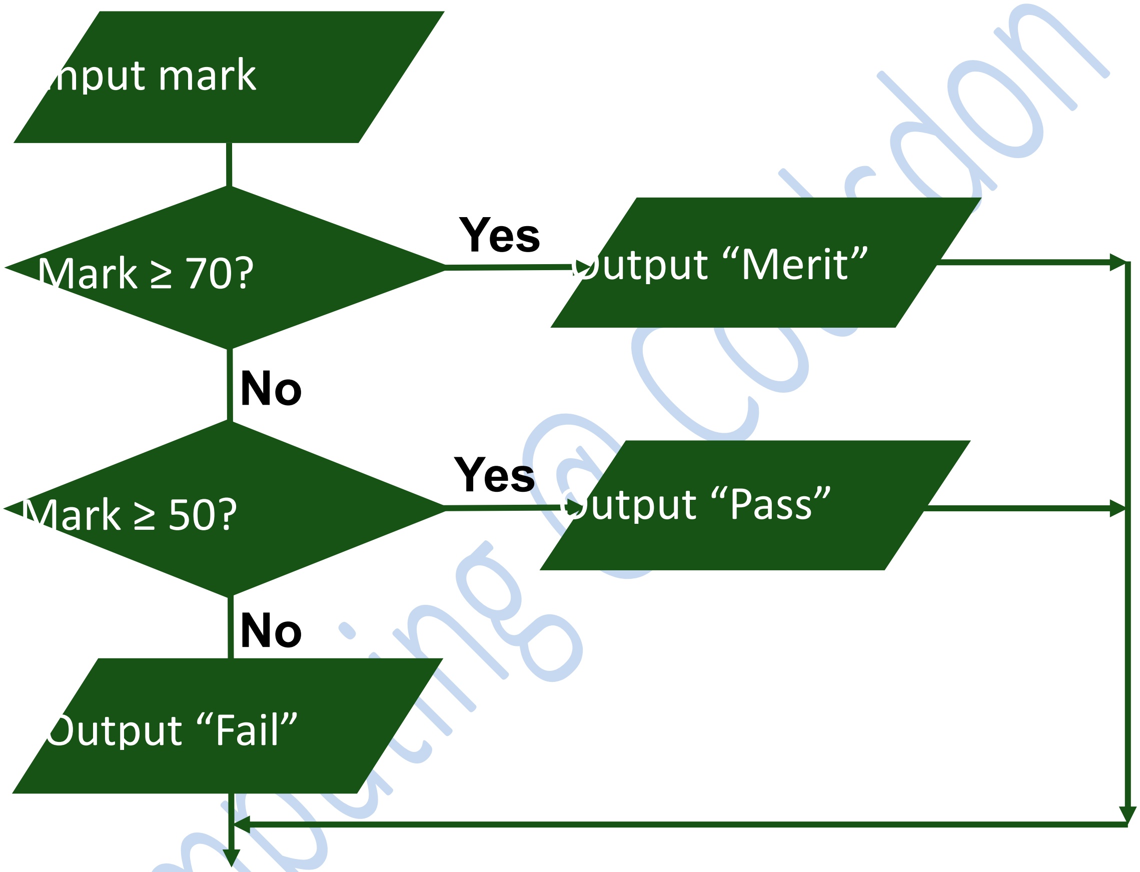
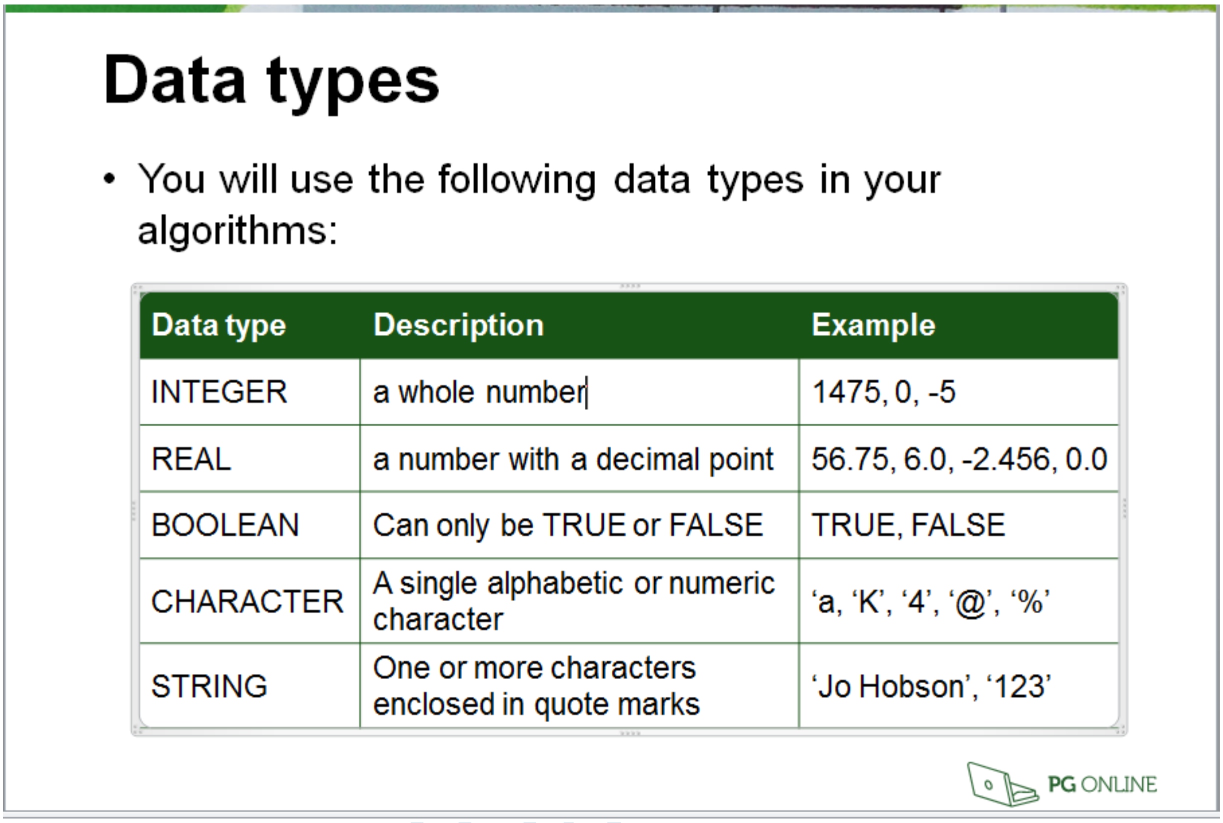
Computing @Coulsdon

Summer 2019



At the end of these tasks you should be able to:

- state what is meant by an algorithm
- state what is meant by abstraction
- state what is meant by decomposition
- state the sequence in which items in a sorted list will be examined in a linear and binary search
- state the advantages and disadvantages of a linear and binary search
- state an advantage and disadvantage of the merge sort compared with the bubble sort
- show the state of a list after the first pass in a bubble sort
- use a flowchart or pseudocode to define the steps in a simple algorithm
- trace through a simple flowchart or pseudocode algorithm to determine the output

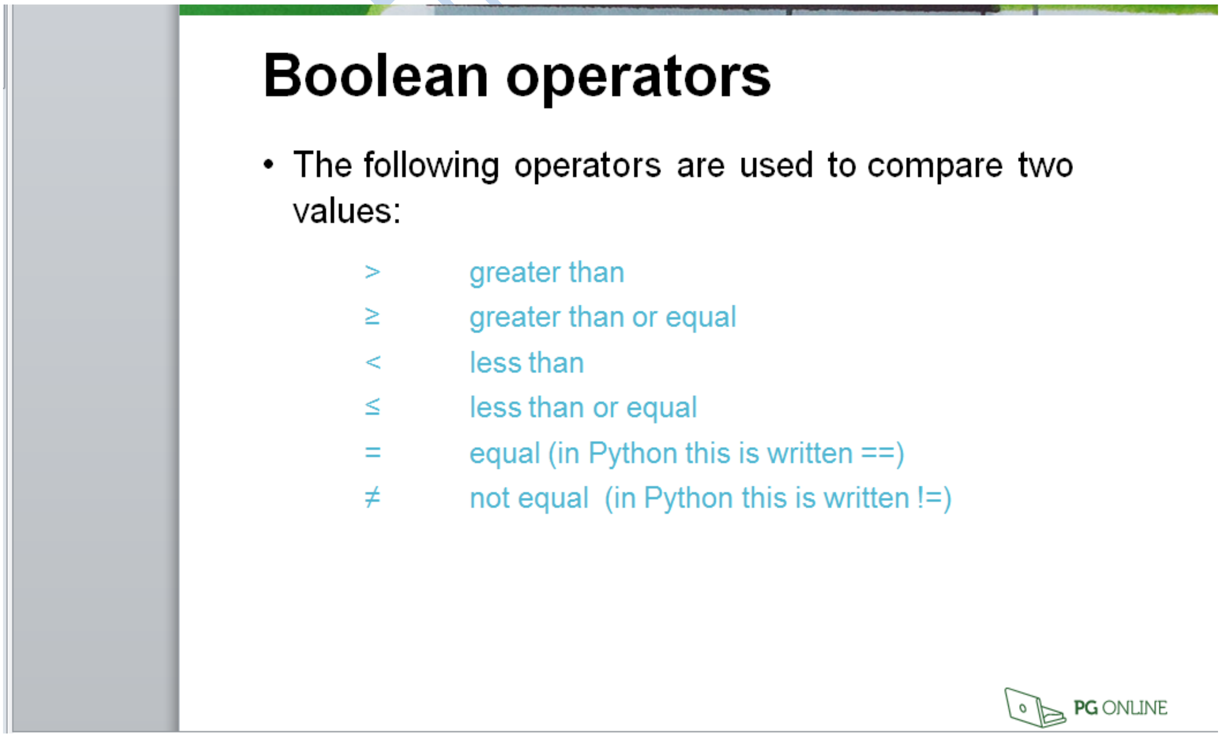


Data types

- You will use the following data types in your algorithms:

Data type	Description	Example
INTEGER	a whole number	1475, 0, -5
REAL	a number with a decimal point	56.75, 6.0, -2.456, 0.0
BOOLEAN	Can only be TRUE or FALSE	TRUE, FALSE
CHARACTER	A single alphabetic or numeric character	‘a, ‘K’, ‘4’, ‘@’, ‘%’
STRING	One or more characters enclosed in quote marks	‘Jo Hobson’, ‘123’

PG ONLINE

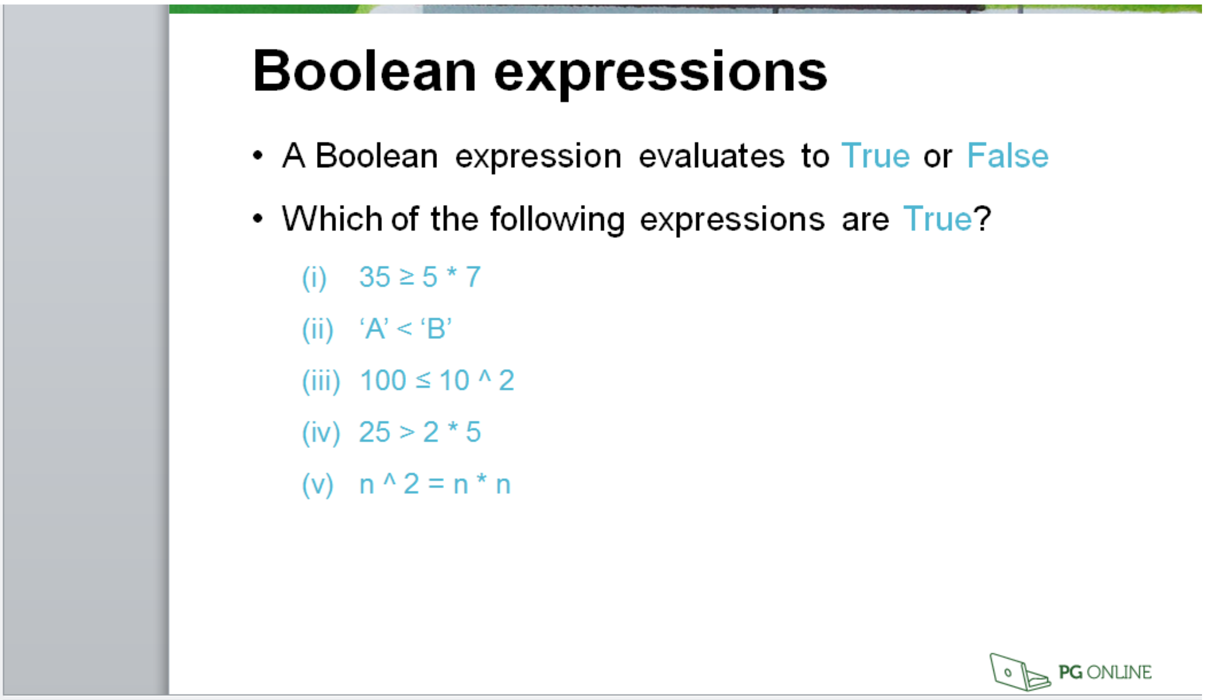


Boolean operators

- The following operators are used to compare two values:

>	greater than
≥	greater than or equal
<	less than
≤	less than or equal
=	equal (in Python this is written ==)
≠	not equal (in Python this is written !=)

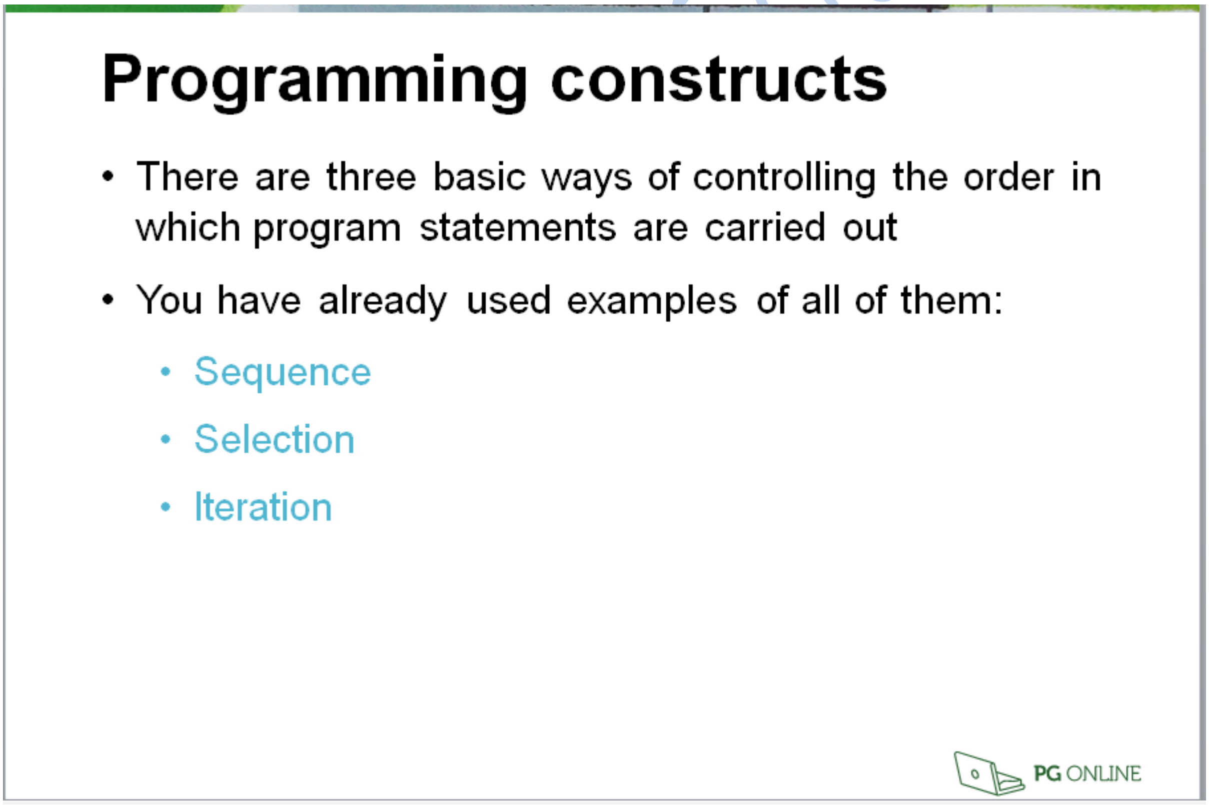
PG ONLINE



Boolean expressions

- A Boolean expression evaluates to True or False
- Which of the following expressions are True?
  - (i) 35 ≥ 5 * 7
  - (ii) ‘A’ < ‘B’
  - (iii) 100 ≤ 10 ^ 2
  - (iv) 25 > 2 * 5
  - (v) n ^ 2 = n * n

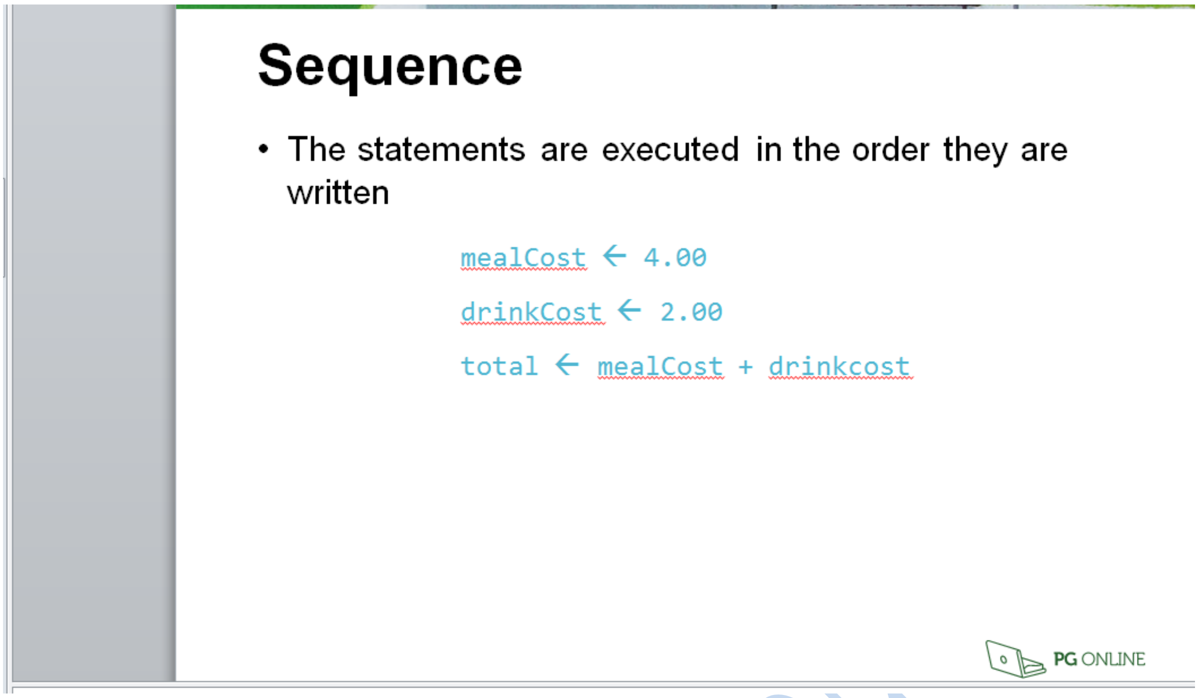
PG ONLINE



Programming constructs

- There are three basic ways of controlling the order in which program statements are carried out
- You have already used examples of all of them:
  - Sequence
  - Selection
  - Iteration

PG ONLINE

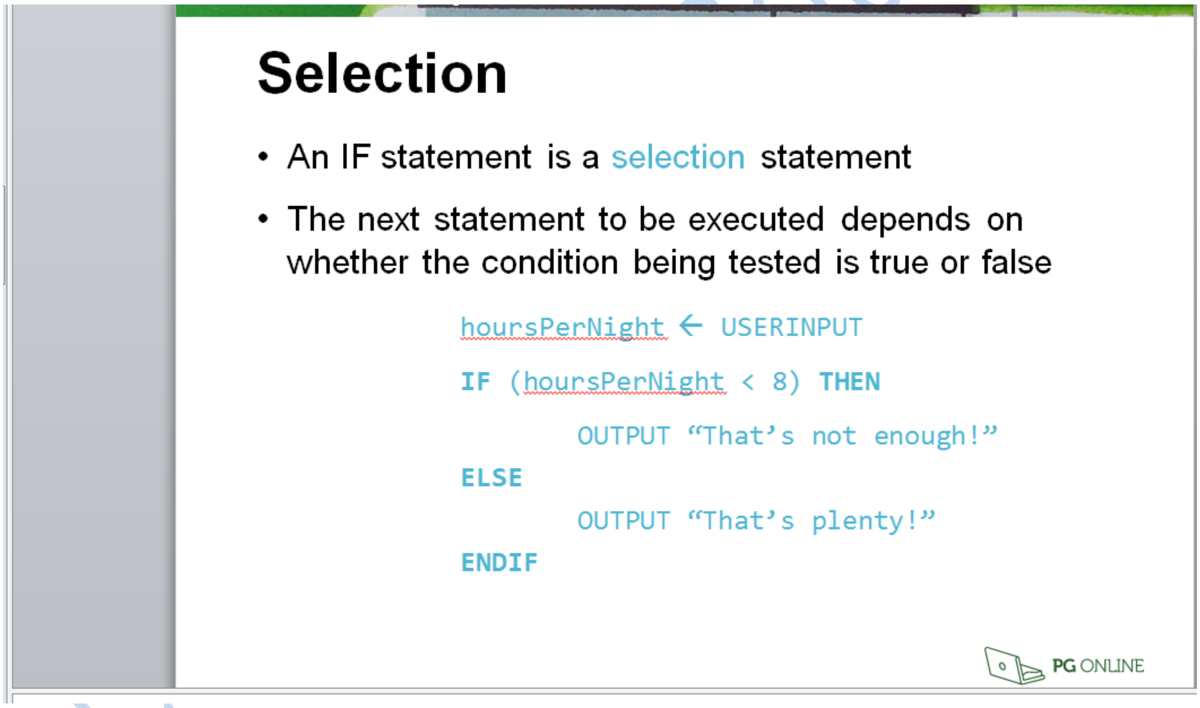


Sequence

- The statements are executed in the order they are written

```
mealCost ← 4.00
drinkCost ← 2.00
total ← mealCost + drinkcost
```

PG ONLINE

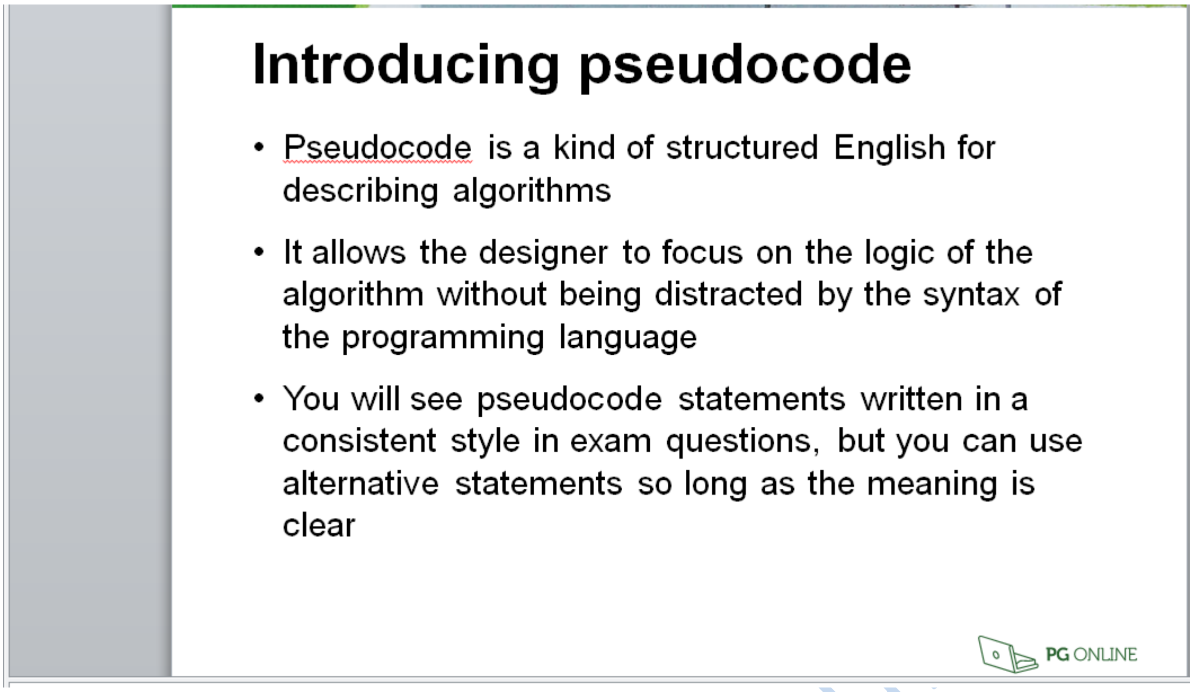


Selection

- An IF statement is a selection statement
- The next statement to be executed depends on whether the condition being tested is true or false

```
hoursPerNight ← USERINPUT
IF (hoursPerNight < 8) THEN
        OUTPUT “That’s not enough!”
ELSE
        OUTPUT “That’s plenty!”
ENDIF
```

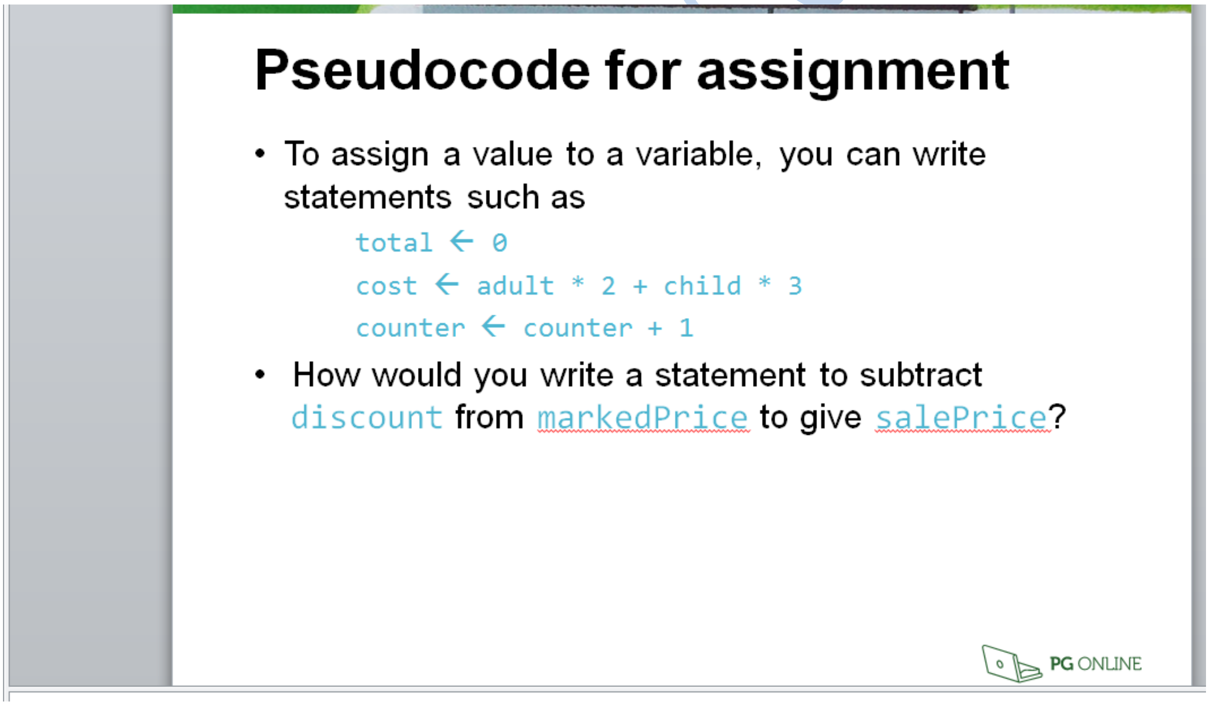
PG ONLINE



Introducing pseudocode

- Pseudocode is a kind of structured English for describing algorithms
- It allows the designer to focus on the logic of the algorithm without being distracted by the syntax of the programming language
- You will see pseudocode statements written in a consistent style in exam questions, but you can use alternative statements so long as the meaning is clear

PG ONLINE



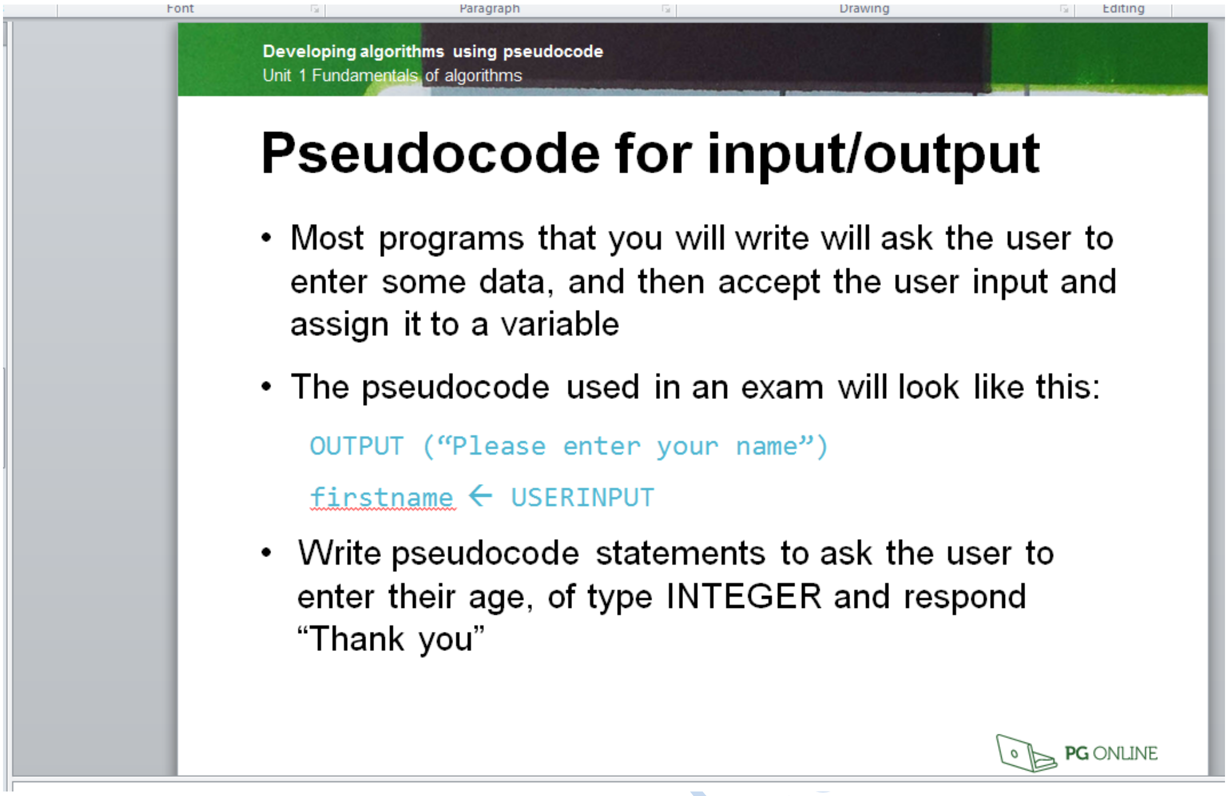
Pseudocode for assignment

- To assign a value to a variable, you can write statements such as

```
total ← 0
cost ← adult * 2 + child * 3
counter ← counter + 1
```

- How would you write a statement to subtract `discount` from `markedPrice` to give `salePrice`?

PG ONLINE



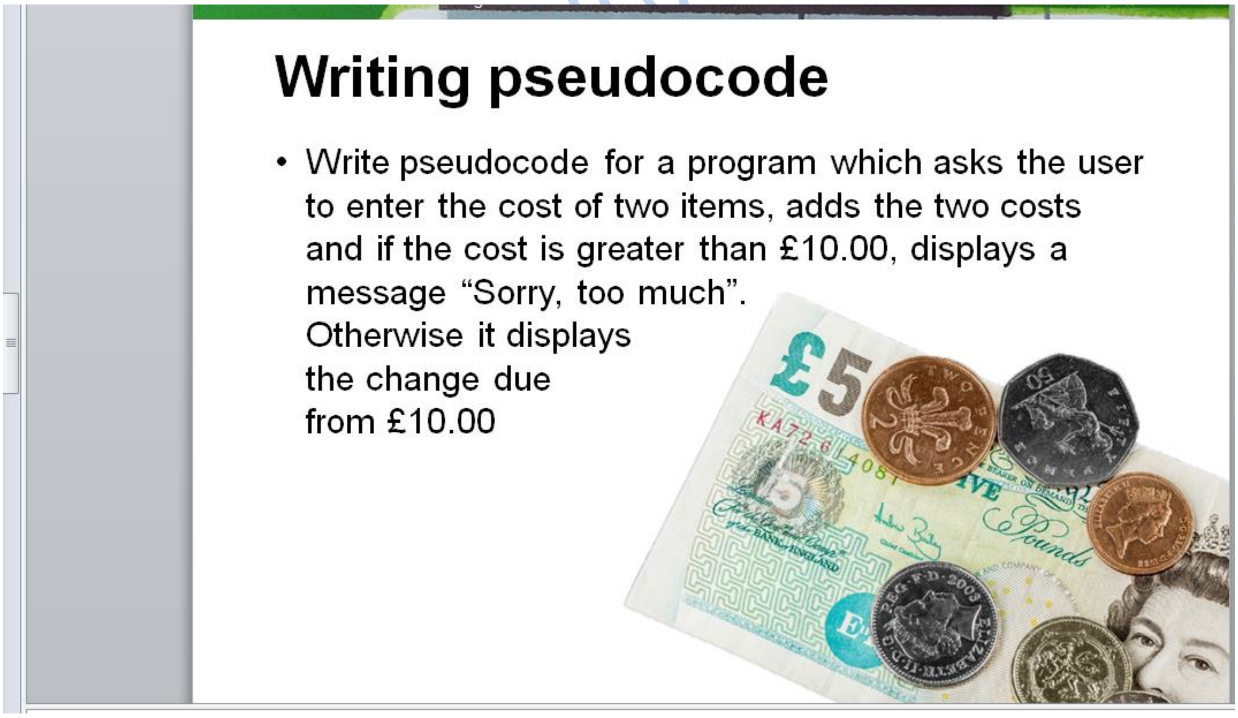
Developing algorithms using pseudocode
Unit 1 Fundamentals of algorithms

Pseudocode for input/output

- Most programs that you will write will ask the user to enter some data, and then accept the user input and assign it to a variable
- The pseudocode used in an exam will look like this:

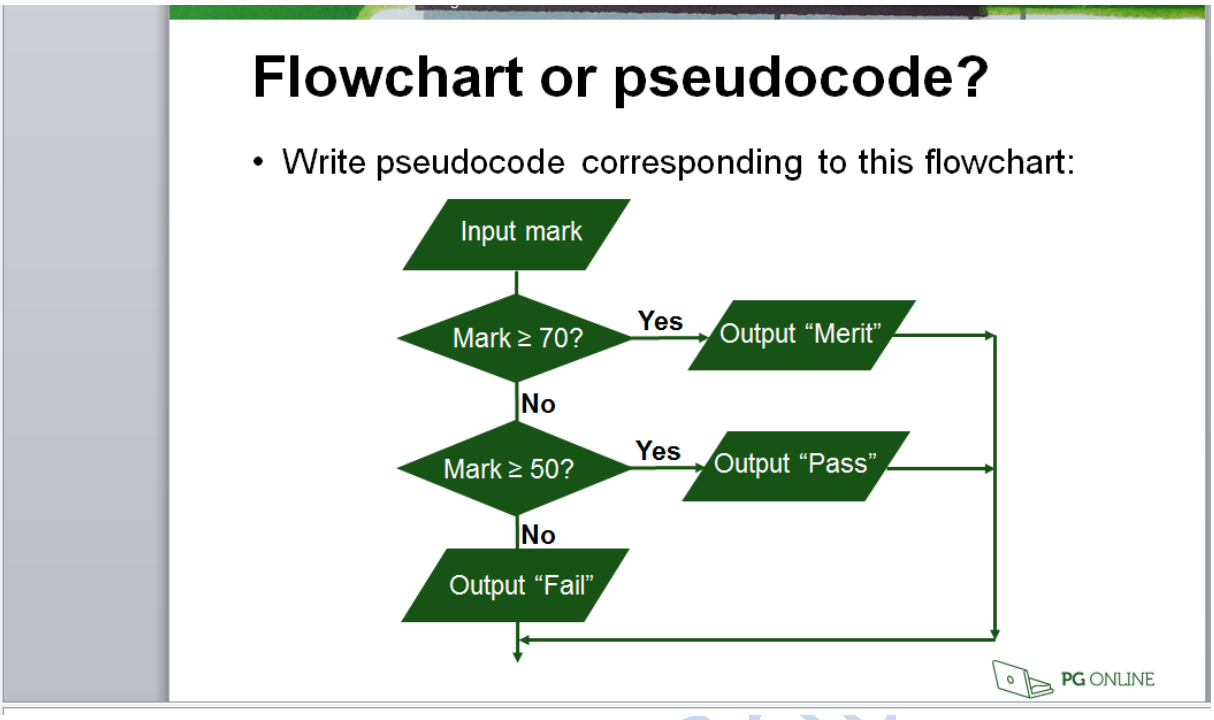
```
OUTPUT ("Please enter your name")
firstname ← USERINPUT
```

- Write pseudocode statements to ask the user to enter their age, of type INTEGER and respond "Thank you"



Writing pseudocode

- Write pseudocode for a program which asks the user to enter the cost of two items, adds the two costs and if the cost is greater than £10.00, displays a message "Sorry, too much". Otherwise it displays the change due from £10.00



Flowchart or pseudocode?

- Write pseudocode corresponding to this flowchart:

Input mark

Mark ≥ 70?

Yes

Output “Merit”

No

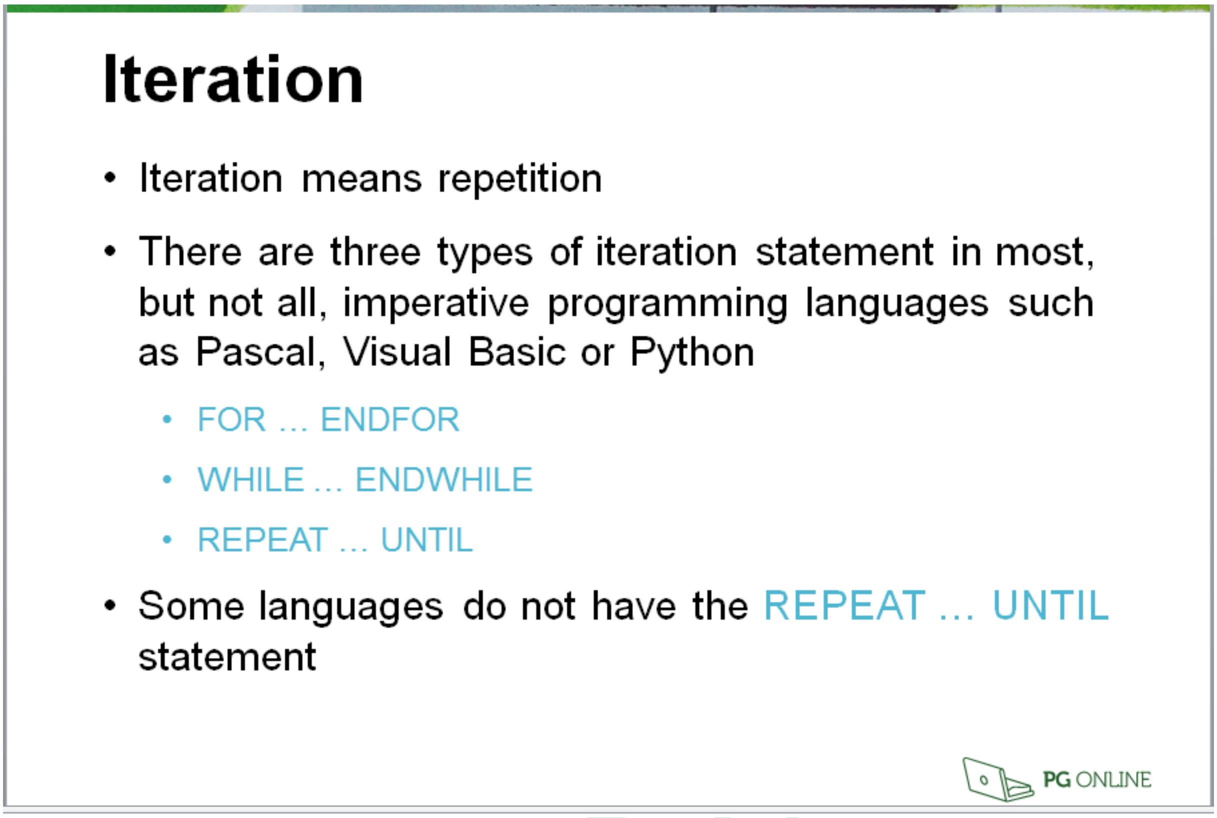
Mark ≥ 50?

Yes

Output “Pass”

No

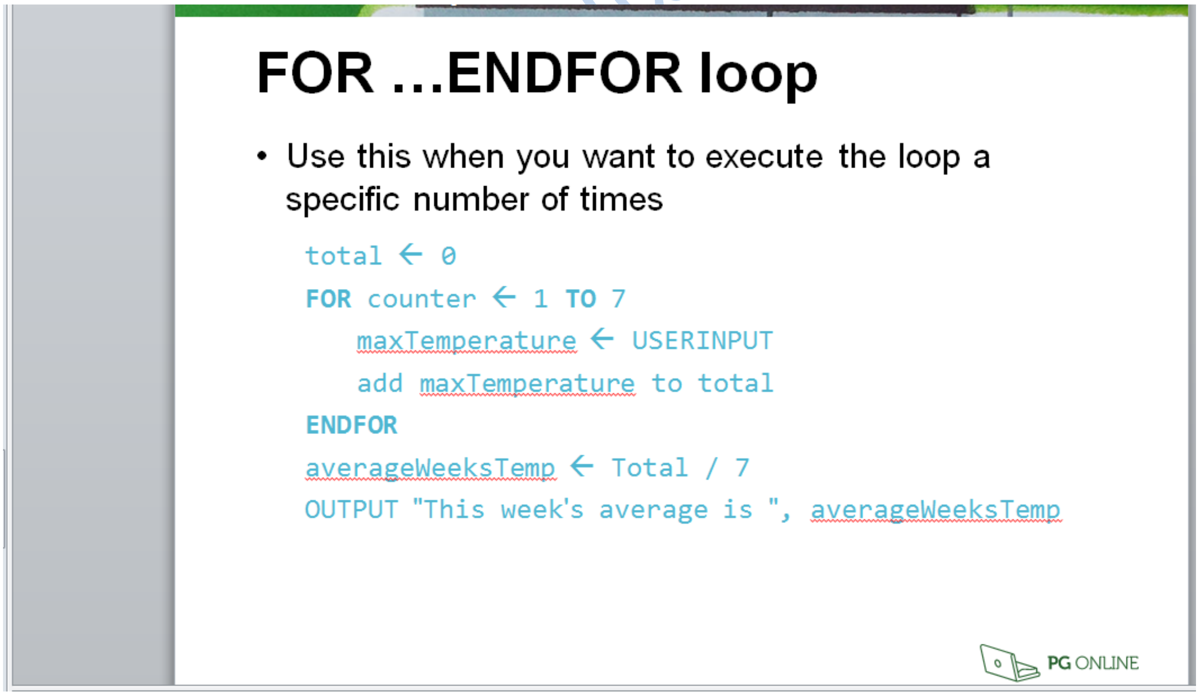
Output “Fail”



Iteration

- Iteration means repetition
- There are three types of iteration statement in most, but not all, imperative programming languages such as Pascal, Visual Basic or Python
  - FOR … ENDFOR
  - WHILE … ENDWHILE
  - REPEAT … UNTIL
- Some languages do not have the REPEAT … UNTIL statement

PG ONLINE

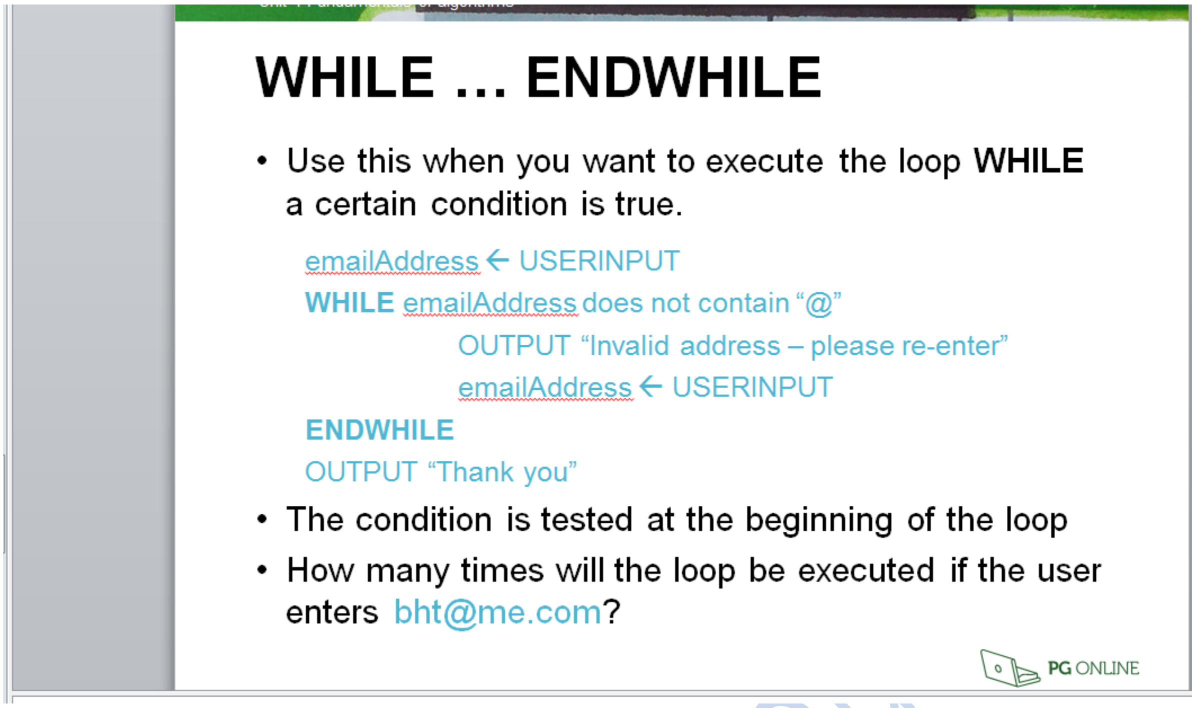


FOR …ENDFOR loop

- Use this when you want to execute the loop a specific number of times

```
total ← 0
FOR counter ← 1 TO 7
    maxTemperature ← USERINPUT
    add maxTemperature to total
ENDFOR
averageWeeksTemp ← Total / 7
OUTPUT "This week's average is ", averageWeeksTemp
```

PG ONLINE

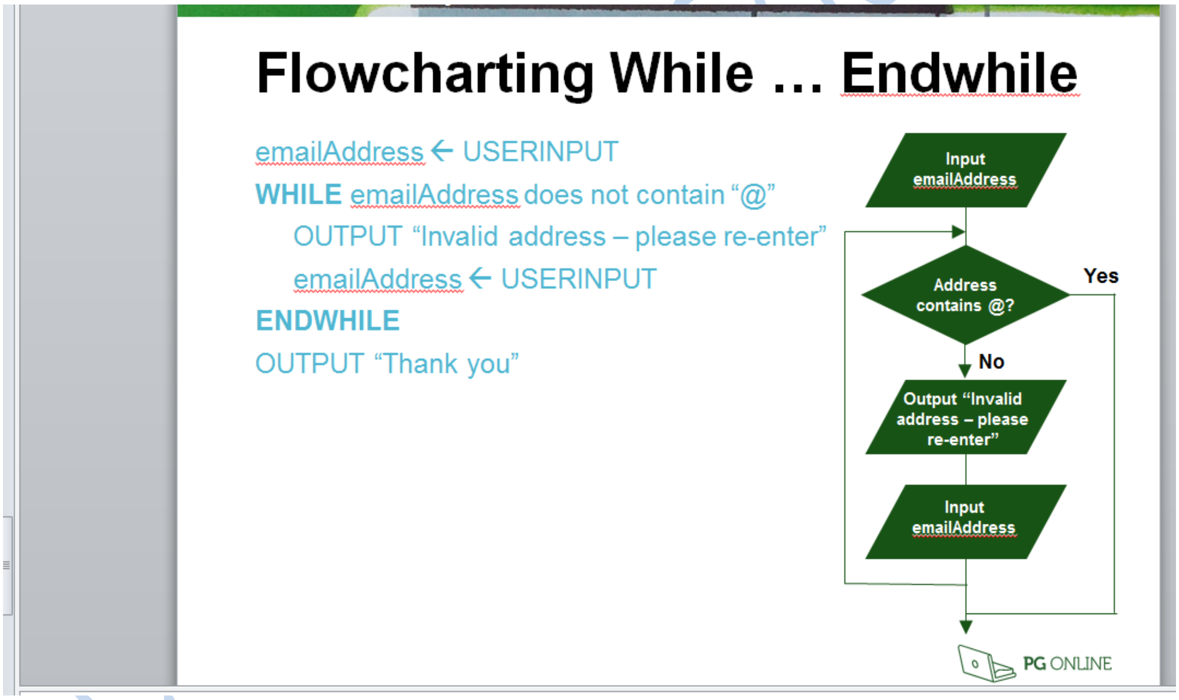


WHILE … ENDWHILE

- Use this when you want to execute the loop **WHILE** a certain condition is true.

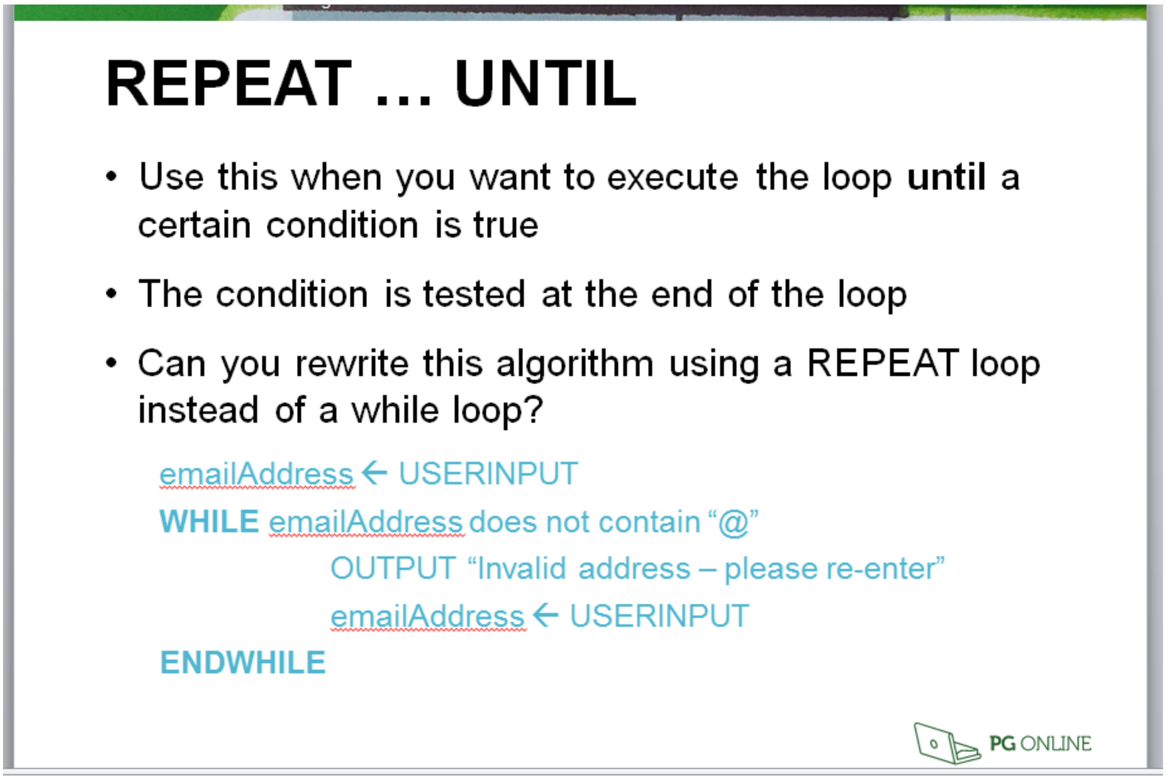
```
emailAddress ← USERINPUT
WHILE emailAddress does not contain "@"
        OUTPUT "Invalid address – please re-enter"
        emailAddress ← USERINPUT
ENDWHILE
OUTPUT "Thank you"
```

- The condition is tested at the beginning of the loop
- How many times will the loop be executed if the user enters bht@me.com?



Flowcharting While … Endwhile

```
emailAddress ← USERINPUT
WHILE emailAddress does not contain "@"
    OUTPUT "Invalid address – please re-enter"
    emailAddress ← USERINPUT
ENDWHILE
OUTPUT "Thank you"
```

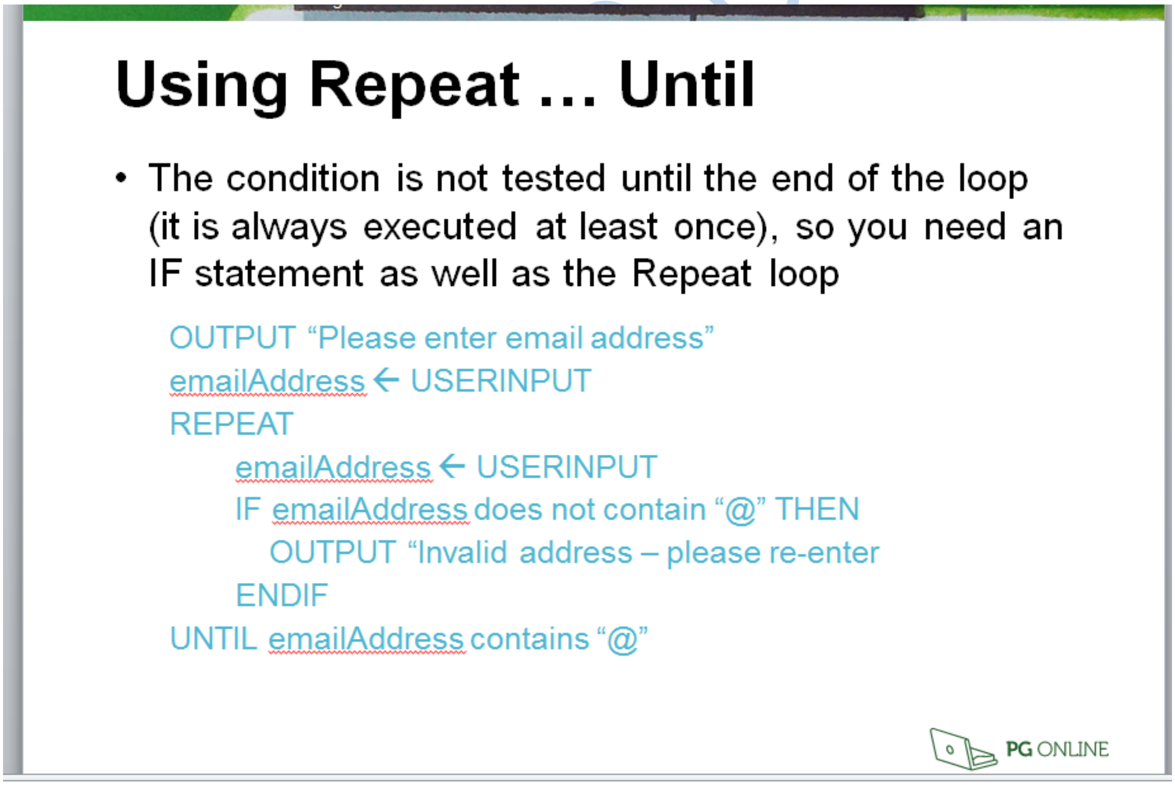


REPEAT … UNTIL

- Use this when you want to execute the loop **until** a certain condition is true
- The condition is tested at the end of the loop
- Can you rewrite this algorithm using a REPEAT loop instead of a while loop?

```
emailAddress ← USERINPUT
WHILE emailAddress does not contain "@"
        OUTPUT "Invalid address – please re-enter"
        emailAddress ← USERINPUT
ENDWHILE
```

PG ONLINE



Using Repeat … Until

- The condition is not tested until the end of the loop (it is always executed at least once), so you need an IF statement as well as the Repeat loop

```
OUTPUT "Please enter email address"
emailAddress ← USERINPUT
REPEAT
    emailAddress ← USERINPUT
    IF emailAddress does not contain "@" THEN
      OUTPUT "Invalid address – please re-enter
    ENDIF
UNTIL emailAddress contains "@"
```

PG ONLINE

1 Developing algorithms using Pseudocode

1. Write a pseudocode algorithm which asks the user to enter a password. If the user enters “NotAtHome!” then print “Welcome”.

   If they enter a different password, display “Wrong password”. [4]

2. The following algorithm accepts 3 integer values from the user, and then prints the maximum value entered. Some lines are missing.

   Complete the program by inserting instructions after Line 4. [4]

```
1.  OUTPUT (“Please enter 3 values. “)
2.  Input (x, y, z)
3.  MaxValue ← x
4.  IF y > MaxValue THEN
        MaxValue ← y
```

3. Write an algorithm for a program which inputs the lengths **a**, **b** and **c** of the three sides of a triangle. The program then determines whether the triangle is right-angled and prints out a message to say whether or not the triangle is right angled. You may assume that **a** is the longest side. The triangle is right-angled if $a^2 = b^2 + c^2$ [4]

[Total 12 Marks]

2 Developing algorithms using pseudocode

Task 1

1. Examine the pseudocode program given below. The operator MOD, e.g. **a MOD b** gives the remainder when integer **a** is divided by integer **b**.

   (a) Which lines show an example of the 'Sequence' programming construct?

   (b) Which lines show an example of the 'Selection' programming construct?

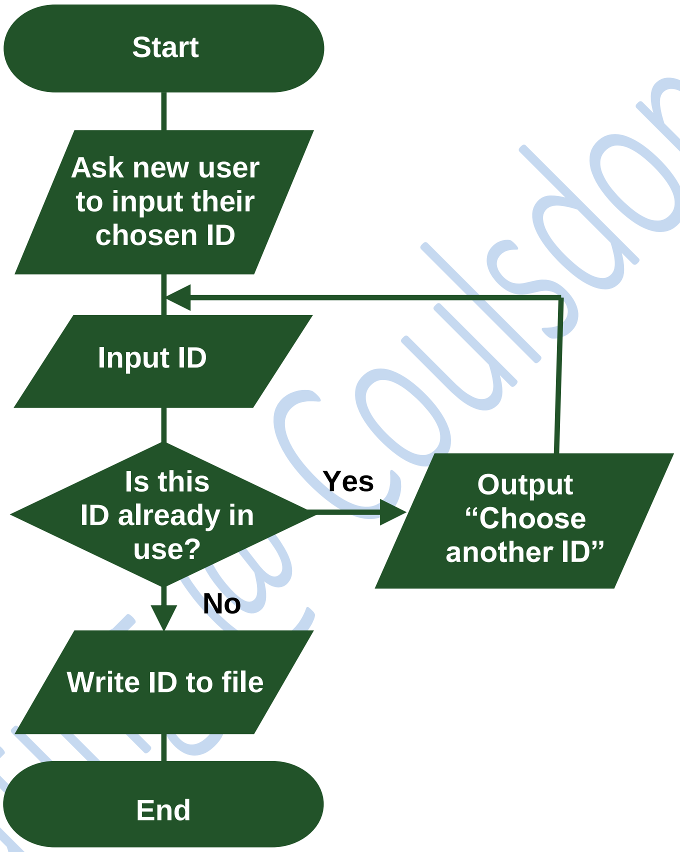
   (c) There are two examples of iteration statements in the program. On which lines do each of the 'Iteration' programming construct begin and end?

   (d) If the user enters 1 and 10 for the first and last numbers in the range, what will be printed out at line 15? Which numbers is the program counting?

```
1.   OUTPUT "This program prints selected numbers in in given range."
2.   anotherGo = "Yes"
3.   WHILE anotherGo = "Yes"
4.         OUTPUT "Please enter the first number in your chosen range."
5.         lowNumber ← USERINPUT
6.         OUTPUT "Please enter the last number in your chosen range."
7.         highNumber ← USERINPUT
8.         x = 0
9.         FOR count = lowNumber TO highNumber
10.          IF count MOD 5 <> 0) AND (count MOD 7<>0)  THEN
11.             OUTPUT (count)
12.             x = x + 1
13.          ENDIF
14.        ENDFOR
```

```
    OUTPUT x, "numbers"
    OUTPUT "Another go?"
    anotherGo ← USERINPUT
ENDWHILE
```

2. Here is a flowchart for an algorithm which asks a user to enter a new ID.

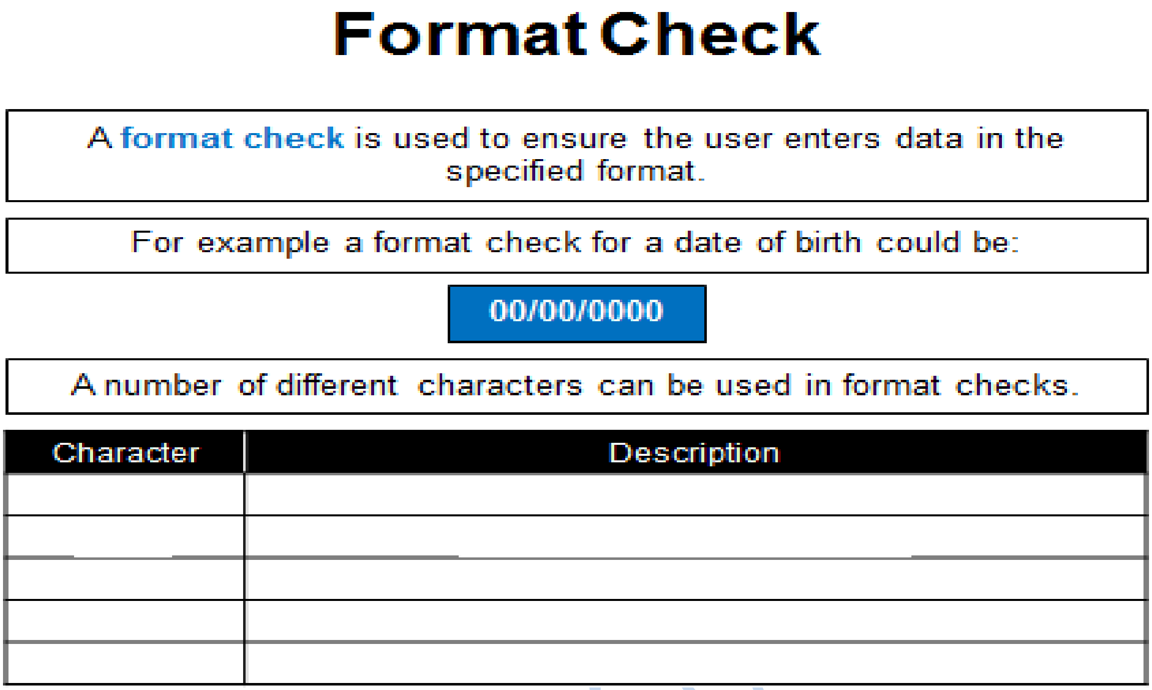

Start
Ask new user to input their chosen ID
Input ID
Is this ID already in use?
Yes
Output "Choose another ID"
No
Write ID to file
End

Write pseudocode equivalent to this flowchart. You can omit the detail of how to check whether the ID is already in use. Test it with the statement

IF IDinUse ….

3. Write a pseudocode algorithm which inputs numeric scores and outputs how many of them are over 100. The end of the data is signalled by a user input of -1.

4. Write a pseudocode algorithm which inputs numeric scores and outputs the average score. The end of the data is signalled by a user input of -1.



Format Check

A **format check** is used to ensure the user enters data in the specified format.

For example a format check for a date of birth could be:

00/00/0000

A number of different characters can be used in format checks.

Character	Description