1. (5 marks) Dominant terms. What is dominant term in the following expressions? Logarithms are base 2.
   a. $0.01n^3 + 2n^5 + 100n$
   b. $3^n + 10n^4 + \log n$
   c. $0.001\log(\log n) + \frac{5}{n} + \log n$
   d. $n! + n^{10}$

2. (5 marks) Arrays and pointers. What are possible outputs of the following code? (Circle all correct answers)

```c
#include <stdio.h>
int main() {
    int m[2] = {5, 8};
    int * n = m;
    printf("m = %d, n = %d\n", m, n);
}
```

a) Program will not compile
b) m = 5, n = 14285714
c) m = 14285714, n = 14285714
d) m = 5, n = 5
e) m = 5, n = 5

3. (5 marks) Character arrays and strings. What are possible outputs of the following code? (Circle all correct answers)

```c
#include <stdio.h>
int main() {
    char x[5] = "Hello world!";
    printf("%s, ", x);
    printf("%c\n", x+7);
}
```

a) Program will not compile
b) Hello world!, w
c) Hello, w
d) Hello, o
4. (20 marks) Recursion. The following unfinished recursive implementation of binary search is supposed to return 1 if target is found in the array arr, and 0 otherwise. Complete the implementation by filling in the base case. Use comments to explain your code.

```c
int BinarySearch(int arr[], int len, int target) {
    if (len<=4) {
        // Fill in code here
    }
    int mid = len/2;
    return BinarySearch(arr,mid,target)+BinarySearch(arr+mid,len-mid,target);
}
```
5. (30 marks) Proof of correctness. Consider the following code and associated output.

<table>
<thead>
<tr>
<th><strong>Code:</strong></th>
<th><strong>Output:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>#include &lt;stdio.h&gt;</td>
<td>1</td>
</tr>
<tr>
<td>void fun(int N) {</td>
<td>4</td>
</tr>
<tr>
<td>int z = 0;</td>
<td>9</td>
</tr>
<tr>
<td>for (int i = 0; i &lt; N; i++) {</td>
<td>16</td>
</tr>
<tr>
<td>z += i*2 + 1;</td>
<td>25</td>
</tr>
<tr>
<td>printf(&quot;%d\n&quot;, z);</td>
<td>36</td>
</tr>
<tr>
<td>}</td>
<td>49</td>
</tr>
<tr>
<td>}</td>
<td>64</td>
</tr>
<tr>
<td>int main() {</td>
<td>81</td>
</tr>
<tr>
<td>fun(10);</td>
<td>100</td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

a) What does the function output, for any given input integer N? (5 marks)

b) Prove your answer in part a) by applying the initialization, maintenance, and termination steps for proving program correctness. (20 marks)

c) What is the best big O estimate for the running time, with respect to the size of the input N? Give a brief justification to your answer. (5 marks)
6. Linked list. An implementation of the linked list is given below.

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct _node {
    int data;
    struct _node * next;
} node_t;

typedef struct {
    node_t * head;
    node_t * tail;
} LL_t;

LL_t * LLcreate() {
    LL_t * ret = malloc(sizeof(LL_t));
    ret->head = NULL;
    ret->tail = NULL;
    return ret;
}

void LLappend(LL_t * intlist, int value) {
    node_t * newNode = malloc(sizeof(node_t));
    newNode->data = value;
    newNode->next = NULL;

    if (intlist->head == NULL) {
        intlist->head = newNode;
        intlist->tail = newNode;
    } else {
        intlist->tail->next = newNode;
        intlist->tail = newNode;
    }
}
```

Currently, two functions have been implemented: LLcreate and LLappend. In this question, you will implement two additional functions. Please use comments to explain your code.
a) (20 marks) Write a function called LLcomp that compares two linked lists. The function should return 1 if the two linked lists are equal in length and contains all the same elements in the same order. Otherwise, the function should return 0. The function prototype is given below:

```c
int LLcomp(LL_t * intlist1, LL_t * intlist2) {
```
b) (15 marks) Write a function called LLupdate that updates the value of a node in a linked list. If the node is not found, the linked list should not change; instead, the user should get the following message: “Node not found!” The function prototype is given below:

```c
void LLupdate(LL_t * intlist, node_t * n, int value) {
```