www.PapaCambridge.com

Booklet 3

This covers the following topics found in section 3 of the computer studies syllabus:

www.PapaCambridge.com

Booklet 3 covers the development of algorithms (both in pseudocode and flowchart form) and also introduces logic gates which is a new topic from 2011.

Introduction to Flowcharts

This section covers the use of flow diagrams (charts) in the production of algorithms. Systems flowcharts are different and these are covered in a different section (Systems analysis).

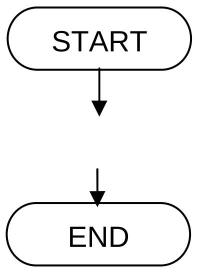
The early part of section 3.1 (i.e. top down design, structure diagrams, menus, libraries of procedures and subroutines) is covered adequately by standard text books.

This section primarily covers four areas:

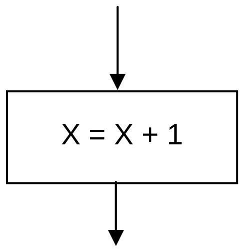
1 Common flow chart symbols
2 Writing flowcharts to solve problems
3 Dry running of flowcharts to determine its function and outputs
4 Exercises to test the above concepts

1 Common flowchart symbols

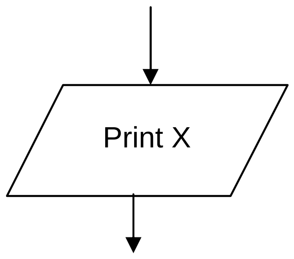
1.1 The start and end box:



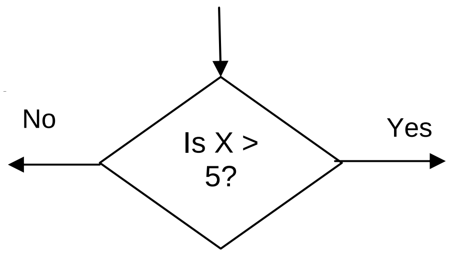
1.2 The process box:



1.3 Input/Output box:



1.4 Decision/query box:



www.PapaCambridge.com

2 Writing flowcharts to solve problems

The following five problems are also covered in section 3.2 where the algorithms are constructed using pseudocode. Candidates may choose to answer questions using either flowcharts or pseudocode but a working knowledge of both techniques is well advised.

2.1 Example 1

A town contains 5000 houses. Each house owner must pay tax based on the value of the house. Houses over \$200 000 pay 2% of their value in tax, houses over \$100 000 pay 1.5% of their value in tax and houses over \$50 000 pay 1% of their value in tax. All others pay no tax. Write an algorithm to solve this problem in the form of a flowchart.

2.2 Example 2

The following formula is used to calculate n: n = (x * x)/(1 – x). The value x = 0 is used to stop the algorithm. The calculation is repeated using values of x until the value x = 0 is input. There is also a need to check for error conditions. The values of n and x should be output. Write an algorithm to show this repeated calculation in the form of a flowchart.

2.3 Example 3

Write an algorithm in the form of a flowchart which takes temperatures input over a 100 day period (once per day) and outputs the number of days when the temperature was below 20C and the number of days when the temperature was 20C and above.

2.4 Example 4

Write an algorithm in the form of a flowchart which:

- inputs the top speeds (in km/hr) of 5000 cars
- outputs the fastest speed and the slowest speed
- outputs the average (mean) speed of all the 5000 cars

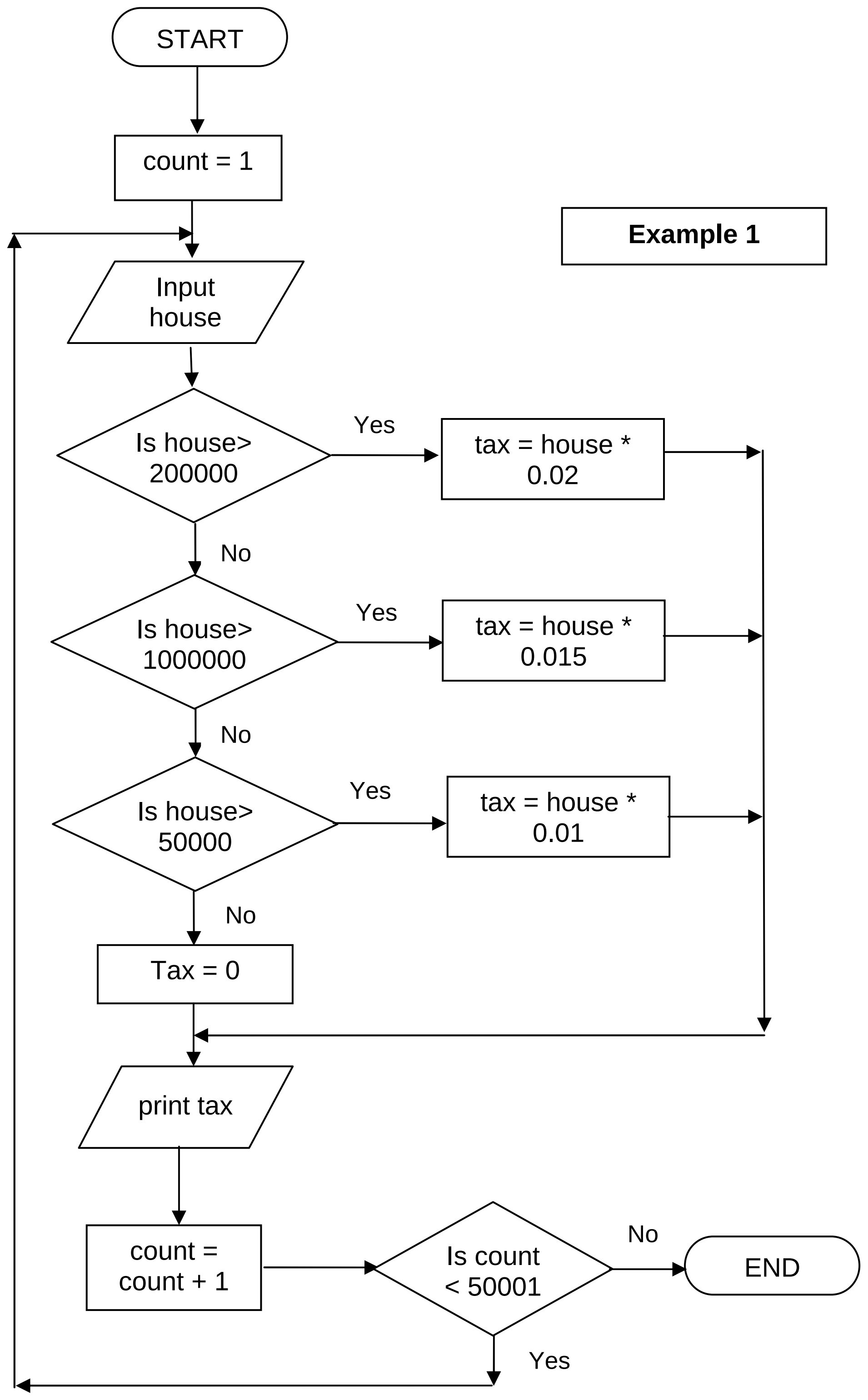
2.5 Example 5

A shop sells books, maps and magazines. Each item is identified by a unique 4 – digit code. All books have a code starting with 1, all maps have a code starting with 2 and all magazines have a code starting with 3. The code 9999 is used to end the algorithm.

Write an algorithm in the form of a flowchart which inputs the codes for all items in stock and outputs the number of books, number of maps and the number of magazines in stock. Include any validation checks needed.

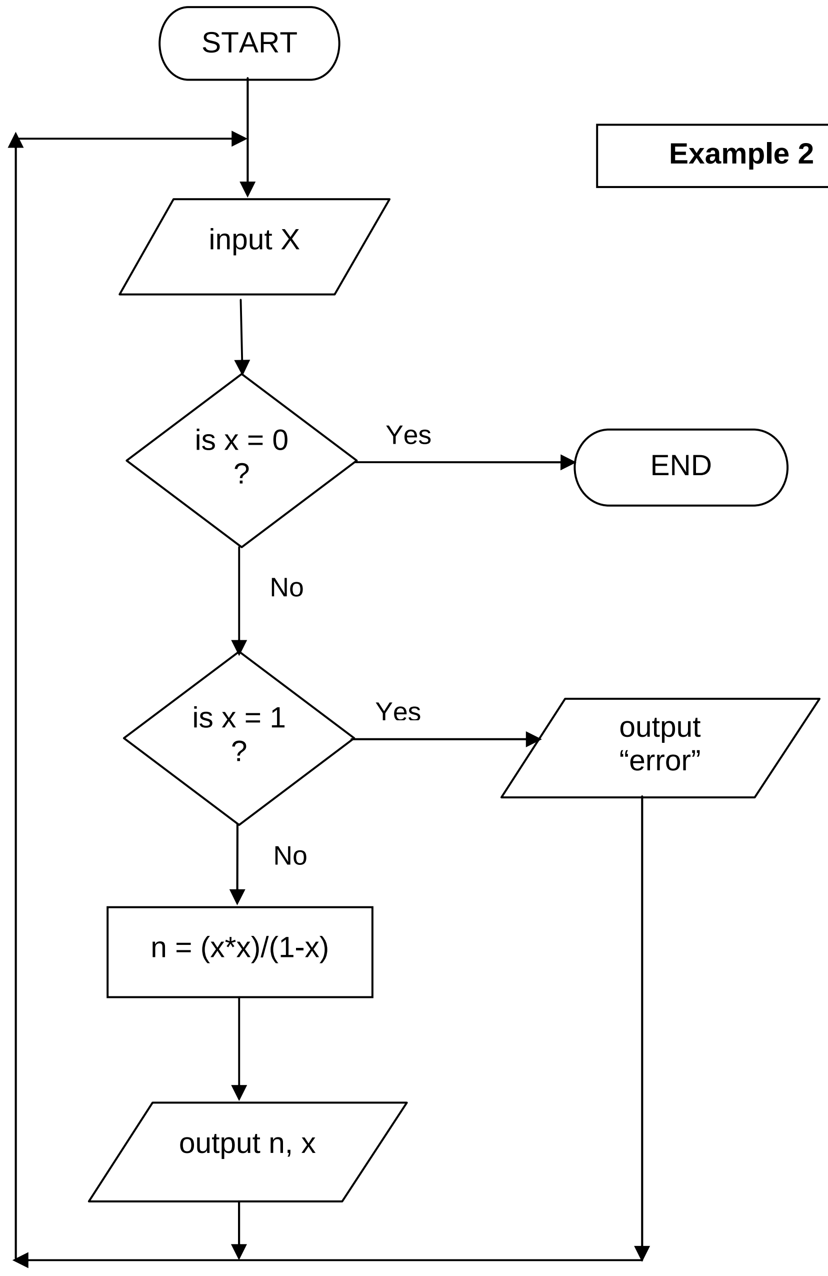
www.PapaCambridge.com

Example 1



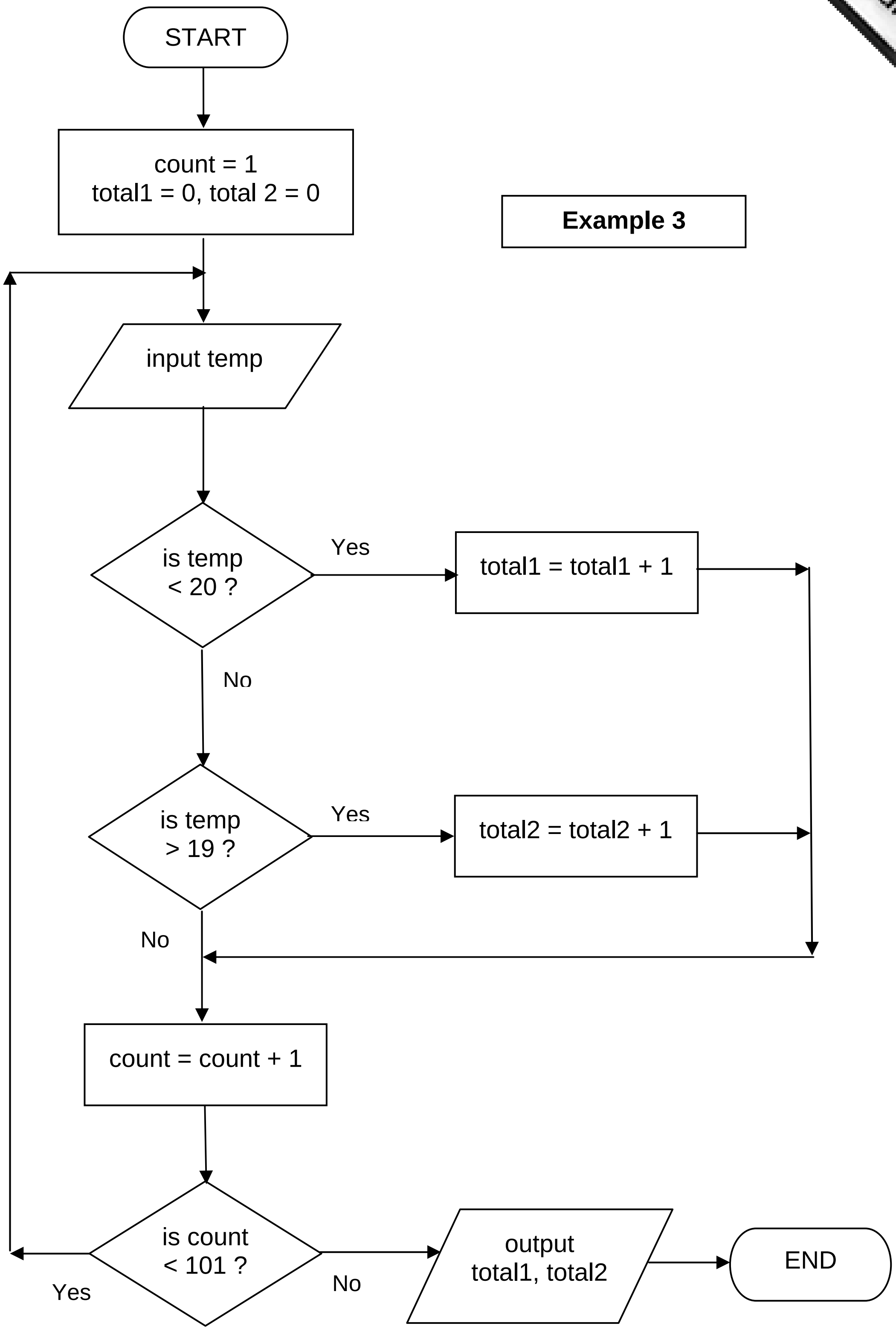
www.PapaCambridge.com

Example 2



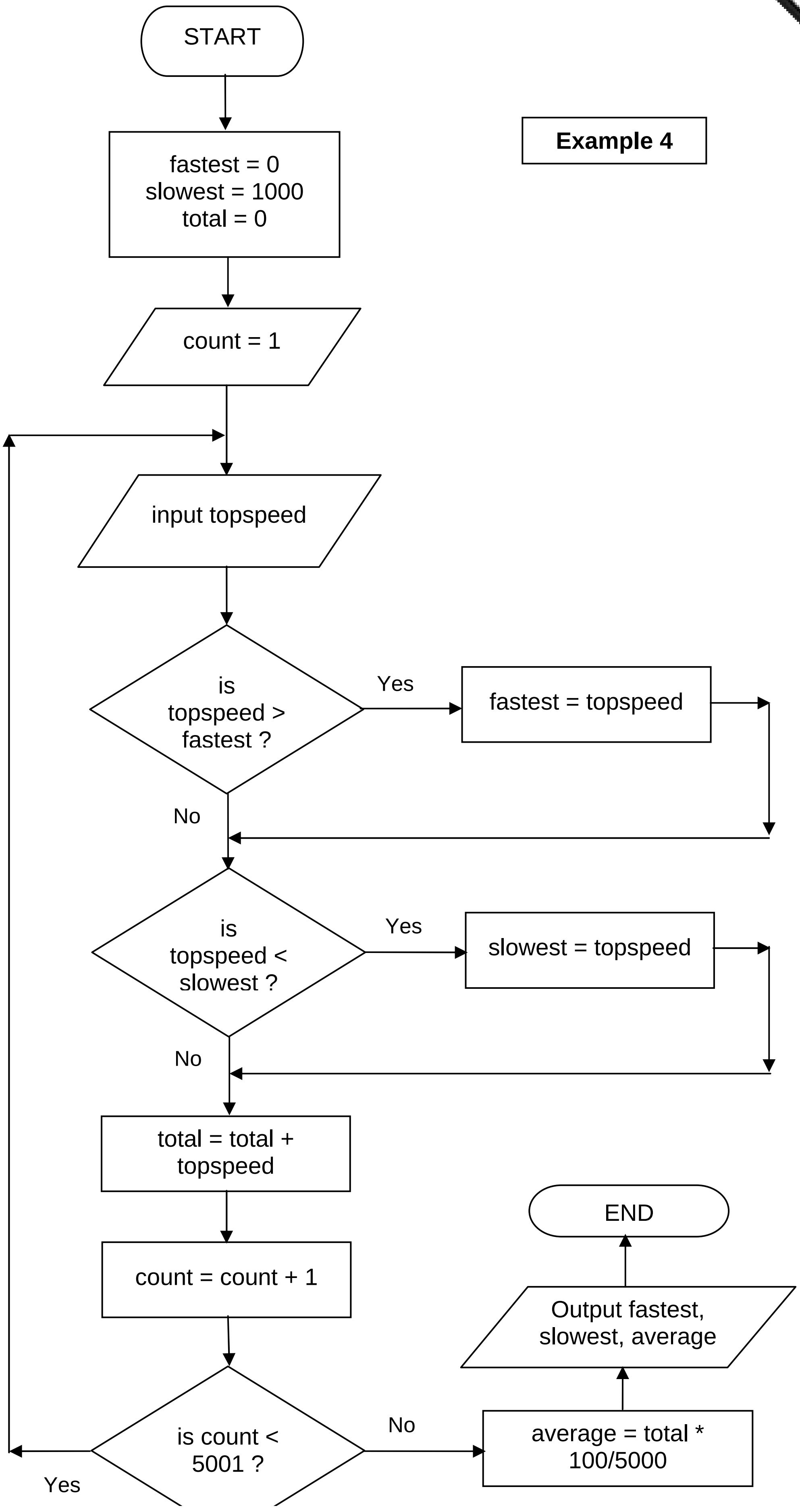
www.PapaCambridge.com

Example 3



www.PapaCambridge.com

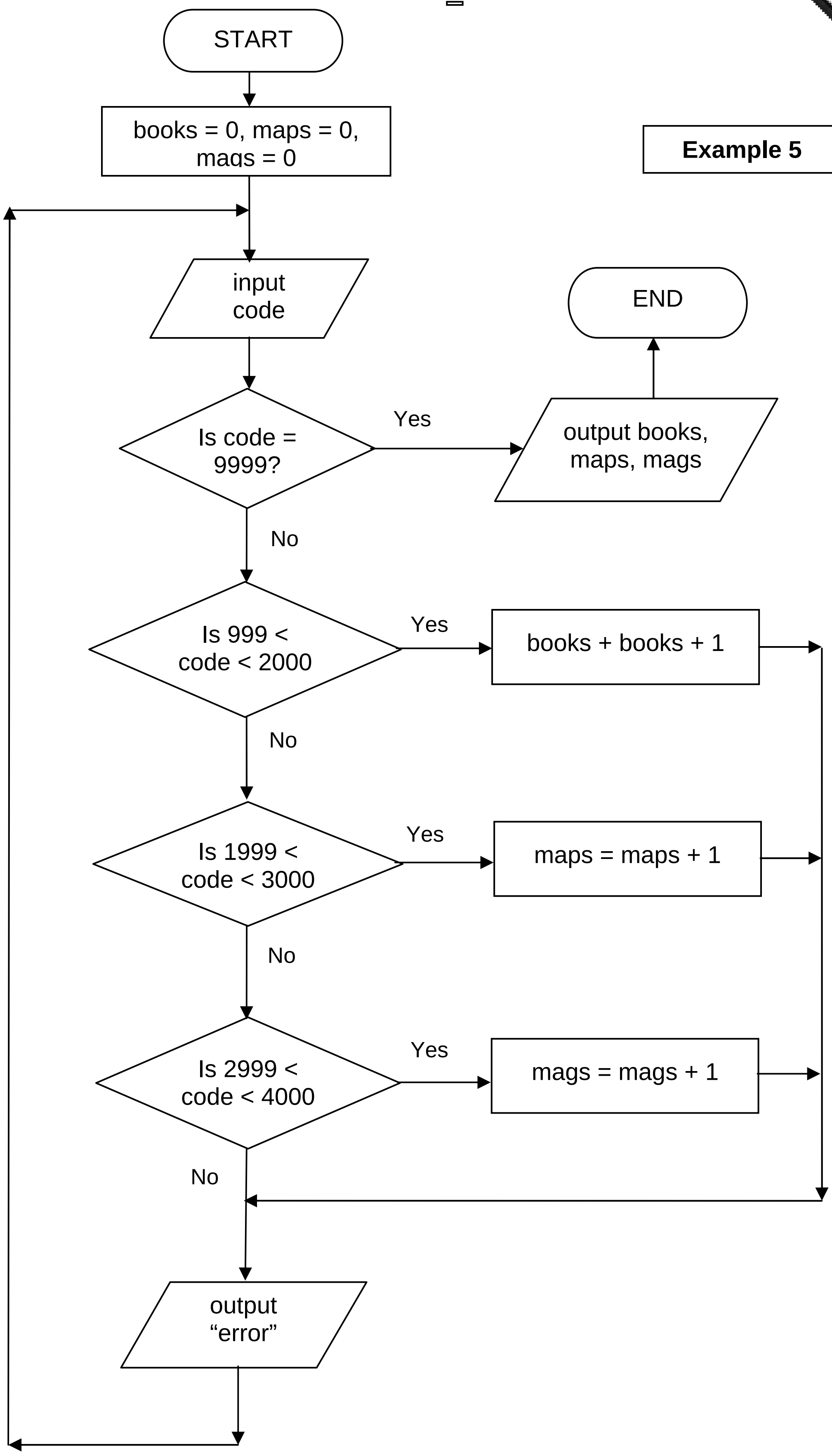
Example 4



www.PapaCambridge.com

www.PapaCambridge.com

Example 5



www.PapaCambridge.com

3 Dry running of flowcharts

Dry running of flowcharts is basically a technique to:

- determine the output for a known set of data to check it carries out the task correctly
- check on the logic of the algorithm
- determine the function of the algorithm

When dry running a flowchart it is advisable to draw up a trace table showing how variables change their values at each stage in the algorithm. The advantages of doing this are:

- if you make a mistake, it is easier to back track to where the error occurred rather than starting from the beginning again
- there is less chance of an error being made
- encourages a more logical approach

The following three examples show all stages in the dry running for the given set of input data:

3.1 Example 1

This algorithm inputs 3 numbers, each number goes through successive division by 10 until its value is less than 1. An output is produced which contains the number input and a value generated by the flowchart processing.

Data to be used: X = 85, 3190, -40

3.2 Example 2

This algorithm inputs 5 values and outputs how many input numbers were negative and how many were positive.

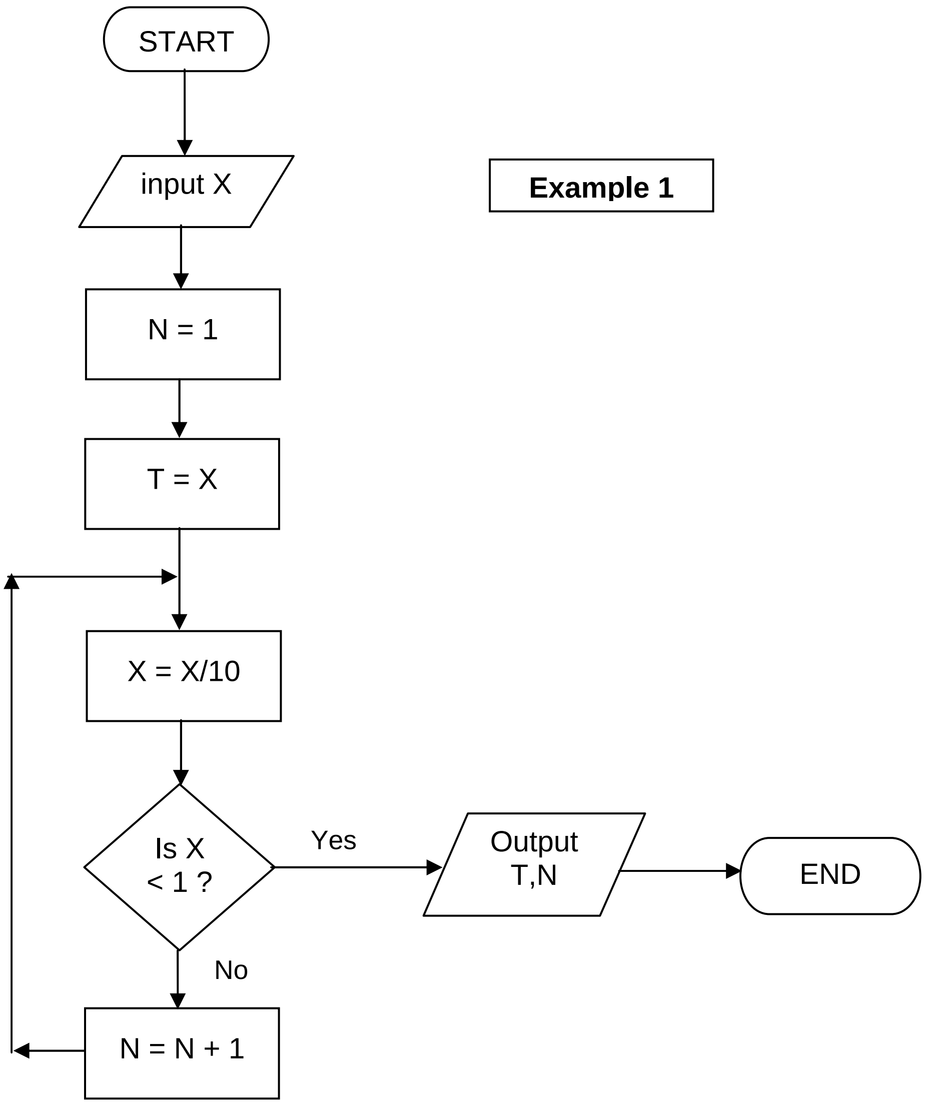
Data to be used: N = 1, -5, 2, -8, -7

3.3 Example 3

This algorithm inputs the number of hours of sunshine recorded each day for a week (7 days). The output is the highest value for hours of sunshine and the average (mean) value for the numbers of hours of sunshine per day.

Data to be used: hours = 9.0, 7.8, 1.2, 4.5, 10.0, 6.4, 3.1

www.PapaCambridge.com

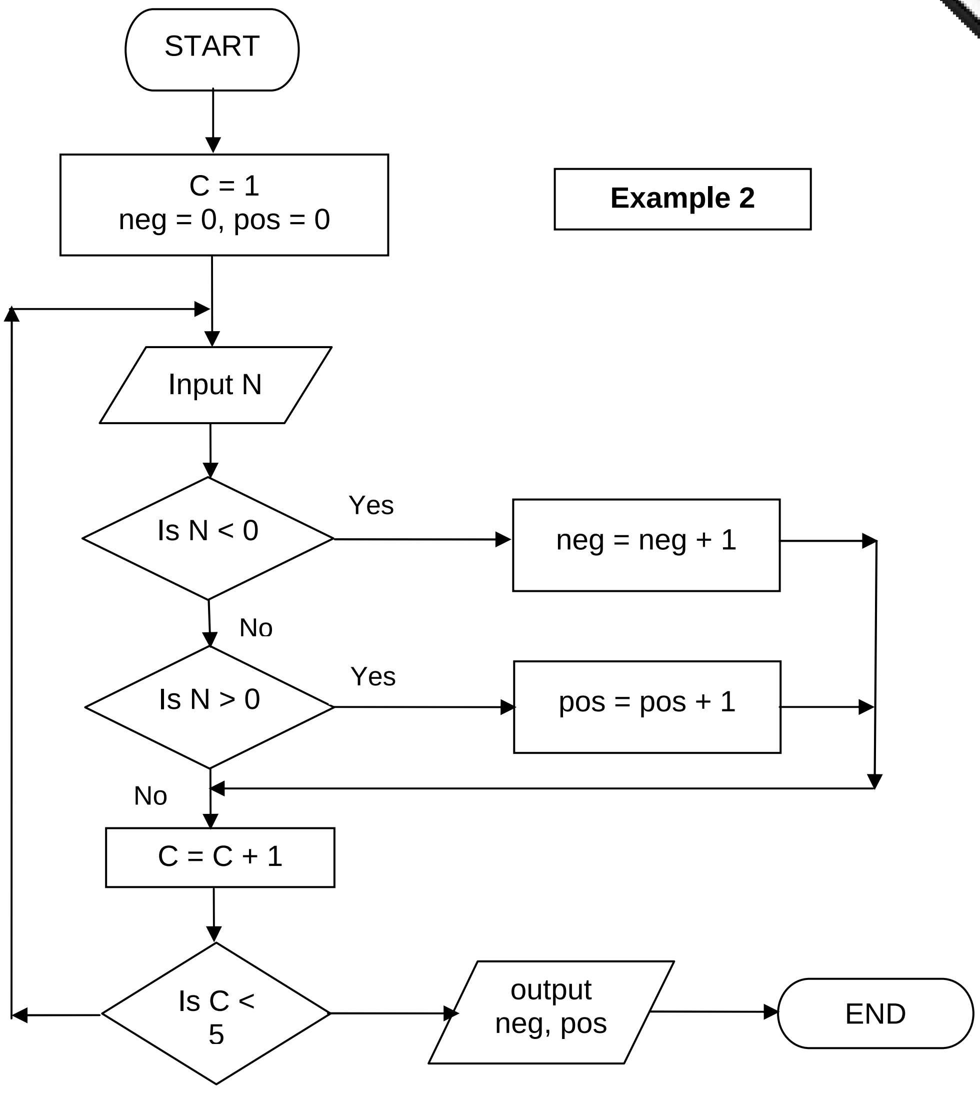


Example 1

Trace Table

X	N	T	Output T, N
85 8.5 0.85	1 2	85	 **85, 2**
3190 319 31.9 3.19 0.319	1 2 3 4	3190	 **3190, 4**
-40 -4	1	-40	 **-40, 1**

www.PapaCambridge.com

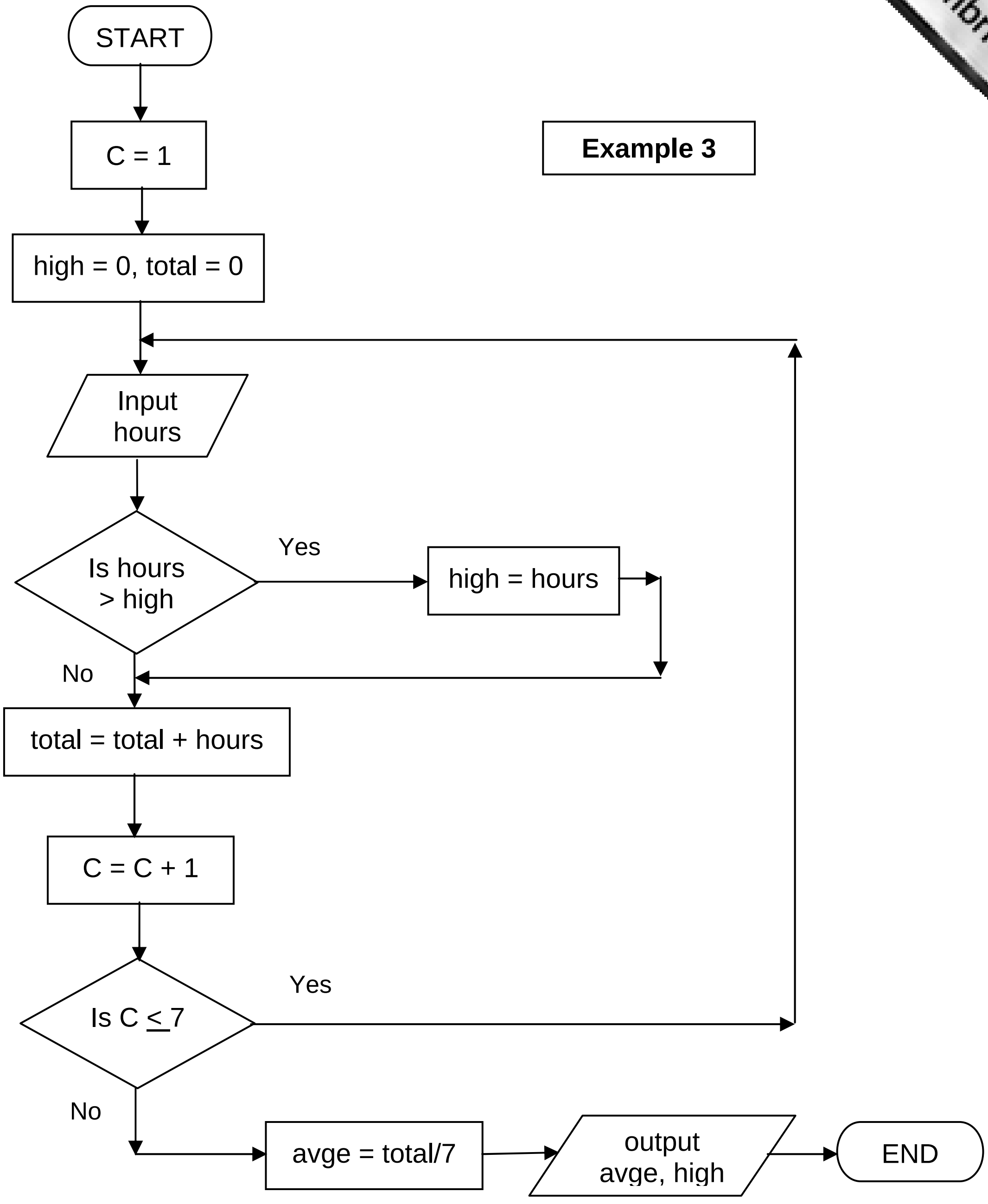


Trace Table

C	N	neg	pos	Output neg, pos
1	1	0	0	
2	-5	1	1	
3	2	2	2	
4	-8	3		
5	-7			
6				**3, 2**

www.PapaCambridge.com

Example 3



Trace Table

C	hours	high	total	avge	Output avge, high
1	9	0	0	6	
2	7.8	9	9		
3	1.2	10	16.8		
4	4.5		18		
5	10		22.5		
6	6.4		32.5		
7	3.1		38.9		
8			42		
					6, 10

www.PapaCambridge.com

4 Problems

Questions 1 to 7 are problems which require an algorithm to be written in the form of a flowchart. Questions 8 to 10 require a trace table to be written and find the expected output for the given set of data. The answers to these questions can be found in booklet 6.

(1) Regis lives in Brazil and often travels to USA, Europe and Japan. He wants to be able to convert Brazilian Reais into US dollars, European euros and Japanese yen. The conversion formula is:

currency value = number of Reais X conversion rate

For example, if Regis is going to USA and wants to take 1000 Reais (and the exchange rate is 0.48) then he would input USA, 1000 and 0.48 and the output would be: 480 US dollars.

Write an algorithm, using a flowchart, which inputs the country he is visiting, the exchange rate and the amount in Brazilian Reais he is taking. The output will be value in foreign currency and the name of the currency.

(2) As part of an experiment, a school measured the heights (in metres) of all its 500 students.

Write an algorithm, using a flowchart, which inputs the heights of all 500 students and outputs the height of the tallest person and the shortest person in the school.

(3) A geography class decide to measure daily temperatures and hours of sunshine per day over a 12 month period (365 days)

Write an algorithm, using a flowchart, which inputs the temperatures and hours of sunshine for all 365 days, and finally outputs the average (mean) temperature for the year and the average (mean) number of hours per day over the year.

(4) A small shop sells 280 different items. Each item is identified by a 3 – digit code. All items that start with a zero (0) are cards, all items that start with a one (1) are sweets, all items that start with a two (2) are stationery and all items that start with a three (3) are toys.

Write an algorithm, using a flowchart, which inputs the 3 – digit code for all 280 items and outputs the number of cards, sweets, stationery and toys.

www.PapaCambridge.com

(5) A company are carrying out a survey by observing traffic at a road junctic Each time a car, bus or lorry passed by the road junction it was noted down.

10 000 vehicles were counted during the survey.

Write an algorithm, using an algorithm, which:

- inputs all 10000 responses
- outputs the number of cars, buses and lorries that passed by the junction during the survey
- outputs the number of vehicles that ***weren't*** cars, buses or lorries during the survey

(6) Speed cameras read the time a vehicle passes a point (A) on the road and then reads the time it passes a second point (B) on the same road (points A and B are 100 metres apart). The speed of the vehicle is calculated using:

$$\text{speed} = \frac{100}{(\text{time at point B} - \text{time at point A})} \quad \text{(metres/sec)}$$

The maximum allowed speed is 100 kilometres per hour. 500 vehicles were monitored using these cameras over a 1 hour period.

Write an algorithm, using a flowchart, which:

- inputs the start time and end time for the 500 vehicles that were monitored
- calculate the speed for each vehicle using the formula above
- outputs the speed for each vehicle and also a message if the speed exceeded 100 km/hour
- output the highest speed of all the 500 vehicles monitored

(7) There are ten stations on a railway line:

1 ------ 2 ------ 3 ------ 4 ------ 5 ------ 6 ------ 7 ------ 8 ------ 9 ------ 10

The train travels in both directions (i.e. from 1 to 10 and then from 10 to 1). The fare between each station is $2.

A passenger inputs the number of the station at the start of his journey and the number of the destination station and the fare is calculated (e.g if a passenger gets on a station 3 and his destination is station 9 his fare will be $12). The calculation must take into account the direction of the train (e.g. a passenger getting on at station 7 and getting off at station 1 will also pay $12 and not a negative value!!).

A discount of 10% is given if 3 or more passengers are travelling together.

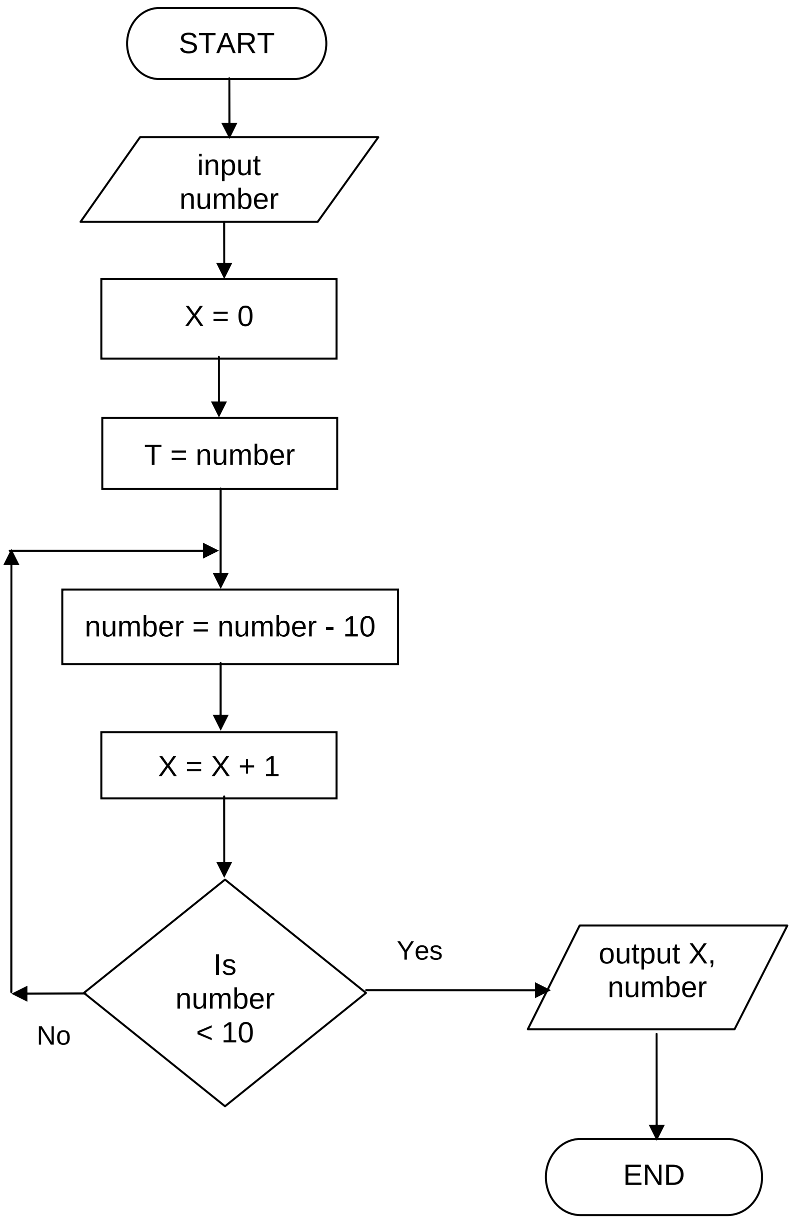
www.PapaCambridge.com

Write an algorithm, using a flowchart, which:

- inputs the number of passengers travelling
- inputs the station number of the starting point and the station number of the destination
- calculates the total fare taking into account the direction of travel
- calculates any discount due
- outputs the cost of the tickets and prints the tickets

(8) Draw the trace table and determine the output from the following flowchart using the following data:

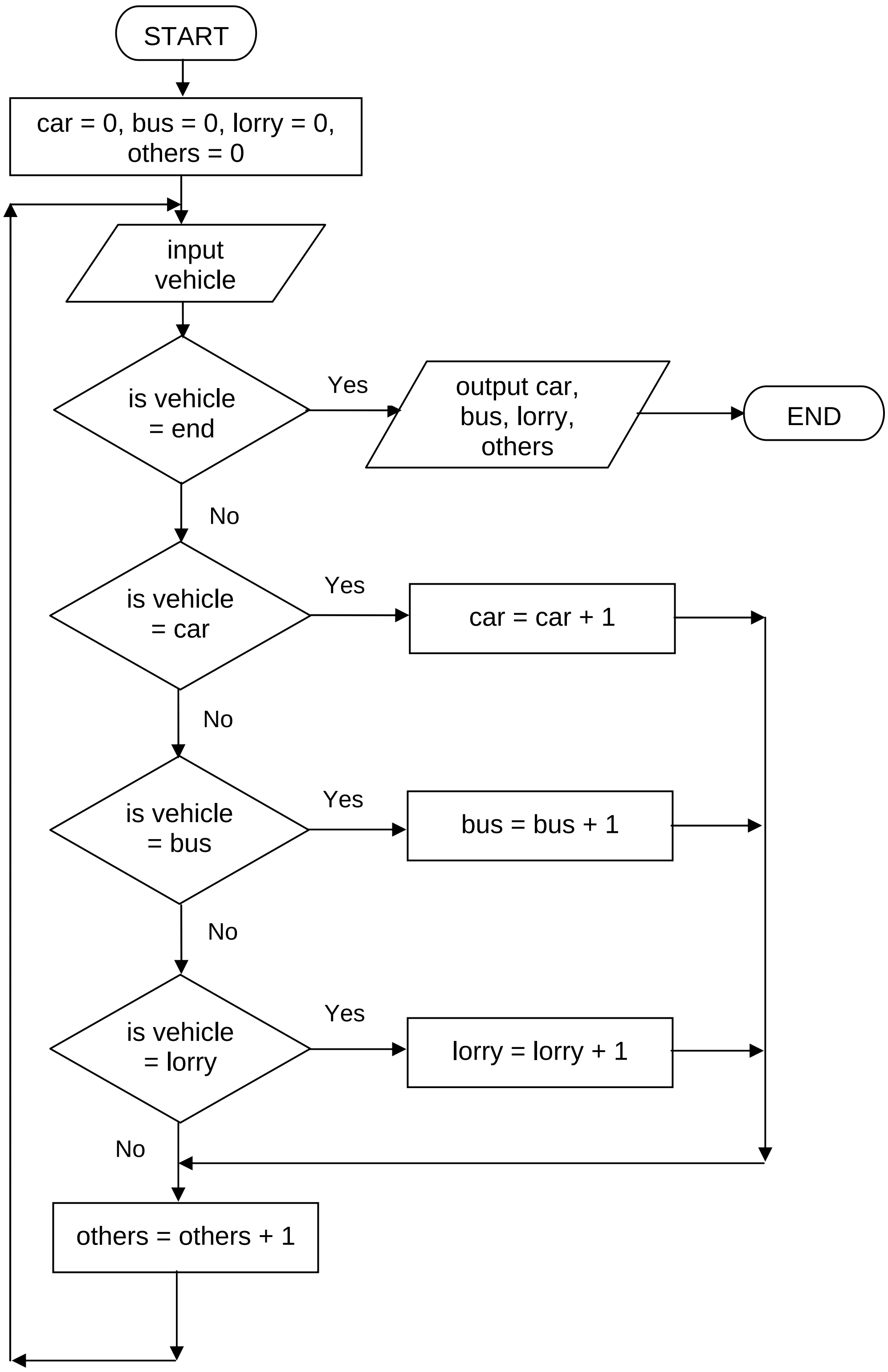
number = 45, -2, 20.5



www.PapaCambridge.com

(9) Draw the trace table and determine the output from the following flowch using the following data (NOTE: input of the word “end” stops the program and outputs results of the survey):

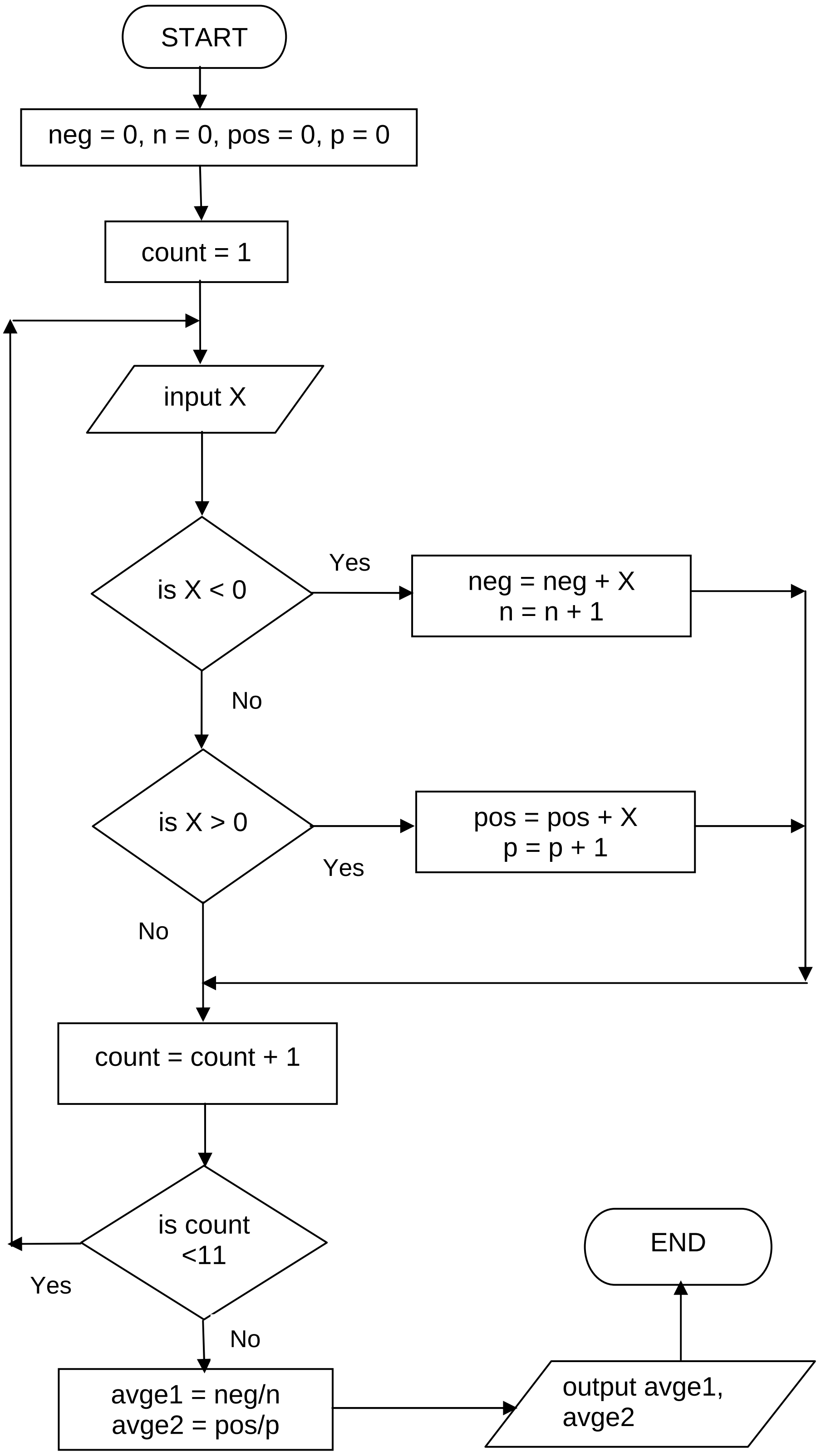
vehicle = car, car, lorry, bus, van, van, car, car, bus, car, end



www.PapaCambridge.com

(10) Draw the trace table and determine the output from the following flowch using the following data:

X = 5, -3, 0, -3, 7, 0, 6, -11, -7, 12



www.PapaCambridge.com

Introduction tp pseudocode

This section covers the use of pseudocode in the production of algorithms. Candidates should use standard computing text books to find out information on the features of programming languages (high level and low level), interpreters, compilers, assemblers, user documentation and technical documentation.

No specific programming language is referred to; development of algorithms using pseudocode uses generic descriptions of looping, branching, data manipulation, input/output, totalling and counting techniques.

The section is broken down into four areas:

1 description of common pseudocode terms
2 writing algorithms using pseudocode
3 finding errors in sections of pseudocode
4 exercises

1 Common pseudocode terms

1.1 counting

Counting in 1s is quite simple; use of the statement **count = count + 1** will enable counting to be done (e.g. in controlling a *repeat* loop). The statement literally means: *the (new) count = the (old) count + 1*

It is possible to count in any increments just by altering the numerical value in the statement (e.g. count = count – 1 counts backwards)

1.2 totalling

To add up a series numbers the following type of statement should be used:

total = total + number

This literally means *(new) total = (old) total + value of number*

1.3 input/output

Input and output indicated by the use of the terms **input number**, **output total**, **print total**, **print "result is" x** and so on.

1.4 branching

There are two common ways of branching:

case of ….. otherwise …... endcase

if ….. then ….. else ….. endif

www.PapaCambridge.com

case of

```
case number of
1:  x = x + 1
2:  y = y + 1
otherwise print "error"
endcase
```

if … then

```
if number = 1 then x = x + 1
   else if number = 2 then y = y + 1
       else print "error"
endif
endif
```

1.5 loops

There are three common ways of performing a looping function:

for … to … next, while … endwhile and **repeat … until**

The following example input 100 numbers and finds the total of the 100 numbers and outputs this total. All three looping techniques are shown:

for … to

```
for count = 1 to 100
   input number
   total = total + number
next
print total
```

while … endwhile

```
while count < 101
  input number
    total = total + number
    count = count + 1
endwhile
print total
```

repeat … until

```
repeat
  input number
    total = total + number
    count = count + 1
until count = 100
print total
```

www.PapaCambridge.com

2 Writing algorithms using pseudocode

The following five examples use the above pseudocode terms. These are the same problems discussed in section 3.1 using flow charts – both methods are acceptable ways of representing an algorithm.

2.1 Example 1

A town contains 5000 houses. Each house owner must pay tax based on the value of the house. Houses over $200 000 pay 2% of their value in tax, houses over $100 000 pay 1.5% of their value in tax and houses over $50 000 pay 1% of their value in tax. All others pay no tax. Write an algorithm to solve the problem using pseudocode.

```
for count = 1 to 5000

    input house

            if house > 50 000 then tax = house * 0.01

                    else if house > 100 000 then tax = house * 0.015

                    else if house > 200 000 then tax = house * 0.02

            else tax = 0

    print tax

 next
```

Notes: (1) a **while** loop or a **repeat** loop would have worked just as well
(2) the use of **endif** isn't essential in the pseudocode

For example,

```
count = 0
while count < 5001
    input house
        if house > 50 000 then tax = house * 0.01
            else if house > 100 000 then tax = house * 0.015
            else if house > 200 000 then tax = house * 0.02
            else tax = 0
                endif
                endif
        endif
    print tax
    count = count + 1
endwhile
```

www.PapaCambridge.com

EXERCISE: Re-write the above algorithm using a **repeat** loop and mod the **if … then … else** statements to include both parts of the house price range.

(e.g. **if** house > 50000 and house <= 100000 **then** tax = house * 0.01)

2.2 Example 2

The following formula is used to calculate n: n = x * x/(1 – x)

The value x = 0 is used to stop the algorithm. The calculation is repeated using values of x until the value x = 0 is input. There is also a need to check for error conditions. The values of n and x should be output. Write an algorithm to show this repeated calculation using pseudocode.

NOTE: It is much easier in this example to input x first and then loop round doing the calculation until eventually x = 0. Because of this, it would be necessary to input x twice (i.e. inside the loop and outside the loop). If input x occurred only once it would lead to a more complicated algorithm.

(Also note in the algorithm that <> is used to represent ≠).

A **while** loop is used here, but a **repeat** loop would work just as well.

```
input x

    while x <> 0  do

        if x = 1 then print “error”

            else n = (x * x)/(1 – x)

            print n, x

        endif

        input x

endwhile
```

www.PapaCambridge.com

2.3 Example 3

Write an algorithm using pseudocode which takes temperatures input over a 100 day period (once per day) and output the number of days when the temperature was below 20C and the number of days when the temperature was 20C or above.

(NOTE: since the number of inputs is known, a **for … to** loop can be used. However, a **while** loop or a **repeat** loop would work just as well).

```
total1 = 0: total2 = 0

for days = 1 to 100

    input temperature

        if temperature < 20 then total1 = total1 + 1

            else total2 = total2 + 1

        endif

next

print total1, total2
```

This is a good example of an algorithm that could be written using the **case** construct rather than **if … then … else**. The following section of code replaces the statements ***if** temperature < 20 **then …… endif**:*

```
        case temperature of

            1:  total1 = total1 + 1

            2:  total2 = total2 + 1

        endcase
```

2.4 Example 4

Write an algorithm using pseudocode which:

- inputs the top speeds of 5000 cars
- outputs the fastest speed and the slowest speed
- outputs the average speed of all the 5000 cars

(NOTE: Again since the actual number of data items to be input is known any one of the three loop structures could be used. It is necessary to set values for the fastest (usually set at zero) and the slowest (usually set at an unusually high value) so that each input can be compared. Every time a value is input which > the value stored in fastest then this input value replaces the existing value in fastest; and similarly for slowest).

www.PapaCambridge.com

```
fastest = 0: count = 0
slowest = 1000
repeat
    input top_speed
    total = total + top_speed
        if top_speed > fastest then fastest = top_speed
            if top_speed < slowest then slowest = top_speed
            endif
        endif
    count + count + 1
until count = 5000
average = total * 100/5000
print fastest, slowest, average
```

2.5 Example 5

A shop sells books, maps and magazines. Each item is identified by a unique 4 – digit code. All books have a code starting with a 1, all maps have a code starting with a 2 and all magazines have a code beginning with a 3. The code 9999 is used to end the program.

Write an algorithm using pseudocode which input the codes for all items in stock and outputs the number of books, maps and magazine in stock. Include any validation checks necessary.

(NOTE: A 4-digit code implies all books have a code lying between 1000 and 1999, all maps have a code lying between 2000 and 2999 and all magazines a code lying between 3000 and 3999. Anything outside this range is an error)

www.PapaCambridge.com

```
books = 0: maps = 0: mags = 0

repeat

    input code

         if code > 999 and code < 2000 then books = books + 1

           else if code > 1999 and code < 3000 then maps = maps + 1

           else if code > 2999 and code < 4000 then mags = mags + 1

           else print “error in input”

           endif:endif:endif

until code = 9999

print books, maps, mags
```

(NOTE: A function called INT(X) is useful in questions like this. This returns the integer (whole number) part of X e.g. if X = 1.657 then INT(X) = 1; if X = 6.014 then INT(X) = 6 etc. Using this function allows us to use the **case** statement to answer this question:

```
books = 0: maps = 0: mags = 0
    repeat
      input code
        x = INT(code/1000)            * divides code by 1000 to give a
          case x of                   * number between 0 and 9
            1: books = books + 1
            2: maps = maps + 1
            3: mags = mags + 1
          otherwise print “error”
          endcase
    until code = 9999
    print books, maps, mags
```

this is probably a more elegant but more complex solution to the problem)

www.PapaCambridge.com

4 PROBLEMS

Questions 1 to 3 contain sections of pseudocode which contain errors. Locate the errors and suggest the correct coding. Questions 4 to 10 are problems which require an algorithm to be written in pseudocode – there is “no right answer” here; as long as the pseudocode works then the solution is acceptable.

(1) The following section of pseudocode inputs 1000 numbers and then outputs how many were negative, how many were positive and how many were zero.

Locate the 3 errors and suggest a corrected piece of code.

```
1   negative = 1: positive = 1
2   for x = 0 to 1000
3       input number
4           if number < 0 then negative = negative + 1
5           if number > 0 then positive = positive + 1
6           endif
7           endif
8   next
9   print negative, positive
```

(2) The following section of pseudocode inputs rainfall (in cm) for 500 days and outputs the average rainfall and the highest rainfall input.

Locate the 3 errors and suggest a corrected piece of code.

```
1   highest = 1000
2   days = 1
3       while days > 0
4           input rainfall
5               if rainfall > highest then highest = rainfall
6               endif
7       total = total + rainfall
8       days = days + 1
9   average = total/500
10  endwhile
11  print average, highest
```

(3) The following section of pseudocode inputs a number, n, multiplies together 1 x 2 x 3 x ……. x n, calculates input number/sum and outputs result of the calculation.

Locate the 3 errors and suggest a corrected piece of code.

```
1   input n
2       for mult = 1 to n
3           sum = 0
4           sum = sum * mult
5           result = n/sum
6       next
7   print result
```

www.PapaCambridge.com

(4) Regis lives in Brazil and often travels to USA, Europe and Japan. He wa to be able to convert Brazilian Reais into US dollars, European euros and Japanese yen. The conversion formula is:

currency value = number of Reais X conversion rate

For example, if Regis is going to USA and wants to take 1000 Reais (and the exchange rate is 0.48) then he would input USA, 1000 and 0.48 and the output would be: 480 US dollars.

Write an algorithm, using pseudocode, which inputs the country he is visiting, the exchange rate and the amount in Brazilian Reais he is taking. The output will be value in foreign currency and the name of the currency.

(5) As part of an experiment, a school measured the heights (in metres) of all its 500 students.

Write an algorithm, using pseudocode, which inputs the heights of all 500 students and outputs the height of the tallest person and the shortest person in the school.

(6) A geography class decide to measure daily temperatures and hours of sunshine per day over a 12 month period (365 days)

Write an algorithm, using pseudocode, which inputs the temperatures and hours of sunshine for all 365 days, and finally outputs the average (mean) temperature for the year and the average (mean) number of hours per day over the year.

(7) A small shop sells 280 different items. Each item is identified by a 3 – digit code. All items that start with a zero (0) are cards, all items that start with a one (1) are sweets, all items that start with a two (2) are stationery and all items that start with a three (3) are toys.

Write an algorithm, using pseudocode, which inputs the 3 – digit code for all 280 items and outputs the number of cards, sweets, stationery and toys.

(8) A company are carrying out a survey by observing traffic at a road junction. Each time a car, bus, lorry or other vehicle passed by the road junction it was noted down.

10 000 vehicles were counted during the survey.

Write an algorithm, using pseudocode, which:

- inputs all 10000 responses
- outputs the number of cars, buses and lorries that passed by the junction during the survey
- outputs the number of vehicles that ***weren't*** cars, buses or lorries during the survey

www.PapaCambridge.com

(9) Speed cameras read the time a vehicle passes a point (A) on the road a then reads the time it passes a second point (B) on the same road (points A and B are 100 metres apart). The speed of the vehicle is calculated using:

$$\text{speed} = \frac{100}{(\text{time at point B} - \text{time at point A})} \quad \text{(metres/sec)}$$

The maximum allowed speed is 100 kilometres per hour. 500 vehicles were monitored using these cameras over a 1 hour period.

Write an algorithm, using pseudocode, which:

- inputs the start time and end time for the 500 vehicles that were monitored
- calculate the speed for each vehicle using the formula above
- outputs the speed for each vehicle and also a message if the speed exceeded 100 km/hour
- output the highest speed of all the 500 vehicles monitored

(10) There are ten stations on a railway line:

1 ------ 2 ------ 3 ------ 4 ------ 5 ------ 6 ------ 7 ------ 8 ------ 9 ------ 10

The train travels in both directions (i.e. from 1 to 10 and then from 10 to 1). The fare between each station is $2.

A passenger inputs the number of the station at the start of his journey and the number of the destination station and the fare is calculated (e.g if a passenger gets on a station 3 and his destination is station 9 his fare will be $12). The calculation must take into account the direction of the train (e.g. a passenger getting on at station 7 and getting off at station 1 will also pay $12 and not a negative value!!).

A discount of 10% is given if 3 or more passengers are travelling together.

Write an algorithm, using pseudocode, which:

- inputs the number of passengers travelling
- inputs the station number of the starting point and the station number of the destination
- calculates the total fare taking into account the direction of travel
- calculates any discount due
- outputs the cost of the tickets and prints the tickets

www.PapaCambridge.com

Introduction to Logic

Many electronic circuits operate using binary logic gates. Logic gates basically process signals which represent ***true*** or ***false*** or the equivalent i.e. ON or OFF, 1 or 0

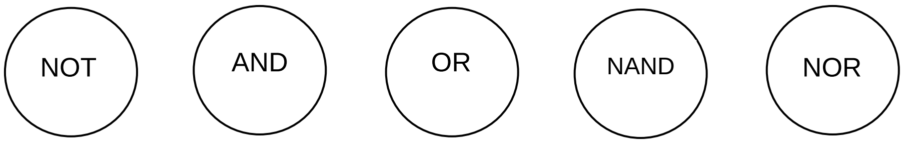
Whilst there are a number of logic gates, only the five simplest are covered in this booklet: NOT gate, AND gate, OR gate, NAND gate and NOR gate.

The following notes describe the function of all five gates, how to produce truth tables, how to design networks using logic gates, and how to determine the output from a logic network.

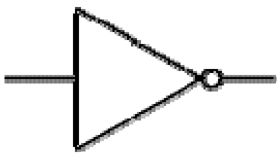
The five main logic gates

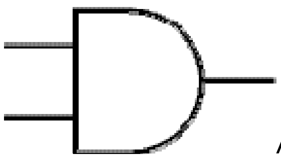
The most common symbols used to represent logic gates are shown below. To avoid confusion the graphical representations will be used in exam questions but candidates may use either set of symbols when answering questions.

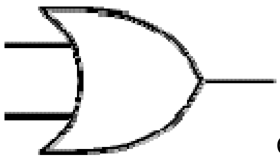
1 simple graphical representations

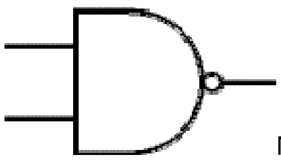


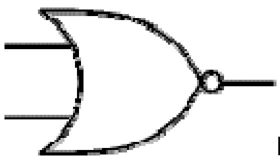
2 MIL symbols used to represent logic gates


NOT gate


AND gate


OR gate


NAND gate


NOR gate

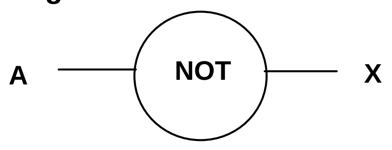
Truth tables are used to show logic gate functions (refer to next section). The NOT gate has only one input (and one output) but the other four gates have two inputs (but still only one output).

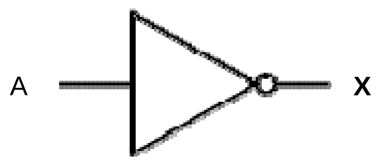
The next section describes the function of all five logic gates.

www.PapaCambridge.com

Description of the function of the five logic gates

The NOT gate



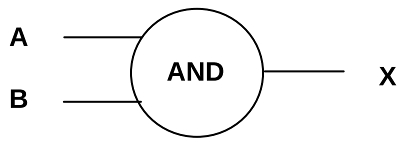


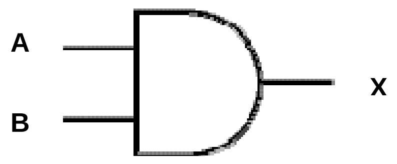
The output (called X) is **true** (i.e. **1** or **ON**) when the **INPUT A** is **NOT TRUE** (i.e. **0** or **OFF**).

X = NOT A

INPUT A	OUTPUT X
1	0
0	1

The AND gate



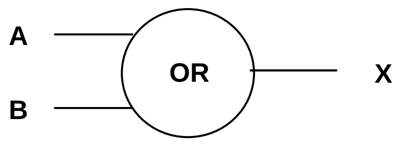


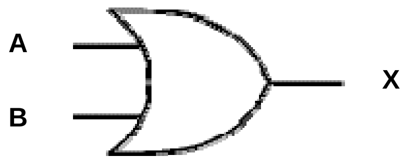
The output (called X) is only **true** (i.e. **1** or **ON**) if the (**INPUT A AND INPUT B**) are both **true** (i.e. **1** or **ON**).

= A AND B

INPUT A	INPUT B	OUTPUT X
1	1	1
1	0	0
0	1	0
0	0	0

The OR gate





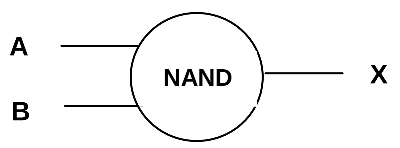
The output (called X) is **true** (i.e. **1** or **ON**) if the (**INPUT A OR INPUT B**) are **true** (i.e. **1** or **ON**).

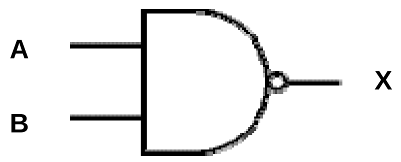
X = A OR B

INPUT A	INPUT B	OUTPUT X
1	1	1
1	0	1
0	1	1
0	0	0

www.PapaCambridge.com

The NAND gate





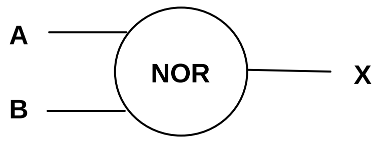
This is basically an **AND** gate with the output X inverted.

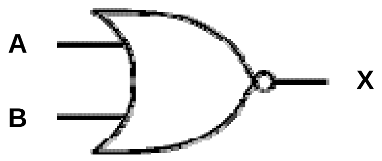
The output (called X) is **true** (i.e. **1** or **ON**) if (**INPUT A AND INPUT B**) are **NOT** both **true** (i.e. **1** or **ON**).

OT (A AND B)

INPUT A	INPUT B	OUTPUT X
1	1	0
1	0	1
0	1	1
0	0	1

The NOR gate





This is basically an **OR** gate with the output X inverted.

The output (called X) is **true** (i.e. **1** or **ON**) if **NOT** (**INPUT A OR INPUT B**) are **true** (i.e. **1** or **ON**).

NOT (A OR B)

INPUT A	INPUT B	OUTPUT X
1	1	0
1	0	0
0	1	0
0	0	1

The tables above containing 1s and 0s are known as **truth tables** and are an integral part of logic gates functionality. These are used extensively throughout this booklet in the design and testing of logic networks built up from logic gates.

www.PapaCambridge.com

Combinations of logic gates

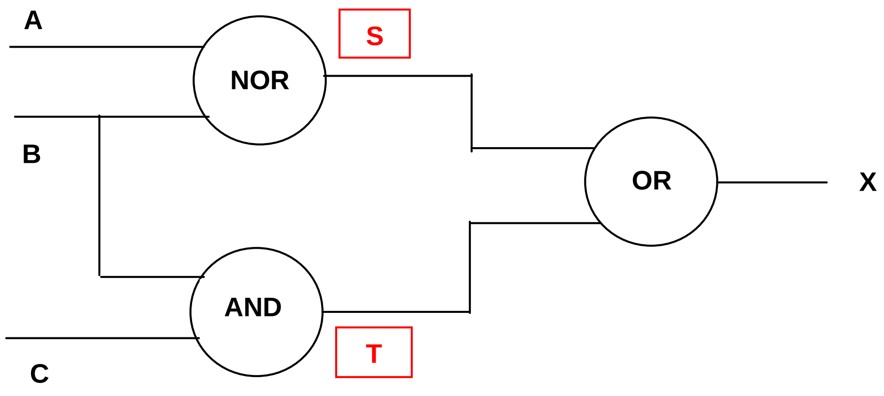
It is possible to combine logic gates together to produce more complex logic networks.

This booklet will only deal with a maximum of three inputs and up to six logic gates. The output from a logic network is checked by producing the truth table (as shown in the examples below).

We will deal with two different scenarios here. The first involves drawing the truth table from a given logic network; the second involves designing a logic network for a given problem and then testing it by drawing the truth table.

Producing the truth table from a given logic network

Consider the following logic network which contains three inputs and three logic gates:



If we now look at the output in two stages. First let us consider the outputs being produced at stages **S** and **T**. To do this we need to draw a truth table. There are three inputs (A, B and C) which gives 2^3 (i.e. 8) possible combinations of 1s and 0s. To work out the outputs at **S** and **T** we need to refer to the truth tables for the NOR gate and for the AND gate. For example, when A = 1 and B = 1 then we have 1 NOR 1 which gives the value of **S** = 0. Continuing doing the same thing for all 8 possible inputs we get the following **interim truth table**:

A

S

T

A	B	C	S	T
1	1	1	0	1
1	1	0	0	0
1	0	1	0	0
1	0	0	0	0
0	1	1	0	1
0	1	0	0	0
0	0	1	1	0
0	0	0	1	0

www.PapaCambridge.com

The final stage involves S **OR** T.

S	T	X
0	1	1
0	0	0
0	0	0
0	0	0
0	1	1
0	0	0
1	0	1
1	0	1

X

This gives the final truth table:

A	B	C	X
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	1

Designing logic networks to solve a specific problem and testing using truth tables

Consider the following problem:

"If button A or button B are on and button C is off then the alarm X goes on"

We can convert this onto logic gate terminology (ON = 1 and OFF = 0):

If (A = 1 OR B = 1) AND (C = NOT 1) then (X = 1)

(Notice: rather than write 0 we use NOT 1)

To draw the logic network, we do each part in brackets first i.e. A = 1 OR B = 1 is one gate then C = NOT 1 is the second gate. These are then joined together by the AND gate. Once the logic network is drawn we can then test it using a truth table. Remember the original problem – we are looking for the output to be 1 when A or B is 1 and when C is 0. Thus we get the following logic network and truth table from the network. Looking at the values in the truth table, we will be able to clearly see that it matches up with the original problem which then gives us confidence that the logic network is correct.

www.PapaCambridge.com

A	B	C	X
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	0

Let us now consider a second problem:

A steel rolling mill is to be controlled by a logic network made up of AND, OR and NOT gates only. The mill receives a stop signal (i.e. S = 1) depending on the following input bits:

INPUT	BINARY VALUE	CONDITION
L	**1** **0**	**Length > 100 metres** **Length ≤ 100 metres**
T	**1** **0**	**Temperature < 1000 C** **Temperature ≤ 1000 C**
V	**1** **0**	**Velocity > 10 m/s** **Velocity ≤ 10 m/s**

A stop signal (S = 1) occurs when:

either Length, L > 100 metres and Velocity, V ≤ 10 m/s

or Temperature, T ≤ 1000 C and Velocity, V > 10 m/s

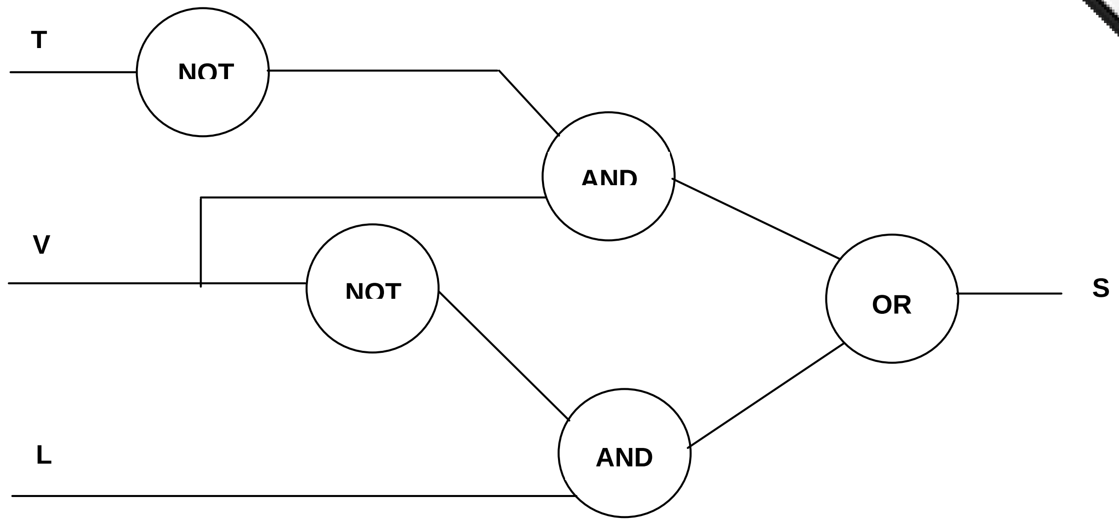
Draw a logic network and truth table to show all the possible situations when the stop signal could be received.

The first thing to do is to try and turn the question into a series of logic gates and then the problem becomes much simplified.

- The first statement can be re-written as: **(L = 1 AND V = NOT 1)** since Length > 100 metres corresponds to a binary value of 1 and Velocity ≤ 10 m/s corresponds to a binary value of 0 (i.e. NOT 1).
- The second statement can be re-written as **(T = NOT 1 AND V = 1)** since Temperature ≤ 1000C corresponds to a binary value of 0 (i.e. NOT 1) and Velocity > 10 m/s corresponds to a binary value of 1
- Both these statements are joined together by OR which gives us the logic statement: **if (L = 1 AND V = NOT 1) OR (T = NOT 1 AND V = 1) then S = 1**

We can now draw the logic network and truth table to give the solution to the original problem (input L has been put at the bottom of the diagram just to avoid crossing over of lines; it merely makes it look neater and less complex and isn't essential):

www.PapaCambridge.com



L	T	V	S
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	1
0	0	0	0

PROBLEMS

In questions 1 to 6, produce truth tables from the given logic networks. Remember that if there are TWO inputs then there will be four (2^2) possible outputs and if there are THREE inputs there will be eight (2^3) possible outputs.

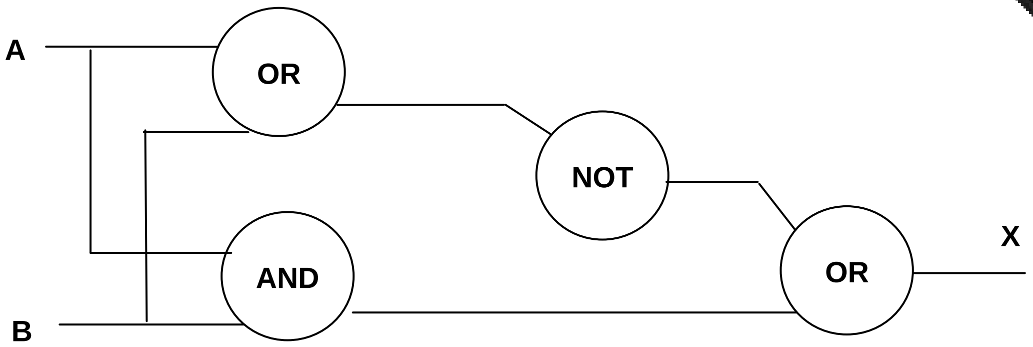
i.e.

A	B	C	X
1	1	1	
1	1	0	
1	0	1	
1	0	0	
0	1	1	
0	1	0	
0	0	1	
0	0	0	

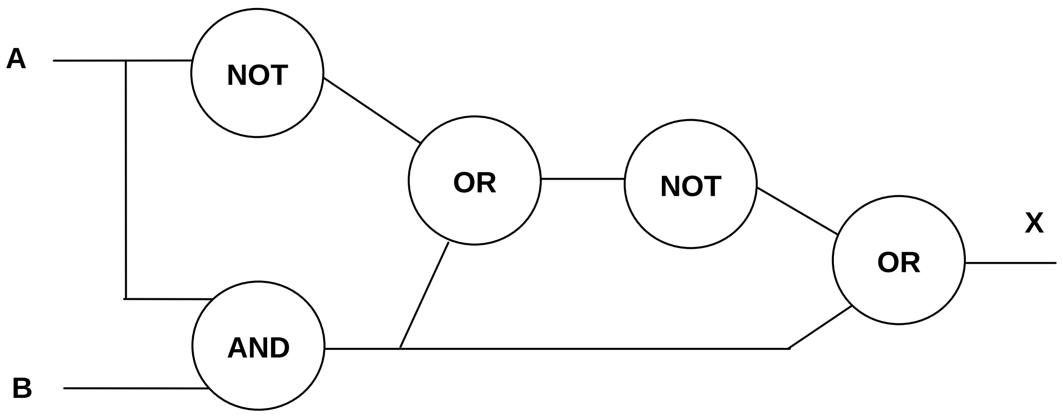
A	B	X
1	1	
1	0	
0	1	
0	0	

www.PapaCambridge.com

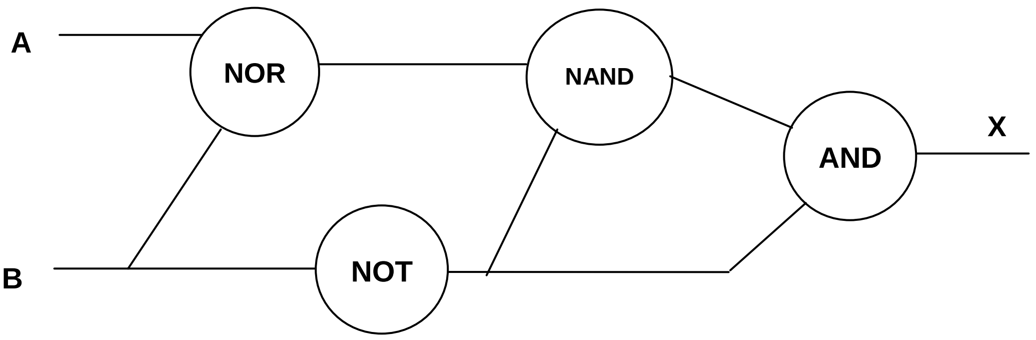
(1)



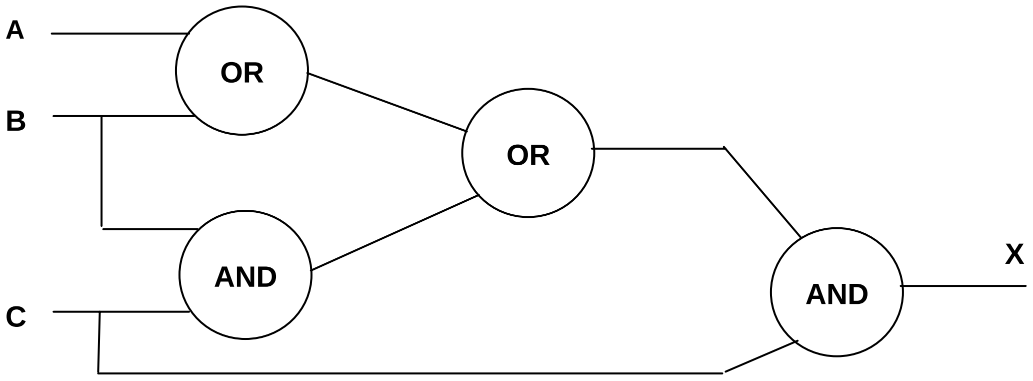
(2)



(3)

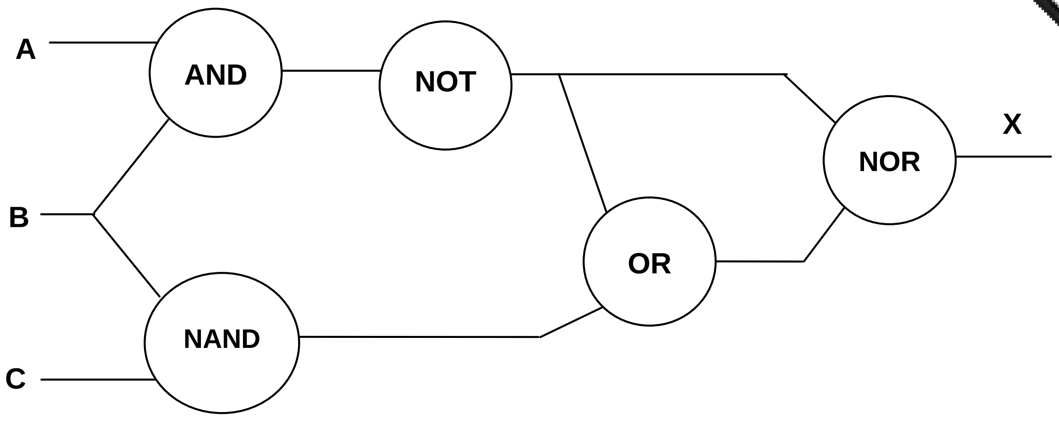


(4)

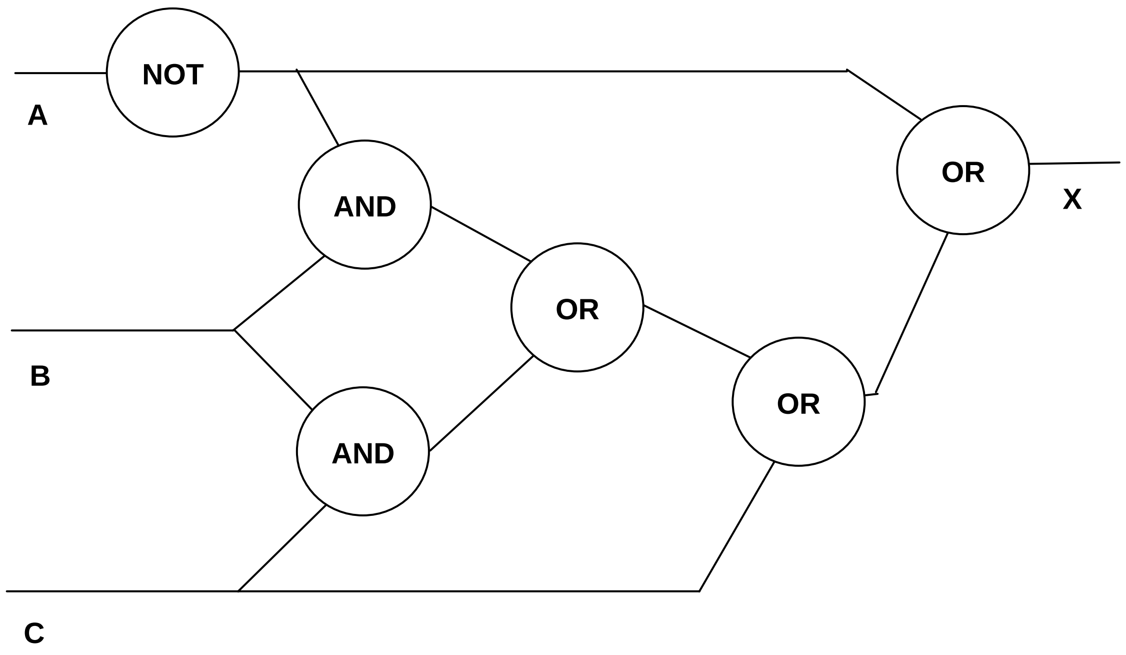


www.PapaCambridge.com

(5)



(6)



Questions 7 to 10 require both the logic network to be created and also the truth table. The truth table can be derived from the logic network, but also from the problem. This is a check that the logic network actually represents the original problem.

(7) A computer will only operate if three switches P, S and T are correctly set. An output signal (X = 1) will occur if R and S are both ON or if R is OFF and S and T are ON. Design a logic network and draw the truth table for this network.

www.PapaCambridge.com

(8) A traffic signal system will only operate if it receives an output signal (D =
This can only occur if:

either (a) signal A is red (i.e. A = 0)
or (b) signal A is green (i.e. A = 1) and signals B and C are both red (i.e. B and C are both 0)

Design a logic network and draw a truth table for the above system.

(9) A chemical plant gives out a warning signal (W = 1) when the process goes wrong. A logic network is used to provide input and to decide whether or not W = 1.

Input	**Binary Value**	**Plant Status**
C	1 0	Chemical Rate = 10 m^3/s Chemical Rate < 10 m^3/s
T	1 0	Temperature = 87 C Temperature > 87 C
X	1 0	Concentration > 2 moles Concentration = 2 moles

A warning signal (W = 1) will be generated if

either (a) Chemical Rate < 10 m^3/s
or (b) Temperature > 87 C and Concentration > 2 moles
or (c) Chemical rate = 10 m^3/s and Temperature > 87 C

Draw a logic network and truth table to show all the possible situations when the warning signal could be received.

(10) A power station has a safety system based on three inputs to a logic network. A warning signal (S = 1) is produced when certain conditions occur based on these 3 inputs:

Input	**Binary Value**	**Plant Status**
T	1 0	Temperature > 120C Temperature $\le$ 120C
P	1 0	Pressure > 10 bar Pressure $\le$ 10 bar
W	1 0	Cooling Water > 100 l/hr Cooling Water $\le$ 100 l/hr

A warning signal (S = 1) will be generated if:

either (a) Temperature > 120C and Cooling Water $\le$ 100 l/hr
or (b) Temperature $\le$ 120C and (Pressure > 10 bar or Cooling Water $\le$ 100 l/hr)

Draw a logic network and truth table to show all the possible situations when the warning signal could be received.