Answer ANY THREE questions.
If you answer more, the best three answers will be taken into account.

1. (a) Associate the word or phrase below, labelled (1)-(10), with the description that best fits taken from the sentences labelled (i)-(x). Each description can be used with only one of the words or phrases. Write your answer in the form "(N) means (a)" where (N) is the number of one of the words or phrases (i.e., 1 to 10) and (a) is the number of the sentence (i.e., i to x):

(1)	assembler	(i)	Location of programs currently being executed, including their data
(2)	operating system	(ii)	Small fast memory close to the CPU
(3)	bit	(iii)	Program that translates symbolic instructions into machine code instructions
(4)	cache	(iv)	Integrates up to hundreds of transistors onto a single chip
(5)	central processor unit	(v)	Abstraction describing the operation of the hardware embodied in the assembly language
(6)	compiler	(vi)	Program that translates higher level programs into assembly language
(7)	memory	(vii)	Program that manages the computer resources and allows interface by a user
(8)	instruction set architecture	(viii)	Basic individual command to a computer
(9)	instruction	(ix)	Binary digit, i.e. can be 0 or 1
(10)	integrated circuit	(x)	Part of the computer that executes instructions and maintains overall control

[10]

[QUESTION 1 CONTINUED OVERLEAF]

[TURN OVER]

[QUESTION 1 CONTINUED]

(b) Draw a diagram of the main functional components of a computer system. Your diagram should include any necessary communications lines.

[11]

(c) Using your diagram in (b) explain the fetch and execute cycle of a computer as it operates to execute a program. Illustrate your answer with a program that fetches two numbers, adds them, and places the result into the memory location of the first number.

[12]

[Total 33]

2. This question refers to the MIPS instruction set.

(a) For each of the following MIPS instructions, write corresponding high level (C++) code that might have been translated to these instructions, giving also an explanation for each one. For example, "move $10,$4" means that the content of register $4 is copied into register $10, so that if variables x and y were associated with $4 and $10 respectively, then this instruction might be the ultimate destination of a C++ assignment **y = x;**

(i)	add $1,$2,$3	(vi)	slt $1,$2,$3
(ii)	lw $1,100($2)	(vii)	j 1200
(iii)	sw $1,100($2)	(viii)	slti $1,$2,36
(iv)	beq $1,$2,L	(ix)	addi $1,$2,24
(v)	bne $1,$2,L	(x)	muli $15,$16,4

[14]

(b) The following instructions are taken from an instruction set of the VAX family of computers. For each one show how it can be translated into a corresponding set of MIPS instructions:

(i)	decl $10	this decreases register $10 by 1: $10 = $10 - 1
(ii)	clrl $10	this sets the contents of register $10 to zero
(iii)	clrl 1200	this sets the contents of memory address 1200 to zero

[9]

[QUESTION 2 CONTINUED OVERLEAF]
[CONTINUED]

[QUESTION 2 CONTINUED]

(c) Suppose that there is an array **int x[100]** with **size=100.** Show how to represent each of the following fragments of C++ code in MIPS. You may assume that the memory address of **x** is represented by **Xstart.**

(i) x[5] = x[6] + 1;

(ii) x[5] += 1;

(iii) int i;
for(i=0; i < size; ++i)   x[i] = 0; [10]

[Total 33]

3. (a) Explain in detail how MIPS procedures (with parameters) are defined and processed. [12]

(b) Suppose that there is a procedure `swap(int &a, int &b)` which swaps the values assigned to `a` and `b.` Write this procedure in MIPS, being sure to include comment lines in your code. [10]

(c) The following is pseudo MIPS code that implements a function with one argument n. We assume that (i) the argument (n) is in $4, and (ii) $11 and $8 are initialised to 1.

```
MAIN :      addi $4,$0,4          #the argument n = 4
            jal fact              #
                                  #would eg print result here

fact:       beq $4,$11,RETURN     #
            push($31)             # no such instruction
            push($4)              # no such instruction
            addi $4,$4,-1         #
            jal fact              #
RETURN:     $4 = pop()            #!
            $31 = pop()           #!
            mult $8,$8,$4         #
            jr $31                #
```

Fill in the comment lines, to explain each instruction. Those instructions with the comment “no such instruction” should be rewritten as correct MIPS instructions. Explain what the function does, for example, when **n** is initialised to the value **3.** [11]

[Total 33]

[TURN OVER]

4. (a) Suppose a computer has a word length of **n,** and uses two's complement arithmetic.

   (i) $D = d_{n-1} d_{n-2} \cdots d_1 d_0$ is a binary number where d_{n-1} is the most significant bit, and d_0 is the least significant bit. Write down an expression for the decimal value of D.

   (ii) From (i) find the largest (positive) and smallest (negative) integer numbers that can be represented (expressed in decimal).

   (iii) What is the binary representation of the decimal number -27 in the case where n=8?

   (iv) Show how to evaluate the subtraction 17-34 using two's complement arithmetic when n=8. [14]

   (b) Using the IEEE 32-bit floating point standard, show how the decimal number -0.25 would be represented in binary. [10]

   (c) The variable x represents a floating point number, and the exponential of x is defined as:

```
exp(x) =  1 + x + (x*x/2!) + (x*x*x/3!) + (x*x*x*x/4!) + ...
```

   Outline how you might implement such a function using the MIPS instruction set (you are not expected to write the complete code). [9]

[Total 33]

5. (a) Use a truth table to prove that `A.(B+C) = (A.B) + (A.C)`.

[8]

   (b) Explain, *with examples*, the internal representation of a floating point number.

[7]

   (c) Provide drawings of circuit diagrams using logic gates to provide the following two parts of a one-bit adder. For each circuit, explain how it works.

   (i) The circuit which, given the two bit inputs A and B, together with the carry-in, produces the carry-out

   (ii) The circuit which, given the two bit inputs A and B, together with the carry-in, produces the sum

[18]

[Total 33]

[END OF PAPER]