A Level Computer Science Summer Work

(AQA GCE Computer Science)

Introduction: Welcome to A Level Computer Science!

If you have done GCSE Computing most of this work will be recognisable as it is topics you have covered previously, but a refresh is always a good idea.

The first section is looking at binary conversions and addition, the second section is completing a research task.

If you are new to Computer Science then this work will give a good base knowledge to get you started on the course. Along with the tasks attached you need to complete the **Python challenge** to give some insight into the programming language, and the binary tasks will give you some insight into the theory section.

You will probably have to spend a few hours in total on these activities.

DUE: 10th September 2018

Theory: Binary numbers

One of the most fundamental concepts in computing is binary numbers. Computers use the binary number system for storing program instructions and data of all types - numbers, characters, sound, and pictures and so on.

Computers use the binary number system as it is much easier to represent than the decimal number system we are familiar with. This is because computers require electricity to function. In Figure 1 we can see what happens when we connect up a light bulb to an electrical source via a switch.

When the switch is turned off no electricity gets through to the light bulb so it is off. When the switch is turned on the light bulb is on.

This can be represented in the following way:

OFF = 0

ON = 1

Put simply you can say that a computer switches on electricity to represent the number 1 and switches off the electricity to represent the number 0.

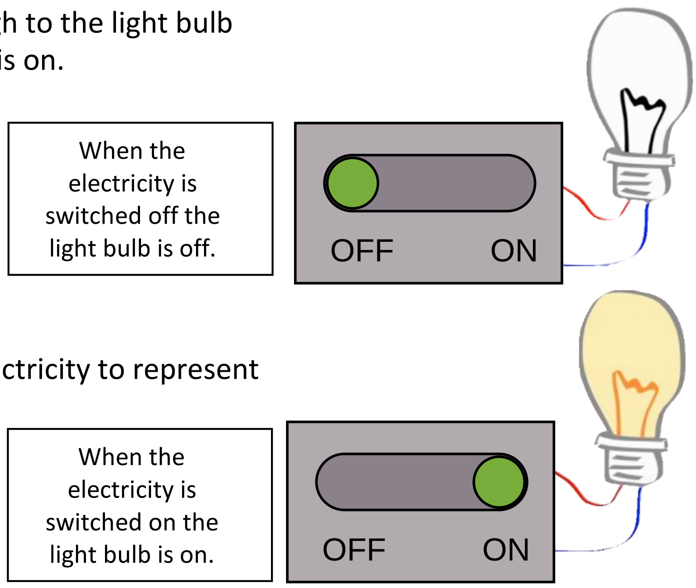

Figure 1: Representation of the two states available in electricity

The numbers 0 and 1 are the only two numbers available in binary.

Representing numbers in decimal

If we only have the numbers 1 and 0 how can computers represent a range of numbers - say 167? To answer this question we must look at how decimal numbers are actually formed.

In the decimal number system there are 10 different symbols available to represent numbers:

0	1	2	3	4	5	6	7	8	9

However, this doesn't mean that we can only represent the numbers 0-9. Consider the number 167:

100	**10**	**1**
1	6	7

Notice that we use the concept of columns (you may remember this from primary school), where each column is 10 times larger than the previous one (right to left). By doing this, the 10 symbols that we have are enough to represent any possible number.

Therefore 167 really says:

- We have 1 lot of 100
- We have 6 lots of 10
- We have 7 lots of 1

Hopefully, this makes sense as we will make use of this to explain how binary can represent numbers greater than 1 or 0.

Let's reconsider the column headings above:

100	**10**	**1**
1	6	7

We could rewrite them so they were:

$\mathbf{10^2}$	$\mathbf{10^1}$	$\mathbf{10^0}$
1	6	7

Understanding this will be useful when we look at the binary representation of columns.

Representing numbers in binary

Sometimes we call the decimal system base 10 as it has 10 symbols that we can use to represent numbers. Likewise, binary is sometimes called base 2 as it has 2 symbols for representation. How can we use these two symbols to represent larger numbers?

If you look back to the decimal column headings above you will notice that the headings are all powers of ten. In binary all of our column headings are powers of two:

2^2	2^1	2^0
1	0	1

You could also write this as:

4	2	1
1	0	1

This means that the binary number 101 is representing the following situation:

- We have 1 lot of 4
- We have 0 lots of 2
- We have 1 lots of 1

If we want to represent larger numbers we just keep adding columns on at the left hand side. These days computers have 64-bit processors, this means that they can work with binary numbers that have 64 columns. In Computing you will generally only be asked to work with 8-bit binary numbers or binary numbers which have 8 columns.

Converting from binary to decimal

In the example above we had the binary number 101, by placing it in a table we can see the column headings for each digit:

4	**2**	**1**
1	0	1

Once we know this we can work out what the equivalent decimal (base 10) representation would be:

(1 x 4) + (0 x 2) + (1 x 1) = 5

What about a larger binary number, say 10011010? This is easy - just work out the column headings above each digit - they are all powers of two! E.g.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	1	0	1	0

This is the same as saying:

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
1	0	0	1	1	0	1	0

Once we have the column headings all we need to do is add up the column values that have a one underneath:

128 + 16 + 8 + 2 = 154

Converting from decimal to binary

What about going the other way? Say we had a decimal number, such as 167, and wanted to convert to binary. This is very straightforward.

Step One - put down your column values

The first thing to do is to write down your column values e.g.

128	64	32	16	8	4	2	1

Step Two - subtract each column value from your decimal number

Starting from the left-hand side work out whether the column value can be subtracted from the number you are trying to convert. If it can be place a 1 under that column and subtract the column value from the original decimal number.

Repeat this until you reach 0.

This will be demonstrated on the following page.

128	64	32	16	8	4	2	1

Ask the following questions to yourself:

1. Is it possible to subtract 128 from 167?
   - 167 - 128 = 39
   - We have a positive answer, therefore place a 1 in the 128 column of the table:

128	64	32	16	8	4	2	1
1							

2. Is it possible to subtract 64 from 39?
   - 39 - 64 = -25
   - we have a negative answer, therefore place a 0 in the 64 column of the table:

128	64	32	16	8	4	2	1
1	0						

3. Is it possible to subtract 32 from 39?
    - 39 - 32 = 7
    - we have a positive answer, therefore place a 1 in the 32 column of the table:

128	64	32	16	8	4	2	1
1	0	1					

4. Is it possible to subtract 16 from 7?
    - 7 - 16 = -9
    - we have a negative answer, therefore place a 0 in the 16 column of the table:

128	64	32	16	8	4	2	1
1	0	1	0				

5. Is it possible to subtract 8 from 7?
    - 7 - 8 = -1
    - we have a negative answer, therefore place a 0 in the 8 column of the table:

128	64	32	16	8	4	2	1
1	0	1	0	0			

6. Is it possible to subtract 4 from 7?
    - 7 - 4 = 3
    - we have a positive answer, therefore place a 0 in the 4 column of the table:

128	64	32	16	8	4	2	1
1	0	1	0	0	1		

7. Is it possible to subtract 2 from 3?
    - 3 - 2 = 1
    - we have a positive answer, therefore place a 0 in the 2 column of the table:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	

8. Is it possible to subtract 1 from 1?

- 1 - 1 = 0
- we have zero left, therefore place a1 in the 1 column of the table:

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
1	0	1	0	0	1	1	1

Therefore, the representation of 167 in binary is 10100111.

Summary

The binary number system is used by computers as it is easy for them to represent - electricity turned on represents a 1 and electricity turned off represents a 0. The binary representation of large numbers uses a column system based on powers of 2, we can use this to covert numbers to and from binary easily.

In the next section we will look at adding binary numbers together.

Adding together decimal numbers

Obviously it is not enough just to be able to represent numbers in binary - we need to be able to perform calculations with them. In this section we will look at how to add binary numbers together.

Consider adding together the decimal numbers 167 and 14:

10^2	**10^1**	**10^0**
1	6	7
	1	4

This is a fairly straightforward sum that you can probably do in your head but let's consider how to perform it in any case.

We would look at each column in turn (right to left) and add the values in each column together, carrying as necessary.

In the first column 7 + 4 = 11 therefore we would place a 1 in the space for the answer and carry a 1:

10^2	**10^1**	**10^0**
1	6	7
	1	4
		1
	1	

In the second column the sum is now 6+1+1 = 8:

10^2	**10^1**	**10^0**
1	6	7
	1	4
	8	1
	1	

In the third column the sum is 1 + 0 = 1 (as there is no value in the third column for the second number and there is no carry from the previous column):

10^2	10^1	10^0
1	6	7
	1	4
1	8	1
	1	

Therefore the answer to the original sum is 167+14 = 181. Addition in binary work in exactly the same way.

Adding together binary numbers

In this section we will add together two binary numbers:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
		1	1	1	1	0	0

Binary addition works in exactly the same way as decimal addition; we will look at each column from right to left in turn. In the first column the sum is 1 + 0 = 1. Therefore, a 1 is place in the space for the answer:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
		1	1	1	1	0	0
							1

In the second column the sum is 1 + 0 = 1. Therefore, a 1 is placed in the space for the answer:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
		1	1	1	1	0	0
						1	1

In the third column the sum is 1 + 1 = 10. This is because 2 in binary is 10. Because of this we place a zero in the space for the answer and carry the 1:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
		1	1	1	1	0	0
					0	1	1
				1			

In the fourth column the sum is 0 + 1 + 1 = 10. Again, we place a zero in the space for the answer and carry the 1:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
		1	1	1	1	0	0
				0	0	1	1
			1	1			

In the fifth column the sum is 0 + 1 + 1 = 10. Again, we place a zero in the space for the answer and carry the 1:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
		1	1	1	1	0	0
			0	0	0	1	1
		1	1	1			

In the sixth column the sum is 1 + 1 + 1 = 11. This is because 3 in binary is 11. Therefore, we place a 1 in the space for the answer and carry a 1:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
	1	1	1	1	1	0	0
		1	0	0	0	1	1
	1	1	1	1			

In the seventh column the sum is 0 + 1 = 1:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
	1	1	1	1	1	0	0
	1	1	0	0	0	1	1
1	1	1	1	1			

In the eighth column the sum is 1 + 0 = 1:

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1
	1	1	1	1	1	0	0
1	1	1	0	0	0	1	1

Therefore, the final answer is 11100011.

Summary

In this section you have seen how to add binary numbers together. The next few pages have exercises which use skills of converting between binary and decimal and adding binary numbers together.

Task 1: Binary conversions

Write the following as decimal numbers showing the working:

Question No.	Binary Value	Decimal Value (with Working)
1	1101	
2	10111	
3	11010	
4	110101	
5	110100	
6	1011100	
7	1101000	
8	11010100	

Task 2: Decimal conversions

Write the following as binary numbers showing the working:

Question No.	Decimal Value	Binary Value (with Working)
1	129	
2	137	
3	196	
4	172	
5	1161	
6	174	
7	185	
8	1148	

Task 3: Binary addition

Add the following binary numbers together showing the working:

Question No.	Sum	Answer (with Working)
1	11010 + 101	
2	1001 + 101	
3	10011 + 1011	
4	10011 + 10011	
5	10110 + 1011	
6	101101 + 101101	
7	101011 + 10110	
8	1101111 + 1011101	

Research

*Not copy and paste.

Research one new technology that has been in the news this year. Give an overview of the technology and your opinion on its development and the impact that it could have.

To start you off have a look at the following links:

http://www.bbc.co.uk/news/technology

https://www.technologyreview.com/lists/technologies/2017/

https://www.newscientist.com/subject/technology/

Programming

https://www.sololearn.com/Course/Python/

Sign up and complete the set tasks to introduce you to the programming language. OR use it to refresh your skills.