**Count Pairs whose sum is equal to X**

Given two linked list **head1**and **head2** with **distinct**elements, determine the **count of all distinct pairs** from both lists whose sum is equal to the given value **x**.

**Note**: A valid pair would be in the form **(x, y)**where **x** is from first linked list and **y** is from second linked list.

**Example 1:**

**Input:**

head1 = 1->2->3->4->5->6

head2 = 11->12->13

x = 15

**Output:** 3

**Explanation:** There are total 3 pairs whose sum is 15 : (4,11) , (3,12) and (2,13)

**Example 2:**

**Input:**

head1 = 7->5->1->3

head2 = 3->5->2->8

x = 10

**Output:** 2

**Explanation:** There are total 2 pairs whose sum is 10 : (7,3) and (5,5)

**Your Task:**  
You only need to implement the given function **countPairs()**that take two linked list **head1** and **head2**and return the **count of distinct** **pairs** whose sum is equal to **x**.

**Expected Time Complexity:** O(length(head1)+lenght(head2)).  
**Expected Auxiliary Space:** O(length(head1)) or O(length(head2)).

**Constraints:**  
1<=length(head1), lenght(head2)<=105  
1 <= Value of elements of  linked lists <= 109  
1<= x <= 109**Note** : All elements in each linked list are unique.

class Solution{

public:

int countPairs(struct Node\* head1, struct Node\* head2, int x) {

unordered\_map<int,int>mp1;

Node\*temp1 = head1;Node\*temp2 = head2;

while(temp1){

mp1[temp1->data]++;

temp1 = temp1->next;

}

int ct = 0;

while(temp2){

if(mp1.find(x-temp2->data)!=mp1.end()){

ct++;

}

temp2 = temp2->next;

}

return ct;

}

};

Link : <https://www.geeksforgeeks.org/problems/count-pairs-whose-sum-is-equal-to-x/1>