

Term	Definition
Data structures	A data structure is a specific way of organising data within memory so it can be processed efficiently.
Arrays	An array is a data structure that contains groups of elements. Normally these elements are all of the same data type.
Records	Records are composed of fields, each of which contains one item of information about the same subject, e.g. one person. A set of records constitutes a file.

Data types

Data types include:

- integer: whole numbers, e.g. 10, 23
- real: numbers with decimal or fractional parts, e.g. 3.142, 99.9
- character: individual characters, e.g. M, F
- string: series of characters, e.g. Fred, Smith
- Boolean: e.g. TRUE or FALSE

Arrays normally include elements of the same data type.

Records include a mixture of data types.

Arrays

Arrays are data structures comprising a collection of elements (values or variables). Each element is referenced by an index.

One-dimensional array example

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
37	11	42	6	26	56	4	76

- 8 elements
- The index always starts at position [0].
- Each element can be accessed using its index.
- The element at index [4] is 26.

Two-dimensional array example:

Index	i	j
0	Fred	Smith
1	John	Davies
2	Sue	Evans
3	Pamela	Wilson

- 4 elements
- The index always starts at 0.
- Each element can be accessed using its index.
- The element at index [1,2] is Davies.
- The element at index [3,1] is Pamela.

Traversing

Print the contents of the array above:

```

1 for i = 0 to 3
2   for j = 1 to 2
3     output myArray [i, j]
4   next j
5 next i
    
```

Insertion

Add data to an element at a given index:

```

1 myArray [4,1] = "Sian"
2 myArray [4,2] = "Jones"
    
```

This would store "Sian Jones" at index 4 of the array.

Deletion

Deleting data from an element at a given index:

```

1 myArray [1,1] = ""
2 myArray [1,2] = ""
    
```

This would leave the memory at index 6 blank.

Using an array to search data

Arrays can be used to search for a data item from a list. The data from the file can be read into an array and then the data can be searched using the index until the required item is found.

```
Declare linearSearch(dataList, searchItem)
```

```
position = 0
found = false
```

```
while position < len(dataList) and found = false
  if dataList[position] = searchItem then
    found = true
  else
    position = position + 1
  end if
end while
```

```
testList = [1,3,21,45,57,17,34,65]
linearSearch(testList, 45)
linearSearch(testList, 20)
```

Records

A record is a group of related data held within the same structure. A record is a grouping of fields within a table that reference one particular object.

An example of the structure of a pupil record is:

Fieldname	Data type	Length	Example
PupilID	String	5	P0001
Firstname	String	20	Fred
Surname	String	20	Smith
House No	Integer		10
Postcode	String	8	CF62 6YX
Gender	Character	1	M
Attendance	Real		94.3

A record comprises multiple data types. Each record will be of the same structure (same fields).

Each record will have a key field which will be a unique identifier for that record. In general, searches will be carried out on the key field.

Reading data records from a file

When data records are read from a file, they can be read into a two-dimensional array to allow them to be manipulated (searched or sorted).

Writing data records to a file

Data can be written to a file via the computer user interface.

```
pupilID= pupilID.txt
firstname = firstname.txt
surname = surname.txt
form = form.txt
DoB = DoB.txt

file.open ("PupilDetails.txt")

file.write (pupilID + firstname +
surname + form+ DoB)

file.close ()

messagebox ("Confirmation", "Pupil
details successfully saved")
```

Fixed and variable records

There are two main types of records; fixed length and variable length records.

Fixed length records

Each record in a file will be the same length with the size of each field remaining consistent.

The Pupil records will all have pupilIDs that are 5 characters in length with the same format P9999.

The firstname field has been set at 20 characters. If a pupil's name is shorter than 20 characters a series of blanks (padding) will be stored in that field, e.g. if Fred is stored in a record, there will be four characters and 16 blanks.

Variable length records

Each record in a file can be of a different length which will be dependent on the length of the data to be stored in that record. Each field is separated from other fields by special characters known as delimiters.

Advantages and disadvantages of fixed and variable length records

Variable length records save storage space but are more difficult to manipulate through code.

Fixed length records can take up considerable space which is used to store padding. However, as the length of each record is known, they are easier to manipulate when programming.