

HINT

Here's how to write comments in Python, Java and C#:

- **Python:** start each line of a comment with the # symbol
- **Java:** start single line comment with // or start a multi-line comment with /* and end it with */
- **C#:** Same as for Java.



▲ Programmers often work in teams; it is vital that any coding they share is clear and error free

Table 2.5 lists the techniques you should use to make your programs easy to read and understand.

TECHNIQUE	DESCRIPTION
Comments	Comments should be used to explain what each part of the program does.
Descriptive names	Using descriptive identifiers for variables, constants and subprograms helps to make their purpose clear.
Indentation	Indentation makes it easier to see where each block of code starts and finishes. Getting the indentation wrong in Python will result in the program not running or not producing the expected outcomes.
White space	Adding blank lines between different blocks of code makes them stand out.

▲ Table 2.5 Techniques for program clarity

SKILLS

CRITICAL THINKING,
PROBLEM SOLVING,
CREATIVITY

ACTIVITY 11

REWRITING AND IMPLEMENTING ALGORITHMS

- 1 Program the following algorithm in a high-level language and make it readable by adding comments and indentation.

```

SET x TO 10
WHILE x >= 0 DO
    IF x > 0
        SEND x TO DISPLAY
    ELSE
        SEND 'Blast Off' TO DISPLAY
    END IF
    SET x TO x -1
END WHILE

```



▲ Calculators have a number of useful functions

SKILLS ➔ REASONING

SKILLS ➔ CRITICAL THINKING

SKILLS ➔ PROBLEM SOLVING

KEY POINT

Readability of code is important. You should always make your code easy to read.

CHECKPOINT

Strengthen

S1 Why it is important to make your code easy to read?

S2 Outline the four techniques that a programmer should use to make code easy to read.

Challenge

C1 Revisit the programs you have already written. Do you still understand what they do and how they work? If not, try to improve their readability.

How confident do you feel about your answers to these questions? If you're not sure you answered them well, look again at Table 2.5.

SUMMARY

- Using comments, descriptive names, indentation and white space makes code easier to read.
- Producing readable code makes it easier to understand what a program does and how it does it.

7 STRINGS

A string is a data type that is used to represent a sequence of characters, such as numbers, text, spaces and punctuation. Strings must be enclosed in quotation marks to distinguish them from variable names. A string could be a user name, a whole sentence or paragraph of data.

LEARNING OBJECTIVES

- Describe what a string is and explain what strings are used for
- Use iteration to traverse a string
- Concatenate and split strings

SUBJECT VOCABULARY

string a sequence of characters. They can be letters, numbers, symbols, punctuation marks or spaces

A character is one of the four basic data types. It can be a single letter, a symbol, a number or even a space. A sequence of characters is called a **string**. Although strings can contain different sorts of characters, including numbers, they are all treated as if they were text.

When a computer executes a program, it needs a way of telling the difference between a string and an instruction. In most programming languages, this is achieved by enclosing strings in quotation marks (e.g. 'johnsmith@mail.com', '10/04/15' or '123'). Both single ' ' and double " " quotation marks are acceptable, as long as they are used in the same way each time. In the example, "It's her book", the apostrophe is treated as part of the string because double quotes have been used to enclose the string.

Strings are very useful when communicating with users. For example, asking them to enter some information into a program or displaying the output of a program in a format that humans can read and understand.

STRING INDEXING

GENERAL VOCABULARY

reference to refer to something

Each character in a string has an index number, with the first character at position 0. You can use the index to **reference** individual characters in a string.

Therefore the index position of the letter 'm' is 2, even though it is the third character, as shown in Table 2.6.

INDEX	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
String	C	o	m	p	u	t	e	r		S	c	i	e	n	c	e

▲ Table 2.6 An example string index

KEY POINT

Computers start counting at 0. Therefore, although the length of the string 'Computer Science' is 16, the indexes of the characters range from 0 to 15.

LENGTH

SUBJECT VOCABULARY

function a subprogram that performs a specific task and returns a value to the main program. High-level programming languages have a number of useful built-in functions. You can also create your own or use functions available in online libraries

The Pearson Edexcel pseudocode has a built-in **LENGTH function**, which you can use to find the number of characters in a string. Therefore:

```
SET numChars TO LENGTH(myText)
SEND numChars TO DISPLAY
```

would print '16'. Table 2.7 gives the method for Python, Java and C#.

LANGUAGE	METHOD	EXAMPLE	OUTPUT
Python	The function to find the length of a string is <code>len()</code> .	<code>string = 'Computer Science'</code> <code>length = len(string)</code> <code>print(length)</code>	16
Java	Each string object has a <code>.length()</code> method.	<code>String s = "Computer Science";</code> <code>int length = s.length();</code> <code>System.out.println(length);</code>	16
C#	The length of a string can be found by accessing the <code>length</code> property.	<code>string subject = "Computer Science";</code> <code>Console.WriteLine(subject.Length);</code>	16

▲ Table 2.7 Finding the length of a string in Python, Java and C#.

SKILLS

PROBLEM SOLVING,
CREATIVITY,
DECISION MAKING

ACTIVITY 12

CHECKING PASSWORD LENGTH

Create and write a program to check the length of a password. If the password entered is less than six characters, the program should output 'The password you have entered is not long enough'; otherwise it should output 'Length of password OK'.

STRING TRAVERSAL

SUBJECT VOCABULARY

string traversal using a loop to cycle through each character in a string

You can use a **FOR** loop to cycle through each of the characters in a string. This is known as **string traversal**.

The following algorithm prints out the word 'monkey' letter by letter, displaying each letter on a separate line.

```
SET animalName TO 'monkey'
FOR index = 0 TO LENGTH(animalName) - 1
    SEND animalName[index] TO DISPLAY
END FOR
```

The loop runs until `LENGTH(animalName) - 1` because string indexing starts at 0 and therefore has to print characters with indexes of 0 to 5.

PYTHON

```
animalName = 'monkey'
for index in range 0, len(animalName):
    print(animalName[index])
```

Using Python you should not use `len(animalName) - 1` as the range command does not include the end number of the loop.

JAVA

```
class Main {
    public static void main(String[] args) {
        String animalName = "monkey";
        for(int index = 0; index < animalName.length(); index++) {
            System.out.println(animalName.charAt(index));
        }
    }
}
```

C#

```
string animalName = "monkey";

for (int index = 0; index < animalName.Length; index++)
{
    Console.WriteLine(animalName[index]);
}
```

SKILLS

PROBLEM SOLVING,
CREATIVITY,
DECISION MAKING

ACTIVITY 13

WHICH CAR?

Write a program that will check if a make of car entered by the user is in the string 'The cars present included Ford, Mercedes, Toyota, BMW, Audi and Renault.'

If the car entered by the user is present, then 'It is present' should be returned or 'It is not present', if not.

It should not matter which case the car name is entered by the user.

OTHER WAYS TO MANIPULATE STRINGS

FINDING A CHARACTER WITH A PARTICULAR INDEX

The index of a character is the position of that character within a string. Note that indexing starts at 0 not 1. This means the first character will have an index of 0 and the second character will have an index of 1.

PYTHON

Python uses square brackets to access elements in a string:

```
string = 'Computer Science'
print(string[11])
```

This code would return the character 'c'.

JAVA

Java allows you to get a character at a particular index using the `charAt` method.

```
class Main {
    public static void main(String[] args) {
        String string = "Computer Science";
        System.out.println(string.charAt(11));
    }
}
```

This code would display the character 'i' (the 12th character at index 11).

C#

```
string text;
text = "Computer Science";
Console.WriteLine(text[3]);
```

This code would return the character 'p'.

CHANGING ALL CHARACTERS TO LOWER CASE**PYTHON**

```
string = 'Computer Science'
string = string.lower()
print(string)
```

This code would return 'computer science'.

JAVA

```
class Main {
    public static void main(String[] args) {
        String string = "Computer Science";
        string = string.toLowerCase();
        System.out.println(string);
    }
}
```

C#

```
string text;
text = "Computer Science";
text = text.ToLower();
Console.WriteLine(text);
```

Would output 'computer science'.

CHANGING ALL CHARACTERS TO UPPER CASE**PYTHON**

```
string = 'Computer Science'
string = string.upper()
print(string)
```

This would return 'COMPUTER SCIENCE'.

JAVA

```
class Main {
    public static void main(String[] args) {
        String string = "Computer Science";
        string = string.toUpperCase();
        System.out.println(string);
    }
}
```

C#

```
string text;
text = "Computer Science";
text = text.ToUpper();
Console.WriteLine(text);
```

This would output 'COMPUTER SCIENCE'.

HINT

Changing strings to upper or lower case is useful when you are comparing two strings and the case does not matter, only the characters.

EXTRACTING CHARACTERS FROM A STRING**PYTHON**

```
string = 'Computer Science'
substring = string[3:6]
print(substring)
```

This code would return 'put' as it selects the characters from index 3 to index 5. Index 6 is not included.

JAVA

```
class Main {
    public static void main(String[] args) {
        String string = "Computer Science";
        String substring = string.substring(3,6);
        System.out.println(substring);
    }
}
```

C#

```
string text;
text = "Computer Science";
text = text.Substring(3,4);
Console.WriteLine(text);
```

This would display 'put' as it selects 4 characters starting at index 3 (remember that the first character is at index 0).

CHECKING A PHRASE IN THE STRING

These examples check to see if the string 'Computer Science' contains the strings 'put' and 'PUT': 'put' is a substring of 'Computer Science' but 'PUT' is not, so these examples will display 'True' then 'False'.

PYTHON

```
string = 'Computer Science'
present = 'put' in string
print(present)
present = 'PUT' in string
print(present)
```

JAVA

```
class Main {
    public static void main(String[] args) {
        String string = "Computer Science";
        boolean present = string.contains("put");
        System.out.println(present);
        present = string.contains("PUT");
        System.out.println(present);
    }
}
```

C#

You can check if a substring is present in a string using the `contains` method. If the substring is present in the string then 'True' is returned, otherwise 'False' is returned.

```
string text;
bool found = false;
text = "Computer Science";
found = text.Contains("put");
Console.WriteLine(found);
```

This would display 'True', as 'put' is in 'Computer Science'.

CONCATENATION

Concatenation involves joining two or more items of information together. Concatenating two strings produces a new string object. It is very useful when displaying text on screen.

In Pearson Edexcel pseudocode concatenation is done in the following way.

```
RECEIVE userName FROM (STRING) KEYBOARD
SEND 'Hello' & userName TO DISPLAY
```

Note that literal text is enclosed in speech marks but the variable name is not. In Python, this would be:

```
userName = input('Please enter your username')
print('Hello' + userName)
```

The '+' character is used for concatenation.

In Python, you cannot concatenate strings with numbers. For example, the following would produce an error message.

```
length = 13
print('The length is' + length)
```

To overcome this, the number must be converted to a string:

```
length = 13
print('The length is' + str(length))
```

In Java, you can concatenate strings with most other data types:

```
Scanner scanner = new Scanner(System.in);
System.out.println("Please enter your username:");
String username = scanner.nextLine();
System.out.println("Hello" + username);

int length = 13;
System.out.print("The length is" + length);
```

In C# this would be:

```
string userName;
Console.WriteLine("Please enter your username:");
userName = Console.ReadLine();
Console.WriteLine("Hello" + userName);
```

In C#, you can concatenate with numbers.

COMPARING STRINGS

Comparing strings in Python and C# can be done using the == operator but in Java you need to use a string object's equals method:

```
String food = "pie";
if(food.equals("pie")) {
    System.out.println("Yum! I like pie");
}
```

SKILLS

PROBLEM SOLVING,
CREATIVITY,
DECISION MAKING

ACTIVITY 14

CONCATENATING AND SLICING STRINGS

A company wants a program to generate usernames for new employees. Each username consists of the first four letters of the employee's last name and the first letter of their first name joined together. If the employee's last name is less than four characters in length a letter 'X' is used to fill in for each of the missing characters. Develop a program that asks the user to input their first and last names and outputs their username.

SKILLS

CRITICAL THINKING

SKILLS

PROBLEM SOLVING,
CREATIVITY

SKILLS

CRITICAL THINKING

SKILLS

PROBLEM SOLVING,
CREATIVITY

CHECKPOINT

Strengthen

S1 How are individual characters in a string referenced?**S2** Develop an algorithm that uses a loop to traverse a string.**S3** How are a string and a non-string concatenated?

Challenge

C1 Develop a program that asks the user to input a sentence and then splits it up wherever a space occurs. Each word should then be displayed on a separate line.

How confident do you feel about your answers to these questions? If you're not sure you answered them well, have another go at the activities in this section.

SUMMARY

- A string is a sequence of characters.
- Each character in a string has a unique index value representing its position in the string. The first character in a string has the index value 0.
- High-level programming languages have a built-in length function that finds the length of a string.
- A loop is used to traverse a string, character by character.
- Concatenation is the process of joining two or more strings together.
- Slicing is the process of extracting part of a string.
- String formatting is used to control the way text is displayed on screen.

8 DATA STRUCTURES

Data should be classified and organised into similar types so it can be easily searched and analysed. They are stored together in data structures, such as records and arrays.

LEARNING OBJECTIVES

- Describe the structure of one- and two-dimensional arrays and give examples of their use
- Create and use one- and two-dimensional arrays in programs
- Describe the record data structure and explain what it is used for
- Design record structures

SUBJECT VOCABULARY

data structure an organised collection of related elements. Arrays and records are two common data structures used in programming

A **data structure** is an organised collection of related elements. There are many different data structures that can store multiple data items used in programming. You have already encountered one of them – strings. In this section we will investigate two more: arrays and records.

ARRAYS

SUBJECT VOCABULARY

array a structure that contains many items of data of the same type. The data is indexed so that a particular item of data can be easily found

index a number that identifies each element of an array in Python and Java

An **array** is an organised collection of related values with a single shared identifier. All the elements in an array are the same data type. Each has a unique **index** value denoting its position in the array.

In the Pearson Edexcel pseudocode, an array is initialised using the **SET** command. For example, this statement initialises an array called **firstNames** with four elements, all of string type. The square brackets denote the start and end of the array.

```
SET firstNames TO [‘Ashura’, ‘Bryn’, ‘Eloise’, ‘Mei’]
```

In most programming languages, arrays are static. A static array has a fixed size and when it is declared the number of items it can hold must be stated.

For example, **array friends [5]**.

As with strings, arrays have a length indicating the number of items. Here are the functions to find the length of an array declared as **myArray**.

PYTHON	JAVA	C#
<code>len(myArray)</code>	<code>myArray.length</code>	<code>myArray.Length</code>

Loops can be used to traverse arrays.

ACTIVITY 15

Declare an array with the elements Ford, Mercedes, Toyota, BMW, Audi and Renault.

In a high-level language, create a program to find the length of the array and then traverse it, printing out each of the items.

PYTHON

In Python, arrays are not commonly used. Instead **lists** are used but they are very similar in the way that they operate. Just like an array, a list is created by adding items, separated by commas, inside square brackets.

Lists are easier to use as they are **dynamic** – they do not have a fixed size and can grow as new elements are added. When they are declared they do not have to be given a size, e.g.:

```
cars = []
```

When adding items to an array the `append` command is used, for example:

```
cars = []
cars.append('Audi')
```

```
print(cars)
```

would return the following:

```
['Audi']
```

Another advantage of lists is that the items do not have to be of the same data type.

```
cars = []
cars.append('Audi')
cars.append(3)
print(cars)
```

would return:

```
['Audi', 3]
```

There are many functions in Python for manipulating arrays.

Investigate the following:

```
max()
min()
slice()
```

JAVA

Arrays are static data structures, which means that you can't change the size of an array to make it bigger or smaller in order to add or remove values.

For example, you can have an array of 5 integers:

```
int[] integers = {1,2,3,4,5};
```

The following is not possible with an array:

```
integers.add(6); or integers.remove(5);
```

In order to add or remove values you need to use a list which is a dynamic data structure.

The most common type of list is an `ArrayList`:

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> integers = new ArrayList<Integer>();
        integers.add(1);
        integers.add(2);
        integers.add(3);
        integers.add(4);
        integers.add(5);
        integers.add(6);
        integers.remove(0); // remove the value at index 0
                            (not the value 0)
        System.out.println(integers);
    }
}
```

This example will display [2,3,4,5,6] because the first value will be removed.

Arrays are supported for any data type in Java. For example, you can have an array of strings:

```
String[] colours = {"red", "green", "blue"};
```

Or an array of integers:

```
int[] numbers = {1,2,3,4};
```

However, you can't have an array that contains both strings and integers (or any other mix of data types).

For this, you need to use a vector:

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        Vector allSorts = new Vector();
        allSorts.add(1);
        allSorts.add("Two");
        allSorts.add(true);
        System.out.println(allSorts);
    }
}
```

This example will output [1, Two true] because the vector `allSorts` contains an integer, a string and a Boolean value.

C#

C# supports lists and arrays. The code below uses an array to hold the items.

```
string[] make = { "Ford", "Mercedes", "Toyota", "BMW",
    "Audi", "Renault" };

foreach (string currentItem in make)
{
    Console.WriteLine(currentItem);
```

SKILLS

CRITICAL
THINKING,
PROBLEM
SOLVING,
CREATIVITY

ACTIVITY 16

DESCRIBING AND IMPLEMENTING A LINEAR SEARCH ALGORITHM

The following code uses a linear search algorithm.

```

SET firstNames TO ['Ashura', 'Bryn', 'Eloise', 'Mei', 'James', 'Irena']
RECEIVE searchName FROM (STRING) KEYBOARD

SET found TO False
SET index TO 0

WHILE found = False AND index <= (LENGTH(firstNames) - 1) DO
    IF searchName = firstNames[index] THEN
        SET found TO True
    END IF
    SET index to index + 1
END WHILE

IF found = True THEN
    SEND searchName & ' is at position' & index & ' in the list' TO DISPLAY
ELSE
    SEND searchName & ' is not in the list' TO DISPLAY
END IF

```

Implement this algorithm in the high-level programming language you are studying.



MULTIDIMENSIONAL ARRAYS

In the array `cars = ['Ford', 'Mercedes', 'Toyota', 'BMW', 'Audi', 'Renault']` there is only one item at each index position: the name of the manufacturer.

A multidimensional array is an 'array of arrays'; each item at an index is another array.

We declared the above array using this statement, `array cars[6]`.

If we wanted to create a two-dimensional array we could declare it as `array cars[3, 2]` so that there is another array at each index to store two items of information.

Here is an extract from an array named `examResults`. It has three rows, each of which stores a set of four exam results. The mark of 47 is located at `examResults [1, 2]` – second row, third element along – and the value 80 at `examResults [0, 0]` – first row, first column.

	0	1	2	3
0	80	59	34	89
1	31	11	47	64
2	29	56	13	91

▲ Table 2.8 `examResults` two-dimensional array

Each item of data has two indexes. An array to hold this data would be declared as `array[3, 4]`.

If the array was printed it would be `[[80, 59, 34, 89], [31, 11, 47, 64], [29, 56, 13, 91]]`. There are square brackets around each set of results and around the whole array.

In Python, the array would be declared and initialised as:

```
Scores = [[80, 59, 34, 89], [31, 11, 47, 64], [29, 56, 13, 91]]
```

In Java, it would be:

```
int[][] Scores = new int[][] {
    {80, 59, 34, 89},
    {31, 11, 47, 64},
    {29, 56, 13, 91}};
```

In C# the array would be declared and initialised as:

```
int[,] Scores = new int[,] { { 80, 59, 34, 89 }, { 31, 11, 47, 64 }, { 29, 56, 13, 91 } };
```

WORKED EXAMPLE

A teacher has stored the surnames and test scores of a class of students in a two-dimensional array, e.g. `results[['Smith', '69'], ['Jackson', '90']]`, etc. Create a program that would print out the names and test scores of all the students who have scored 50 or over in the test.

Python

```
results = [['Smith', 69], ['Jackson', 90], ['Dubois', 30]]
for index in range(0, len(results)):
    if results[index][1] >= 50:
        print(results[index][0] + str(results[index][1]))
```

In Python each element is written as `results[0][1]` and not `results[0, 1]`.

Java

```
class Main {
    public static void main(String[] args) {
        String[][] results = {
            {"Smith", "69"},
            {"Jackson", "90"},
            {"Dubois", "30"}};
```

```

for(int index = 0; index < results.length; index++) {
    int score = Integer.parseInt(results[index][1]);
    String name = results[index][0];
    if(score >= 50) {
        System.out.println(name + score);
    }
}
}
}
}
}

```

In Java, a two-dimensional array has two indexing expressions (e.g., `results[0][1]`). The first index is the row number and the second index is the column number. Both index expressions count from 0. This means `results[0][1]` will get the value in the second column (index 1) of the first row (index 0), which in the above example would be '69'.

C#

```

string[,] results = new string[,] { { "Smith", "69" },
{ "Jackson", "90" }, { "Dubois", "30" } };

for (int index = 0; index <= results.GetUpperBound(0);
index++)
{
    if (int.Parse(results[index,1]) >= 50)
    {
        Console.WriteLine(results[index,0] + " " +
results[index,1]);
    }
}

```

In C# an array only consists of one data type, hence the student scores are in double quotes. To be able to check if the student scores are above 50, the score in the array must be cast as an integer using `int.Parse`.

SKILLS

CRITICAL THINKING,
PROBLEM SOLVING,
CREATIVITY

HINT

All the data values stored in an array must be of the same data type. In this case they are all strings. You will have to convert the scores to integers to find the highest. But as Python uses lists this is not necessary. Strings and integers can be combined within the same list.

ACTIVITY 17

USING TWO-DIMENSIONAL ARRAYS

- 1 Develop a program that creates and initialises an array to hold these five sets of marks.
80, 59, 34, 89
31, 11, 47, 64
29, 56, 13, 91
55, 61, 48, 0
75, 78, 81, 91
- 2 Extend your program so that it calculates and displays the highest mark, the lowest mark and the average mark achieved.

ACTIVITY 18

This two-dimensional array holds the highest score achieved by each player in each of the three levels of an online game.

- 1 Develop a program that initialises the array and then searches through it to find the player with the highest score in each of the three levels.

PLAYER	LEVEL	SCORE
Alexis	1	19
Seema	1	29
Seema	2	44
Lois	1	10
Alexis	2	17
Alexis	3	36
Dion	1	23
Emma	1	27
Emma	2	48

▲ Table 2.9 Example of two-dimensional array

RECORDS

SUBJECT VOCABULARY

record a data structure that stores a set of related values of different data types

field an individual element in a record

We have already said that the elements of an array must all be the same data type. In contrast, the **record** data structure stores a set of related values of different data types.

Each element in a record is known as a **field** and is referenced using a field name.

Table 2.10 shows how the record data structure works. Each row of the table holds a set of information about a particular learner (these are the records). Each column stores one item of information about the learner – their learner number, their age, their form, etc. (these are the fields). All the values in a column have the same data type – `learnerNum` and ‘age’ are integers; `firstName`, `lastName` and ‘form’ are strings.

Programming languages vary in the way they handle the record data structure.

learnerNum	firstName	lastName	AGE	FORM
1	Isla	Smith	15	10H
2	Shinji	Fujita	14	10B
3	Anita	Khan	15	10A
4	Abdur	Rahman	15	10G

▲ Table 2.10 Example of record data structure

SKILLS

CRITICAL THINKING,
PROBLEM SOLVING,
CREATIVITY

▲ It is essential to have a way to sort through the vast amounts of recorded music available

SKILLS

CRITICAL THINKING

SKILLS

REASONING

SKILLS

REASONING

SKILLS

REASONING

SKILLS

CRITICAL THINKING,
PROBLEM SOLVING

SKILLS

CRITICAL THINKING,
PROBLEM SOLVING

ACTIVITY 19

USING RECORDS TO STORE MUSIC DETAILS

- 1 A record data structure is to be used to store the details of music albums. Provide the appropriate data type for these fields:
 - a the title of the album
 - b the name of the artist
 - c the year of release
 - d the genre.
- 2 Develop a program that uses a record structure for storing the details of music albums. It must:
 - a have fields for title, artist, year of release and genre
 - b allow the user to input the details of new albums
 - c allow the user to search for an album by name and display its details.

CHECKPOINT

Strengthen

S1 What is the index of the first element in a one-dimensional array?

S2 How does a linear search algorithm find an element in a one-dimensional array?

S3 How is an element stored in a two-dimensional array referenced?

S4 How is a nested IF used to traverse a two-dimensional array?

Challenge

C1 Develop a program for a simple address book that uses a two-dimensional array to store a set of names and email addresses, and allows the user to search for a person by name and returns their email address.

C2 Develop a program that uses a two-dimensional array to represent a treasure map consisting of a grid of 4 rows and 4 columns. A random number function should be used to establish the location of the treasure. The user must hunt for the treasure by repeatedly entering the coordinates of squares. The program should tell them when they have found the treasure and help them in their search by indicating how close they are.

How confident do you feel about your answers to these questions? If you're not sure you answered them well, try redoing activities 15–19.

SUMMARY

- A data structure is an organised collection of related elements. Arrays and records are common data structures.
- A one-dimensional array is a list of elements, each of which has a unique index value representing its position in the list.
- A two-dimensional array is a matrix of rows and columns. Each element in the array has a unique pair of indices, one to identify the row and one the column in which it is located.
- All the elements in an array have the same data type (Python uses lists).
- A record consists of a collection of fields. The values stored in a record can be of different data types.

9 INPUT/OUTPUT

All program data has to be entered (input) and information is output. The input could be automatic, e.g. from a sensor, but is often provided by human users. It is essential that this input is checked to ensure that it is what could reasonably be expected and falls within a certain range. Incorrect input could cause problems or even catastrophes.

LEARNING OBJECTIVES

- Explain the need for validation
- Write code that validates user input
- Write code that accepts and responds appropriately to user input
- Write code that reads to and writes from a text file

USER INPUT

Most programs require some form of input either from a user or from a file. You already know how to receive user input from a keyboard.

A program can be made much more ‘user friendly’ by displaying helpful messages informing users of what they are expected to enter and confirming successful input.

VALIDATION

It is important to ensure that data entered by the user is valid, as invalid data can cause a program to behave unexpectedly or even stop altogether. If the data entered into a program is incorrect, the output it produces will also be wrong. This is sometimes called the Garbage In, Garbage Out (GIGO) principle.

Any program that requires data entry should have appropriate forms of **validation** built in. But validation can’t guarantee that the data entered is correct. It can only make sure that it is reasonable.

There are a number of different types of validation.

RANGE CHECK

A range check is used to ensure that the data entered is within a specified range. Study this algorithm, written in Pearson Edexcel pseudocode which checks that the number entered is between 1 and 10.

```

BOOLEAN valid
SET validNum TO False
WHILE validNum = False DO
    SEND 'Please enter a number between 1 and 10:' TO
    DISPLAY
    RECEIVE number FROM (INTEGER) KEYBOARD
    IF number >= 1 AND number <= 10 THEN
        SET validNum TO True
    END IF
END WHILE
SEND 'You have entered:' & number TO DISPLAY

```

The algorithm uses a Boolean variable named `validNum` as a status flag. It is initially set to False. The `WHILE` loop continues running until `validNum` is equal to True. An `IF` statement determines if the value of `validNum` should be set to True.

SKILLS

CRITICAL THINKING,
PROBLEM SOLVING

ACTIVITY 20

IMPLEMENTING RANGE CHECKS

Implement the range check algorithm in the high-level programming language you are studying.



SKILLS

CRITICAL THINKING,
PROBLEM SOLVING

ACTIVITY 21

IMPLEMENTING A PRESENCE CHECK

Implement the presence check algorithm in the high-level programming language you are studying.

GENERAL VOCABULARY

predefined already defined before the start of something

LOOK-UP CHECK

A look-up check is used to test that a value is one of a **predefined** set of acceptable values. The list of acceptable values can be stored in a one-dimensional array.

This algorithm stores a list of valid form names in an array. It compares the form name entered by the user with the values in the array.

```
SET arrayForms TO ['7AXB', '7PDB', '7ARL', '7JEH']
RECEIVE form FROM (STRING) KEYBOARD
SET valid TO False
SET index TO 0
SET length TO LENGTH(arrayForms)
WHILE valid = False AND index < length DO
    IF form = arrayForms[index] THEN
        SET valid TO True
    END IF
    SET index TO index + 1
END WHILE
```