

Name: _____

University of California, San Diego
CSE 30 – Computer Organization and Systems Programming
Practice Final
Dr. Diba Mirza

Name	
Student ID	

Name of person to your left	
Name of person to your right	

Please put your name on the top of every page and complete the all of the above information.

Problem	Possible	Score
1	15	
2	40	
3	25	
4	25	
5	25	
6	40	
Total	170	

Name: _____

This page was intentionally left blank.

Name: _____

Problem 1: (15 points) (10 minutes) You, Detective Jack Clouseau, are guarding the coveted Pink Panther diamond, which resides in a safe with a special combination lock. The lock has four dials – each dial consists of numbers 0-9 and letters a, b.

a) **(5 points)** How many unique combinations are there in your special combination lock?

b) **(5 points)** Meanwhile, the Phantom is trying to steal the diamond. If it takes him a second to check each combination, and you check on the safe every 15 minutes, what are your chances of catching him? Justify your answer.

Name: _____

- c) **(5 points)** You decide that you want to make a new lock with hundred times the number of possible combinations as the old one. If each dial consists of the same numbers and letters as before, what is the minimum number of dials you need on the new lock?

Name: _____

Problem 2: (40 points) (30 minutes) C Medley

(a) (20 points) Fun with Pointers

For the C code below make the following assumptions

- i. The base address of `array` is `0x10000000`
- ii. The byte ordering is Little Endian
- iii. `int` is 4 bytes

```
int array[3]={-1,1,2};
int main(){
    unsigned char *ptr1 = (((unsigned char *) array) + 1);
    unsigned char val;
    int *ptr2 = array + 2;
    val = *ptr1;
    return 0;
}
```

- i) **(10 points)** Show the byte level representation of `array` in memory. You have to mark the increasing direction of memory addresses.

- ii) **(5 points)** What is the value in `*ptr2` before the return statement?

- iii) **(5 points)** What is the value of the variable `val` before the return statement?

Name: _____

(b) (20 points) Buggy Buggy Debuggy

Consider the following C code.

```
int main()
{
    short disappear[]={1,2,3};
    short index=2;

    do
    {
        disappear[index]=0;

    }while(index-- >=0);
}
```

- i) **(10 points)** Show the entire contents of the array `disappear` at the end of each iteration of the do-while loop, for three iterations.

- ii) **(5 points)** There is a bug in the above code. Identify it and explain its possible effects on the compilation and execution of the program

Name: _____

- iii) **(5 points)** Change the code to correct for the above bug. What are the final values of `index` and the contents of the array `disappear`, right before `main` returns?

Name: _____

Problem 3: (25 points) (20 minutes) Fill in the code

(a) (15 points) Data Structures

Consider the following definition of a node in a doubly linked list.

```
typedef struct Node ListNode;  
  
struct Node {  
    int data;           // this is the data in this node element  
    ListNode* next;     // this is a pointer to the next element in the list  
    ListNode* prev;     // this is a pointer to the previous element in the list  
};
```

The following C function should completely reverse the order of the elements in the list. A part of the function is implemented using recursion. Fill in the missing implementation. The function is initially called with a pointer to the first `ListNode` object of the list. It should ultimately return a pointer to the new first element of the list (which was previously the last element)

```
ListNode * reverse(ListNode *aNode)  
{  
    /* Your code goes here */  
  
    _____  
    _____  
    _____  
    _____  
    _____  
    _____  
    _____  
  
    if(aNode->prev == NULL)    //stop if end of list  
        { return aNode; }    // has been reached  
    else  
        { return reverse(aNode->prev); } //otherwise continue  
                                           // recursively  
}
```

Name: _____

(b) (10 points) C Strings: The following function should copy n characters from string *source* to string *dest* starting with the first element of *source*. Fill in the blanks to complete the code. There should only be one expression in each while loop, meaning you must only fill in the right hand side of the assignment statement in the first while loop, and the left hand side of the assignment statement in the second while loop. You should not add any additional statements to the code.

```
/* strncpy: copy n characters from source to dest */
void strncpy(char * dest, char * source, int n)
{
    while(*source && n-- > 0)
    {
        *dest++ = _____;
    }

    while(n-- > 0)
    {
        _____ = 0;
    }
}
```

Name: _____

Problem 4: (25 points) Bit Manipulation (30 minutes) Write ARM code for the function `isolateRangeOfBits`. This function isolates the bits between lower and upper and returns only those bits. You must follow the ARM Procedure Call Standard. You must comment your code extensively. Code that is not adequately commented will be penalized.

```
isolateRangeOfBits(0x12345678, 0, 8) returns 0x78
isolateRangeOfBits(0x12345678, 8, 12) returns 0x6
isolateRangeOfBits(0x12345678, 30, 32) returns 0x0

/* Precondition: lower <= upper */
int isolateRangeOfBits(int number, unsigned int lower,
unsigned int upper);

isolateRangeOfBits:
```

Name: _____

Name: _____

Problem 5: (25 points) Compilation (30 minutes)

The following function performs a finite impulse response (FIR) filter using input from array z and coefficients found in integer array h .

```
int compute_FIR( int* h, int* z, int ntaps, int input)
{
    int ii;
    int accum;

    /* store input at the beginning of the delay line */
    z[0] = input;

    /* calc FIR and shift data */
    accum = h[ntaps - 1] * z[ntaps - 1];
    for (ii = ntaps - 2; ii >= 0; ii--) {
        accum += h[ii] * z[ii];
        z[ii + 1] = z[ii];
    }
    return accum;
}
```

Write the above function in assembly using ARM instructions. Specify the mapping between variables and registers in your code. You must follow the ARM Procedure Call Standard.

Mapping:

compute_FIR:

Name: _____

Name: _____

Problem 6: (40 points) NO DISASSEMBLE! REASSEMBLE!

What kind of pants does Mario wear? DENIM, denim, denim

Consider the following code:

```
main:      MOV r0, #16
           SUB r0, r0, #1
           MOV r1, #5
           ADD r1, r1, r1, LSL #2
           BL mario
           BX lr

mario:     CMP r0, r1
           BEQ yoshi
           BLT luigi
           SUB r0, r0, r1
           B mario

luigi:     SUB r1, r1, r0
           B mario

yoshi:     BX lr
```

- a) (5 points)** How many functions does the above code contain? What is the return value of the `mario` function?

Name: _____

- b) (10 points)** Assume the processor that runs the above code implements a five-stage pipeline. Are there any data hazards in the above code, if no special mechanisms were in place to deal with data hazards? If yes, identify them and justify your answer.
- c) (5 points)** List two mechanisms to alleviate data hazards. If the processor implemented each of these mechanisms, how many cycles does the execution of the `main` function take until (but excluding) the call to `mario`?

Name: _____

- d) (5 points)** How many bytes of code does the `mario` function use? Assume the normal ARM instruction set as we discussed in class (i.e., no Thumb instructions).

- e) (10 points)** Rewrite the function using the skeleton code below. You only need to add three instructions. How many bytes of code does your new implementation of `mario` use?

```
mario          CMP r0,r1
               _____
               _____
               _____

yoshi          BX lr
```

- f) (5 points)** Translate the `mario` function into C. Your function header should list the types of any arguments and return values. Also, your code should be as concise as possible, without any `gotos`.

Name: _____