8 A program is being developed to implement a game for up to six players.

During the game, each player assembles a team of characters. At the start of the game there are 45 characters available.

Each character has four attributes, as follows:

Attribute	Examples	Comment
Player	0, 1, 3	The player the character is assigned to.
Role	Builder, Teacher, Doctor	The job that the character will perform in the game.
Name	Bill, Lee, Farah, Mo	The name of the character. Several characters may perform the same role, but they will each have a unique name.
Level	14, 23, 76	The skill level of the character. An integer in the range 0 to 100 inclusive.

The programmer has defined a record type to define each character.

The record type definition is shown in pseudocode as follows:

```
TYPE CharacterType
   DECLARE Player : INTEGER
   DECLARE Role : STRING
   DECLARE Name : STRING
   DECLARE Level : INTEGER
ENDTYPE
```

The `Player` field indicates the player to which the character is assigned (1 to 6). The field value is 0 if the character is **not** assigned to any player.

The programmer has defined a global array to store the character data as follows:

```
DECLARE Character : ARRAY[1:45] OF CharacterType
```

At the start of the game all record fields are initialised, and all `Player` field values are set to 0

The programmer has defined a program module as follows:

Module	Description

`Assign()`	• called with two parameters:
	  ○ an integer representing a player
	  ○ a string representing a character role
	• search the `Character` array for an unassigned character with the required role
	• If found, assign the character to the given player and output a confirmation message, for example:
	`"Bill the Builder has been assigned to player 3"`
	• If no unassigned character with the required role is found, output a suitable message.

(a) Write pseudocode for module `Assign()`.

[7]

(b) A new module will store the contents of the `Character` array in a text file.

The module is defined as follows:

Module	Description
`Save()`	• form a string from each record with fields separated by the character `'^'` • write each string to a separate line of the new file named `SaveFile.txt`

Complete the pseudocode for module `Save()`.

```
PROCEDURE Save()
```

```
ENDPROCEDURE
```

[7]

(c) The program is changed and the record type definition is modified as follows:

```
TYPE CharacterType
   DECLARE Player : INTEGER
   DECLARE Role : STRING
   DECLARE Name : STRING
   DECLARE Level : INTEGER
   DECLARE Status : BOOLEAN
ENDTYPE
```

Describe how the additional Boolean field may be stored with the rest of the fields on one line of a text file.

[2]

(d) The save operation is to be extended so that multiple files may be saved as the game progresses. This will allow the user to restore the game from any saved position. The filename must reflect the sequence in which the files are saved.

Describe a method that would allow multiple files to be saved **and** give an example of **two** consecutive filenames.

Method

Example

[2]