations are carried out:

```
CreateQueue

AddName("Alex")

AddName("Iyaloo")

AddName("Betty")

AddName("Ethan")

RemoveName

AddName("Tonata")

RemoveName
```

Add appropriate labels to the diagram to show the final state of the queue. Use the space on the left as a workspace. Show your final answer in the node shapes below: Include the head pointer and the tail pointer for the queue.



[3]

(b) Using pseudocode, a record type, Node, is declared as follows:

```
TYPE Node

          DECLARE Name    : STRING

          DECLARE Pointer : INTEGER

ENDTYPE
```

The statement

`DECLARE Queue : ARRAY[1:6] OF Node` reserves space for 6 nodes in array Queue.

The `CreateQueue` operation links all nodes and initialises the three pointers that need to be used: `HeadPointer`, `TailPointer` and `FreePointer`.

Complete the diagram to show the value of all pointers after `CreateQueue` has been executed.

HeadPointer

TailPointer

FreePointer

	Queue	
	Name	**Pointer**
[1]		
[2]		
[3]		
[4]		
[5]		
[6]		

[4]

8 A road operator transports passengers to some towns in Namibia.

The road operator provides an app for passengers to purchase tickets. An array called **`RoadNetwork`** is used to store the names of the towns on the network. Passengers must enter a departure station into the app.

The current contents of the array **`RoadNetwork`** are shown:

1	Grootfontein
2	Katima Mulilo
3	Okahandja
4	Ondangwa
5	Oshakati
6	Otjiwarongo
7	Rundu
8	Swakopmund
9	Walvis Bay
10	Windhoek

A binary search is used to check if the entered departure station exists in **`RoadNetwork`**.

(a) (i) A binary search algorithm is used to find a specific value in the array **`RoadNetwork`** above. Explain why an array needs to be sorted before a binary search algorithm can be used.

.....

.....

.....

..... [2]

(ii) Explain how a binary search will find the value Rundu in **`RoadNetwork`**.

.....

.....

.....

.....

.....

.....

..... [4]

(b) Complete this procedure to carry out a binary search on the **RoadNetwork** shown above.

```
PROCEDURE BinarySearch   (RoadNetwork,  SearchValue)
     DECLARE Midpoint,   First, Last  :  INTEGER
     DECLARE Found  :  BOOLEAN

     First ← 1

     Last  ← ARRAYLENGTH (.................................)

     Found  ←  FALSE
     WHILE  (First <=  Last)  AND NOT  (FOUND)

          Midpoint ← ..................................... .
               IF RoadNetwork [Midpoint]  =  SearchValue
               THEN

                    Found ← TRUE
               ELSE
                    IF SearchValue < RoadNetwork  [Midpoint]
                    THEN

                              Last ← ................
                    ELSE

                              First ← ...................
                    ENDIF
               ENDIF
     ENDWHILE
ENDPROCEDURE
```

[4]

(c) A user enters the departure station "**Mariental**".

Explain how a serial search would check if the departure town exists in the array `RoadNetwork`.

.....

.....

.....

.....

.....

.....

.....

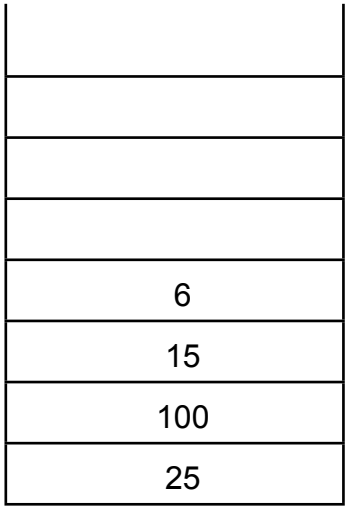
..... [4]

For Examiner's Use

9 A computer program stores a rugby team's scores on a stack named `ScoreData`. The stack has two sub-programs to add and remove data items from the stack. The stack is implemented as a 1D array, `ScoreArray`.

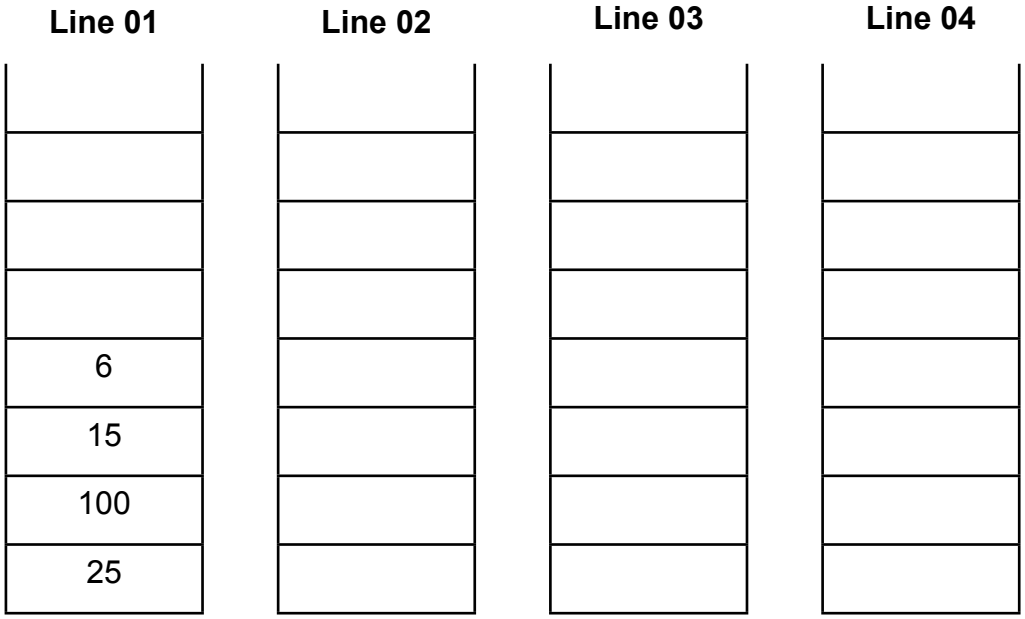
Sub-program	Description
push ( )	The parameter is added to the top of the stack
push ( )	The element at the top of the stack is removed

The current contents of `ScoreData` are shown:



(a) Show the contents of the stack `ScoreData` after each line of the following lines of code are run

```
01 push (30)
02 pop ()
03 push (85)
04 push (42)
```



[4]

(b) The main program asks a user to push or pop an item from the stack. If the user enters 'push', the data item is added to the stack. If the user chooses 'pop', the next item is removed from the stack, multiplied by 5 and output.

The main program is shown:

```
01 Answer = INPUT ("Is your choice push or pop?")
02 If Answer = "push"
03 THEN
04   push  (INPUT  ("Enter data item"))
05 Else
06   PRINT (pop    ()  * 5)
07 EndIf
```

(i) Before the sub-programs, push() and pop(), can add or remove items from the stack, a selection statement is used to decide if each action is possible.

Describe the decision that needs to be made in each sub-program and how this impacts the next process.

push()

.....

.....

.....

.....

pop()

.....

.....

.....

.....

[4]

(ii) The algorithm does not work when the user enters "PUSH" or "Push". The algorithm needs to be changed in order to accept these inputs. Identify the line number to be changed and state the change that should be made.

Line number .....

Change .....

.....

[2]

(c) The stack is implemented as a 1D array, `ScoreArray`.

Describe how a 1D array can be set up and used to push and pop items as a stack.

.....

.....

.....

.....

.....

..... [3]

s

BLANK PAGE

BLANK PAGE