I. Assume that the function Mystery has been defined as follows:

```c
int Mystery(int m, int n) {
    if (n == 0) {
        return 1;
    } else {
        return m * Mystery(m, n - 1);
    }
}
```

(a) (4 marks) What is the value of Mystery(2, 3)?

This function computes $m^n$.

The value of Mystery(2, 3) is 8.

(b) (4 marks) What is the computational complexity of Mystery(m, n) in term of the big-Ω notation, where $N$ is the value of the argument $n$, assuming $n \geq 0$?

$\Omega(n)$
2. Draw a heap-stack diagram showing how memory is allocated for each of the following problems. Use both explicit addresses and arrows.

(a) (5 marks)

```c
void main() {
    int n = 3;
    int array[2];
    array[1] = 1
    int *list = new int[n];
    list[2] = 2;
}
```

<----- diagram at this point

![Heap and Stack Diagram](image)

Continued on Page 3
(b) (5 marks)

```c
struct rectangleT {
    float length;
    float width;
}

rectangleT MakeRectangle(float l, float w);

int main() {
    rectangleT *rec = new rectangleT[2];
    rec[1] = MakeRectangle(4.5, 2.5);
    return 0;
}

rectangleT MakeRectangle(float l, float w) {
    rectangleT rect;
    rect.length = l;
    rect.width = w;
    return rect;
}
```

---

**Diagram**

Heap:
- `1000`
- `1004`
- `1008`
- `100C`

Stack:
- `0x1000`
- `0x1004`
- `0x1008`
- `0x100C`

Objects:
- `rect`
- `l` (4.5)
- `w` (2.5)
- `overhead`

---

Continued on Page 4
3. (10 marks) Complete the following implementation of the selection sort algorithm.

    /*    
    * Function Sort
    *       ---------
    * Sorts a array of effective size n using
    * the selection sort algorithm.
    */

    void Sort(int array[], int n) {
        int rh;
        int lh;

        for (lh = 0; lh < n-1; lh++) {
            rh = lh;
            for (int i = lh + 1; i < n; i++) {
                if (array[i] < array[rh]) {
                    rh = i;
                }
            }

            int tmp = array[lh];
            array[lh] = array[rh];
            array[rh] = tmp;
        }
    }

    Continued on Page 5
4. (10 marks) Assuming the type definition:

    struct cellT {
        int value;
        cellT *next;
    };

    Write a function `PrintLinkedList(cellT *head)` which takes a pointer to a linked list (without dummy cell) as input and prints the value of each cell of the list in order.

    void PrintLinkedList(cellT *head) {
        cellT *ptr;
        for (ptr = head; ptr != NULL; ptr = ptr->next) {
            cout << ptr->value << endl;
        }
    }
5. (7 marks) Assuming the type definition:

```c
struct cellT {
    char ch;
    cellT *next;
};
```

The following figure represents an editor buffer implemented by linked list.

```
... A | B ...
```

Complete the function

```c
void EditorBuffer::insertCharacter(char ch) {
    cellT *cp = new cellT;
    cp->ch = ch;
    cp->next = cursor->next;
    cursor->next = cp;
    cursor = cp;
}
```