

- 3 The file, `HASHEDDATA.TXT`, holds details of the names and telephone numbers of 250 people.

There are a total of 500 lines in the file, and a number of these lines are empty of name and telephone number.

An index is stored for each line of the file.

The format of the data in the file is:

`<Index>,<PersonName>,<TelephoneNumber>`

The first 10 lines from the file are shown as follows:

```
0,,
1,,
2,,
3,Boon Keng V.,07492 546415
4,,
5,,
6,Ahmad Yusof,07439 778665
7,Durno Peter,07662 863518
8,Batisah Wong,07362 156265
9,,
```

The values in the file are separated by the comma character.

A record structure is used to store a name and telephone number. A data structure of 500 records is needed to store all the names and telephone numbers. Each line in the file is written to a corresponding position in the data structure.

The records with index six to eight from the data structure are:

Index	PersonName	TelephoneNumber
6	Ahmad Yusof	07439 778665
7	Durno Peter	07662 863518
8	Batisah Wong	07362 156265

### Task 3.1

Use program code to create a:

- record structure to hold the name and telephone number for one person
- data structure, using this record structure to store 500 records.

### Evidence 7

Your program code.

[6]



**Task 3.2**

Write program code to:

- read the lines from the file
- extract the <Index>, <PersonName> and <TelephoneNumber> values
- store these values in the data structure.

Create a procedure called `DisplayValues` that will loop through the data structure and display the index, name and telephone number for every record where the name is present.

Ensure your procedure uses headings to identify the data displayed.

**Evidence 8**

Your program code.

[13]

**Evidence 9**

A screenshot showing the output.

[1]

A hashing function was used to create the file. The same hashing function can be used to search the data structure for a particular name.

The hashing function generates a hash. This is calculated as follows:

```
Get SearchName
Set HashTotal to 0
FOR each Character in SearchName
    Get the ASCII code for Character
    Multiply the ASCII code by the position of Character in SearchName
    Add the result to the HashTotal
Calculate Hash as HashTotal MOD 500
RETURN Hash
```

**Task 3.3**

Add the program code for the hashing function. Use the following specification:

```
FUNCTION GenerateHash(SearchName : STRING) : INTEGER
```

The function has a single parameter `SearchName` and returns an integer value.

Write additional code for your program to allow you to test the implementation of this function.

The following test data will assist you.

“Tait Davinder” should return a hash of 87

“Anandan Yeo” should return a hash of 156



**Evidence 10**

Your program code.

[8]

**Evidence 11**

A screenshot (or screenshots) of your program to show the results of the hash calculation for both the given test data values.

[2]

The hash calculated from the `SearchName` can be used to find a corresponding record in the data structure.

If the `SearchName` is not found in the record given by the hash **and** the record is not empty:

- compare `SearchName` with the next record
- until the `SearchName` is found or an empty record is found.

If an empty record is found then the program will report that the name is "NOT FOUND".

If the record is found, the program will output the index, name and telephone number.

**Task 3.4**

Add program code to implement the search as described.

**Evidence 12**

Your program code.

[7]

**Evidence 13**

A screenshot (or screenshots) of your program showing the result of the following searches:

Search 1: Charlie Love

Search 2: Chin Tan

Search 3: John Barrowman

[3]

