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.
Skill level	14, 23, 76	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 SkillLevel : INTEGER
ENDTYPE
```

The `Player` field indicates the player to which the character is assigned (1 to 6). This 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` fields are set to 0

The programmer has defined a program module as follows:

Module	Description

`Count()`	• called with two parameters:
	  ○ an integer representing a player
	  ○ a string representing a character role
	• searches the `Character` array for characters with the given role that are assigned to the given player
	• counts the number of assigned characters and sums their total skill level
	• outputs the result of the search if characters with the given role are found, for example:
	`"Player 3 has 4 characters with the role of Teacher and the total skill level is 65"`
	• if no characters with the given role are found, outputs:
	`"No characters with that role are assigned to this player"`

(a) Complete the pseudocode for module `Count()`.

```
PROCEDURE Count(ThisPlayer : INTEGER, ThisRole : STRING)
```

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

```
ENDPROCEDURE
```

[7]

(b) The `Character` array data has been saved in the text file `SaveFile.txt`
Each line of the file contains one element of the array (one record).

New modules are defined:

Module	Description

`Extract()` (already written)	• called with two parameters:
	  ○ a string representing a complete line from the text file
	  ○ an integer representing a field number (see structure below)
	• returns a string representing the required field

`Restore()`	• opens the text file `SaveFile.txt` • reads lines from the file and assigns values to each record in the `Character` array using data from each line of the file

As a reminder, the record structure is repeated here:

```
TYPE CharacterType
   DECLARE Player : INTEGER       //Field number 1
   DECLARE Role : STRING          //Field number 2
   DECLARE Name : STRING          //Field number 3
   DECLARE SkillLevel : INTEGER   //Field number 4
ENDTYPE
```

Write pseudocode for module `Restore()`.

You must use the module `Extract()`.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [7]

(c) The game can last for several days and users often find that they have to close and rerun the game program many times in order to complete it.

Describe the benefit of using the file `SaveFile.txt` as described in part **(b)**.

....

....

....

.... [2]