ECE 2035 Programming HW/SW Systems Fall 2015

4 problems, 6 pages Exam One Solutions 16 September 2015

Problem 1 (2 parts, 30 points) **Loops**

Part A (10 points) The array Temps holds a set of 200 temperatures, represented as integers. Write a **for loop** in C that computes the average of the boiling temperatures in Temps (those greater than or equal to 100) and stores the average in the variable Avg (ignore any remainder). Assume there is at least one boiling temperature in Temps. *For maximum credit, declare and initialize variables as needed.*

```
int Temps[200] = {4, -1, 103, …, -36, 125};
int Avg = 0;
int i, Count = 0;

for (i=0; i < 200; i++)
     if (Temps[i] >= 100) {
        Avg = Avg + Temps[i];
        Count++;
     }
Avg = Avg / Count;
```

Part B (20 points) Write a MIPS code fragment that is equivalent to the code you wrote in Part A. **Store the average in $1**. *For maximum credit, include comments and use a minimal number of instructions.*

Label	Instruction	Comment
	.data	
Temps:	**.word 4, -1, 103, …, -36, 125**	**# alloc & init Temps[200]**
	.text	
AvgBoiling:	addi $1, $0, 0	# offset = i*4
	addi $10, $0, 800	# offset < 800
	addi $2, $0, 0	# int Avg=0;
	addi $3, $0, 0	# int Count=0
ForLoop:	beq $1, $10, EndFor	# for (i=0; i < 200; i++)
	lw $4, Temps($1)	# Temps[i]
	slti $11, $4, 100	# Temp[i] >= 100;!Temp[i] < 100
	bne $11, $0, EndIf	
	add $2, $2, $4	# Avg = Avg + Temp[i]
	addi $3, $3, 1	# Count++
EndIf:	addi $1, $1, 4	# offset = offset + 4
	j ForLoop	
EndFor:	div $2, $3	# Avg = Avg / Count
	mflo $1	

Problem 2 (3 parts, 25 points) **Conditionals & Compound Predicates**

For this problem, assume these registers hold the values of these variables:

$2: Answer	$4: Hs	$6: He	$7: Ss	$9: Se	$10: temp	$11: Max

Part A (9 points) Write a MIPS code fragment that computes the maximum of `Ss` and `Hs` and stores it in `$11`. Use `$10` as a predicate register. Use a minimal number of additional registers as needed. *For maximum credit, include comments.*

Label	Instruction	Comment
	add $11, $0, $7 slt $10, $4, $7 bne $10, $0, endif add $11, $0, $4	# Answer = Ss # if (Hs >= Ss) # answer = Hs
endif:		

Part B (10 points) Consider the following MIPS code fragment.

Label	Instruction
	slt $10, $9, $4
	bne $10, $0, Below
	slt $10, $6, $7
	bne $10, $0, Below
	add $2, $0, $11
	j End
Below:	addi $2, $0, 0
End:	...

What is the equivalent C code fragment in terms of `Hs, He, Ss, Se, Max,` and `Answer`? For maximum credit, use a compound logical predicate wherever possible. Assume the variables are all of type `int`.

```
if  (Se  < Hs || He < Ss)
      Answer = 0;
else
      Answer = Max;
```

Part C (6 points)

If `Ss` and `Se` are integers representing years on a timeline, where `Ss < Se`, and `Hs` and `He` are years on another timeline where `Hs < He`, draw an example of a case where `Answer = 0 ($2=0)` computed in the code in Part B (select values for `Hs, He, Ss,` and `Se` to illustrate this case).

```
          (1992)                     (1994)
H:      Hs -------------------------- He
S:                                              Ss --------------------------- Se
                                               (1996)                        (1998)
```

Problem 3 (2 parts, 20 points) **MIPS Controller and Instructions**

Part A (8 points) Suppose the instruction "`jal Foo`" is executed which changes the values of the following registers to:

Register	**Value**
$31	2032
PC	2056

What is the address of the first instruction of the subroutine `Foo` and what is the address of the `jal Foo` instruction?

Subroutine `Foo` starts at address: 2056

Address of `jal Foo` instruction: 2028

Part B (12 points) For each of the following, write a single MIPS instruction to implement the C fragment? Assume variables `A, B, C,` and `D` are of type `int` and are stored in registers `$1, $2, $3,` and `$4`.

`A = 0xAB020000;`	**`lui $1, 0xAB02`**
`B = C & 3;`	**`andi $2, $3, 3`**
`C = D / 512;`	**`sra $3, $4, 9`**

Problem 4 (2 parts, 25 points) **More Loops and Conditionals**

Part A (15 points) Suppose A is an array of 100 integers that might contain duplicate elements and x and position are variables of type integer. Write a **do while** loop that determines whether x is an element of A and if so, sets position to the smallest index of A at which x appears. Otherwise, if x is not in A, it sets position to -1. Declare and initialize any additional variables you need. For full credit, **do not** use the **break** statement.

```
int i = 0;
int position = -1;
do {
    if ( x == A[i]) position = i;
    i++;
} while ((position == -1) && (i < 100));
```

Part B (10 points) What does the following code fragment print?

```
int A[10] = {1990, 3, 1992, 5, 1999, 1, 2001, 3, 2005, 9};
int B[10] = {1991, 2, 1993, 3, 1998, 7, 2002, 3, 2007, 6};
int i,j;
for(i=1; i<10; i=i+2)
  {
    for (j=1; j<10; j=j+2)
      {
        if (A[i] == B[j])
          {
            printf(“C: %d, As: %d, Bs: %d.\n”, A[i], A[i-1], B[j-1]);
            break;
           }
      }
  }
```

```
C: 3, As: 1990, Bs: 1993.

C: 3, As: 2001, Bs: 1993.
```