1. (a) Write the binary number 11010 as a sum of distinct powers of 2. Describe explicitly the way in which the answer is obtained. (**3** marks)

   (b) A binary fraction has the form $a.b$, where a and b are finite bit strings, the dot is the radix point and at least one of a, b is not empty. Write the binary fraction 1001.11 as a sum of distinct powers of 2. Describe explicitly the way in which the answer is obtained. (**4** marks)

   (c) Add the binary fractions 10.1 and 110.01. Show the working including any carries. (**4** marks)

2. (a) Subtract the binary number 110 from the binary number 11011. Describe explicitly the way in which the answer is obtained. In particular, identify any places in the calculation where a borrow occurs. (**3** marks)

   (b) Multiply the binary numbers 11 and 1011. Describe explicitly the way in which the answer is obtained. If the calculation involves an addition, then describe any carries explicitly. (**3** marks)

   (c) The hexadecimal system for representing integers has 16 digits, namely the usual digits 0 to 9 and the letters A to F. Each hexadecimal digit corresponds to a binary bit string of length 4. For example 1 corresponds to the bit string 0001 and A corresponds to the bit string 1010. Convert the binary numbers 10011101 and 110110 to hexadecimal numbers. Describe explicitly the way in which the answers are obtained. (**5** marks)

3. (a) The Brookshear floating point representation for a binary fraction x consists of eight bits, labeled s, e_1, e_2, e_3, m_1, m_2, m_3, m_4 from left to right. If x is zero, then all eight bits are zero. If x is strictly negative, then the sign bit s is 1. If x is strictly positive, then the bit s is zero. Next, suppose x is not zero. To obtain the remaining seven bits, x is written in the form

$$\pm 2^r * 0.t$$

   where r is an integer and t is a bit string such that the leftmost bit of t is 1. The bits e_1, e_2, e_3 together comprise the three bit excess notation for r and the bits m_1, m_2, m_3, m_4 of the mantissa are the leftmost four bits of t.

   Obtain the decimal fraction with the Brookshear representation 11011101. Show the sign, the mantissa and the decimal representation of the exponent r. Describe explicitly the way in which the answers are obtained. (**7** marks)

   (b) Find the largest positive decimal number that has an exact Brookshear representation. Justify your answer. (**4** marks)

4. (a) State one disadvantage that pseudo-code has with respect to conventional programming languages.

(**2** marks)

(b) Consider the following pseudo-code.

```
m = 11
n = 7
q = 0
r = m
while r ≥ n
    r = r − n
    q = q + 1
endWhile
print(q, r)
halt
```

Identify that portion of the code that constitutes a while loop. (**2** marks)

(c) The expression $r \geq n$ in part (b) of this question can have the value $True$ or the value $False$. Explain what happens if the value is $True$. Next, explain what happens if $r \geq n$ has the value $False$. (**4** marks)

(d) How many times is the expression $r \geq n$ in part (b) of this question evaluated?

(**3** marks)

5. (a) Explain what is meant by the expression ”binary Boolean operation”. (**2** marks)

(b) Write out the truth table for the Boolean expression A XOR NOT(B). (**4** marks)

(c) Use the truth table in part (b) of this question to evaluate the Boolean expression

$$(2 > 1) \text{ XOR NOT}(6 < 4)$$

Describe explicitly the way in which the truth table is used. (**2** marks)

(d) Evaluate the following Boolean expressions, given that C is $True$ (1), D is $False$ (0) and the value of E is unknown. Describe explicitly the way in which the answer to (ii) is obtained, even though the value of E is unknown.

i) C OR NOT(D)

ii) D AND $(C$ OR $E)$

(**3** marks)

6. The table included below in this question describes instructions of length 16 bits, made by concatenating an op-code and an operand. The first four bits record the op-code. The remaining 12 bits record the operand. Four bits are required to specify a register R and eight bits are required to specify a memory location XY. Each register holds eight bits and each memory location holds eight bits.

Each 16 bit instruction is coded by four hexadecimal digits. For example, the four hexadecimal digits $37A9$ specify an instruction with op-code 3, in which the 7 refers to register 7 and $A9$ refers to the memory cell $A9$. The registers are numbered in hexadecimal from 0 to F. All memory addresses in this question are given in hexadecimal notation.

Op-code	Operand	Description
1	RXY	Load register R with the bit pattern in memory cell XY.
2	RXY	Load register R with the bit pattern XY.
3	RXY	Store the bit pattern in register R at memory cell XY.
4	$0RS$	Move the bit pattern in register R to register S.
5	RST	Add (two's complement) the bit patterns in registers S and T. Put the result in register R.
6	RST	Add (floating point) the bit patterns in registers S and T. Put the result in register R.
7	RST	Or the bit patterns in registers S and T. Put the result in register R.
8	RST	And the bit patterns in registers S and T. Put the result in register R.
9	RST	Exclusive Or the bit patterns in registers S and T. Put the result in register R.
A	$R0X$	Rotate the bit pattern in register R one bit to the right X times.
B	RXY	Jump: place the address XY in the program counter if the bit pattern in register R is equal to the bit pattern in register 0.
C	000	Halt.

(a) Write a series of instructions to exclusive or the contents of memory cells 6C and 6D and to put the result in memory cell 6E. Ensure that the program halts. Describe the action of each instruction in words. (**5** marks)

(b) Adapt the instructions in part (a) of this question such that the exclusive or of the contents of the cells 6C and 6D is stored in cell 6E if and only if cell 6C does not contain 00. If cell 6C does contain 00, then the contents of cell 6E are unchanged. Ensure that the program halts. Assume if necessary that the halt instruction is stored in memory cells 11 and 12, i.e. cell 11 contains C0 and cell 12 contains 00. Write out the whole program, including the relevant instructions from part (a). Describe the actions of any instructions that do not appear in the answer to part (a) of this question. (**6** marks)

7. Array indexing begins with 0. A two dimensional array A of integers is defined by

$$A = \begin{bmatrix} 4 & 1 & 0 \\ 2 & 8 & 3 \\ 3 & 9 & 6 \end{bmatrix}$$

(a) What is the value of $A[2, 2]$? (**2** marks)

(b) Find the values of the indices i, j such that $A[i, j]$ is equal to 8. (**2** marks)

(c) Suppose that a computer memory consists of cells with addresses in the range 0 to C in hexadecimal notation. Suppose that each memory cell can contain exactly one integer. Describe explicitly how the array A can be stored in this memory in row major order. (**2** marks)

(d) The address polynomial for the row major storage of the array A is $3*i+j$, where 0 is the address of the memory cell that contains $A[0, 0]$. Consider the machine code used in question 6. Suppose that i is stored in register 1 and j is stored in register 2. Write a program using the machine code in question 6 to evaluate the address polynomial and put the result in register 3. Describe the actions of the instructions. (**5** marks)

8. Consider the following pseudo-code for a function.

```
function greetings(count)
   while count < 3
      print("Hello")
      count = count + 1
   endWhile
endFunction
```

(a) Describe in detail the execution of the code when $greetings$ is called with parameter value 2. (**4** marks)

(b) Give an example of a function call to $greetings$. (**2** marks)

(c) Rewrite the pseudo-code for $greetings$ to product a new function called $greetings1$. If the integer $count$ is strictly less than 3 then $greetings1$ has the same behaviour as $greetings$. If $count$ is greater than or equal to 3, then $greetings1$ prints the string "Hello" exactly once. Write out all of the pseudo-code for $greetings1$. (**5** marks)

9. The following pseudo-code defines a function f. The terms A, B are 1-dimensional arrays with integer entries. Array indexing begins with the index 0. The symbol $!=$ is not equals.

```
function f(A, B)
   if length(A) != length(B)
      return False
   endIf
   i = 0
   while i < length(A)
      if A[i] != B[i]
         return False
      endIf
      i = i + 1
   endWhile
   return True
endFunction
```

(a) What is returned by the function f if it is called with the following values of A and B? Note that [] is the empty array, i.e. the array with no elements. In each case state clearly the reason why the function f returns an appropriate value.

i) $A = [2, 3, 1]$, $B = [2, 3, 1]$

ii) $A = [1, 2, 3]$, $B = [1, 2]$

iii) $A = [1]$, $B = []$

iv) $A = []$, $B = []$

(**4** marks)

(b) Rewrite the function f to obtain a new function $f1$. Let A, B be 1-dimensional arrays with integer entries as in part (a) of this question. The function call $f1(A, B)$ returns $True$ if B is obtained by rotating A one place to the left with wrap round. In all other cases $f1(A, B)$ returns $False$. For example, if $A = [3, 4, 5]$ and $B = [4, 5, 3]$ then $f1(A, B)$ returns $True$. Note that the first entry of A becomes the last entry of B. Write out the pseudo-code for $f1(A, B)$ in full. (**8** marks)