

3) (10 pts) DSN (Linked Lists)

Write a function, `mode`, that takes in a pointer to the front of a linked list storing integers, and returns the mode of the list of integers. Recall that the mode of a list of values is the value that occurs most frequently. You may assume that all of the integers in the list are in between 0 and 999, inclusive. If there is more than one mode, your function must return the smallest of all of the modes. (For example, if the list contains the values 2, 4, 3, 2, 2, 4, 1, and 4, your function must return 2 and should NOT return 4, since both 2 and 4 occur three times in the list but 2 is smaller than 4.) Hint: declare an auxiliary array inside of the mode function. *You may assume that the list pointed to by front is non-empty.*

Use the struct definition provided below.

```
#include <stdlib.h>
#include <stdio.h>
#define MAX 1000

typedef struct node {
    int value;
    struct node* next;
} node;

int mode(node* front) {

    int freq[MAX], i;
    for (i=0; i<MAX; i++)
        freq[i] = 0;

    while (front != NULL) {
        freq[front->value]++;
        front = front->next;
    }

    int res = 0;
    for (i=1; i<MAX; i++)
        if (freq[i] > freq[res])
            res = i;

    return res;
}
```

Grading: 3 pts initializing frequency array, 4 pts filling frequency array, 3 pts finding index of maximum value of frequency array. Only take off 1 pt if ties are broken incorrectly.

Note: Please readjust points for solution ideas different than this as necessary.