

ECE 199 Exam 2 Fall 2002

Tuesday, November 5th, 2002

Name:

1. Be sure your exam booklet has 7 pages.
2. Write your name at the top of each page.
3. This is a closed book exam.
4. You are allowed one 8.5 x 11 sheet of notes.
5. Absolutely no interaction between students is allowed.
6. Show all of your work.
7. Don't panic, and good luck!

Problem 1	20 points	_____
Problem 2	20 points	_____
Problem 3	20 points	_____
Problem 4	20 points	_____
Problem 5	20 points	_____
Total	100 points	_____

Name: _____

Problem 1 (20 points): Short Answer

Part A. What is the purpose of the Ready bit in the Keyboard Status Register? Be brief.

Part B. What is the PC of the instruction that executes immediately after the TRAP instruction in the code below?

```
.ORIG x3000
LD    R0, DATA
STI   R0, TOP
TRAP  x30
ADD   R0, R0, #1

DATA  .FILL  x1234
TOP   .FILL  x30
```

Part C. The following subroutine can be used to output a digit to the monitor. But it has a serious bug. What is it?

```
OUTPUT      ST  R0, SAVE_R0
            ST  R1, SAVE_R1
            LD  R1, ASCII_ZERO
            ADD R0, R0, R1
            OUT
            LD  R1, SAVE_R1
            LD  R0, SAVE_R0
            RET

ASCII_ZERO  .FILL  x30           ; x30 is ASCII for zero ('0')
SAVE_R0     .FILL  x0000
SAVE_R1     .FILL  x0000
```

Name:

Problem 2 (20 points): The Stack Data Structure

Write an LC-2 subroutine to reverse the contents of a stack. That is, if the stack contained the data items, **A**, **B**, **C**, **D** (where **A** is at the top), after the call to `REVERSE`, the stack will contain the items **D**, **C**, **B**, **A** (where **D** is at the top).

To assist you in writing your code, you will make use of the `PUSH` and `POP` routines (assume they are provided) to manipulate the stack. `POP` will return the value at the top of the stack in `R0` and will set `R1 = 1` if the stack was empty (i.e., underflow). `PUSH` will add the value in `R0` to the stack and will set `R1 = 1` if the stack is full (i.e., overflow).

If you need to use additional memory for `REVERSE`, you can use the area starting at location `x4000`. Assume the `PUSH` and `POP` subroutines are on the same page as `REVERSE`.

REVERSE:

Name:

Problem 3 (20 points): LC-2 to C

Provide the C statements expressed by the provided LC-2 code fragments. Use the provided C code as a starting point for your answer.

Part A

<pre>LDR R0, R6, #3 LDR R1, R6, #4 STR R0, R6, #4 STR R1, R6, #3</pre>	<pre>main() { int v1; int v2; int temp; }</pre>
------------------------------------------------------------------------	-------------------------------------------------------------

Part B

<pre>LDR R0, R6, #3 NOT R0, R0 LDR R1, R6, #4 NOT R1, R1 AND R2, R0, R1 NOT R2, R2 STR R2, R6, #5</pre>	<pre>main() { int v3; int v4; int v5; }</pre>
---------------------------------------------------------------------------------------------------------	-----------------------------------------------------------

Name:

Part C

<pre> AND R0, R0, #0 ADD R0, R0, #15 STR R0, R6, #4 FL1 AND R0, R0, #0 STR R0, R6, #3 FL1_2 LDR R0, R6, #3 LDR R1, R6, #4 NOT R1, R1 ADD R1, R1, #1 ADD R0, R0, R1 BRzsp DONE F LDR R0, R6, #5 ADD R0, R0, #1 STR R0, R6, #5 LDR R0, R6, #3 ADD R0, R0, #2 STR R0, R6, #3 BRnzp FL1_2 DONE </pre>	<pre>main() { int i; int j; int k; }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------

Name:

Problem 4 (20 Points): Iteration in C

Provide the output generated by each piece of C code below.

Part A:

```
int x = 20;
int y = 10;

while ((x > 10) && (y & 15)) {
    y = y + 1;
    x = x - 1;
    printf("*");
}
```

Part B:

```
int x;

for (x = 10; x ; x = x - 1)
    printf("*");
```

Part C:

```
int x;

for (x = 0; x < 10; x = x + 1) {
    if (x % 2)
        printf("*");
}
```

Part D:

```
int x = 0;
int i;

while (x < 10) {
    for (i = 0; i < x; i = x + 1)
        printf("*");
    x = x + 1;
}
```

Name:

Problem 5 (20 Points): Using Functions in C

```
#include <stdio.h>

int Swap(int a, int b);

main()
{
    int a;
    int b;

    scanf("%d", &a);
    scanf("%d", &b);

    Swap(a, b);

    printf("a = %d\n", a);
    printf("b = %d\n", b);
}

/* This function swaps the value of the two input parameters */
int Swap(int a, int b)
{
    int x;

    x = a;
    a = b;
    b = x

    return x;
}
```

Part A. If the user enters **23 45** for the input, what is the output of this program?

Part B. Suggest a way to fix this program using only the concepts described in class thus far?