CO200 – COA. Tutorial 3. Duration: 60 minutes. Max: 30 – Aug – 2017

1. Write the MIPS instructions for the following 2 tasks. Use the global pointer.
a. Access the first integer in the global data section.

b. Access the last possible integer in the global data section.

2. From the memory layout of a process in a MIPS machine, what are the sizes of:

(a) Static/global data section	(b) Code section

3. Encode/decode the following MIPS instructions:

a. add.d F0, F2, F4	b. 0x0

4. a. For the MIPS assembly instructions below, what is the corresponding C statement? Assume that the variables f, g, h, i, and j are assigned to registers $s0 , $s1 , $s2 , $s3 , and $s4 , respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7 , respectively.

MIPS Code	C Statements

sll $t0, $s0, 2 # $t0 = f * 4
add $t0, $s6, $t0 # $t0 = &A[f]
sll $t1, $s1, 2 # $t1 = g * 4
add $t1, $s7, $t1 # $t1 = &B[g]
lw $s0, 0($t0) # f = A[f]
addi $t2, $t0, 4
lw $t0, 0($t2)
add $t0, $t0, $s0
sw $t0, 0($t1)	

b. For the MIPS assembly instructions in 4.a, rewrite the assembly code to minimize the number if MIPS instructions (if possible) needed to carry out the same function.

5. Object files of 3 procedures (A, B, C) are shown below. These 3 are fed to the linkage editor to create the final a.out. Fill in the details for the object file of the final a.out file. The following details are given:

1. Procedure A declares X and is referenced in Procedure A and Procedure B.
2. Procedure C declares Y and is referenced in Procedure C.
3. Both X and Y are the first variables in A and C respectively.
4. Procedure A calls B, and procedure B calls C.

Any other information, if required, may be assumed. Fill in all the relevant boxes in the object files. Fill in all the boxes in the a.out file (next page). (2+2+2+4)

Header	Name	Procedure A			Procedure B			Procedure C
	Text Size	0x100			0x200			0x50
	Data Size	0x40			0x20			0x8
Text Seg.	Address	Instruction		Address	Instruction		Address	Instruction
	0	`lw $a0, Unknown1($gp)`		...	...		0	`lw $a0, Unknown5($gp)`
	4	`jal Unknown2`		12	`jal Unknown3`			
	...	...		16	`sw $a1, Unknown4($gp)`			
Data Seg.	Unknown2	(X)						
	...	...		...	...		...	...
Reloc. Info.	Address	Instruction Type	Dependency	Address	Instruction Type	Dependency	Address	Instruction Type	Dependency
Sym. Tab.	Label	Address		Label	Address		Label	Address	
		--							
		--							

Header	*Name*	*a.out*	
	Text Size		
	Data Size		
Text Segment	*Address*	*Instruction*	
		`lw $a0, Unknown1($gp)`	
		`jal  Unknown2`	
Data Segment	*Address*		