COMP.2030 Test 1 9/27/21

**CLOSED BOOK/NOTE/LAPTOP/Cell Phone/Calculator.**

**NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. (5 pt) What is base-5 representation of decimal value 25 ?
2. (5 pt) The decimal value 250 is known to be 372 in octal. What is the hexadecimal representation of value 250?
3. (20 pt) When x and y are 8 bits long, and have 0xFF and 0x00, respectively, show byte values in ***hexadecimal*** when each of the following operations is applied independent of others from the original values of x and y:

x & !y ~x || ~y ~x & y ~x && y

1. (20 pt) Suppose that **the data segment starts at the address of 0x00008080**, which is also the address of label ‘array.’ Consider the following sequence of MIPS instructions in the text segment, which are executed in sequence. What are the values of the registers in the comment fields after the corresponding instructions on the left completes execution one after another ?

.data

array: .word 0x8090, 7, 4, 8, 16, 10, 8,0,4,7

.text

la $t9, array # $t9 =

lw $t0, ($t9) # $t0 =

lw $t2, ($t0) # $t1 =

1. (10 pt) With the data segment in question 1 above,

write a MIP instruction(s) to fetch the value of the fourth value of array (value 8) into register $s7.

write a MIP instruction(s) to replace the second value of array (value 7) by the value in register $s7.

1. (10 pt) A C function prob() below is compiled to MIPS codes on the right. Write the C function prob() in ONE or TWO C statements.

# function prob():

# argument p1 and p2 are passed by

# $a0 and $a1, respectively

prob: lw $t0, ($a0)

lw $t1, ($a1)

add $t0, $t0, $t1

sw $t0, ($a1)

jr $ra

void prob(int \*p1, int \*p2){

1. (10 pt) Function ff() on the left is compiled into MIPS codes on the right. Fill in the underlined to make the C function equivalent to the MIPS codes (C operator for xor is “^”).

# x in $a0

ff: li $t0,0

xx: beq $t0, $a0, yy

xor $t0, $t0, $a0

srl $a0, $a0, 1

b xx

yy: jr $ra

int ff (unsigned x) {

int val = 0;

while (\_\_\_\_\_\_\_\_\_\_\_\_\_) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

return val;

}

1. (10 pt) An integer array ‘X’ with at least one element. The size of ‘X’ is passed as an argument in variable limit. Write the C function func() that performs the same operation as the MIPS code on the right with a ‘for’ loop.

int func(int limit){

# $t9 has limit

# return value is via $a0

func: lw $a0, X($zero)

li $t0, 1

rept: bge $t0, $t9, exit

sll $t0, $t0, 2

lw $a1, X($t0)

bgt $a0, $a1, next

move $a0, $a1

next: sra $t0, $t0, 2

addi $t0, $t0, 1

b rept

exit: jr $ra

int i, temp;

return val;

}

1. (10 pt) Convert the C-code on the right to its pseudo-C code with ‘goto.’

void main () {

void main () {

int arr[10] = {12, -1, 8, 0, 6, 85, -74, 23, 99};

int i, size=9, sum=0, pos=0, neg=0;

for (i = 0; i < size; i++) {

sum += arr[i];

if (arr[i] > 0)

pos += arr[i];

else if (arr[i] < 0)

neg += arr[i];

}

}

int arr[10] = {12, -1, 8, 0, 6, 85, -74, 23, 99};

int i, size=9, sum=0, pos=0, neg=0;

1. (30 pt) Convert the following C-code to its pseudo-C code with gots’s,

void cond(int a, int \*p){

if (p && a >0)

\*p += a;

}

and **write MIPS codes** by referencing pseudo-C code.

Pseudo-C

MIPS