CS 261 Machine Organization (Spring 2020) - Homework 4

Name: Adrian Aguada UIN: 674500288

Due Date: February 25, 5 p.m. *Late Due Date: February 26, 5 p.m.*
Please submit the completed fillable pdf to Gradescope. *Total points: 100*

Problem A

[4 points per C code blank, 2 points per table entry = 40 points]

Consider the following blocks of code:

Assembly Code:	**C Code**

.file "hw4a.c"
.text
.globl main
.type main, @function
main:
pushq %rbp
movq %rsp, %rbp
movl $1889785064, -16(%rbp)
movl $0, -12(%rbp)
movl $-1, -4(%rbp)
movl $0, -8(%rbp)
jmp .L2
.L4:
movl -16(%rbp), %eax
andl $1, %eax
movl %eax, -4(%rbp)
cmpl $0, -4(%rbp)
jne .L3
addl $1, -12(%rbp)
.L3:
shrl $4, -16(%rbp)
addl $1, -8(%rbp)
.L2:
cmpl $7, -8(%rbp)
jle .L4
movl $0, %eax
popq %rbp
ret
	int main(){
	unsigned x = 1889785064;
	int c = 0;
	int s = -1 ;
	for(int i = 0; i <= 7 ; i++){
	s = x & 1 ;
	if(s == 0){
	c++ ;
	}
	x = x >> 4 ;
	}
	return 0;
	}

The assembly code on the left came from the skeleton C file on the right after optimizing with -O0. ("gcc -O0 -S -fno-asynchronous-unwind-tables", the last option was used to disable cfi directives). Complete the C code based on the Assembly code given. It may help to first fill out the table below to map the registers/addresses to the C variables

Assembly address/register	**C variable**
-16(%rbp)	x
-12(%rbp)	c
-4(%rbp)	s
-8(%rbp)	i

Problem B

[2 points per destination, 4 points per result = 60 points]

Consider the following initial values for the registers and memory. Values are reset back to the values shown in the table before each operation. Determine the result of the following operations and state where the result is stored.

* Destination should be formatted after doing all address calculations in order to be counted for full credit
* Result should match the size specified by the destination

Address	Value
0x08	0x88
0x56	0xC1
0x57	0xD2
0x80	0xC3
0x81	0x00
0xB8	0x32

Register	Value
%rax	0xF4E3B2A1
%rbx	0xFACE
%r8	0xB0
%r9	0x0A
%r10	0x36

Instruction	Destination	Result
movw $12(%r10,%r9,2),%ax	%ax	0xC1
movb $10(%r9,%r10,2),%bl	%bl	0xC3
incl %eax	%eax	0xF4E3B2A2
subw $-2(%r9),%bx	%bx	
decb $8(%r8)	%r8	0x31
imulw $8, -0x30(%r8)	%r8	0x618
sarb $5, %bl	%bl	0xFE
shrw $1,$6(,%r9,8)	%r9	0x60
andl %ebx,%eax	%eax	0xB280
negq %r8	%r8	0x4F