Handout for CS16—Midterm Exam E02

W15, Phill Conrad, UC Santa Barbara

The following struct definitions are used in some questions on this exam.

```
struct StudentGpa {
  int perm;
  double gpa;
};

struct Student {
  int perm;
  Student *next;
};

struct StudentList {
  Student *head;
  Student *tail;
};
```

Function `addPermToList`

The function `addPermToList` takes two parameters:

- `sList` is a pointer to a StudentList struct with the head and tail of a linked list of Student structs.
- `perm` is an integer that is a perm number that should be placed in a new Student struct and added at the tail of the list.

The function does not return anything. It simply allocates a new Student struct on the heap, puts perm in it, and adds that struct at the tail of the list.

Function `honorsStudentList`

The function `honorsStudentList` takes three parameters:

- `students` is an array of StudentGpa structs
- `numStudents` is the number of elements in that array (the occupancy)
- `sList` is a pointer to a StudentList struct, that initially has an empty list (i.e. head and tail are both pointing to null before the function is called.)

The function does not return anything. Instead, as a side effect of calling the function, the `head` and `tail` members of the `StudentList` struct passed in are set to point to a linked list of `Student` structs, each one containing the perm of a student that has a gpa of 3.5 or higher.

To put it another way, at the end of the function call, for each `StudentGPA` struct that was in the array `students`, where the gpa was 3.5 or higher, there should be a corresponding `Student` struct in the linked list pointed to by `list`. The elements in the linked list should be in the same order that they appeared in the array, except ONLY the ones with a 3.5 gpa or higher appear in the linked list.