Checkpoint Task

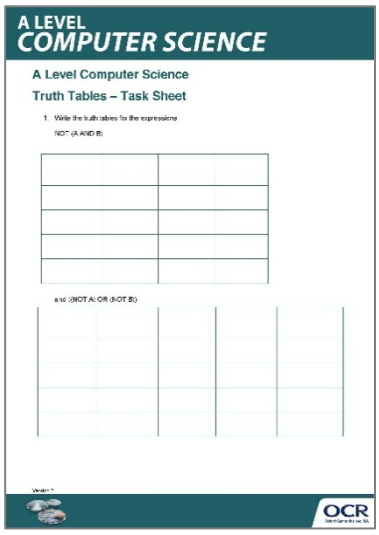
This Checkpoint Task should be used in conjunction with the KS4-KS5 Transition Guide – Binary.

A Level Computer Science

Truth Tables

Instructions and answers for teachers

This Checkpoint Task should be used in conjunction with the KS4-KS5 Transition Guide – Binary.


A LEVEL
COMPUTER SCIENCE
A Level Computer Science
Truth Tables – Task Sheet
1 Write the truth tables for the expressions
NOT (A AND B)
and (NOT A) OR (NOT B)
OCR

The Activity:

Ask the student to complete the questions and task on the activity sheet.

This activity offers an opportunity for English skills development.

Associated materials:

'Truth Tables' activity.

Ask the student to complete the questions and task on the activity sheet.

They can use a suitable high level programming language but should provide a suitable plan for their work, eg flowchart or pseudocode to explain their thinking. They should justify their choice of test data to cover all the possibilities.

Students should notice the truth tables are the same and should be encouraged to deduce that **NOT (A AND B)** is the same as **((NOT A) OR (NOT B))** (De Morgan's law). They should be encouraged to look at **NOT (A OR B)** and suggest what the answer might be before checking their prediction. This demonstrates their ability to generalise, a fundamental computational thinking skill.
It is important the knowledge can be applied so it is important they recognise what form the output from this will take and what the actual value returned is.

Activity

1. Write the truth tables for the expressions
   NOT (A AND B)

A	B	A AND B	NOT(A AND B)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

and ((NOT A) OR (NOT B))

A	B	NOT A	NOT B	(NOT A) OR (NOT B)
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

2. What do you notice about these tables?

They are the same. NOT(A AND B) = (NOT A) OR (NOT B)

3. Design and create a program to output the value of a after the statement

IF (a < b) OR (b < c) THEN a = b has been executed.

Program in python might be

```
a=int(input('value for a '))
b=int(input('value for b '))
c=int(input('value for c '))
if a<b or b<c:
    a=b
print(a)
```

4. Decide on suitable test data for this program giving a reason for each combination of values for a, b and c, give your expected result and the actual result for each.

Values	Reason	Expected	Actual
a=1, b=2, c=3	a<b, b<c	a=b=2	2
a=1, b=1, c=1	a NOT<b, bNOT<c	a NOT=b, a=1	1
a=4, b=3, c=5	a NOT<b, b<c	a=b=3	3
a=4, b=5, c=3	a <b, bNOT<c	a=b=5	5
Other tests could include negative values, zeros and non-numeric values eg			
a=-3, b=-2, c=-1	Test negative values	a=b=-2	-2
a=1, b=frank, c=1	Test non numeric values	Error	Error
a=0.5, b=0.4, c=0.3	Test non integer values	Error, integers expected	Error
To fix the last error the program could be modified to use real values casting input as float instead of int			
a=0.5, b=0.4, c=0.3	a NOT<b, bNOT<c	a NOT=b, a=0.5	0.5
a=0.3, b=0.4, c=0.5	a<b, b<c	a=b=0.4	0.4

Revised program

```
a=float(input('value for a '))
b=float(input('value for b '))
c=float(input('value for c '))
if a<b or b<c:
    a=b
print(a)
```