University of California, San Diego
Dept. of Computer Science and Engineering
CSE 30 – Computer Organization and Systems Programming
Problem Set #2 Rubric

Problem 1: Overflow (15 points)

a) Number 3 (the third operation) (5 points)
b) none (5 points)

Problem 2: String Manipulation (20 points)

-1 for any of the following:

- use of strlen(const char *)

-2 for any of the following:

- not returning -1 when search fails
- not returning any value
- using .length which is invalid in C
- using sizeOf() to determine the length of the string
- misuse of * or &
- use of “true”--true is not defined since many C compilers (notably ANSI C) don’t have a boolean type
- Index Out of Bounds Exception or Off by One Error

-5 for any of the following:

- finding the first occurrence of target in the string instead of finding the last occurrence
- returning a pointer to the last occurrence of target

-10 for any of the following:

- infinite loops
- implementing the wrong function (i.e. returning the total occurrences of target instead of the index of the last occurrence of target)

Problem 3: Pointers (20 points)

-8 Pts.

```
Arrow for py points to the wrong place or to multiple locations.
```

-3 Pts. for any of the following (-6pts total if all wrong)

```
Arrow for px points to the wrong place or to multiple locations.
Arrow for pz points to the wrong place or to multiple locations.
```

-2 Pts. for any of the following (-6pts if all wrong)

```
x holds the wrong value
y holds the wrong value
z holds the wrong value
```

-5 Pts

Not Folowing instructions: No Diagram/ Not using arrows to connect the pointers' boxes to the x,y,z boxes on the other side of the diagram.

Problem 4: Fibonacci Errors (10 points)

…..
int fib[24];
int i;
fib[0] = 0;
fib[1] = 0;
……

-3 points - incorrect initialization of one of fib[0] or fib[1]
for example: fib[0] = 1
-5 points - incorrect initialization of both fib[0] and fib[1]

Problem 5: **Average Errors (15 points)**

```
void average (int [] arr, int n)   // should be "int arr[]"
{
   for (i = 0; i ≤ n; i++)   // need to declare 'i' at the beginning
   {                         // of the function, "i < n"

       sum = arr[i] + sum;  // need to declare and initialize sum
   }
   // 1. change result to a double and declare it at the beginning
   // and cast sum and n separately
   // 2. keep result as an int but declare at the beginning
   int result = sum/n;

   // if result is a double, just add a comma after the ending quotes
   // if result is an int, change "%lf" to "%d" and add a comma after
   // the ending quotes
   printf ("The average is: %lf" result);
}
```

one possible working code:

```
void average (int arr[], int n)
{
   int i, sum = 0;
   double result;

   for (i = 0; i < n; i++)
   {
       sum = arr[i] + sum;
```

```
    }
    result = (double) sum / (double) n;
    printf ("The average is: %lf", result);
}
```

- 2 pts total: not declaring variables at the beginning
- 2 pts: not initializing sum to be 0
-2 pts: not casting (correctly)
- 1 pt: no comma
- 1 pt: didn't change $\leq$ to <
- 1 pt: didn't rewrite code

Problem 6: **Nested Loops (15 points)**

a) What are the values in `result_output` array?

0 0 0 0 0 0 6 7 8 00 11 12 13 0 0 16 17 18. Rest of them is zero

b) What are the values in the `result_output` if you change `result[m*N+n]` to `result[(m - 1) * (N - 2) + n - 1]` in the `filter` function ?

6 7 8 11 12 13 16 17 18. Rest of them is zero

Problem 7: Binary Manipulation (5 points)

-1 sytax

-1 incorrect looping

-2 not able to find odd digit

-2 not able to count 1s