3) (10 pts) DSN (Linked Lists)

Write a function, `moveFrontToBack`, that takes in a pointer to the front of a ***doubly*** linked list storing an integer, moves the first node of the list to the back of the list and returns a pointer to the new front of the list. If the list contains fewer than two elements, the function should just return the list as it is. (Note: prev points to the previous node in the list and next points to the next node in the list.)

Use the struct definition provided below.

```
typedef struct dllnode {
    int value;
    struct dllnode* prev;
    struct dllnode* next;
} dllnode;

dllnode* moveFrontToBack(dllnode* front) {

    if (front == NULL || front->next == NULL)      // 2 pts
        return front;

    dllnode* newfront = front->next;

    dllnode* back = newfront;                      // 3 pts iterating to back
    while (back->next != NULL)
        back = back->next;

    back->next = front;                            // 1 pt
    front->prev = back;                            // 1 pt
    front->next = NULL;                            // 1 pt
    newfront->prev = NULL;                         // 1 pt

    return newfront;                               // 1 pt
}
```

Grading conceptually: 2 pts base cases, 3 pts iterating to back, 5 pts reattaching things and returning.