Part I: Write and Test C code/program

Suppose the following struct and variable declarations for linked list with pointer, where the linked list is ordered by the ABC123 ID. Each node is represented by the following Node definition:

```c
typedef struct{
    char szAbc123Id[7];
    double dGPA;
} Student;
typedef struct Node
{
    Student student;       // This is different from last week since
    // student is in the node instead of
    // directly containing szAbc123Id and dGpa.
    struct Node *pNext;
} Node;
Node *pHead;
Node *pFind, *pWorst, *pPrecedes;
```

Create a linked list with the following 5 elements:

```
   pHead    "apg238"  3.76    "byi761"  3.21    "ijx218"  3.52    "jnv088"  3.11    "uyx181"  3.85    NULL
```

Implement and test the following functions.

- **void printHighGPA(Node *pHead)**, which prints the ABC123 ID and GPA for students having a GPA \(\geq 3.5\) of students in the linked list that begins at pHead.
- **Node *searchLL(Node *pHead, const char szMatchId[], Node ** ppPrecedes)** that uses the ABC123 ID and find the specified student in the linked list. If it is found, return the pointer to the node containing that ID. If it is not found, return NULL. Also, return (via the parameter list) the pointer to the node that precedes that node. If there isn’t a preceding node, this should be NULL.
- **Node *getWorstStudent(Node *pHead)**, which is passed the pointer to a linked list. It returns a pointer to the node for the student with the lowest GPA. Return NULL if there aren’t any nodes in the linked list.