

1.4.1

DATA TYPES

TOPIC WISE EXAM QUESTIONS

A-LEVEL

OCR

- a) Primitive data types, integer, real/ floating point, character, string and Boolean.
- b) Represent positive integers in binary.
- c) Use of sign and magnitude and two's complement to represent negative numbers in binary.
- d) Addition and subtraction of binary integers.
- e) Represent positive integers in hexadecimal.
- f) Convert positive integers between Binary Hexadecimal and denary.
- g) Representation and normalisation of floating point numbers in binary.
- h) Floating point arithmetic, positive and negative numbers, addition and subtraction.
- i) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR.
- j) How character sets (ASCII and UNICODE) are used to represent text.

Candidates need to have an understanding of programming data types such as integer, real, Boolean, character string etc. Candidates need to be able to choose appropriate data types for a situation or given data. Candidates should have experience of programming solutions using these data types. Candidates should have knowledge of how to convert from one data type to another. Candidates should understand how and why computers store data as binary using switches, and that a binary number can have a variety of different interpretations depending on what is being stored (e.g. numeric, text, image, sound).

Candidates should be able to convert positive whole numbers to binary and from binary to decimal/denary.

Candidates should know how to store negative numbers using Sign and Magnitude and two's complement. Candidates should be able to convert denary numbers to sign and magnitude, and two's complement – and vice-versa.

Candidates should be able to perform addition and subtraction on integer binary numbers. (These numbers could be positive or negative using two's complement representation.)

Candidates need to have an understanding of the purpose and potential uses of Hexadecimal for example where and why they are used instead of binary and the benefits of using Hexadecimal over alternatives such as binary. Candidates should be able to convert denary numbers to hexadecimal and vice-versa and from binary to hexadecimal and vice-versa.

Candidates should have an understanding of how (positive and negative) real numbers are represented in a binary floating-point representation, and should be able to convert between a denary number and a real binary number. (NB the representation used for the exam is the mantissa and exponent both represented using two's complement.)

Candidates should understand the need for normalised floating point numbers. Candidates should be able to normalise a floating point number.

Candidates should be able to perform addition and subtraction floating point arithmetic including addition and subtraction of both positive and negative numbers.

Candidates should be able to perform right and left logical shifts. Candidates should understand the effect of right and left shifts on a binary numbers.

Candidates should understand the purpose of using masks with bitwise operators, and should have experience of applying masks using AND, OR and XOR.

Candidates should have an understanding of how characters are represented in a computer and in binary. Candidates should understand the need for a character set and how a computer makes use of a character set. Candidates should be aware of the ASCII and UNICODE character sets and be able to explain the differences between these and how these impact what can be stored.

Candidates should be able to use a character set, or part of a character set, to translate characters into binary and vice-versa. (Candidates are not expected to memorise any values in a character set).

3 (a) (i) Convert the denary number **189** to hexadecimal.

.....
.....
.....
..... [1]

(ii) Convert the unsigned binary number **1010101111** to hexadecimal.

.....
.....
..... [1]

(b) Negative binary values can be represented using either sign and magnitude or two's complement.

(i) Convert the denary number **-107** to an 8-bit binary number using sign and magnitude.

.....
.....
..... [1]

(ii) Convert the denary number **-107** to an 8-bit binary number using two's complement.

.....
.....
..... [1]

(iii) Give **one** advantage of storing values using two's complement instead of sign and magnitude.

.....
..... [1]

3 (a) Describe what is meant by the term 'character set'.

.....
.....
.....
..... [2]

(b) (i) Convert the hexadecimal number 66 into a denary number. Show your working.

.....
.....
.....
..... [2]

(ii) State **two** reasons why a programmer would prefer to use hexadecimal numbers rather than binary numbers.

1
.....
2
..... [2]

(c) Show the denary value 6.25 as a floating point binary number representing the mantissa and exponent. Both of these should be stored in two's complement representation.

You should use as few bits as possible.

Show your working.

.....
.....
.....
.....
.....
..... [3]

(d) State the benefit of using a normalised form when representing data as a floating point number.

.....
..... [1]

3 (a) (i) Convert the hexadecimal value **B7E** to a binary number.

.....
..... [1]

(ii) 110010101 is a binary number that is represented using sign and magnitude.

Convert this binary number to a denary number.

.....
..... [1]

(iii) Complete this binary subtraction. Both numbers are 8-bit integer values represented using two's complement.

Show the result in the same format and show your working.

$$\begin{array}{r} 0110\ 1101 \quad - \\ 0011\ 0100 \end{array}$$

.....
.....
.....
.....
.....
..... [3]

- 5 (e) The programmer creates another function to count and return how many capital letters are in a string that is passed into the function as a parameter.

The `asc()` function takes in a character and returns its ASCII value. For example `asc("A")` returns 65. Capital letters have ASCII values between 65 and 90 inclusive.

- (i) Complete the function below.

```
function countCapitals(text)
    // initialise counter to 0
    capCount = 0
    // loop through each character in the string passed in
    for x = 0 to text.length-1
        c = text.subString(x, 1)
        // check if character is a capital
        if asc(c) >= 65 .....
            // if so, increment counter
            .....
        endif
    next x
    .....
endfunction
```

[3]

- (ii) Give **one** similarity between ASCII and Unicode.

.....
..... [1]

- (iii) Give **two** differences between ASCII and Unicode.

Difference 1

.....

Difference 2

.....

[2]

4 (a) State why computer systems store data in binary.

.....
..... [1]

(b) (i) Convert the denary number 97 into an 8-bit binary number.

.....
.....
.....
..... [1]

(ii) Convert the denary number 171 into a hexadecimal number.

.....
.....
.....
..... [1]

(iii) Convert the denary number -97 into an 8-bit binary number using two's complement.

.....
.....
.....
..... [1]

(iv) Convert the denary number -17 into an 8-bit binary number using sign and magnitude.

.....
.....
.....
..... [1]

(c) State **one** advantage of using two's complement instead of sign and magnitude.

.....
.....
.....
..... [2]

(d) Calculate the addition of these two 8-bit (unsigned) binary numbers.

Show your working.

$$\begin{array}{r} 11011011 \\ 10001001 + \\ \hline \end{array}$$

2 (a) (i) Convert the denary number 231 to an unsigned 8-bit binary number.

.....
.....
.....
..... [1]

(ii) Convert the hexadecimal number 6F to an unsigned 8-bit binary number.

.....
.....
.....
..... [1]

(b) The floating point number 0011010100 is stored using 6 bits for the mantissa and 4 bits for the exponent, both in two's complement. This number is not normalised.

(i) Give the normalised version of this number, showing your working.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
..... [4]

- 3 (a) (i) Convert the denary number -119 to an 8-bit binary number with two's complement representation.

.....
.....
.....
..... [1]

- (ii) Convert the unsigned binary number 1101101 to hexadecimal.

.....
.....
.....
..... [1]

- (iii) Convert the denary number 171 to hexadecimal.

.....
.....
.....
..... [1]

- (iv) Convert the hexadecimal number $A6$ to binary.

.....
.....
.....
..... [2]

5 (a) Convert the binary number 11101100 into a denary number.

.....
.....
.....
..... [1]

(b) Convert the binary number 10110011 into a hexadecimal number.

.....
.....
.....
..... [1]

(c) Convert the two's complement binary number 10011011 into a denary number.

.....
.....
.....
..... [1]

(d) Calculate the subtraction of the following two 8-bit binary numbers.

You must show your working.

$$\begin{array}{r} 11010011 \\ \underline{01111001} - \end{array}$$

[2]

8 A simple program is shown below.

```
//Program to calculate number of times
//a number goes into 100
count = 0
num = int(input("Enter a number"))
while (count*num)<=100
    count=count+1
endwhile
count=count-1 //Take one off as gone over
print(str(num) + " goes into 100 " + str(count) + " times.")
```

Fig. 8.1

(a) State the output of the program when the number 30 is entered.

.....
..... [1]

(b) State the most suitable data type of the variable `count`

.....
..... [1]

(c) State the data type of the result of the expression `(count*num)<=100`

.....
..... [1]

(d) State the data type of the result of the expression

```
str(num) + " goes into 100 " + str(count) + " times."
```

.....
..... [1]

10 (a) Show how the binary number 01011110 is represented in hexadecimal.

.....
.....
.....
..... [1]

(b) Show how the hexadecimal number 9B is represented in denary.

.....
.....
..... [2]

(c) Show how the denary number -87 is represented in sign and magnitude binary.

.....
.....
..... [2]

(d) Complete the following binary subtraction. Show your working.

$$\begin{array}{r} 01001001 - \\ \underline{00101111} \end{array}$$

(e) The floating point binary number 010011 011 consists of a 6-bit mantissa and 3-bit exponent, both represented in two's complement. Convert the number to denary, showing your working.

..... [2]
.....
.....
..... [3]

(f) Show the denary number -5.25 in floating point binary form representing the mantissa and exponent in two's complement, using as few bits as possible. Show your working.

.....
.....
.....
.....
..... [4]

2 Variables in programs contain specific types of data.

(a) Complete the table below to suggest a suitable data type for each piece of data.

Data	Data Type
'H'	Character
"Hello"	
35	
-2.625	Real
True	

[3]

(b) Show the denary number 35 as an 8-bit (unsigned) binary number.

.....
..... [1]

(c) The character 'A' in the ASCII character set is represented by the denary value 65. Write the binary representation for the ASCII character 'H'. Show your working.

.....
.....
.....
..... [2]

(d) Show the denary number $-2\frac{3}{8}$ as a floating-point binary number with a 6-bit mantissa and 4-bit exponent, both stored using two's complement representation.

.....
.....
.....
.....
.....
..... [3]

9 (a) Demonstrate how the bytes below are added together. Show your working.

01101010
00111111+

[2]

(b) Demonstrate how the bottom byte below is subtracted from the top byte. Show your working.

11001111
00111001 -

[2]

(c) Convert the binary number shown below to hexadecimal.

0011011100001111

.....
.....
.....
..... [2]

(d) The number below is represented in floating point format with a 5-bit mantissa in two's complement followed by a 3-bit exponent in two's complement. Calculate the denary value of the number, showing your working.

01001 010

.....
.....
.....
.....
.....
..... [3]

- (e) The numbers below are represented in floating point format with a 5-bit mantissa in two's complement followed by a 4-bit exponent in two's complement. Normalise the numbers shown below, showing your working.

00011 0010

.....
.....
.....
..... [2]

11100 0110

.....
.....
.....
..... [2]

- (f) Show the byte below after having an AND applied with the masking byte.

Byte	1	0	1	1	1	0	0	1
AND	1	1	1	1	1	1	1	1
Result								

[1]

- (g) Show the byte below after having an OR applied with the masking byte.

Byte	1	0	1	1	1	0	0	1
OR	1	1	1	1	1	1	1	1
Result								

[1]

5 (a) Convert the denary number 72 to an unsigned 8-bit integer.

.....
.....
.....
..... [1]

(b) Convert the unsigned binary number 10000101 to denary.

.....
.....
.....
..... [1]

(c) Convert the denary number 104 to hexadecimal.

.....
.....
.....
..... [1]

(d) The following floating point binary number is represented using 6 bits for the mantissa and 4 bits for the exponent, both using two's complement.

Mantissa	Exponent
0100101	0100

Convert the number to denary, showing your working.

.....
.....
.....
.....
.....
.....
..... [3]

(e) Given that computers store everything in binary, explain how they are able to represent text.

.....
.....
.....
..... [2]

6 (a) (i) Convert the denary number 188 to an unsigned 8-bit binary number.

.....
.....
.....
..... [1]

(ii) Convert the denary number 188 to hexadecimal.

.....
.....
.....
..... [1]

(b) (i) Convert the denary number -44 to an 8-bit binary number with sign and magnitude representation.

.....
.....
.....
..... [1]

(ii) Convert the denary number -44 to an 8-bit binary number with two's complement representation.

.....
.....
.....
..... [1]

(c) Explain how, using bit shift, the unsigned binary number 00101100 can be divided by 4.

.....
.....
.....
..... [2]

5 (a) Convert the binary number 01101111 to a hexadecimal number.

.....
..... [1]

(b) Convert the denary number -19 to an 8-bit number using:

(i) Two's complement representation.

.....
..... [1]

(ii) Sign and Magnitude representation.

.....
..... [1]

(c) The two values below are stored using unsigned binary. Calculate the subtraction of 01110010 from 11000011. Show your working.

$$\begin{array}{r} 11000011 \\ - 01110010 \\ \hline \end{array}$$

[2]

(d) Convert the denary number $1\frac{5}{8}$ (i.e. 1.625) to a normalised floating point binary number using 5 bits for the mantissa and 3 bits for the exponent. Show your working.

.....
.....
.....
.....
.....
.....
..... [3]

5 (a) Give the number 55 in binary as an 8-bit unsigned integer.

.....
.....
.....
..... [2]

(b) Represent the number 55 in normalised floating point binary notation, using 8 bits for the mantissa followed by 8 bits for the exponent, both in two's complement binary.

.....
.....
.....
..... [2]

(c) Represent the number 55 in normalised floating point binary notation, with the mantissa and exponent both in two's complement binary, using as few bits as possible.

.....
.....
.....
..... [2]

(d) State why a programmer might choose to declare a variable as a floating point number.

.....
.....
.....
..... [1]

3(a). An AND operation with the mask 10101010 is applied to the binary number 01010101. Show the result.

01010101
10101010 AND

[1]

(b). An OR operation with the mask 10101010 is applied to the binary number 01010101. Show the result.

01010101
10101010 OR

[1]

4. Two floating point numbers are shown below. Calculate the answer of the second number subtracted from the first. You must show your working and ensure your answer is normalised.

01001100 0011 - 01001010 0010

[5]

**If you found this
useful, drop a follow
to help me out!**

THANK YOU!

GCST