Q–1 For each of the following 8051 instructions, specify if the given usage is correct or not. If it is correct, specify what the instruction will do. Otherwise state why the usage is illegal.

a)	MOV	R1, R7
b)	SUB	A, @R0
c)	MUL	AB
d)	CJNE	B, #21, label
e)	XCHD	A, B
f)	JMP	@A+DPTR
h)	MOV	A, @R3
i)	DJNZ	@R0, label

Q–2 Write a code fragment in 8051 assembly language using bit addressing instructions to evaluate
P3.4+F0$\cdot(C \oplus \overline{B})$ Where C is the value of carry flag and B is the value stored at the most significant bit of byte address 28H. If it evaluates to '1', P3.5 should be complemented.

Q–3 A program calls a function written in 8051 assembly language with a number between 0 and 3 (both inclusive) stored in register A. We want the called function to execute the corresponding code fragment (starting with labels L0, L1, L2, L3) and then return to the main program.
Write the assembly language code for the called function, by using a jump table consisting of an array of AJMP instructions.
(For purposes of illustration, let the code fragments labelled L0, L1, L2 and L3 replace the value in A by the byte read from ports P0, P1, P2 or P3 respectively).

Q–4 **a)** What is the advantage of PC relative addressing? Give examples of instructions of 8051 which use PC relative addressing.

b) How does decimal adjust work after addition in 8051? Show how it produces the correct decimal result after a binary addition.

c) Under what conditions is the overflow flag in PSW of 8051 set after add/subtract and after mul/div instructions.

d) What does the parity flag in 8051 indicate? When is it updated?

e) The 8052 has 256 Bytes of internal RAM, of which the top 128 occupy the same address space as the special function registers. Show how these are distinguished by using an example in which you read a byte from the memory address 80H and write it to a special function register at the address 80H.

f) A system with 8051 processor has an external ROM connected to it. It encounters the instruction MOVC A, @A+DPTR
How does it decide whether the constant is to be fetched from internal ROM or from external ROM?

Q–5 **a)** What problem can occur if one wants to read the count from a running 16 bit timer in 8051? Show how this problem can be avoided.

b) How can we use the timer T0 as a counter for external events? (Give all the hardware and software requirements). What is the highest rate of events which can be counted?

c) We want to implement a real time clock using an 8051 based system with a crystal frequency of 11.059 MHz. The clock display should be updated every second. This delay is obviously too long to implement using a timer directly. Show how to set up an interrupt service routine for timer 0 such that it calls a function ClockUpd every second as accurately as possible.

Q–6 Show the timing diagram for placing data and latching it in an SPI interface with CPOL = 0 and CPHA = 1 (mode 1). Is it possible to connect a device using mode 1 with another using mode 3? If so, how?

Q–7 What is the addressing mechanism used in I2C interface? Give the sequence of signaling by the master in order to address a slave device in order to write to it and to read from it.

Q–8 When is clock stretching used in an I2C interface? How is it implemented?

Q–9 Between SPI and I2C, which serial interface can run faster and why? In the case where multiple slave devices should be connected to a master, which of the two interfaces requires lower pin count and less pc board area? (Give reasons)

Q–10 A fixed program microprocessor carries out the following operations repeatedly: I) Fetch an instruction from ROM, II) Decode it, III) Fetch an operand from data memory and another from a register in parallel, IV) execute the instruction and V) Write the result to data memory.

Every read from ROM or read/write operation from/to data memory takes 3 clock cycles, while decoding, register read/write and execution takes a clock cycle each.

We want to assess the effect of placing an on-chip instruction cache. The probability of finding the next instruction in the cache is P and fetching it from the cache takes just one clock cycle. Evaluate the expected speed up factor for the processor with cache as a function of P. Find the limiting values of speed up as P tends to zero and as P tends to 1.

Q–11 What is the advantage of using segmented addresses as is done by the 8086 processor? Describe the different models used by 8086 based programs with respect to the size of their code and data areas.
How does the compiler/assembler use the information about the model while compiling a program?

Q–12 8086 uses a full descending stack. It takes more clock cycles to push a general purpose register than to pop it. What can be the reason for this imbalance, assuming that memory read and write times are identical?

Q–13 Write an 8086 assembly program using string instructions to find the length of a string terminated by '
0' as used in C. You can assume that the maximum length of the string is 130 characters.

Q–14 MIPS provides a minimal set of natively implemented instructions. The assembler can construct instructions and addressing modes not provided by the processor. A register

($at) is reserved for the use of the assembler for this purpose.
Show how the assembler constructs an instruction to load a 32 bit immediate value into a register using multiple native instructions.
Also, show how it can construct the based indexed addressing mode where the effective address is the sum of two registers and an immediate constant.