W9/ L1

6th November 2019

LO: Develop ability to convert a denary number into binary number and vice versa

- Take notes about the denary and binary number systems
- Learn the difference between a bit, a nibble and a byte
- Practise converting denary numbers into binary

Do Now: Write down the units of file size from bytes to terabytes

Bits and bytes

Bits can be grouped together to make them easier to work with. **A group of 8 bits is called a byte**.

Other groupings include:

- **Nibble** - 4 bits (half a byte)
- **Byte** - 8 bits
- **Kilobyte** (KB) - 1000 bytes
- **Megabyte** (MB) - 1000 kilobytes
- **Gigabyte** (GB) - 1000 megabytes
- **Terabyte** (TB) - 1000 gigabytes

Denary

- Denary is the number system that we use every day of our lives.
- Denary uses ten digits (0,1,2,3,4,5,6,7,8 and 9) to represent all numbers.
- For this reason, denary is also known as base-10.
- When we put digits together, each digit is worth ten times the one to its right.
- This is complicated to think about, but much easier to understand when we use a table...

Denary

- We can work out what the digits 5 1 6 2 means by using place values (which increase by the power of 10)

Place value	1000	100	10	1
	5	1	6	2

5 x1000 =	5000
1 x 100 =	+100
6 x 10 =	+60
2 x 1 =	+2
Total	5162

Memory Map

two digits (0,1,2,3,4,5,6,7,8 and 9) to represent all numbers

known as base-10

Binary

e.g. 5162 with place values

Place value	8	4	2	1
	5	1	6	2

1000 x 5 =	5000
100 x 1 =	+100
10 x 6 =	+60
1 x 2 =	+2
Total	5162

each digit is worth ten times the one to its right (increases by the power of 10)

What is Binary?

- Binary is a number system used by computers
- There are only two possible digits used:

1 and 0

- For this reason, binary is also known as base-2.

Binary

- We can work out what the digits 1 0 0 1 means by using place values (which increase by the power of 2)

Place value	8	4	2	1
	1	0	0	1

1 x8 =	**8**
0 x 4 =	**0**
0 x 2 =	**0**
1 x 1 =	**1**
Total	**9**

So binary number 1001 is equivalent to the denary number 9

Memory Map

two digits (0 and 1) to represent all numbers

known as base-2

Binary

e.g. 1001 with place values

Place value	8	4	2	1
	1	0	0	1

1 x 8 =	8
0 x 4 =	+0
0 x 2 =	+0
1 x 1 =	+1
Total	9

each digit is worth 2 times the one to its right (place values increase by the power of 2)

Bits, Nibbles and Bytes

- The binary number 1001 has 4 bits (binary digits).
- A 4-bit binary number is known as a nibble.
- An 8-bit binary number is known as a byte.
- The place values for a byte are:

128	64	32	16	8	4	2	1

- So to convert 10010111 into denary:

IMPORTANT:
You must show your working out!

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

128 + 16 + 4 + 2 + 1

= 151

Denary to Binary

- To convert from the denary number 49 into binary, we write a 1 underneath each place value that we need to add up to 49.
- So working from the left, we ask ourselves "Can we use 128 to get 49?"
- The answer is "No, it's too big" because so a 0 goes underneath the 128 place value.
- We do the same for the 64. Again, it's too big so a 0 goes under the 64.
- But 32 will go into 49 so we put a 1 under it. Now we have 17 left.
- We put a 1 under each number that will make 17.

IMPORTANT:
You must show your working out!

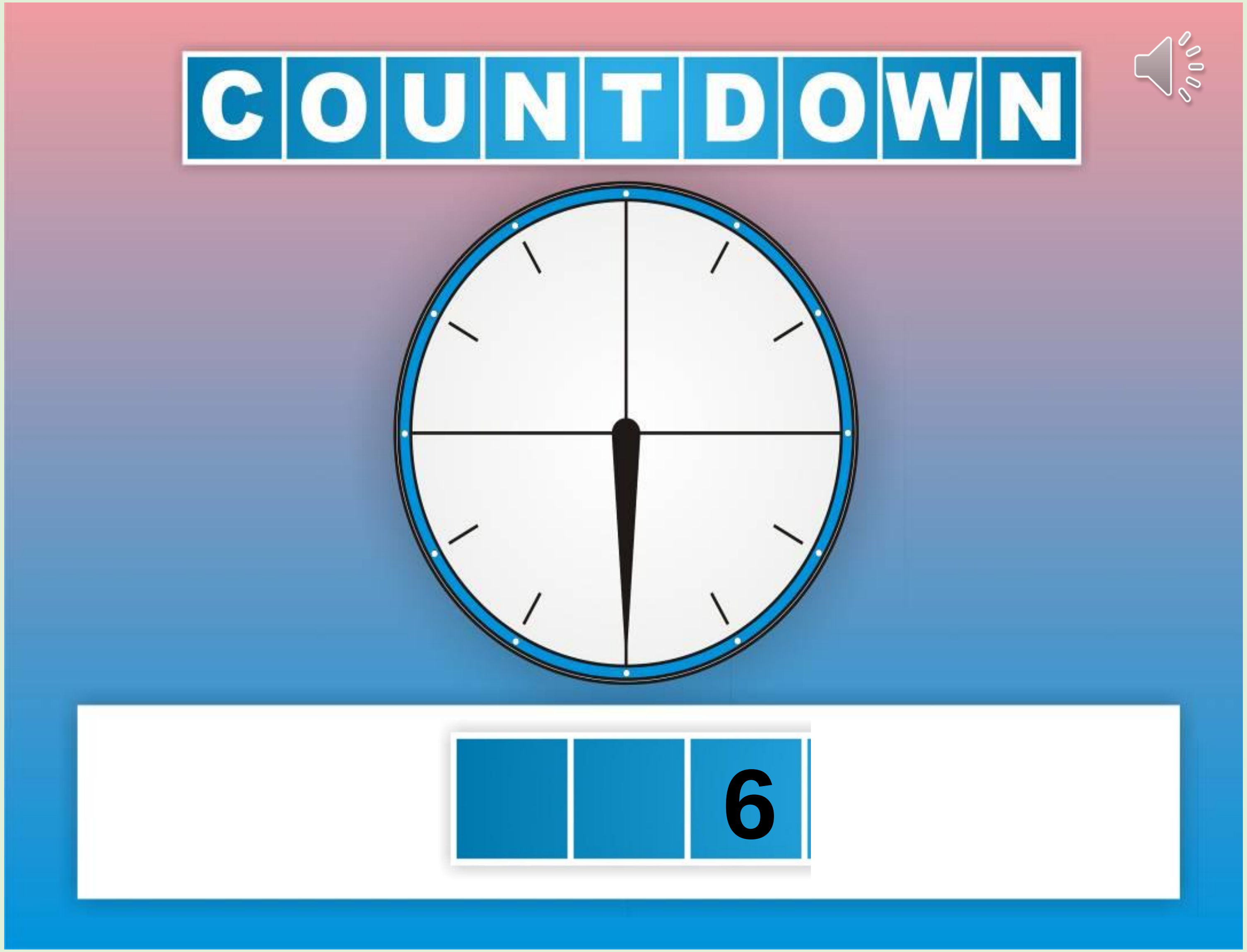
128	64	32	16	8	4	2	1
0	0	1	1	0	0	0	1

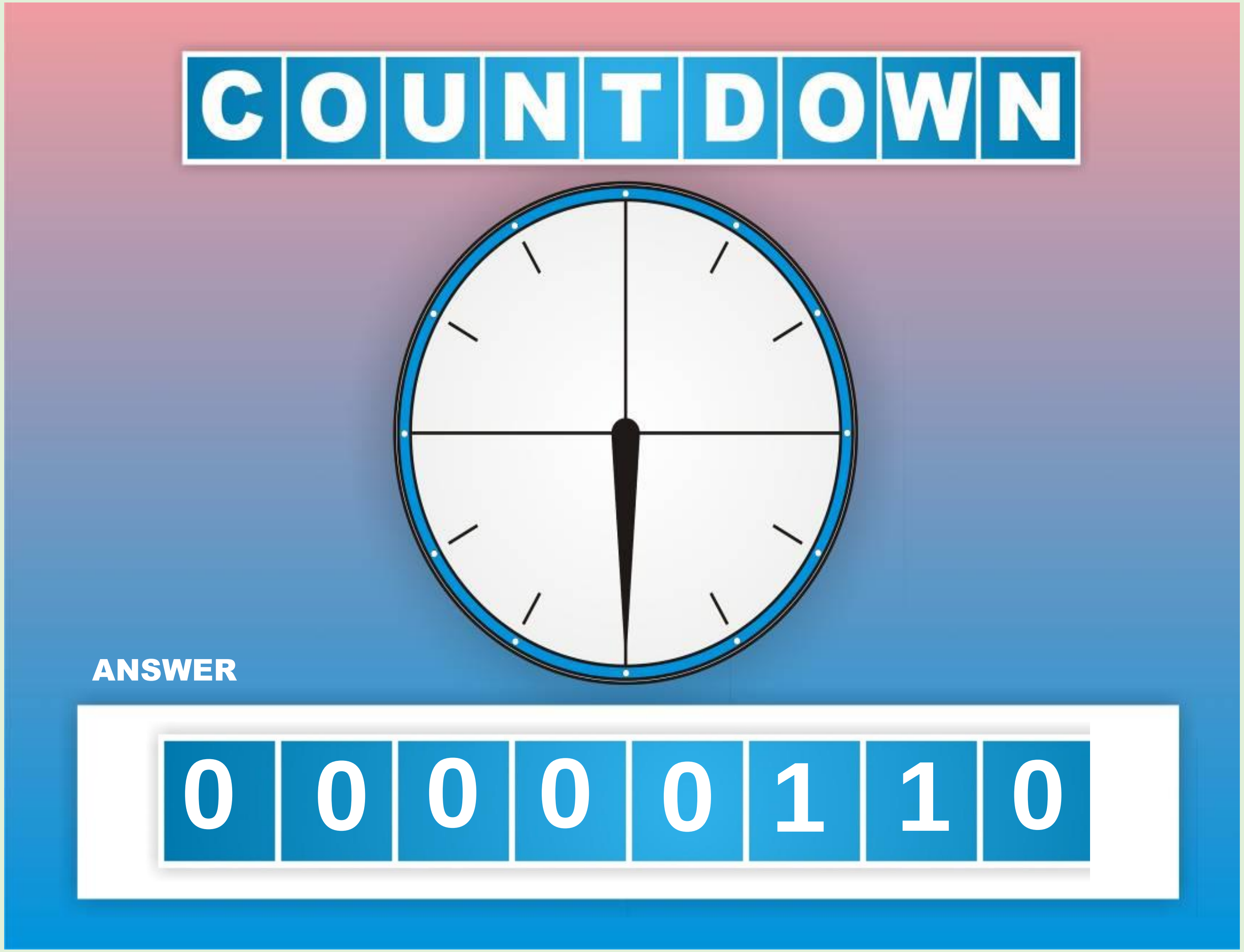
32 + 16 + 1

= 49

COUNTDOWN

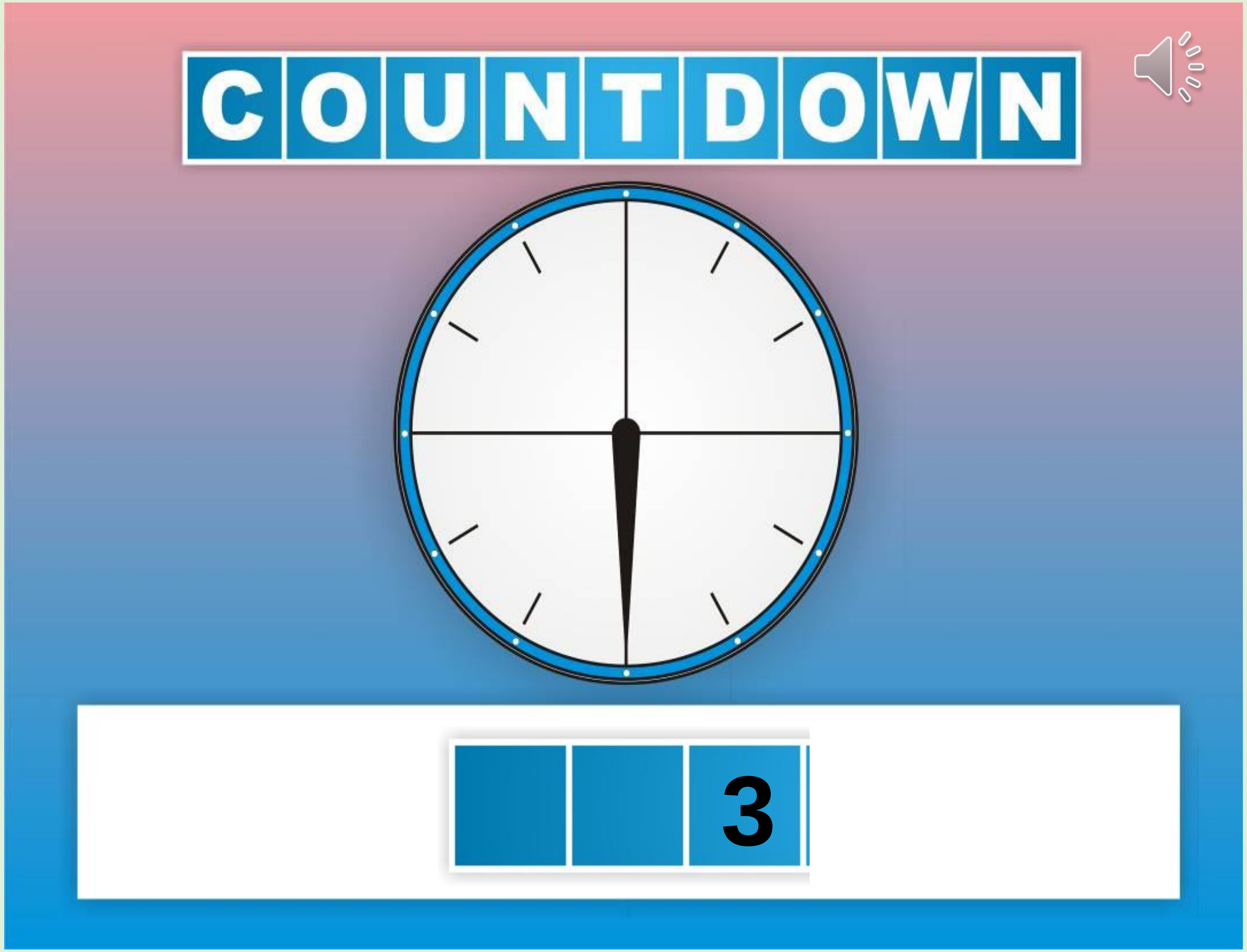
Ready!

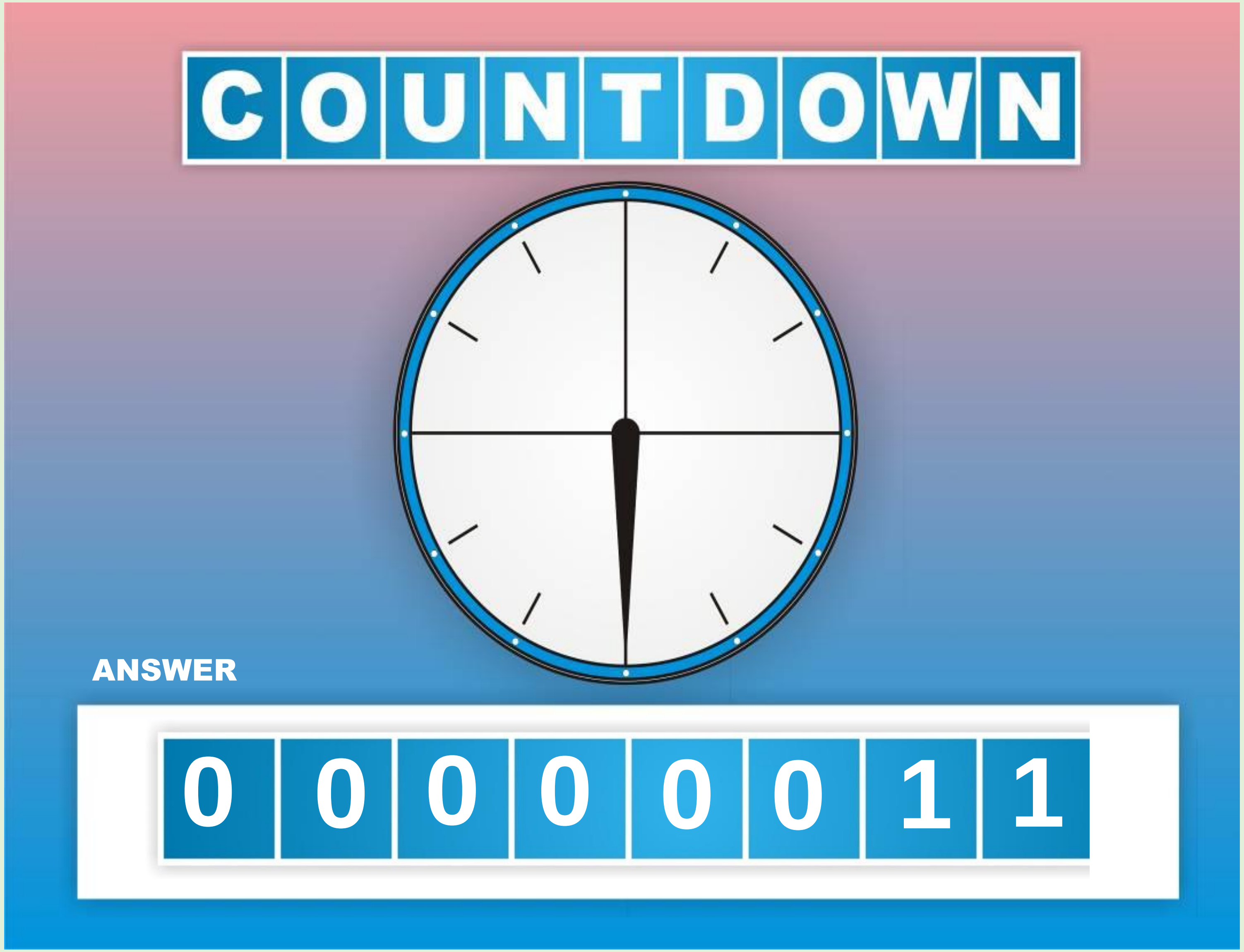

COUNTDOWN
6


COUNTDOWN
ANSWER
0 0 0 0 0 1 1 0

COUNTDOWN

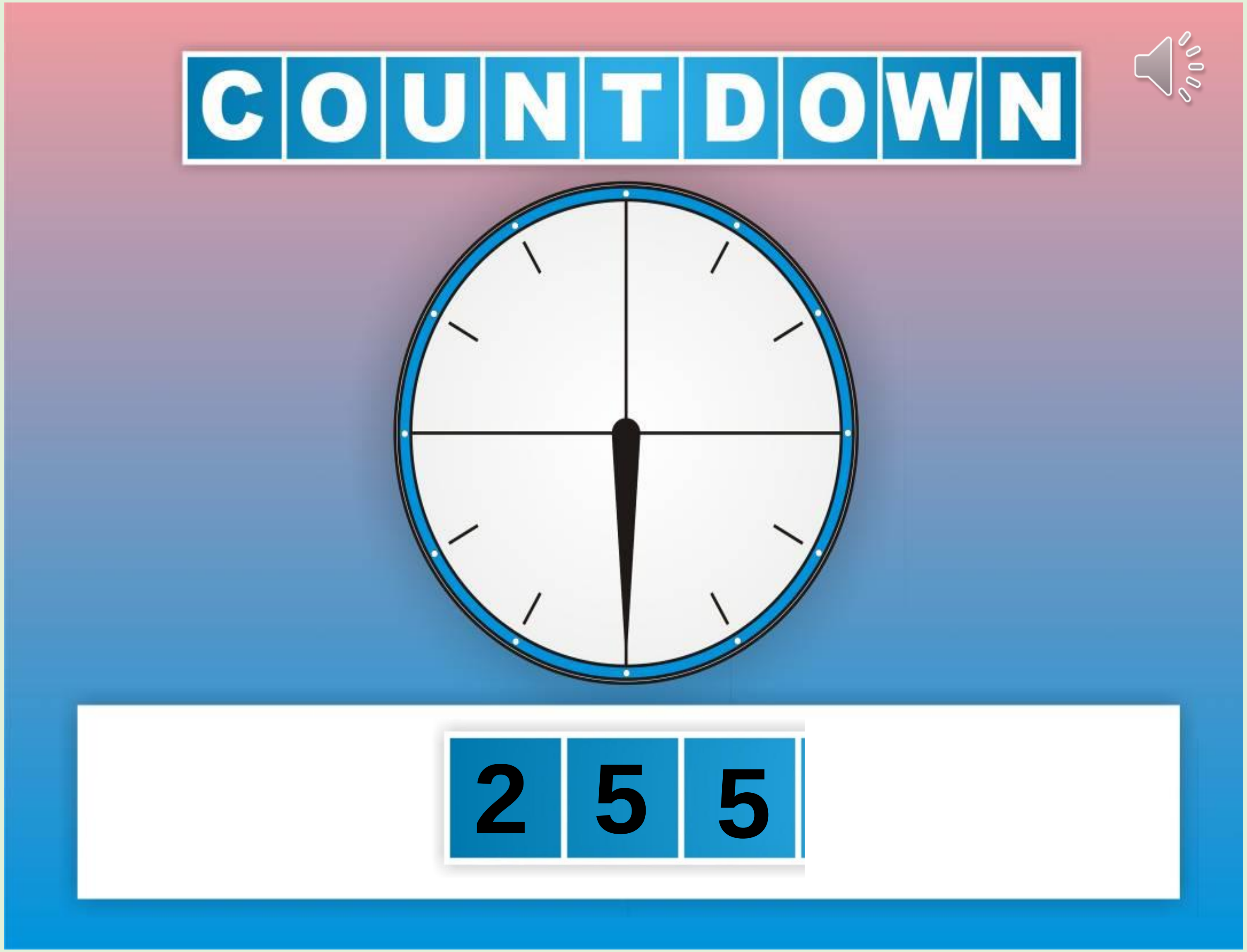
Ready!

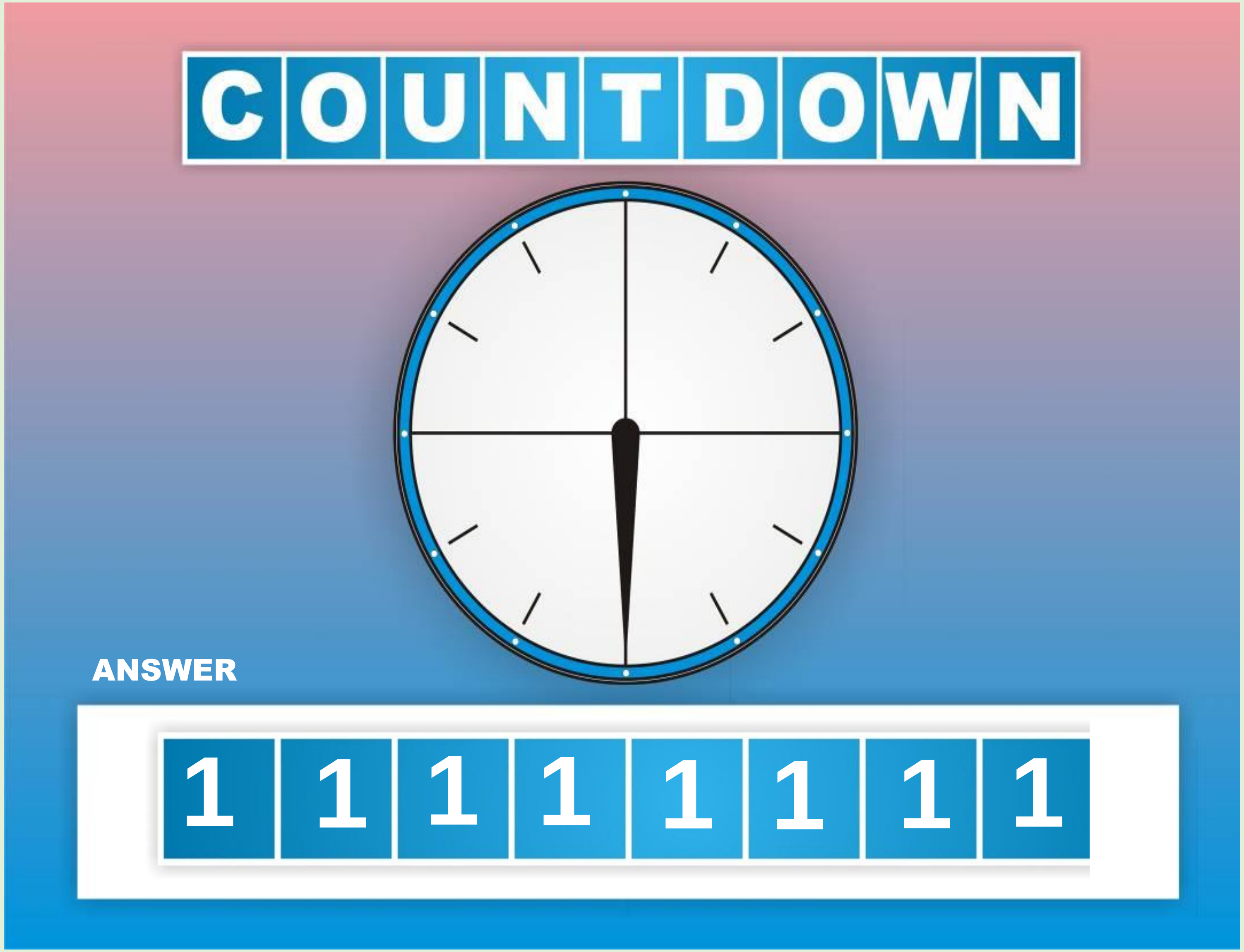

COUNTDOWN
3


COUNTDOWN
ANSWER
0 0 0 0 0 0 1 1

COUNTDOWN

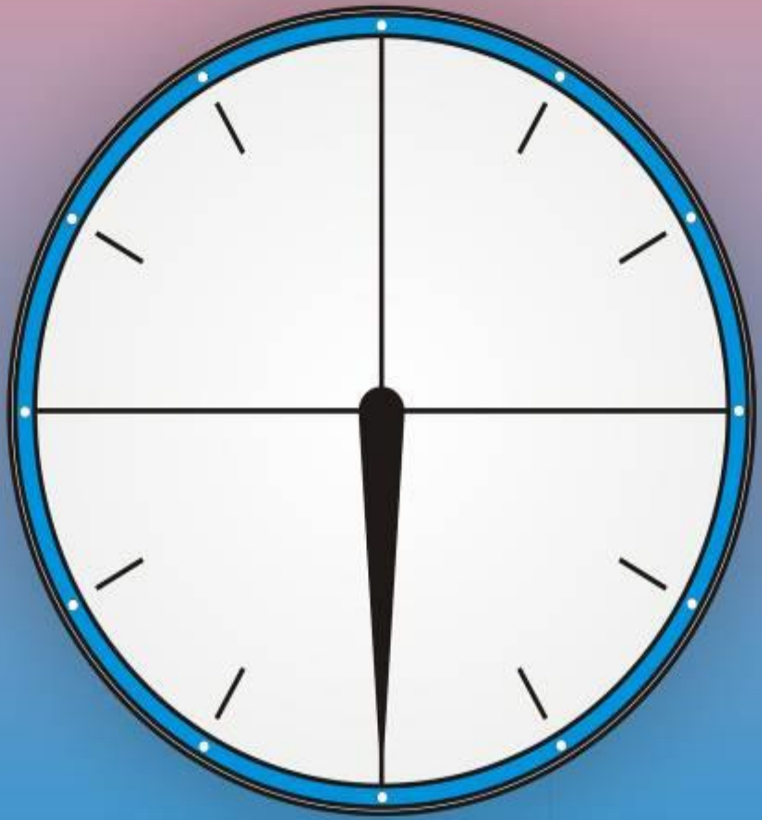
Ready!


COUNTDOWN
2 5 5


COUNTDOWN
ANSWER
1 1 1 1 1 1 1 1

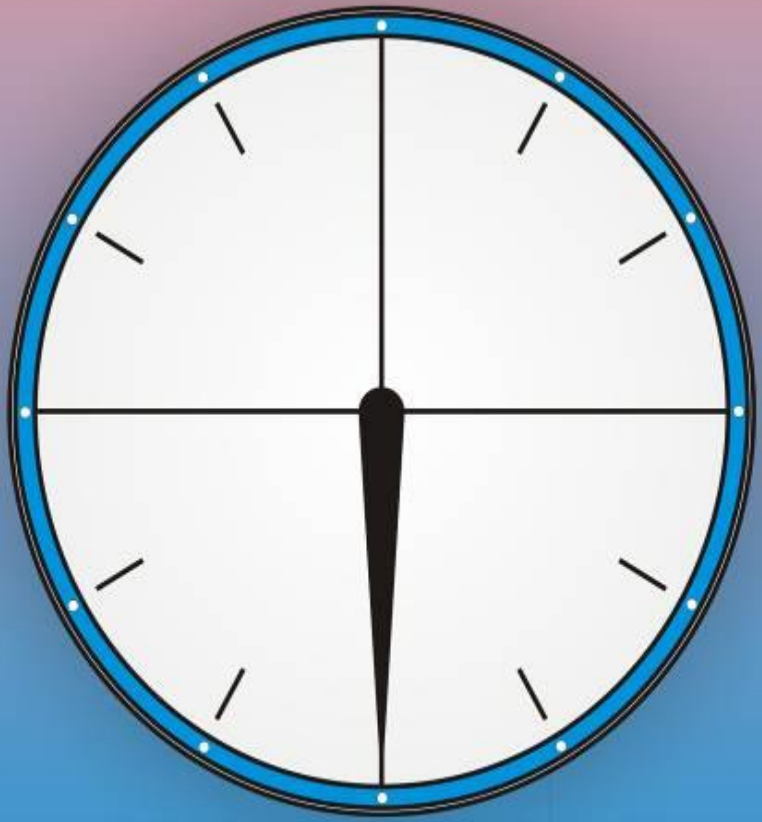
Ready!

COUNTDOWN



1 0 0 0 0 0 0 1

COUNTDOWN



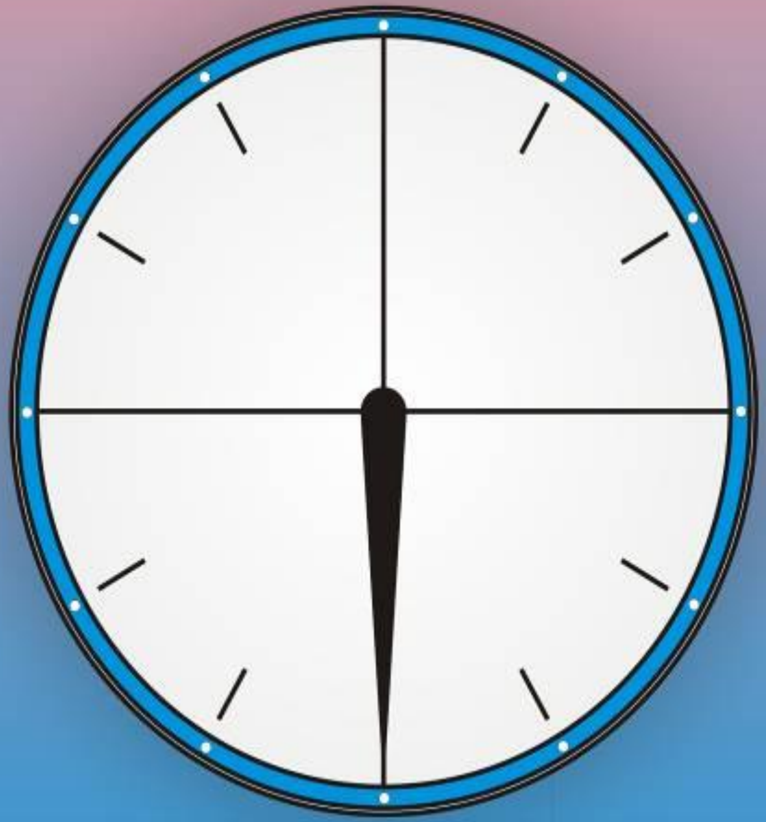
ANSWER

129

COUNTDOWN

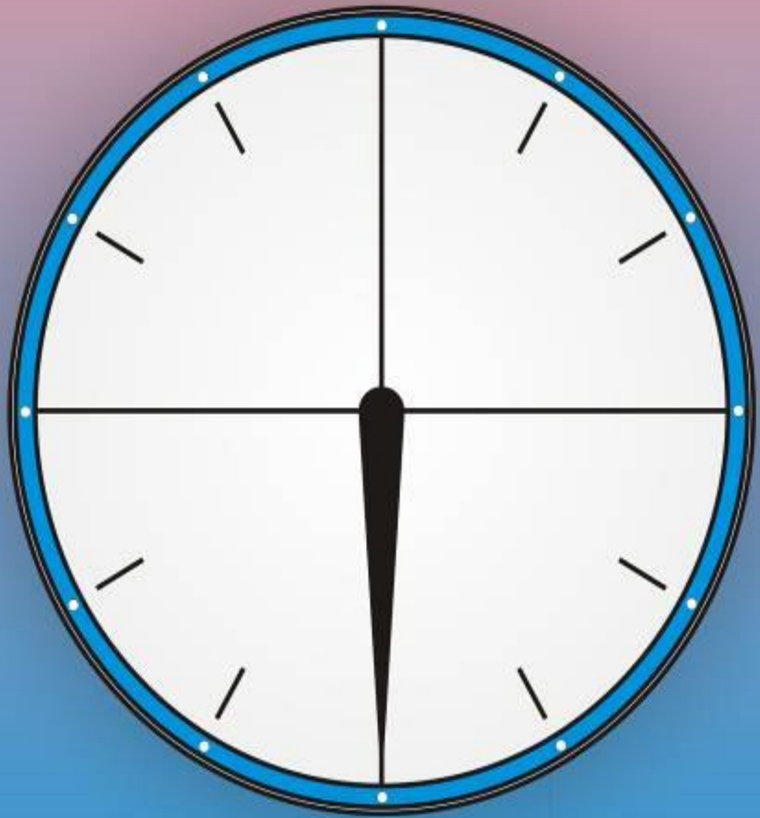
Ready!

COUNTDOWN



0 0 0 1 0 1 0 1

COUNTDOWN

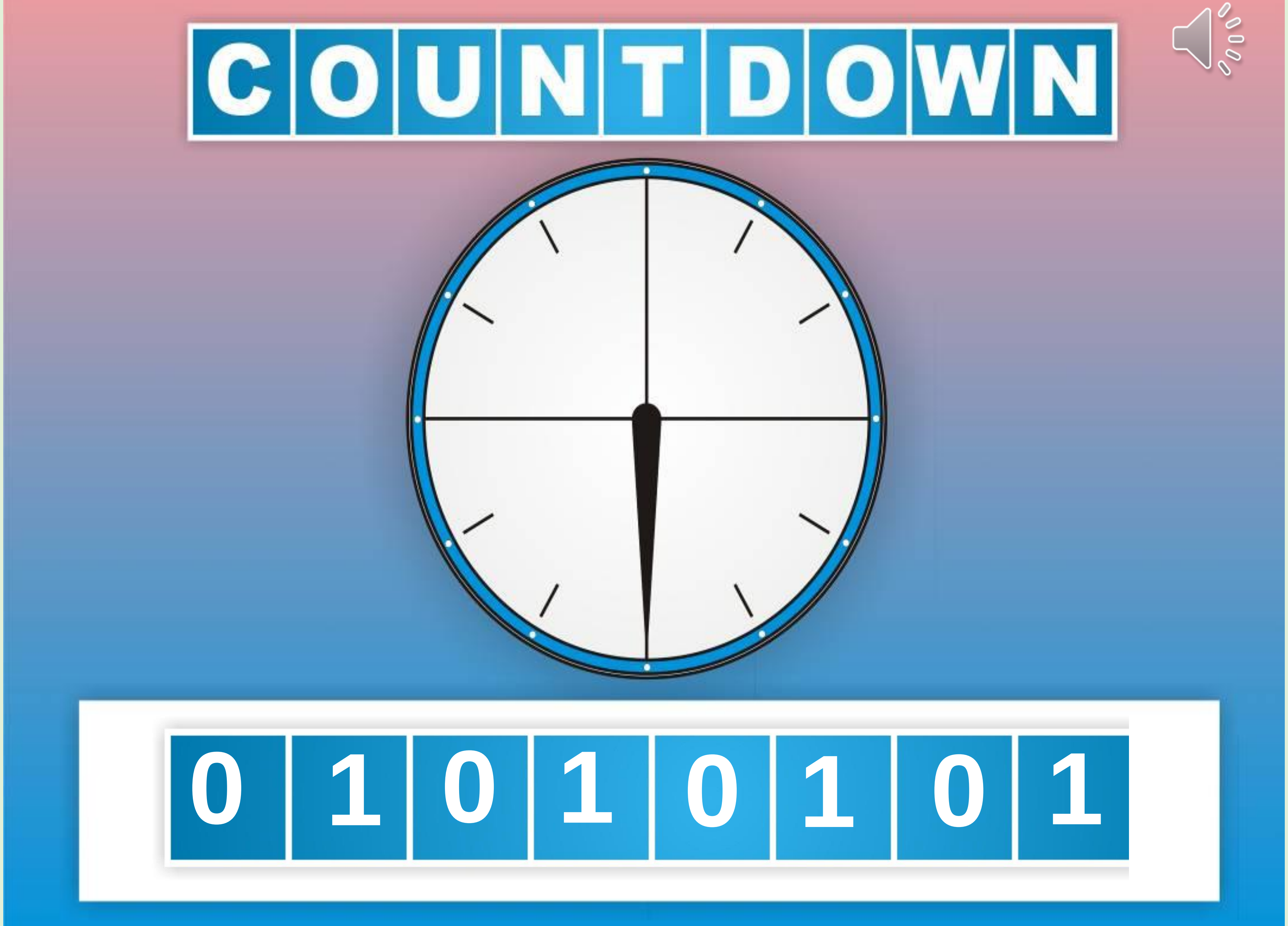


ANSWER

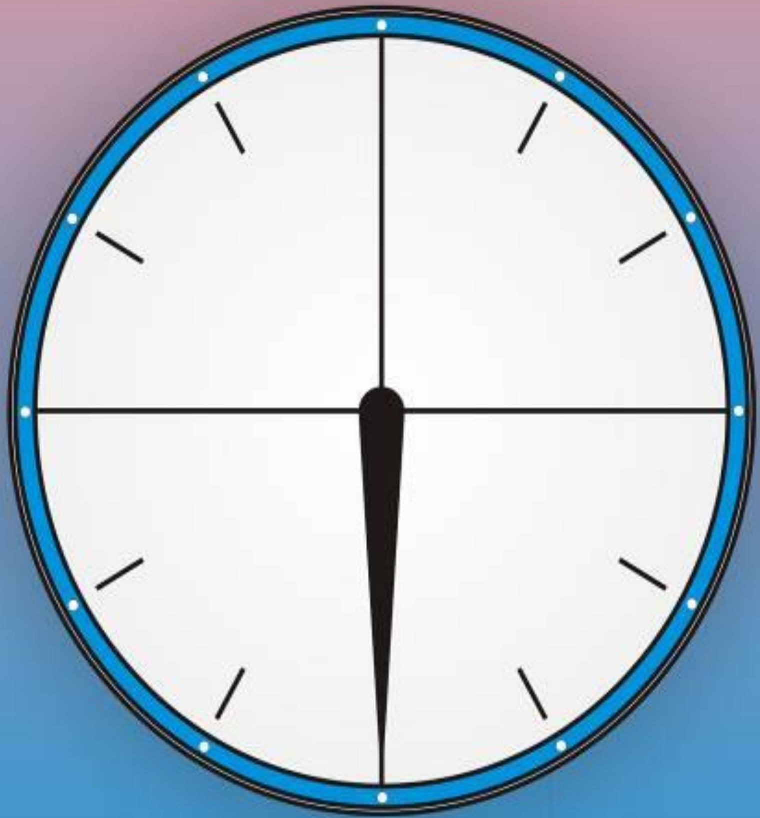
2 1

COUNTDOWN

Ready!


COUNTDOWN
0 1 0 1 0 1 0 1

COUNTDOWN



ANSWER

8 5

W10/ L2

14th November 2019

LO: Develop ability to add binary numbers

Make notes on the rules for **binary addition**

Practise using the rules to add binary numbers

Carry out a binary addition that results in **overflow**

Do Now: Log into Seneca so I know you have your username and password! Request password from Seneca if you have forgotten.

Add Binary Numbers

- The rules for **binary addition**:
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 1 = 0 carry 1
  - 1 + 1 + 1 = 1 carry 1
- Add the binary equivalents of denary 4 + 5.

Denary	Binary			
4	0	1	0	0
5	0	1	0	1
= 9	1	0	0	1
Carry	1			

What is the rule?

- I will give you a simple binary equation to perform
- You have to give the answer (using binary addition rules)
- So the start you off…

0 + 1 is…?

Another example

Denary	Binary			
3	0	0	1	1
10	1	0	1	0
= 13	1	1	0	1
Carry		1		

Another example

Denary	Binary			
14	1	1	1	0
4	0	1	0	0
= 18	0	0	1	0
Carry	1			

1 to carry but no space available in the nibble so the data is lost!

This is known as an **overflow error**.

Problem with a Byte

- Binary addition can run into problems.
- Suppose we only have eight bits in each memory location.
- When we add the binary equivalent of denary 150 + 145:

Denary	Binary								
150		1	0	0	1	0	1	1	0
+145		1	0	0	1	0	0	0	1
= 39		0	0	1	0	0	1	1	1
Carry	1			1					

overflow

- There is no room for a carry so it is lost and we get the wrong answer, 39 instead of 295.
- When there isn't enough room for a result, this is called ***overflow*** and produces an **overflow error**.

W10/ L3

15th November 2019

LO: Develop understanding of the hexadecimal numbering system

Make notes on the hex number system

Practise converting denary to hex

Practise converting hex to denary

Do Now: Write down up to 3 things you can remember from last lesson and then check in your notes

Hexadecimal

- Large binary numbers are quite difficult to remember so programmers often write numbers down in **hexadecimal (hex)** form.
- Hexadecimal numbers are based on the number 16.
- They have 16 different digits:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DENARY	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Binary to Hexadecimal

- To convert from binary to hexadecimal is straightforward.
- Simply take each group of four binary digits, (nibble) starting from the right and translate into the equivalent hex number.

Place values	*8*	*4*	*2*	*1*		*8*	*4*	*2*	*1*
Binary	**1**	**1**	**1**	**1**		**0**	**0**	**1**	**1**
Hex	F					3			

Hexadecimal to Binary

- To convert from hexadecimal to binary simply reverse the process, though you may prefer to go via denary. Treat each digit separately.

Hex	**D**					**B**			
Denary	13					11			
Binary	1	1	0	1		1	0	1	1
Place values	*8*	*4*	*2*	*1*		*8*	*4*	*2*	*1*

- So DB in hexadecimal is 11011011 in binary.

Progress Point – Converting Hex

At the back of your notebooks:

1. Convert AE to binary
2. Convert 11010000 to HEX

Show your working out!

Why use Hexadecimal?

- Large binary numbers are hard to remember
- Programmers use hexadecimal values because:
  - each digit represents exactly 4 binary digits;
  - hexadecimal is a useful shorthand for binary numbers;
  - hexadecimal still uses a multiple of 2, making conversion easier whilst being easy to understand;
  - converting between denary and binary is relatively complex;
  - hexadecimal is much easier to remember and recognise than binary;
  - this saves effort and reduces the chances of making a mistake.
- <u>PICK 2 TO REMEMBER</u>

W11/ L1

20th November 2019

LO: Develop understanding of arithmetic shift

Make notes on binary shift left → Make notes on binary shift right → Practise multiplying and dividing binary numbers

<u>Do Now</u>: Write down up to 3 things you can remember from last lesson and then check in your notes

Multiplying and dividing binary numbers using binary shifts

- https://www.bbc.co.uk/bitesize/guides/zd88jty/revision/4

What is arithmetic shift?

- A method of multiplying/dividing a binary number.
- **To multiply a number**, an arithmetic binary shift moves all the digits in the binary number along to the left and fills the gaps after the shift with 0:
- to multiply by two, all digits shift one place to the left
- to multiply by four, all digits shift two places to the left
- to multiply by eight, all digits shift three places to the left
- and so on

Example of multiplying a binary number by 2

Start with binary 00011100, which is denary 28:

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

Arithmetic shift left starting with the MSB:

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

The left-most value in a binary number is called the
<u>most significant bit (MSB)</u>. It holds the highest place value.

Example of multiplying a binary number by 2

Start with binary 11111111, which is denary 255

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

Arithmetic shift left

256	128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1	0

End with binary 00111000, which is denary 510

Example of multiplying a binary number by 2

Start with binary 11111111, which is denary 255

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

Arithmetic shift left

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

End with binary 11111110, which is denary 254 – overflow error!

What is arithmetic shift?

- **To divide a number**, an arithmetic binary shift moves all the digits in the binary number along to the right and fills the gaps after the shift with the previous MSB value:
- to divide by two, all digits shift one place to the right
- to divide by four, all digits shift two places to the right
- to divide by eight, all digits shift three places to the right
- and so on

The left-most value in a binary number is called the **most significant bit (MSB)**. It holds the highest place value.

Example of multiplying a binary number by 2

Start with binary 00111000, which is denary 56

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

Arithmetic shift right

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

End with binary 00011100, which is denary 28

Progress Point

Perform an arithmetic shift on the numbers below and state the effect of each of these operations:

- Arithmetic shift left by one place on 01010101

2 marks

- Arithmetic shift right by one place on 10010110

2 marks

W11/ L2

21st November 2019

LO: Develop understanding of the role of the operating system

- Create a mind map on six roles of the OS
- Create images to help you recall the roles
- Check recall using your visual memory

Do Now: Write down up to 3 things you can remember from last lesson and then check in your notes

The role of the operating system

FEATURE	FAT32	NTFS
Max. File Name	8.3 Characters	255 Characters
Max. File Size	4GB	16TB
File/Folder Encryption	No	Yes
Fault Tolerance	No	Auto Repair
Security	Only Network	Local and Network
Compression	No	Yes
Conversion	Possible	Not Allowed
Compatibility	Win 95/98/2K/2K3/XP	Win NT/2K/XP/Vista/7

- **Manages periph**
  - The OS mana … scanner
  - The OS mana … .g. a microphone
- **Manages printi**
  - Data sent to p
- **Manages storag**
  - Ensures that … the filing system (e.g. FAT or N
- **Manages memo**
  - Ensures that … mory locations
- **Manages securi**
  - Allows creatio
  - Allows users t
- **Provides user in … sitive, voice-driven and menu-driven)**
  - Allows users to interact with a computer system through graphical icons.

W11/ L3

22nd November 2019

LO: Develop ability to explain sound sampling

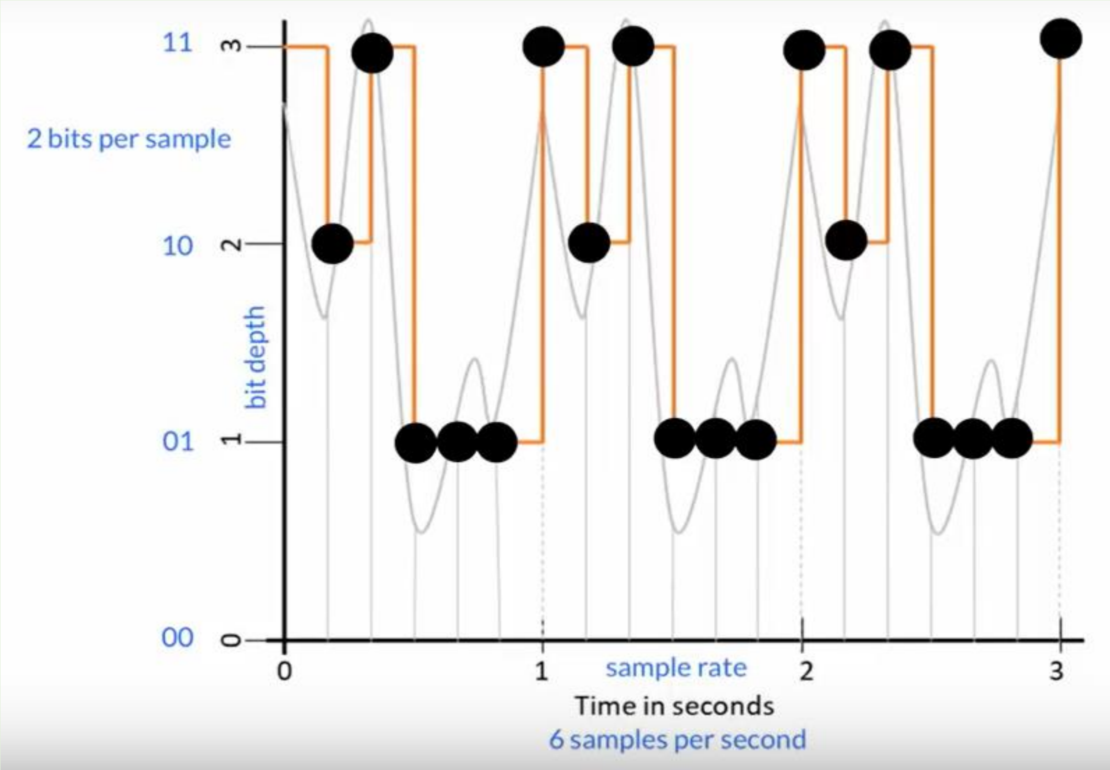
- Watch a video about sound sampling
- Make notes on sound sampling
- Practise exam question on sound sampling

<u>Do Now</u>: Write down up to 3 things you can remember from last lesson and then check in your notes

Watch a video!

- https://student.craigndave.org/videos/ocr-gcse-slr2-6-sound

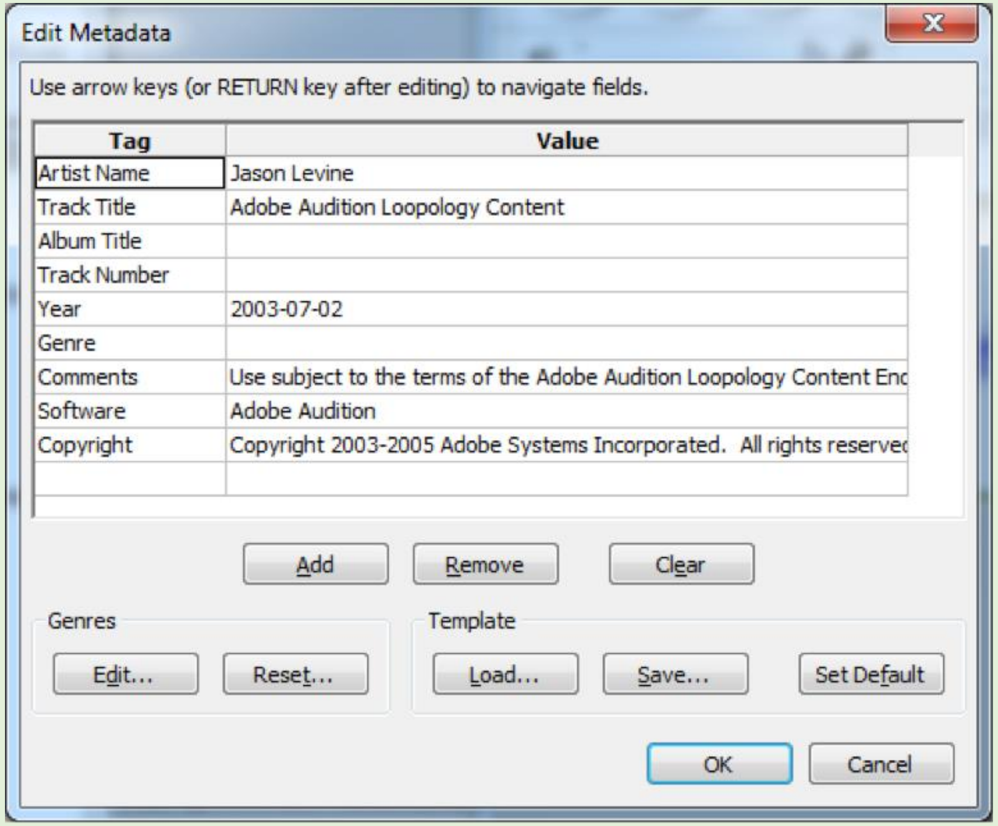
What is sound sampling?



- A method of converting an analogue sound signal into a digital file
- At specific intervals (**frequency**) a measurement of the amplitude (**bit depth**) of the signal is taken - this measurement is called the **sampling rate**
- The bit depth is the number of bits stored per sample (e.g. 2 bits in the diagram)
- The higher the sampling rate/bit depth, the better the quality of the sound file
- The **sample size** can be calculated:
  - **Bit depth** x **sample rate** x **No of seconds**

What is metadata?

- Sound files usually also contain metadata
- Metadata is data that describes data – and in the case of a sound file, it provides information about the sound, including:
  - Artist name
  - Album name
  - Duration
  - Sample rate
  - Bit depth
  - Date created



W12/ L1

27th November 2019

LO: Develop understanding of compression techniques

- Make notes on the purpose of compression
- Watch a video on compression techniques
- Make notes on the difference between lossy and lossless

Do Now: Write down up to 3 things you can remember from last lesson and then check in your notes

What is compression

- Compression is used to reduce file size, which is particularly useful when files need to be downloaded or streamed from the Internet.
- There are 2 main techniques:
  - Lossy
  - Lossless
- Let's learn more by watching a video…
- https://student.craigndave.org/videos/ocr-gcse-slr2-6-compression

How to calculate compression ratio

$$Compression\ ratio = \frac{Original\ file\ size}{Compressed\ file\ size}$$

For example:

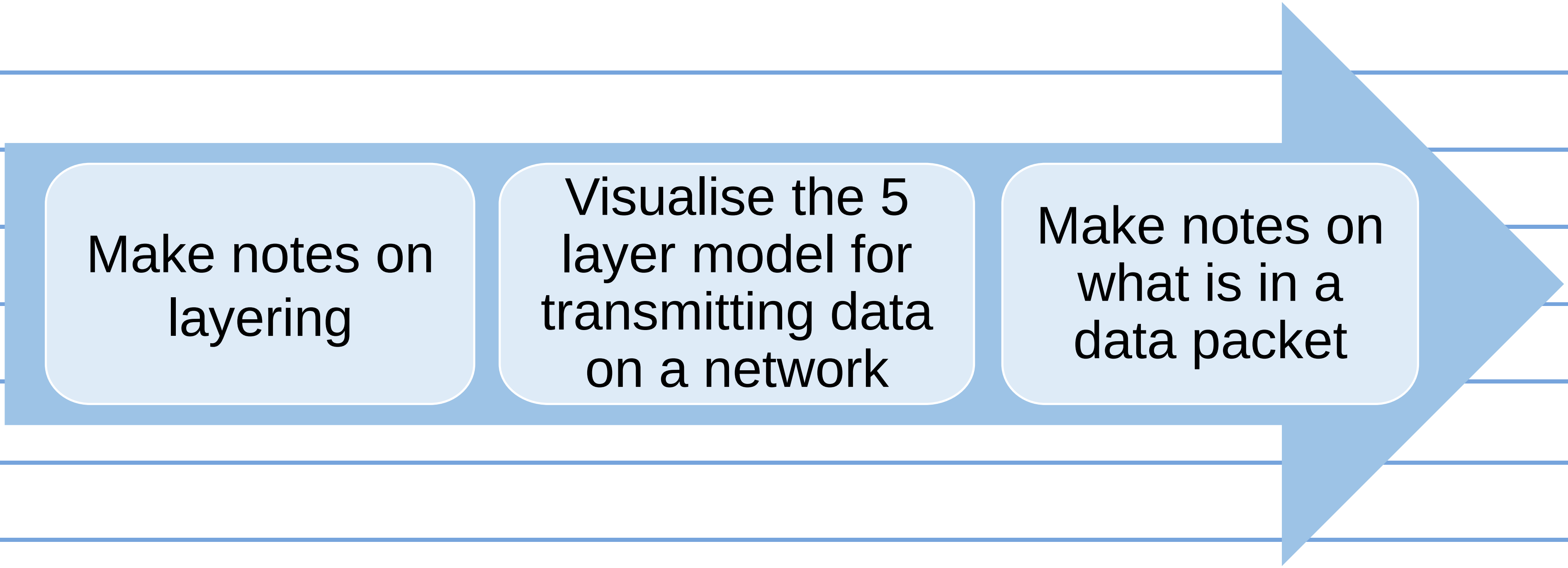
$$Compression\ ratio = \frac{960\ KB}{80\ KB}$$

$$Compression\ ratio = \frac{12\ KB}{1\ KB} = 12 : 1$$

W12/ L2

28th November 2019

LO: Develop understanding of the TCP/IP 5 layer model



Do Now: Write down up to 3 things you can remember from last lesson and then check in your notes

Network layering

- A network is made when 2 or more computers are connected in order to exchange messages.
- Networking is a very complex operation so we need rules for how the computers communicate with each other.
- **Layering** is a term used to describe the processes involved in sending messages across a network.
- The **TCP/IP 5 Layer Model** is used describe how messages are sent over the Internet.
- You need to know the function of each of the 5 layers and the protocols (rules) used at each layer.

TCP/IP 5 Layer Model

- **Application Layer** (protocols include HTTP, FTP, SMTP, and POP3) - This layer encodes the message into a form that will be understood by the recipient device.
- **Transport Layer** (TCP protocol) - This layer breaks down the message into small pieces called **packets**. Each packet is given a packet number and the total number of packets (also called a *checksum*). The recipient uses this information to assemble the packets in the correct order. It also allows the recipient to see if there are any missing packets.
- **Internet Layer** (IP protocol) - This layer adds the **source** IP address and **destination** IP addresses to the data packets. The network then knows where to send them, and where they came from, so they can be routed successfully.
- **Data Link Layer** (Ethernet protocol) - This layer formats (or *frames*) the data packets ready for the physical layer to transmit them.
- **Physical Layer** - This layer transfers the data over a physical connection (i.e. using wires and a NIC or router).

Typical contents of a data packet

<table>
<tr><td>Source Address</td><td>Destination Address</td></tr>
<tr><td colspan="2">Packet order number</td></tr>
<tr><td colspan="2">Other tracking information</td></tr>
<tr><td>The data itself</td><td>Total number of packets/checksum</td></tr>
</table>