CS130 : Introduction to Computer Science
Writing Assignment

Answer the following questions. Your answers should not be more than 2 A4-sheets per question. ***Ensure that answer to each question is written on separate sheets. Please write the answers neatly****. Answers which violate the above guidelines will not be evaluated. Please give independent answers, while discussion is okay. Copying shall be dealt with strictly.*

1 a) Compare the various formats of representing numbers in computers. Discuss briefly with the help of examples. Mention which is the most popular number representation and why do you think so? (10 marks)

b) What do you mean by precision of representing a floating point number in the IEEE standard? Explain with a small example. (5 marks)

2 Write a 2 page note on your notion of the subject "Computer Science". Subdivide the answer into the following 5 sections:

i) A brief summary on the topic. Mention the difference between Computer Sc and Computer Engg.
ii) Mathematical Foundations and Concepts required.
iii) Relationship to rest of Engineering.
iv) Applications in the modern day.
v) Where do you think CS is heading to?

(5 x 3=15 marks)

3. Recall that as a Computer Scientist, each problem that you would like to solve using a computer program is a function from a domain to a range. The domain is used to model the set of inputs to the program and the range is the set of outputs expected from the program. For example, for the sorting problem, the domain is the set of number sequences, and the range is the set of sorted number sequences. As a special case for this question let us consider the domain to be the set of {0,1}-sequences, and the range to be the set {0,1}. How many functions exist between the given domain and range? Given our axiom that computer programs evaluates functions, the next question is, how many C-programs exist? What can you conclude from the

answers to the above two questions?

(15 marks)

4. **Good Processor, Bad Processor:** Suppose that n processors in a space ship have been subjected to cosmic radiation. Fewer than half of them were corrupted. We call the corrupted chips ``bad". The uncorrupted chips are called ``good". Your mission is to identify a chip that you are sure is good. You can use the chips to test each other: choose two chips *T* and *S* and use *T* to test *S*. If *T* (the tester) is good, then it will correctly report whether *S* (the subject) is good or bad. If *T* is bad, however, it will answer any way it wishes. Think of a bad chip as answering *adversarily* about *S*, which is to say, it will give whatever answer is least helpful to your testing strategy. In other words, imagine that an intelligent adversary is determining the behavior of the bad tester---an adversary who has access to and understands your algorithm, and is choosing each bad tester's answer in such a way that your algorithm takes as long as possible.

Design an algorithm that uses as few tests as possible to find *one* good chip. Show that your algorithm is correct and find its worst-case running time (i.e., number of tests). A single test is the operation of querying one processor about the “goodness” of a different processor.

Ground Rules

The algorithm should be sequential (one test at a time) and deterministic (no use of randomness).

The typeset answer must contain three clearly defined sections. The first section (algorithm description) should clearly specify your algorithm. The second section (correctness) should prove that your algorithm is correct and in doing so should help us understand it. The third section (timing) should show why your algorithm works within the timing bounds you state.

Make sure that your proof is clear and concise. It is important that you be careful, but not overly verbose. If you can say the same thing in 10 words as in 100, do so.

Grading

Assuming that your answer follows the ground rules and you prove your algorithm's correctness and timing bounds, you will receive a grade that depends on its running time. The grading scale is:

Running time	Base Grade
Quadratic, $\Theta(n^2)$	7.5
Subquadratic, $o(n^2)$ (an algorithm that is $O(n^2)$ but not $\Theta(n^2)$)	10.5
Linear, O(n)	12
4n	12.6
2n	13.5
1.5n	13.8
n-1	14.4
n-2	15

(15 marks)

5) a) It was discussed in class that the two main elements that need to be specified for a learning agent are the data and the bias. There is a third component that needs to be specified: a learning algorithm. What is the need for a learning algorithm? [Hint: Try to relate to the properties of an algorithm.]

(6 marks)

b)You are building an agent to play tic-tac-toe, where the objective is to get three pieces in a row (row, column or diagonal). The agent employs the following heuristic:

- For every unblocked two in a row for self: +2
- For every unblocked two in a row for opponent: -2
- Win: +5
- Loss: -5
- Draw: 0 (neither agent has three pieces in a row, and the board is full)

The agent employs a two step look ahead, (one self move and one opponent move, assuming that both are using the same heuristic) before picking the a move at the current board position. Given the above heuristic, what is the

move the agent that plays 'X' will pick from the board position shown below? Show clearly the moves the agent will explore.

O - -
 - - -
X O –

(9 marks)

6. A set of logic gates can be called **universal** if they can collectively realize the three Boolean operations **AND**, **OR** and **NOT**. For example, the set comprising the three gates *AND, OR* and *INVERTER* is clearly universal. Find out if the following sets are universal or not. Prove your answers using truth tables or circuit diagrams.

a) A = {*NAND*}
b) B = {*NOR*}
c) C= {*XOR*}
d) D = {*XOR, OR*}
e) E = {*MUX*}. A *multiplexer* (called *MUX* in short) is a three-input gate. It takes three inputs, *in0, in1*and *sel* and produces one output *f.* Input sel selects one of the other two inputs to pass on to the

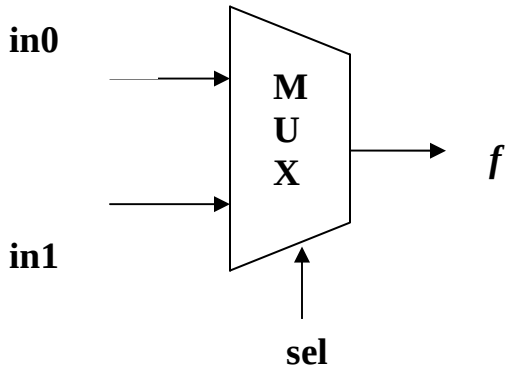

Fig 1

output. If sel=0, f=in0 else f=in1. Logically, *f* = (sel ' **AND** in0) **OR** (sel **AND** in1). *MUX* is pictorially represented by the symbol depicted in **Fig 1**.

(15 marks)