

## ECE 190: Introduction to Computing Systems

Spring 2006 Exam II

Apr. 4, 2006

Name: \_\_\_\_\_

**Note that the questions are not weighted equally. Budget your time accordingly, and do not work too long on any one problem.**

Score:

1. \_\_\_\_\_ (16 pts.)

2. \_\_\_\_\_ (5 pts.)

3. \_\_\_\_\_ (8 pts.)

4. \_\_\_\_\_ (7 pts.)

5. \_\_\_\_\_ (5 pts.)

6. \_\_\_\_\_ (15 pts.)

7. \_\_\_\_\_ (44 pts.)

Total: \_\_\_\_\_(100 pts.)

Section:

9:00am Kelm

12:00pm Nichols

1:00pm Becker

3:00pm Wang

3:00pm Becker

**Problem 1 (16 points)**

- A.** (4 pts.) Consider the LC-3 instruction `TRAP x23`. Explain in one sentence the role of the constant `x23` in this instruction.
- B.** (8 pts.) Suppose that memory mapped I/O is used to interface the LC-3 to a mouse. The mouse has one data register and one status register. Bit 15 of the status register is used to indicate when a mouse click has occurred (i.e, bit 15 is set to a one when a mouse click occurs). In the space below, write an LC-3 subroutine, `GetClick`, that waits for a mouse click, and then returns the value of the mouse data register in `R0`. Your subroutine may change the values of registers `R0-R3`, but it may not change the values of any other registers or memory locations. The addresses of the status and data registers are defined using `.FILL` directives.

```
MSR .FILL xFE10
MDR .FILL xFE12
GetClick
```

- C.** (4 pts.) If the following C code is run on the LC-3, how many calls are made to `TRAP x21` (NOTE: the assembler name for `TRAP x21` is `OUT`)?

```
x = 0;
printf("x is %d",x);
```

**Problem 2** (5 points)

For this problem, you will write the C functions to push a single integer onto a stack. The stack is stored in the global array `Stack`, and the variable `SP` is the stack pointer. Your function should return the value 1 if the item is successfully pushed onto the stack. If the item is not successfully pushed onto the stack, your function should return 0 and not alter the stack.

```
#define StackSize 1000
int Stack[StackSize];
int SP = 0;
```

```
int PUSH(int item)
{
```

```
}
```

**Problem 3** (8 points)

The C code below generates a run-time error. Explain the cause of this error and how it could be fixed.

```
char alphabet[26];
main()
{
    char *str;
    int i = 0;

    for (c = 'a'; c <= 'z'; c++)
        alphabet[i++] = c;

    for (i = 0; i < 26; i++,str++)
        *str = alphabet[i];
}
```

**Problem 4** (7 points)

Write the function `strlen` that returns the length of a NULL-terminated string.

```
int strlen(char *str)
{
```

```
}
```

**Problem 5** (5 points)

The function below should return the value 1 if the global array `List` is sorted, and 0 otherwise. There is a bug in this function. Explain the bug, and describe what can be done to fix the problem. Assume that there are no duplicate entries in the array.

```
#define N 100
int List[N];

int is_List_sorted()
{
    int i;

    for(i = 0; i < N; i++)
        if (List[i] > List[i+1]) return 0;
    return 1;
}
```

**Problem 6** (15 points)

A. What output is generated by the following C program?

```
#include <stdio.h>
main()
{
    int i;
    int x = 5;

    for (i = 0; i = x; i++,x--)
        printf(" %d ",i);
}
```

B. What output is generated by the following C program?

```
#include <stdio.h>
main()
{
    int i;
    int x = 5;

    for (i = 0; i <= x; i++,x--)
        printf(" %d ",i);
}
```

C. What output is generated by the following C program?

```
#include <stdio.h>
main()
{
    int i;
    int x = 5;

    for (i = 0x7FFF; i > 0; i = i >> x)
        printf(" %x ",i);
}
```

**Problem 7** (~~44~~ points)

All parts of this question refer to the C code given on the last page of this exam. You may detach the last page of the exam. The code you write for this problem may alter the values of registers R0-R3.

**A.** (8 pts.) Write the LC-3 assembly language code to implement the assignment in Line A. Assume that the variable x is stored in register R0.

**B.** (4 pts.) Write the LC-3 assembly language code to implement the assignment in Line B. Assume that the variable x is stored in register R0.

**C.** (8 pts.) The assembly language code required to implement the return in Line C must place the value of x in the appropriate memory location, and then perform various other tasks. Write below **only** the LC-3 code required to store x. Assume that x is stored in memory (i.e., it is not currently available in a register).

- D.** (4 pts.) Write the LC-3 assembly language code to implement the assignment in Line D. Assume that the variable `y` is stored in register `R0`.
- E.** (4 pts.) What value is assigned to the variable `x` in line E of main?
- F.** (4 pts.) What value is assigned to the variable `x` in line F of main?
- G.** (4 pts.) What value is assigned to the variable `i` in line G of main?
- H.** (4 pts.) What value is assigned to the variable `i` in line H of main?
- I.** (4 pts.) What is a *dynamic link*, and how it is stored when the function `fn1` is called.

## This code is to be used for Problem 7

```
int A[10];

int fn1(int *B);
void fn2(int x);
void fn3(int *x);

int main()
{
    int A[20];
    int i;
    int x = 5;

    for (i = 0; i < 20; i++)
        A[i] = i;
    x = fn1(A);    /* Line E */
    x = A[2];      /* Line F */
    x = 34;
    fn2(x);
    i = x;         /* Line G */
    fn3(&A[10]);
    i = A[10];    /* Line H */

    return 0;
}

int fn1(int *B)
{
    int x;
    x = *(B+2);   /* Line A */
    x = A[2];     /* Line B */
    *(B+2) = 15;
    A[2] = 17;
    return x;     /* Line C */
}

void fn2(int x)
{
    int y;
    y = x;        /* Line D */
    x = 33;
}

void fn3(int *x)
{
    *x = 25;
}
```