6. A wildlife society stores data on mammals around the world.

This mammal data is held in a database. Whenever new readings become available they are stored in a text file. The text file is read into a program to update the mammal data. The data is then analysed to identify any mammals at risk of extinction.

A sample of the data in the database is shown below.

Table name: MammalData			
mammal	**numberRemaining**	**readingNumber**	**readingDate**
Tiger	3890	2	12/05/2024
Blue Whale	5000	2	12/05/2024
Polar Bear	31000	1	12/05/2024
Rhinoceros	5600	1	12/05/2024
Orangutan	105000	1	12/05/2024
Blue Whale	7632	1	05/05/2024
Tiger	4432	1	05/05/2024
African Wild Dog	3000	1	05/05/2024

Part of the top-level design of the program is shown.

1. declare data structures used in the program
2. read mammal data from database and store in array of records
3. update array of records using data from the readings file
4. add new readings to database
5. analyse the data
6. …

Some of the code used to implement step 1 of the design is shown below.

```
RECORD Animal IS { STRING mammal, INTEGER numberRemaining,
INTEGER readingNumber, STRING readingDate }

DECLARE animalList AS ARRAY OF Animal INITIALLY [NULL] * 1000
```

(a) Using pseudocode, refine step 2 of the design to show the steps needed to read the data from the database server and store it in the array of records called `animalList`. **3**

(b) The number of each mammal is recorded regularly. Whenever new data about a mammal becomes available, the name of the mammal, number remaining and reading number are stored in a text file `readings.txt`.

The contents of the latest `readings.txt` are shown below:

24/05/2024
Polar Bear,29140,2
Giant Panda,1864,1
Tiger,3745,3
African Wild Dog,3012,2

Step 3 of the top-level design processes the readings file.

The refinement for this step is shown below.

3.1 declare temporary array `temp` as array of `Animal`
3.2 read data from `readings.txt` file into `temp`
3.3 sort temp into descending order of `readingNumber`
3.4 move existing records in `animalList` array to create empty rows required
3.5 copy contents of `temp` into empty rows of `animalList` array

Once step 3.2 has been completed, the contents of the array `temp` will be:

mammal	numberRemaining	readingNumber	readingDate
Polar Bear	29140	2	24/05/2024
Giant Panda	1864	1	24/05/2024
Tiger	3745	3	24/05/2024
African Wild Dog	3012	2	24/05/2024

(i) The incomplete design of the insertion sort used to arrange the contents of the `temp` array into descending order of `readingNumber` at step 3.3 is shown below.

```
3.3.1   procedure insertionSort(list)                IN/OUT: list
3.3.2       set value = 0
3.3.3       set index = 0
3.3.4       loop from n = 1 to length(list)–1
...         ...
...         ...
...         ...
...         ...
...         end loop
...     end procedure
```

Write the pseudocode needed to complete the insertion sort algorithm. **3**

(ii) At step 3.4, the data stored in the array `temp` is copied into the `animalList` array.

The table below shows the contents of the `animalList` array **before** step 3.4 is executed.

mammal	numberRemaining	readingNumber	readingDate
Tiger	3890	2	12/05/2024
Blue Whale	5000	2	12/05/2024
Polar Bear	31000	1	12/05/2024
Rhinoceros	5600	1	12/05/2024
Orangutan	105000	1	12/05/2024
Blue Whale	7632	1	05/05/2024
Tiger	4432	1	05/05/2024
African Wild Dog	3000	1	05/05/2024

Step 3.4 requires the contents of the `animalList` array from index 0 to be moved to accommodate the data currently stored in the `temp` array.

The table below shows the contents of the `animalList` array **after** step 3.4 is executed.

mammal	numberRemaining	readingNumber	readingDate
Tiger	3890	2	12/05/2024
Blue Whale	5000	2	12/05/2024
Polar Bear	31000	1	12/05/2024
Rhinoceros	5600	1	12/05/2024
Orangutan	105000	1	12/05/2024
Blue Whale	7632	1	05/05/2024
Tiger	4432	1	05/05/2024
African Wild Dog	3000	1	05/05/2024

Using pseudocode, refine step 3.4 of the top-level design to show the steps needed to move the rows of data within the `animalList` array. **4**

6. (b) (continued) **MARKS**

Once the update has taken place, the contents of the `animalList` array are:

mammal	numberRemaining	readingNumber	readingDate
Tiger	3745	3	24/05/2024
Polar Bear	29140	2	24/05/2024
African Wild Dog	3012	2	24/05/2024
Giant Panda	1864	1	24/05/2024
Tiger	3890	2	12/05/2024
Blue Whale	5000	2	12/05/2024
Polar Bear	31000	1	12/05/2024
Rhinoceros	5600	1	12/05/2024
Orangutan	105000	1	12/05/2024
Blue Whale	7632	1	05/05/2024
Tiger	4432	1	05/05/2024
African Wild Dog	3000	1	05/05/2024

(iii) At step 5 of the top-level design, the program will analyse the contents of the updated `animalList` array to identify any mammals at risk of extinction.

A mammal at risk of extinction is one where the number remaining in the most recent update represents a reduction of 5% or more when compared with the previous reading.

Using pseudocode, write an algorithm to display the name and percentage reduction of any mammal at risk of extinction. **3**