Activity 1.6

Convert the following denary numbers into hexadecimal:

a	98	f	1000
b	227	g	2634
c	490	h	3743
d	511	i	4007
e	826	j	5000

1.1.3 Use of the hexadecimal system

As we have seen, a computer can only work with binary data. Whilst computer scientists can work with binary, they find hexadecimal to be more convenient to use. This is because one hex digit represents four binary digits. A complex binary number, such as 1101001010101111 can be written in hex as D2AF. The hex number is far easier for humans to remember, copy and work with. This section reviews four uses of the hexadecimal system:

- error codes
- MAC addresses
- IPv6 addresses
- HTML colour codes

The information in this section gives the reader sufficient grounding in each topic at this level. Further material can be found by searching the internet, but be careful that you don't go off at a tangent.

Error codes

Error codes are often shown as hexadecimal values. These numbers refer to the memory location of the error and are usually automatically generated by the computer. The programmer needs to know how to interpret the hexadecimal error codes. Examples of error codes from a Windows system are shown below:

```
e HexErrorCode ErrorDescription
- ------------- ----------------
0 0x0           Success
1 0x1           Incorrect function.
2 0x2           The system cannot find the file specified.
3 0x3           The system cannot find the path specified.
4 0x4           The system cannot open the file.
5 0x5           Access is denied.
6 0x6           The handle is invalid.
7 0x7           The storage control blocks were destroyed.
8 0x8           Not enough storage is available to process this command.
9 0x9           The storage control block address is invalid.
0 0xa           Unit test error string
1 0xb           An attempt was made to load a program with an incorrect format.
2 0xc           The access code is invalid.
3 0xd           The data is invalid.
4 0xe           Not enough storage is available to complete this operation.
5 0xf           The system cannot find the drive specified.
6 0x10          The directory cannot be removed.
7 0x11          The system cannot move the file to a different disk drive.
8 0x12          There are no more files.
9 0x13          The media is write protected.
0 0x14          The system cannot find the device specified.
1 0x15          The device is not ready.
2 0x16          The device does not recognize the command.
3 0x17          Data error (cyclic redundancy check).
4 0x18          The program issued a command but the command length is incorrect.
5 0x19          The drive cannot locate a specific area or track on the disk.
6 0x1a          The specified disk or diskette cannot be accessed.
7 0x1b          The drive cannot find the sector requested.
8 0x1c          The printer is out of paper.
9 0x1d          The system cannot write to the specified device.
0 0x1e          The system cannot read from the specified device.
1 0x1f          A device attached to the system is not functioning.
2 0x20          The process cannot access the file because it is being used by another process.
3 0x21          The process cannot access the file because another process has locked a portion of the file.
4 0x22          The wrong diskette is in the drive....
6 0x24          Too many files opened for sharing.
8 0x26          Reached the end of the file.
9 0x27          The disk is full.
0 0x32          The request is not supported.
```

▲ **Figure 1.3** Example of error codes

Find out more

Another method used to trace errors during program development is to use memory dumps, where the memory contents are printed out either on screen or using a printer. Find examples of memory dumps and find out why these are a very useful tool for program developers.

Media Access Control (MAC) addresses

Media Access Control (MAC) address refers to a number which uniquely identifies a device on a network. The MAC address refers to the network interface card (NIC) which is part of the device. The MAC address is rarely changed so that a particular device can always be identified no matter where it is.

A MAC address is usually made up of 48 bits which are shown as 6 groups of two hexadecimal digits (although 64-bit addresses also exist):

```
NN – NN – NN – DD – DD – DD
```

or

```
NN:NN:NN:DD:DD:DD
```

where the first half (NN – NN – NN) is the identity number of the manufacturer of the device and the second half (DD – DD – DD) is the serial number of the device. For example:

00 – 1C – B3 – 4F – 25 – FE is the MAC address of a device produced by the Apple Corporation (code: 001CB3) with a serial number of: 4F25FE. Very often lowercase hexadecimal letters are used in the MAC address: 00-1c-b3-4f-25-fe. Other manufacturer identification numbers include:

```
00 – 14 – 22  which identifies devices made by Dell

00 – 40 – 96  which identifies devices made by Cisco

00 – a0 – c9  which identifies devices made by Intel, and so on.
```

Link

Refer to Chapter 3 for more detail on MAC addresses.

Find out more

Try to find the MAC addresses of some of your own devices (e.g. mobile phone and tablet) and those found in the school.

Internet Protocol (IP) addresses

Each device connected to a network is given an address known as the **Internet Protocol (IP) address**. An IPv4 address is a 32-bit number written in denary or hexadecimal form: e.g. 109.108.158.1 (or 77.76.9e.01 in hex). IPv4 has recently been improved upon by the adoption of IPv6. An IPv6 address is a 128-bit number broken down into 16-bit chunks, represented by a hexadecimal number. For example:

```
a8fb:7a88:fff0:0fff:3d21:2085:66fb:f0fa
```

Note IPv6 uses a colon (:) rather than a decimal point (.) as used in IPv4.

Link

Refer to Chapter 3 for more detail on IP addresses.

Find out more

Try to find the IPv4 and IPv6 addresses of some of your own devices (e.g. mobile phone and tablet) and those found in the school.

HyperText Mark-up Language (HTML) colour codes

HyperText Mark-up Language (HTML) is used when writing and developing web pages. HTML isn't a programming language but is simply a mark-up language. A mark-up language is used in the processing, definition and presentation of text (for example, specifying the colour of the text).

HTML uses `<tags>` which are used to bracket a piece of text for example, <h1> and </h1> surround a top-level heading. Whatever is between the two tags has been defined as heading level 1. Here is a short example of HTML code:

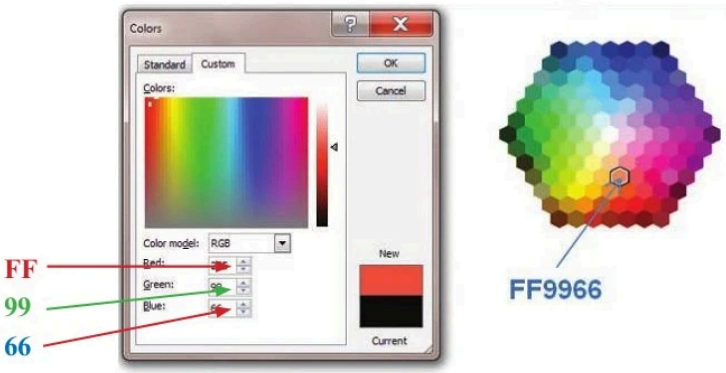
```
<h1 style="color:#FF0000;">This is a red heading</h1>

<h2 style="color:#00FF00;">This is a green heading</h2>

<h3 style="color:#0000FF;">This is a blue heading</h3>
```

▲ **Figure 1.4**

HTML is often used to represent colours of text on the computer screen. All colours can be made up of different combinations of the three primary colours (red, green and blue). The different intensity of each colour (red, green and blue) is determined by its hexadecimal value. This means different hexadecimal values represent different colours. For example:

- # FF 00 00 represents primary colour red
- # 00 FF 00 represents primary colour green
- # 00 00 FF represents primary colour blue
- # FF 00 FF represents fuchsia
- # FF 80 00 represents orange
- # B1 89 04 represents a tan colour,

and so on producing almost any colour the user wants. The following diagrams show the various colours that can be selected by altering the hex 'intensity' of red, green and blue primary colours. The colour ***'FF9966'*** has been chosen as an example:



▲ **Figure 1.5** Examples of HTML hex colour codes

The # symbol always precedes hexadecimal values in HTML code. The colour codes are always six hexadecimal digits representing the red, green and blue components. There are a possible 256 values for red, 256 values for green and 256 values for blue giving a total of 256 × 256 × 256 (i.e. 16 777 216) possible colours.