

# ECE 199 Exam 2 Spring 2003

Tuesday, April 8<sup>th</sup>, 2003

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

- **Be sure your exam booklet has 8 pages.**
- **Write your name at the top of each page.**
- **This is a closed book exam.**
- **You are allowed one 8.5 x 11 sheet of notes.**
- **Absolutely no interaction between students is allowed.**
- **Show all of your work.**
- **Don't panic, and good luck!**
- **Challenge questions are marked \*\*\***

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 (8 points).** The C statement to multiply a number by 10 can be done in the LC-2 using only 5 instructions, none of which are branches. Assuming that the number to be multiplied is in R1, write a sequence of 5 LC-2 instructions or less that will compute  $R1 * 10$  and store the result into R0. Do not use any branches.

**Part B (5 points).** Consider the C statement, `printf("The value of x is %d", x);`. Say the value of `x` is 6372 when this statement is executed. If this code is run on an LC-2, how many calls are made to TRAP x21?

**Part C (7 points).** The following function performs a simple operation. Describe it in twenty words or less.

```
int PropertyX(int number)
{
    int x = 0;

    while (number > 0) {
        x = x + (number % 10);
        number = number / 10;
    }

    return x;
}
```



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

### Problem 3 (20 Points): Iteration in C

**Part A (10 points):** What shape is printed out by the following function? Be specific.

```
void PrintShape(int height)
{
    int i, j;
    for (i = 1; i <= height; i++)
    {
        printf("*");
        for (j = 1; j <= (height - 2); j++)
        {
            if ( (i == 1) || (i == height) )
                printf("*");
            else
                printf(" ");
        }
        printf("*\n");
    }
}
```

**\*\*\*Part B (5 points):** The following function takes an integer height input and prints the letter 'V' using the asterisk character (\*). Fill in the missing code. Three example outputs are shown below. Assume that  $height > 1$  and that the leftmost '\*' appears in the leftmost column on your monitor.

PrintV(2);

```
* *
*
```

PrintV(3);

```
* *
* *
*
```

PrintV(4);

```
* *
* *
* *
*
```

```
void PrintV(int height)
{
    int i, j;
    for(i = 0; i < height; i++)
    {
        for(j = height; j > (height - i); j--)
            printf(" ");

        printf("*");

        for(j = 0; j < _____ ; j++)
            printf(" ");

        if (i < _____ )
            printf("*");
        printf("\n");
    }
}
```

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

**Problem 3, continued:**

**Part C (5 points):** The function below takes an integer height input and prints the letter 'W' using the '\*' character. The code provided is taken from the PrintV function of Part B (note that the last printf("\n") statement was not included). You must finish the function using compilable C code. There is no need to fill in the blank lines as they are the same as from Part B. Example outputs for PrintW are provided below. You may assume that height > 1 and that the output is left-justified.

PrintW(2);

```
* * *
* *
```

PrintW(3);

```
*   *   *
* * * *
*   *
```

PrintW(4);

```
*       *       *
*     * * *     *
* *   * * *   *
*       *       *
```

```
void PrintW(int height)
{
    int i, j;

    for (i=0; i < height; i++)
    {
        for(j = height; j > (height - i); j--)
            printf(" ");

        printf("*");

        for(j = 0; j < _____ ; j++)
            printf(" ");

        if (i < _____)
            printf("*");

        /* Your code begins here; do not change anything above this line.*/

    }
}
```

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

**Problem 4 (20 points): LC-2 Assembly Programming**

**Part A (15 points):** The following LC-2 function `ByOne` performs a simple operation on `R0`. For this part of the question, you will write a function that calls `ByOne` in order to perform the operation “`ByN`”. That is, it performs `ByOne` `n` times on the value in `R0`. The value `n` is passed to `ByN` in `R1`.

**ByOne:**

```
AND R3, R3, 0
ADD R3, R3, #15
```

```
LOOP:  ADD R2, R0, #0
        BRn ADD_ONE
        ADD R0, R0, R0
        BRnzp UPDATE
```

**ADD\_ONE:**

```
ADD R0, R0, R0
ADD R0, R0, #1
```

**UPDATE:**

```
ADD R3, R3, #-1
BRp LOOP
```

```
RET
```

**ByN:**

**\*\*\*Part B (5 points):** For this question, you will write a C function that performs an equivalent operation to `ByN`. The trick here is to realize that the function performed by `ByOne` can be performed with a simple expression in C. Full credit will be given if you can identify this expression.

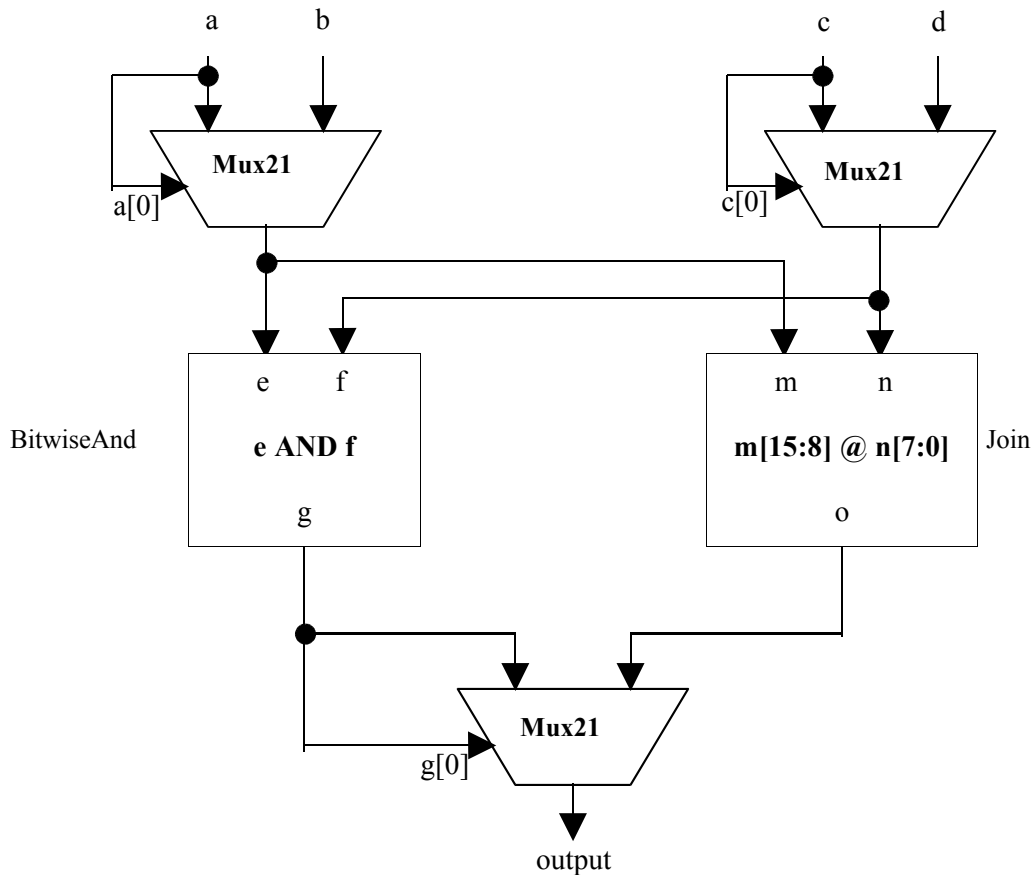
Name: \_\_\_\_\_

**Problem 5 (20 Points): C Programming**

**Part A (15 points).** The following C code implements a basic logic component that we've discussed in class. Name the component and be sure to explain its inputs and outputs. Assume 'a' can only take on the values 0, 1, 2, or 3.

```
int UnknownComponent(int a)
{
    if (a == 0) {
        return 0x01;
    } else if (a == 1) {
        return 0x02;
    } else if (a == 2) {
        return 0x04;
    } else if (a == 3) {
        return 0x08;
    }
}
```

**Part B (5 points).** Now, write C code to implement the logic diagram below. You **must write and use all** of the functions below. Additionally, complete `main()` using the functions you wrote.



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

```
int Mux21( int x, int y, int select )
{

}

int BitwiseAnd( int e, int f )
{

}

int Join( int m, int n )
{

}

int main()
{
    int a, b, c, d, output;

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

    printf( "Result: %d\n", output );
    return 0;
}
```