1.

Algorithm:

Union(L1,L2)

1) Declare node pointer output, output Tail as NULL

2) Repeat steps 3 to 9 while L1!=NULL AND L2!=NULL

3) Make a newNode and set its next = NULL

4) If L1->data < L2->data then

Set newNode->data = L1->data

Set L1 = L1->next

5) Else if L1->data > L2->data then

Set newNode->data = L2->data

L2 = L2->next

6) Else

i) Set Data = L1->data

ii) Set newNode->data = Data

iii) Repeat steps a) and b)

while L1!=NULL AND L2!=NULL AND L1->data == Data AND L2->data == Data

a) Set L1 = L1->next

b) Set L2 = L2->next

7) If output == NULL then

Set Output = outputTail = newNode

8) Else

a) Set outputTail->next = newNode

b) Set outputTail = outputTail->next

9) Repeat steps 10 to 14 while L1!=NULL

10) Make a newNode

11) Set outputTail->next = newNode

12) Set outputTail = outputTail->next

13) Set outputTail->data = L1->data

14) Set L1 = L1->next

Repeat steps 15 to 19 while L2!=NULL

15) Make a newNode

16) Set outputTail->next = newNode

17) Set outputTail = outputTail->next

18) Set outputTail->data = L2->data

19) Set L2 = L2->next

Return output

Intersection(L1,L2)

1. If L1 or L2 is NULL then return NULL

2. Declare node pointers output, outputTail as null.

3. Repeat steps 4 to 6 while L1!=NULL AND L2!=NULL

4. If L1->datadata then

Set L1 = L1->next

5. Else If L2->datadata then

Set L2 = L2->next

6. Else

a) Declare and set data = L1->data

b) Make a newNode

c) Set newNode->data = data and newNode->next = NULL

d) If output == null then

i. Set output = outputTail = newNode

e) Else

i. Set outputTail->next = newNode

ii. Set outputTail = outputTail->next

f) Repeat steps i and ii

while L1!=NULL AND L2!=NULL AND L1->data == data AND L2->data == data

i. Set L1 = L1->next

ii. Set L2 = L2->next