CS 109: Introduction to Computing
Sukhatme
Spring 2017

Homework Assignment 1

Assigned: 8/28/17 via Blackboard
Due: 23:59 (i.e. 11:59:00 pm) on 9/11/2017 via Blackboard

Notes:

a. This is the first homework assignment of CS 109. It is worth 7.5% of your overall grade for this class. You must solve all problems correctly to obtain full credit.

b. You are free to discuss the problems on this assignment with your classmates. However, to receive credit, you must write up and submit solutions completely on your own. You are responsible for understanding your answers. The purpose of discussing the questions with your classmates is to deepen your understanding of the material – not simply to obtain answers from them. To get the most out of the class, you are strongly encouraged to make a serious effort to understand and solve the problems on your own before discussing them with others.

c. On the work you turn in, you must list the names of everyone with whom you discussed the assignment.

d. All answers must be typed, not handwritten. The homework must be turned in as a single **PDF** document on Blackboard. Other formats will not be graded.

Problem 1: Universal Computation [0.5 points]

NAND(**x**, **y**) – defined as NOT(AND(**x**, **y**)) – is a computationally universal operation, capable of implementing any other logical operation. Which of the following is the correct representation of AND(**x**, **y**) just using (a combination of) NAND ?

a. NAND(**x**, **x**)

b. NAND(**x**, NAND(x, **y**))

c. NAND(NAND(**x**, **x**), NAND(**y**, **y**))

d. NAND(NAND(**x**, **y**),NAND(**x**, **y**))

Problem 2: Information [0.5 points]

Assume there are 200 countries in the world. Suppose you want to be able to encode any country as a binary number. At least how many bits (base 2 digits) must there be in such an encoding to enable this? Next, suppose you want to encode any country in a ternary (base 3) system. At least how many trits (base 3 digits) must there be in such an encoding to enable this?

Problem 3: Computers [2.5 points]

Matching (give the one best match for each vocabulary word):

State	1. A specific example of a concept
Abstraction	2. A way of making data more secure
Encoding	3. The condition of a system at a point in time
Discrete	4. A simplified or higher-level representation of a concept
Binary	5. Individually separate and distinct
	6. Every possible condition of a system
	7. A way of making a concept more confusing
	8. An exact description of a specific concept
	9. The practice of writing instructions for a computer
	10. The symbolic expression used to represent an object or concept
	11. A system of numerical notation that uses base 2

Problem 4: Early Programming [1.5 points]

In class, we discussed how the Jacquard Loom was programmed. We said that such programs were limited because they “didn’t scale.”

List the three reasons why Loom programs “don’t scale.” Each item should be a single sentence long.

Problem 5: Architecture & low-level programming [2.5 points]

In class, we discussed the Von Neumann architecture, and showed how to run low-level programming commands. We used the following instruction set (for simplicity, we leave out the ALU, even though that is responsible for the subtraction/addition/comparison):

LOAD *Mi Ri*	Copy from memory location Mi to register Ri
WRITE *Ri Mi*	Write from register Ri to memory location Mi
MOVE Ri Rj	Copy contents from register Ri to register Rj
SET *Ri value*	Set register Ri to given value
ADD *Ri Rj Rk*	Ri + Rj, store in Rk
SUBTRACT *Ri Rj Rk*	Ri - Rj, store in Rk
GOTO *Mi*	Jump to instruction at Mi
COND_GOTO *Ri Rj Mi*	If Ri is **more** than Rj, jump to instruction at Mi

A. In what follows you will need to run through a program and keep track of the state of the computer to show the complete execution of the program. Some memory locations are preloaded with data, this is shown in the first row of the table. Again, the first few rows have been filled in to get you started. Return the full table.

Program:

1	SET R1 3
2	WRITE R1 104
3	SET R2 1
4	SET R3 1
5	WRITE R2 103
6	ADD R2 R3 R2
7	COND_GOTO R1 R2 11
8	MOVE R2 R3
9	ADD R2 R3 R1
10	WRITE R1 102
11	SUBTRACT R1 R2 R1
12	LOAD 104 R2
13	SET R3 3
14	COND_GOTO R2 R1 5
15	SUBTRACT R1 R2 R1
16	END

State:

Program Counter	register			memory			
	R1	R2	R3	101	102	103	104
Before program starts				1	3	2	
1	3			1	3	2	
2	3			1	3	2	3
3	3	1		1	3	2	3
...							

(extend the above table to fill in the full execution of the program)

B. How many instructions were executed?

C. The number of instructions that are executed could be greatly reduced if a different value was used for R1 in instruction 1. What is the maximum value that should be given to R1 to achieve this? Explain your answer.

D. Consider the following program. After how many execution steps will it end? Explain your answer.

Program:

1	SET R1 3
2	SET R2 5
3	SUBTRACT R2 R1 R3
4	WRITE R1 102
5	WRITE R2 103
6	ADD R1 R2 R3
7	LOAD 102 R3
8	COND_GOTO R2 R1 3
9	LOAD 103 R1
10	END