Name:....

Q13

CS2123 Data Structures

Q1. (10pt) We use the following cell structure to store the IDs and names of some students in a **single linked list** data structure

```
typedef struct cell {
    int ID;
    char name[20];
    struct cell *next,;
} cellT, *cellPtrT;
```

Suppose somehow we have created the following two single linked lists, namely M and S, to store the IDs and names of the students who are taking Math and Science, respectively. Lists are **sorted** w.r.t. IDs.



Now we are interested in creating a new list, say B, to store the IDs and names of the students who are taking **both** Math and Science. In other words, we want to find the **intersection** set of M and S. The resulting list B should be sorted w.r.t. IDs, as the original lists. So list B will look like as follows.



You are asked to implement a function, cellT *intersection (cellT *M, cellT *S), which can be called as follows to find the intersection of the given two sets.

```
cellT *M, *S, *B; // cellPtrT M, S, B;
/* somehow the lists pointed by M and S are created */
B = intersection(M, S);
```

After your function, M and S should be intact. So do not remove the cells from M or S! When needed, create new cells for B and copy the ID and name from the cells in M or S.

In another design choice, call it as: intersection2(&B, M, S);



```
Name:.....
                       Answer question 1:
cellT *intersection(cellT *M, cellT *S)
                                           void intersection2(cellT **Bptr,
                                                cellT *M, cellT *S)
//cellPtrT intersection(cellPtrT M,
                                           // void intersection2(cellPtrT *Bptr,
11
                        cellPtrT S)
                                                      cellPtrT M, cellPtrT S)
                                           11
{
                                           {
```