**Linked List**

**Student complete all method in ListPhone.java (f0-f8)**

**void f0\_information();//** Display your RollNumber (Student number)

**void f1\_addHead(Phone x);**//check if type of x is Op or price >40 do nothing otherwise add x to the head of list

*Output:*

(Ip, 8Plus, 3.52);(Ip, 8, 3.0);(Ip, 7Plus, 3.0);(Ip, 7, 2.0);(Ip, 14, 33.0);(Ip, 12Pro, 19.0);(SS, Note21, 17.0);(Ip, 13, 33.0);(Ip, 12, 18.0);(Ip, 11Pro, 15.0);(SS, Note20, 14.0);(Ip, 11, 10.0);

**void f2\_addtail(Phone x)**;// check if type of x is Op or price >40 do nothing otherwise add x to the tail of list

*Output:*

*(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

**Node f3\_InsertBefore()**;//insert Phone x before first Phone has max price.

*Input:*

*(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

*Result:*

*(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);****(Iphone, X, 0.0);(****Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

**Node f4\_InsertAfter()** //insert Phone x after last Phone has max price.

*Input:*

*(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

*Result:*

*(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);****(Iphone, X, 0.0);(****Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

**Node f5\_removeOne()** //remove first Phone has price <20

*Input:*

*(Ip, 14, 32.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

*Result:*

*(Ip, 14, 32.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);*

**Node f6\_removeAll()** //remove all Phones have price <20

Input:

(Ip, 13, 20.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 14, 33.0);

Result:

(Ip, 13, 20.0);(Ip, 13, 33.0);(Ip, 14, 33.0);(Ip, 14, 33.0);

**Node f7\_Sort1()** // sort list ascending by price

Input:

(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);

Result:

(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, 8, 3.0);(Ip, 8Plus, 3.52);(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(SS, Note21, 17.0);(Ip, 12, 18.0);(Ip, 12Pro, 19.0);(Ip, 13, 33.0);(Ip, 14, 33.0);

**Node f8\_sort2()** //sort by price increment from head to last maximum price

Input:

(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(Ip, 12, 18.0);(Ip, 13, 33.0);(SS, Note21, 17.0);(Ip, 12Pro, 19.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, X, 6.0);(Ip, 8Plus, 3.52);

Result:

(Ip, 11, 10.0);(SS, Note20, 14.0);(Ip, 11Pro, 15.0);(SS, Note21, 17.0);(Ip, 12, 18.0);(Ip, 12Pro, 19.0);(Ip, 13, 33.0);(Ip, 14, 33.0);(Ip, 7, 2.0);(Ip, 7Plus, 3.0);(Ip, X, 6.0);(Ip, 8Plus, 3.52);