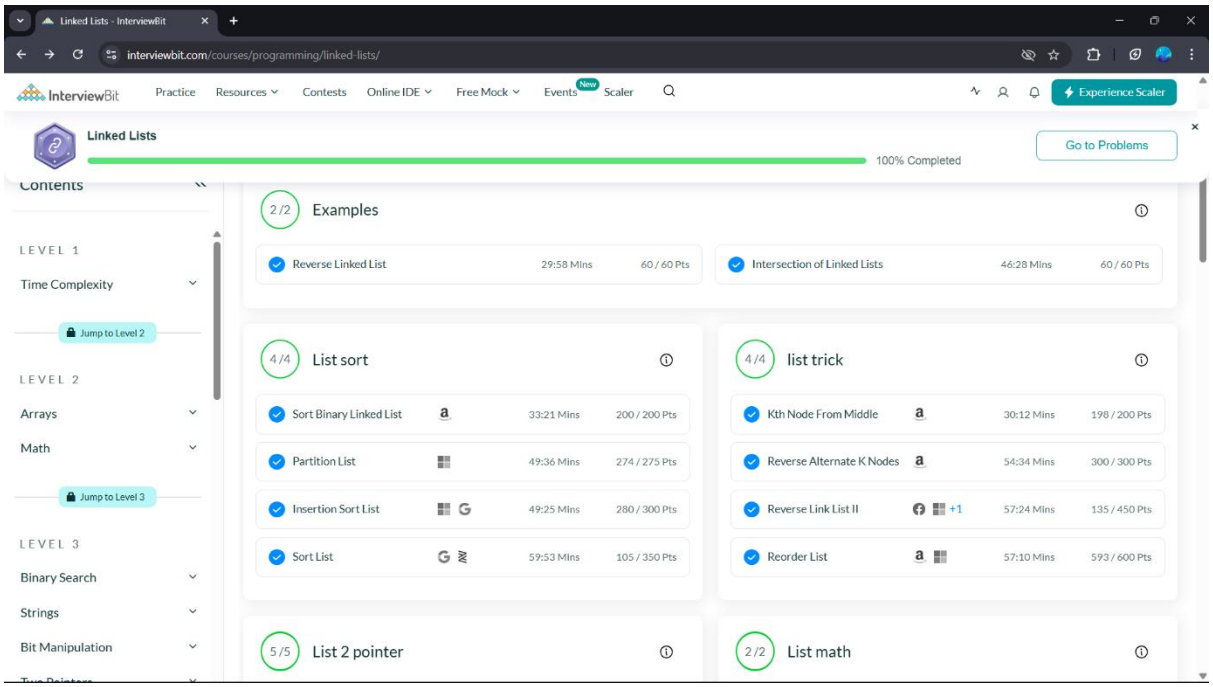
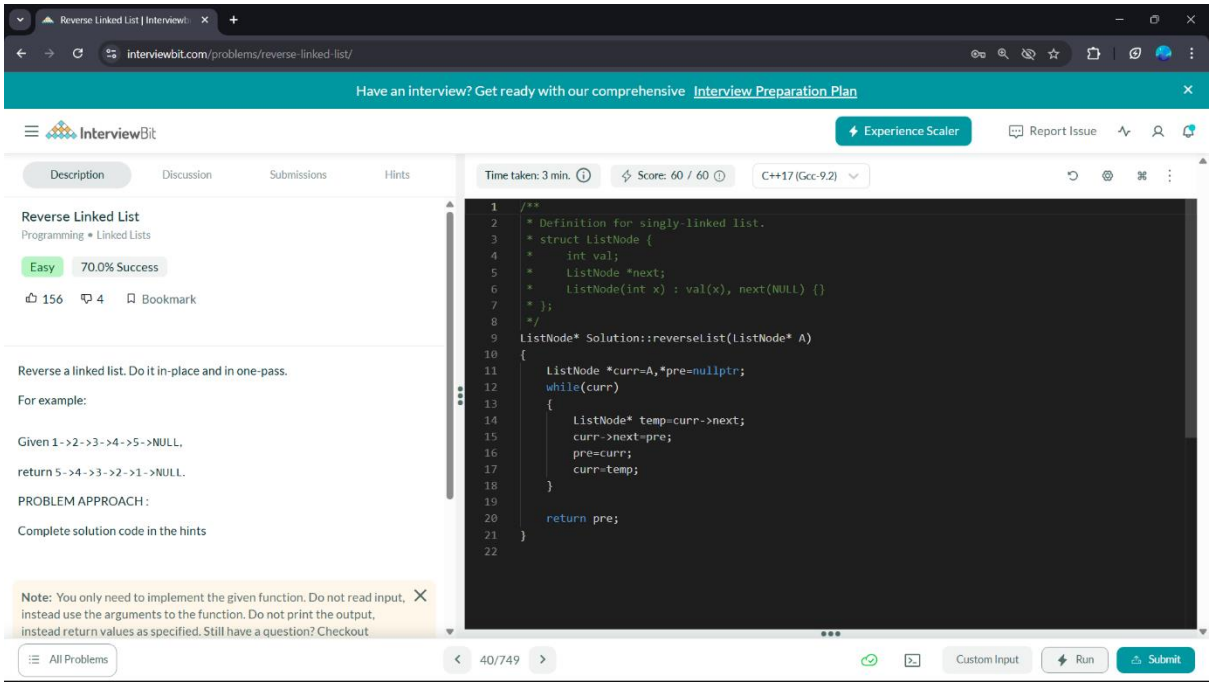
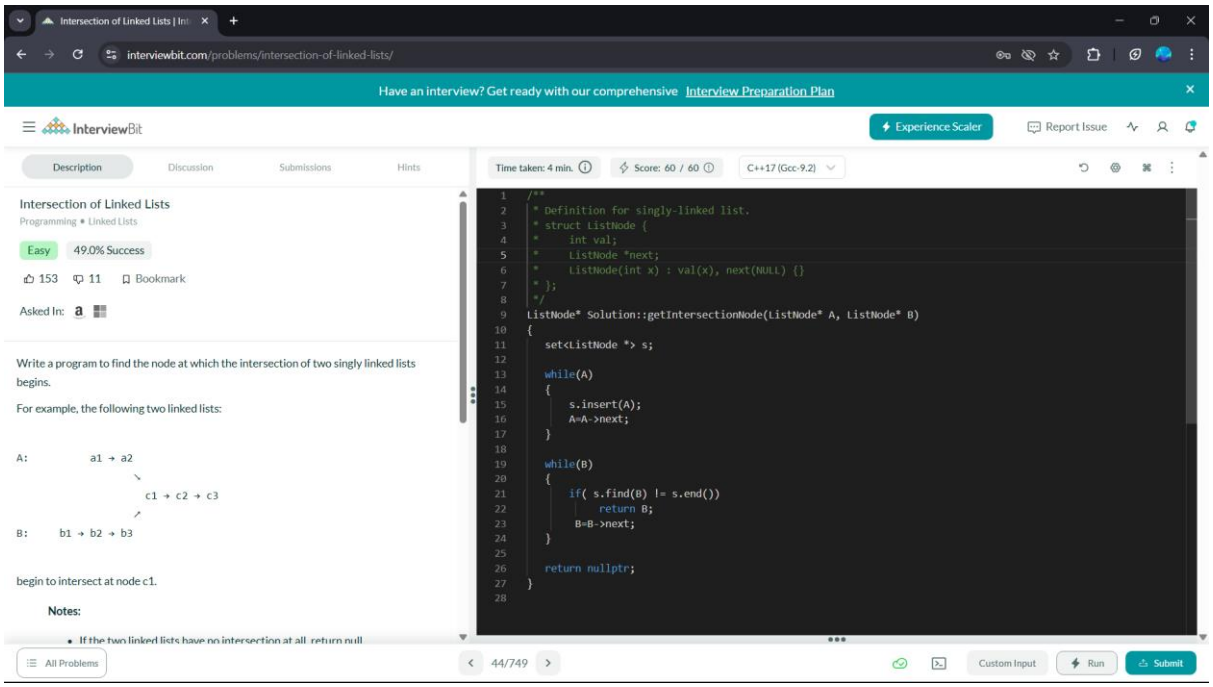
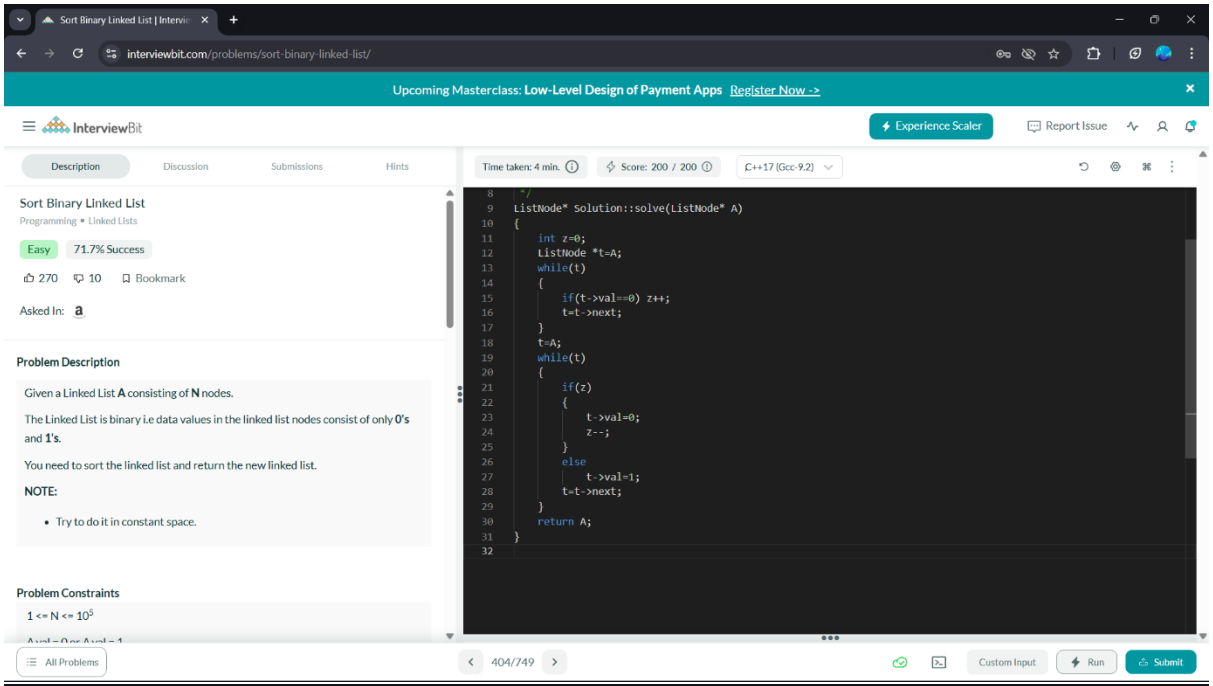
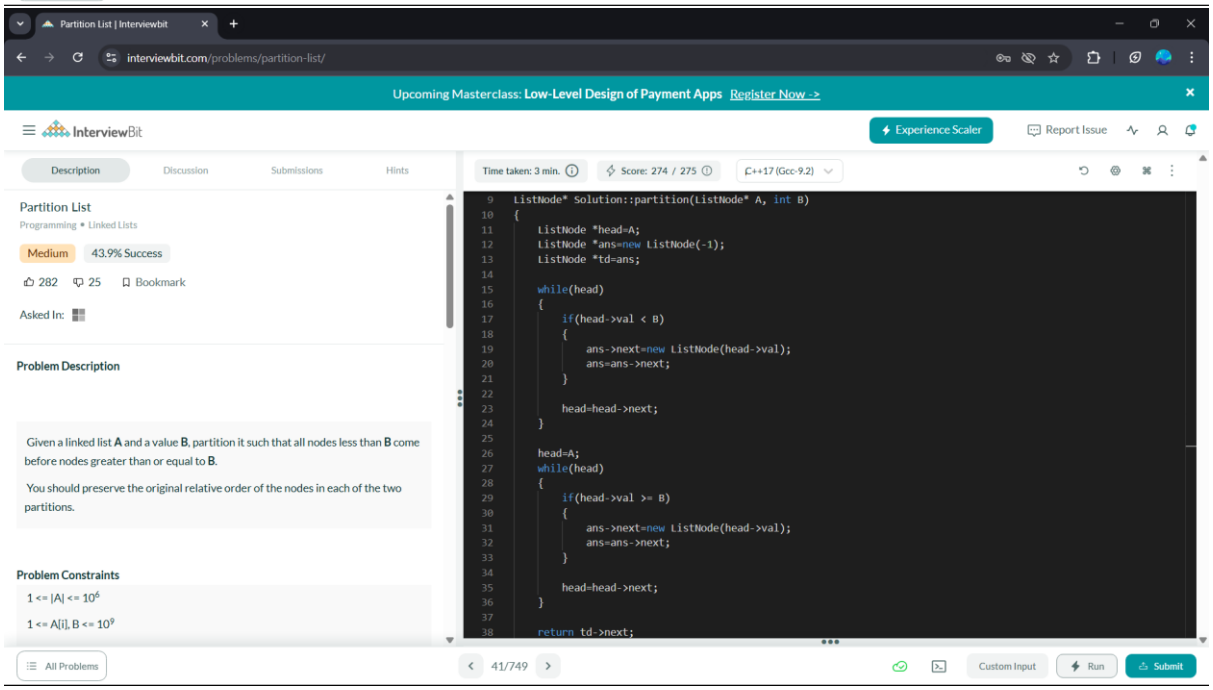
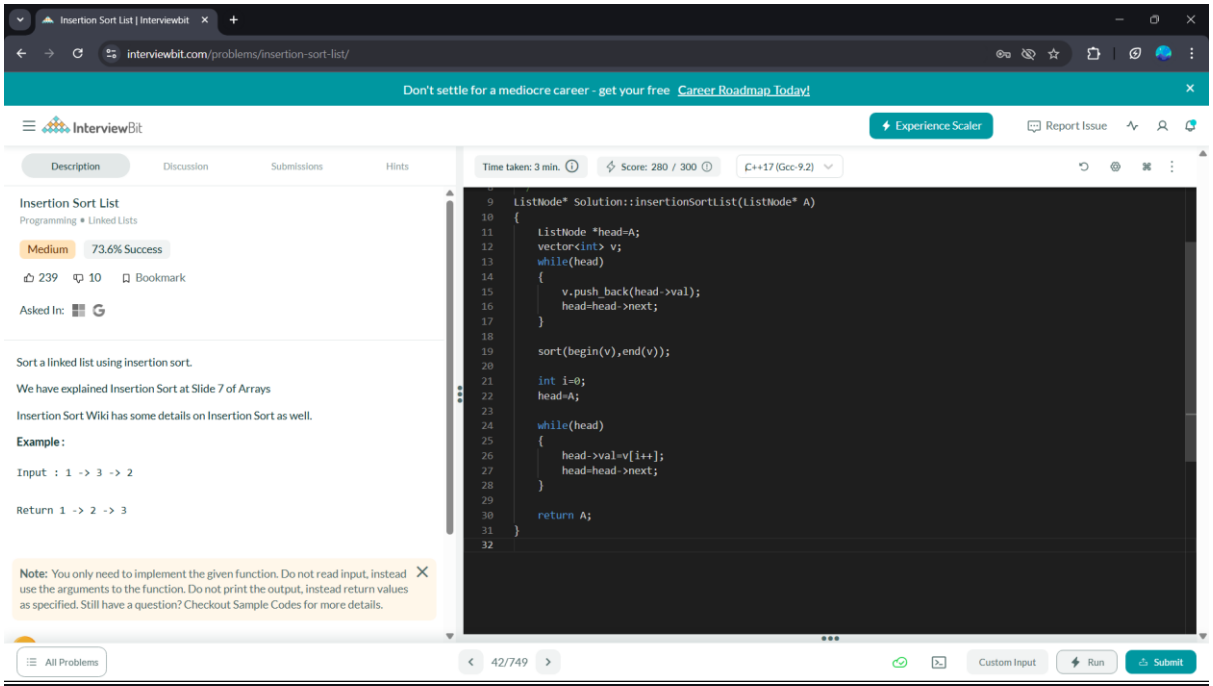

Linked Lists - InterviewBit
interviewbit.com/courses/programming/linked-lists/
InterviewBit
Practice
Resources
Contests
Online IDE
Free Mock
Events
New
Scaler
Experience Scaler
Linked Lists
100% Completed
Go to Problems
Contents
LEVEL 1
Time Complexity
Jump to Level 2
LEVEL 2
Arrays
Math
Jump to Level 3
LEVEL 3
Binary Search
Strings
Bit Manipulation
2/2 Examples
Reverse Linked List 29:58 Mins 60 / 60 Pts
Intersection of Linked Lists 46:28 Mins 60 / 60 Pts
4/4 List sort
Sort Binary Linked List 33:21 Mins 200 / 200 Pts
Partition List 49:36 Mins 274 / 275 Pts
Insertion Sort List 49:25 Mins 280 / 300 Pts
Sort List 59:53 Mins 105 / 350 Pts
4/4 list trick
Kth Node From Middle 30:12 Mins 198 / 200 Pts
Reverse Alternate K Nodes 54:34 Mins 300 / 300 Pts
Reverse Link List II +1 57:24 Mins 135 / 450 Pts
Reorder List 57:10 Mins 593 / 600 Pts
5/5 List 2 pointer
2/2 List math

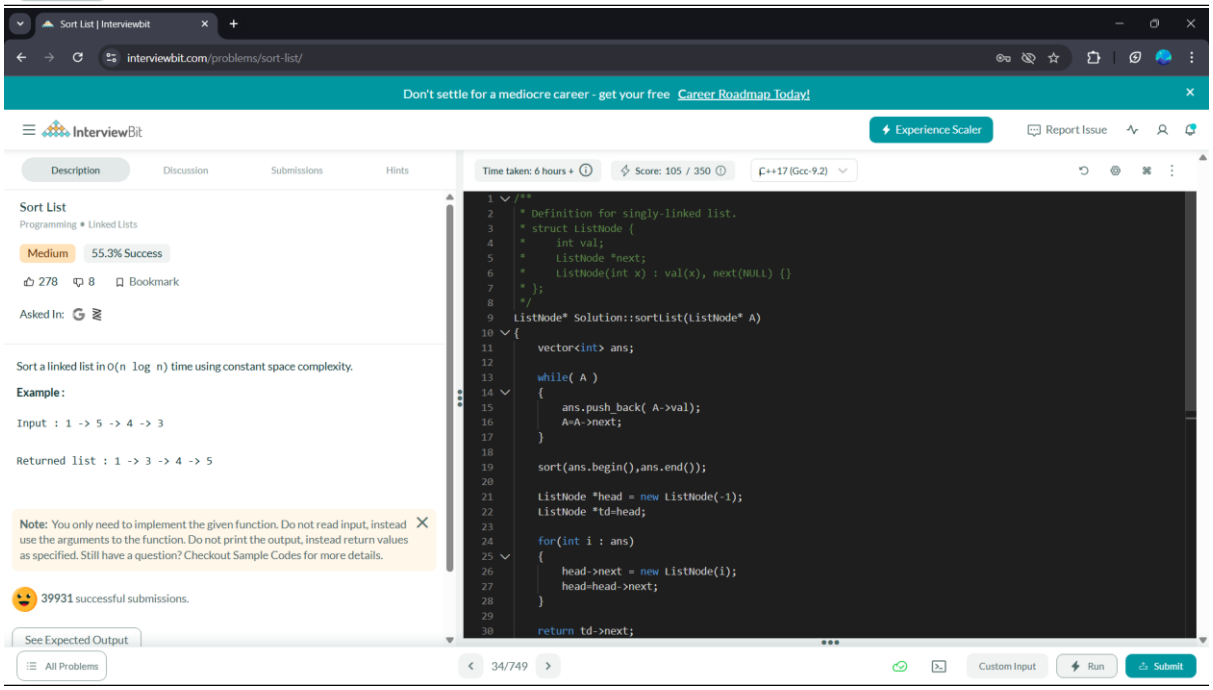

Reverse Linked List | Interview
interviewbit.com/problems/reverse-linked-list/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 3 min.
Score: 60 / 60
C++17 (Gcc-9.2)
Reverse Linked List
Programming • Linked Lists
Easy
70.0% Success
156
4
Bookmark
Reverse a linked list. Do it in-place and in one-pass.
For example:
Given 1->2->3->4->5->NULL,
return 5->4->3->2->1->NULL.
PROBLEM APPROACH :
Complete solution code in the hints
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::reverseList(ListNode* A)
{
ListNode *curr=A,*pre=nullptr;
while(curr)
{
ListNode* temp=curr->next;
curr->next=pre;
pre=curr;
curr=temp;
}
return pre;
}
All Problems
40/749
Custom Input
Run
Submit

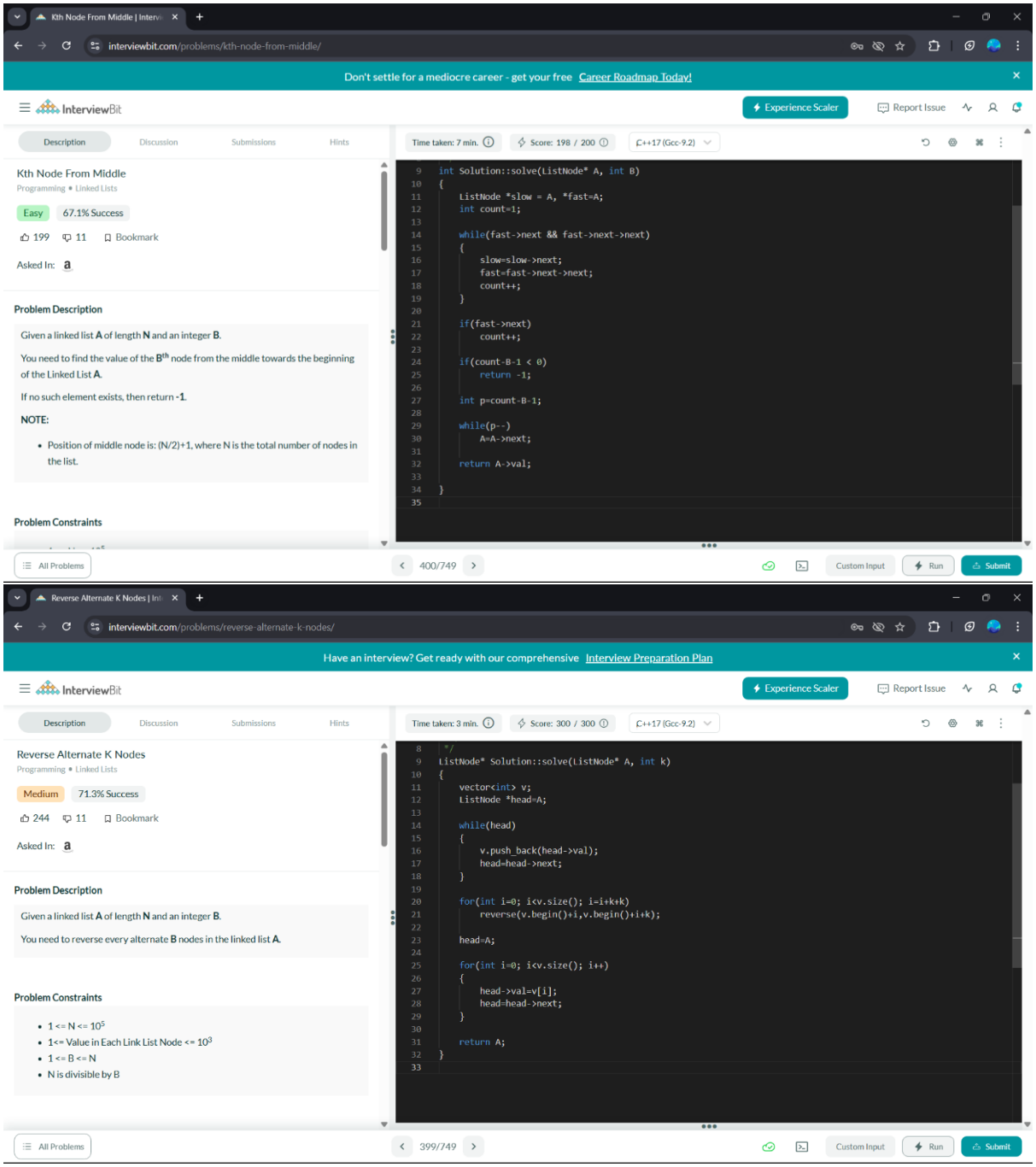

Intersection of Linked Lists | In
interviewbit.com/problems/intersection-of-linked-lists/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 4 min.
Score: 60 / 60
C++17 (Gcc-9.2)
Intersection of Linked Lists
Programming • Linked Lists
Easy
49.0% Success
153
11
Bookmark
Asked In:
Write a program to find the node at which the intersection of two singly linked lists begins.
For example, the following two linked lists:
A: a1 → a2
c1 → c2 → c3
B: b1 → b2 → b3
begin to intersect at node c1.
Notes:
If the two linked lists have no intersection at all, return null
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::getIntersectionNode(ListNode* A, ListNode* B)
{
set<ListNode *> s;
while(A)
{
s.insert(A);
A=A->next;
}
while(B)
{
if(s.find(B) != s.end())
return B;
B=B->next;
}
return nullptr;
}
All Problems
44/749
Custom Input
Run
Submit


Sort Binary Linked List | Interviewbit
interviewbit.com/problems/sort-binary-linked-list/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 4 min.
Score: 200 / 200
C++17 (Gcc-9.2)
Sort Binary Linked List
Programming • Linked Lists
Easy
71.7% Success
270
10
Bookmark
Asked In:
Problem Description
Given a Linked List A consisting of N nodes.
The Linked List is binary i.e data values in the linked list nodes consist of only 0's and 1's.
You need to sort the linked list and return the new linked list.
NOTE:
Try to do it in constant space.
Problem Constraints
1 <= N <= 10^5
*/
ListNode* Solution::solve(ListNode* A)
{
int z=0;
ListNode *t=A;
while(t)
{
if(t->val==0) z++;
t=t->next;
}
t=A;
while(t)
{
if(z)
{
t->val=0;
z--;
}
else
t->val=1;
t=t->next;
}
return A;
}
All Problems
404/749
Custom Input
Run
Submit

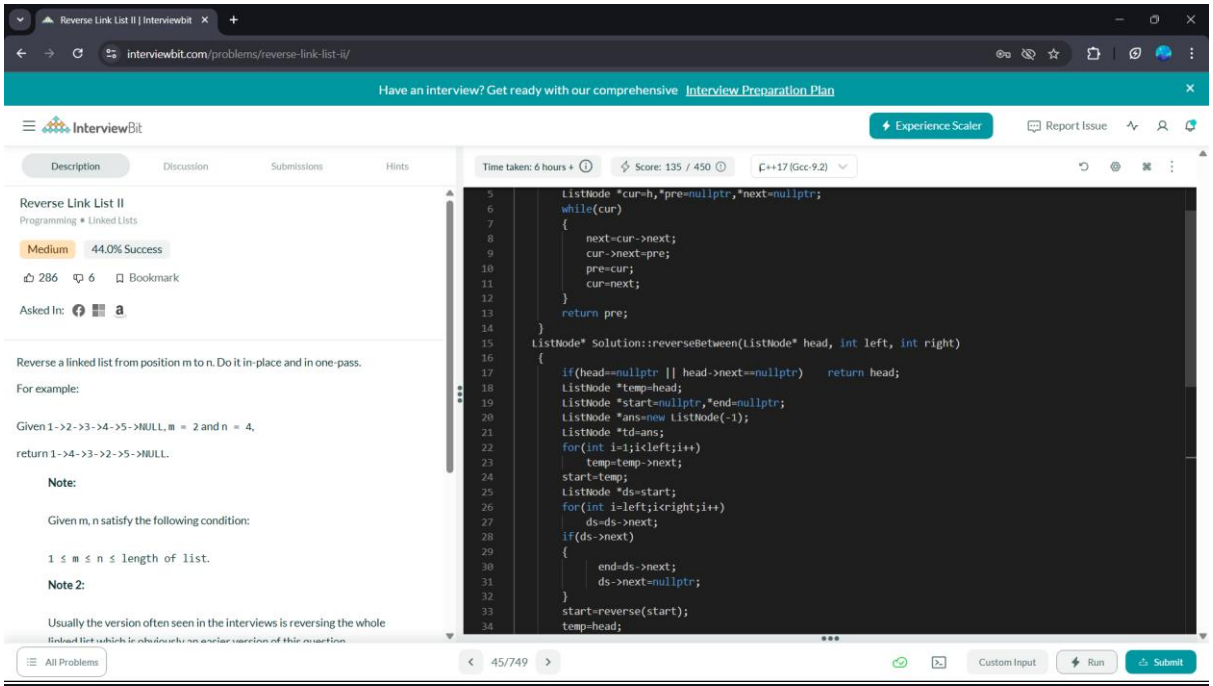

Partition List | Interviewbit
interviewbit.com/problems/partition-list/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 3 min.
Score: 274 / 275
C++17 (Gcc-9.2)
Partition List
Programming • Linked Lists
Medium
43.9% Success
282
25
Bookmark
Asked In:
Problem Description
Given a linked list A and a value B, partition it such that all nodes less than B come before nodes greater than or equal to B.
You should preserve the original relative order of the nodes in each of the two partitions.
Problem Constraints
1 <= |A| <= 10^6
1 <= A[i], B <= 10^9
ListNode* Solution::partition(ListNode* A, int B)
{
ListNode *head=A;
ListNode *ans=new ListNode(-1);
ListNode *td=ans;
while(head)
{
if(head->val < B)
{
ans->next=new ListNode(head->val);
ans=ans->next;
}
head=head->next;
}
head=A;
while(head)
{
if(head->val >= B)
{
ans->next=new ListNode(head->val);
ans=ans->next;
}
head=head->next;
}
return td->next;
All Problems
41/749
Custom Input
Run
Submit

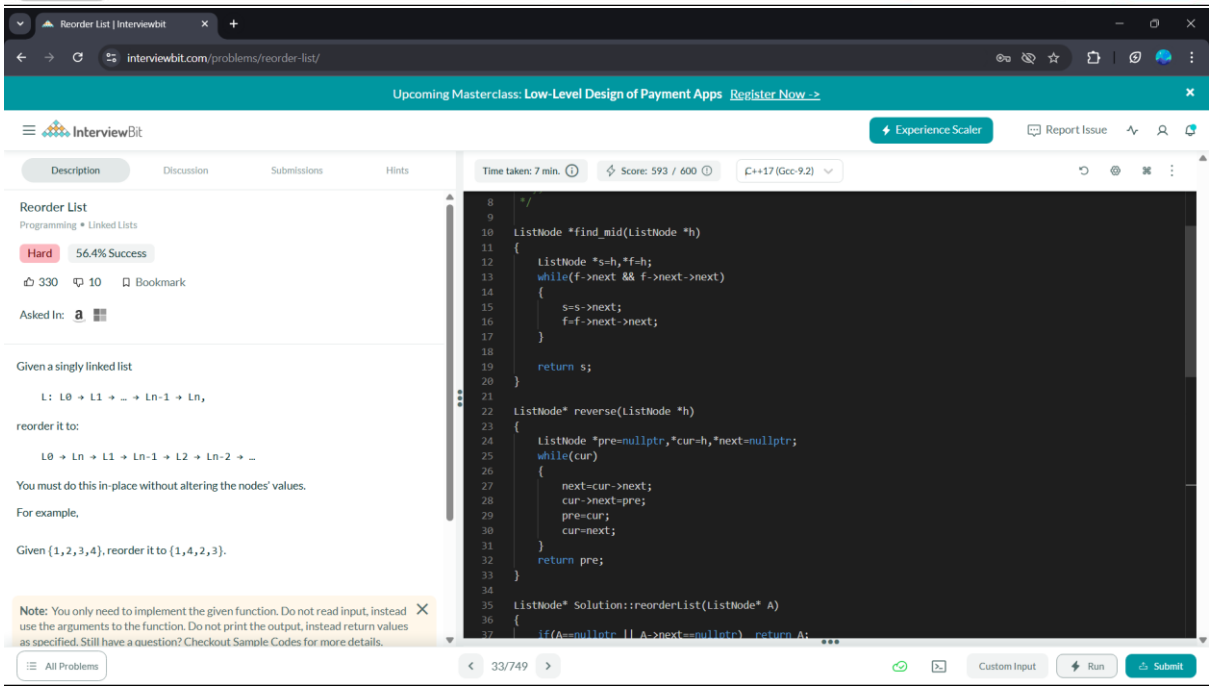

Insertion Sort List | Interviewbit
interviewbit.com/problems/insertion-sort-list/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 3 min.
Score: 280 / 300
C++17 (Gcc-9.2)
Insertion Sort List
Programming • Linked Lists
Medium
73.6% Success
239
10
Bookmark
Asked In:
Sort a linked list using insertion sort.
We have explained Insertion Sort at Slide 7 of Arrays
Insertion Sort Wiki has some details on Insertion Sort as well.
Example :
Input : 1 -> 3 -> 2
Return 1 -> 2 -> 3
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
ListNode* Solution::insertionSortList(ListNode* A)
{
ListNode *head=A;
vector<int> v;
while(head)
{
v.push_back(head->val);
head=head->next;
}
sort(begin(v),end(v));
int i=0;
head=A;
while(head)
{
head->val=v[i++];
head=head->next;
}
return A;
}
All Problems
42/749
Custom Input
Run
Submit

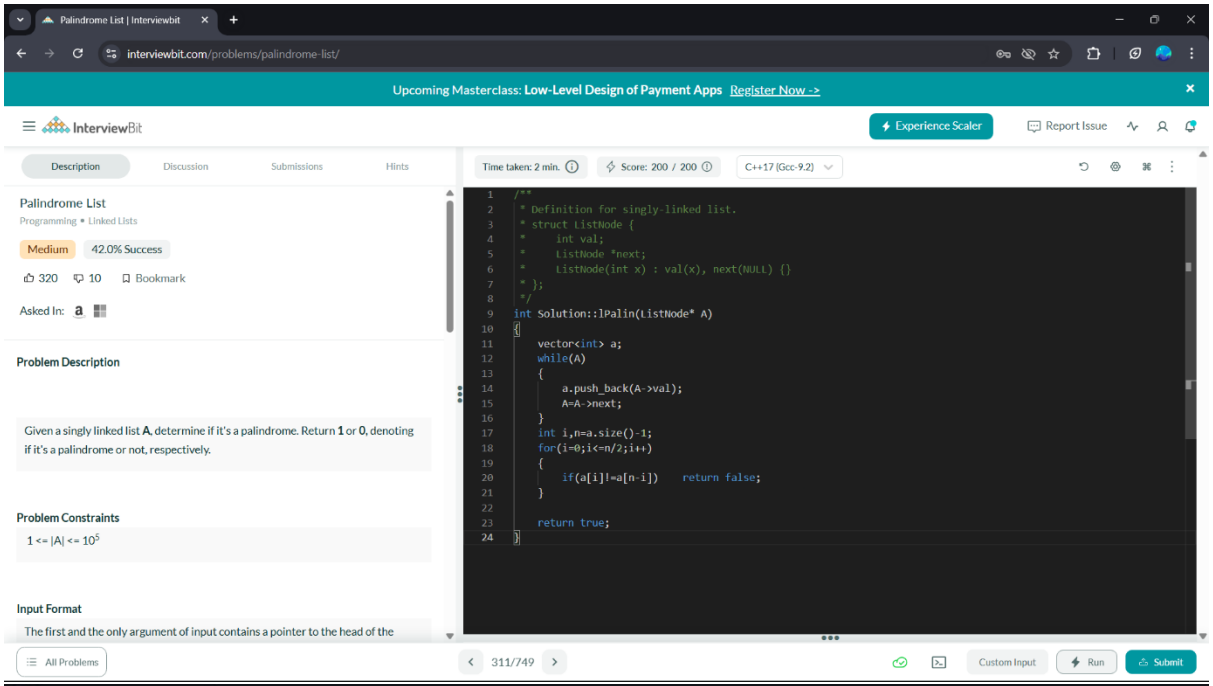

Sort List | Interviewbit
interviewbit.com/problems/sort-list/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 6 hours +
Score: 105 / 350
C++17 (Gcc-9.2)
Sort List
Programming • Linked Lists
Medium
55.3% Success
278
8
Bookmark
Asked In:
Sort a linked list in O(n log n) time using constant space complexity.
Example :
Input : 1 -> 5 -> 4 -> 3
Returned list : 1 -> 3 -> 4 -> 5
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
39931 successful submissions.
See Expected Output
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::sortList(ListNode* A)
{
vector<int> ans;
while(A)
{
ans.push_back(A->val);
A=A->next;
}
sort(ans.begin(),ans.end());
ListNode *head = new ListNode(-1);
ListNode *td=head;
for(int i : ans)
{
head->next = new ListNode(i);
head=head->next;
}
return td->next;
All Problems
34/749
Custom Input
Run
Submit

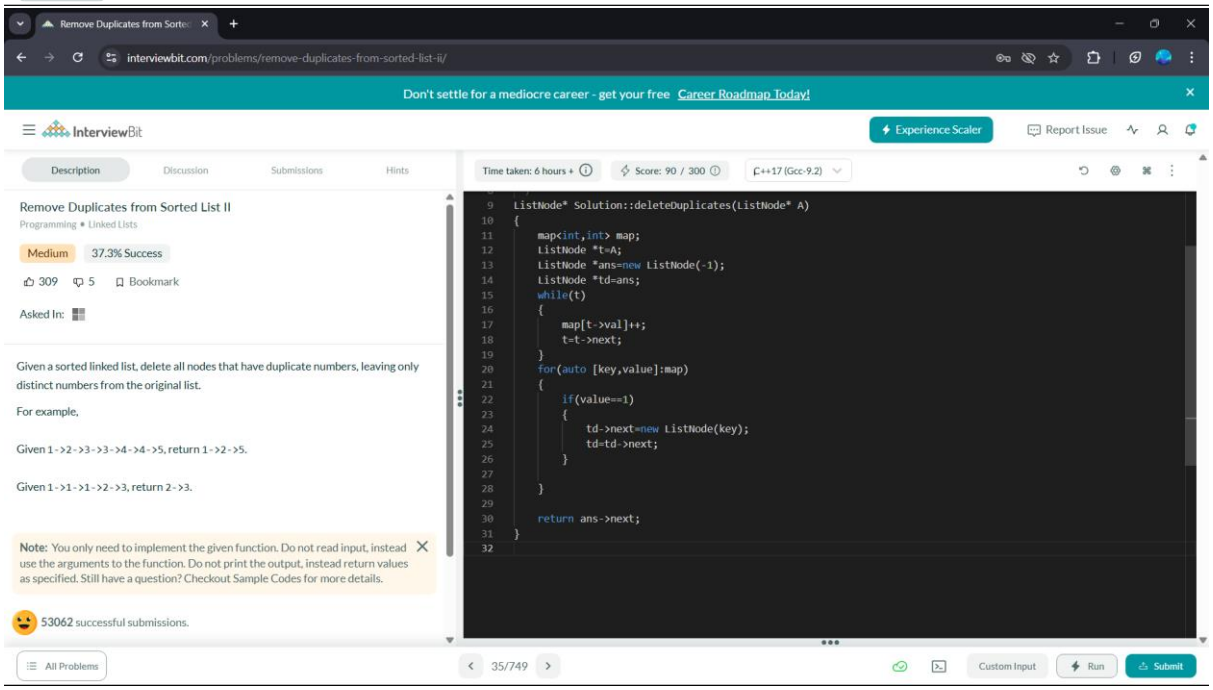

Kth Node From Middle | Interv
interviewbit.com/problems/kth-node-from-middle/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 7 min.
Score: 198 / 200
C++17 (Gcc-9.2)
Kth Node From Middle
Programming • Linked Lists
Easy
67.1% Success
199
11
Bookmark
Asked In:
Problem Description
Given a linked list A of length N and an integer B.
You need to find the value of the B^th node from the middle towards the beginning of the Linked List A.
If no such element exists, then return -1.
NOTE:
Position of middle node is: (N/2)+1, where N is the total number of nodes in the list.
Problem Constraints
int Solution::solve(ListNode* A, int B)
{
ListNode *slow = A, *fast=A;
int count=1;
while(fast->next && fast->next->next)
{
slow=slow->next;
fast=fast->next->next;
count++;
}
if(fast->next)
count++;
if(count-B-1 < 0)
return -1;
int p=count-B-1;
while(p--)
A=A->next;
return A->val;
}
All Problems
400/749
Custom Input
Run
Submit

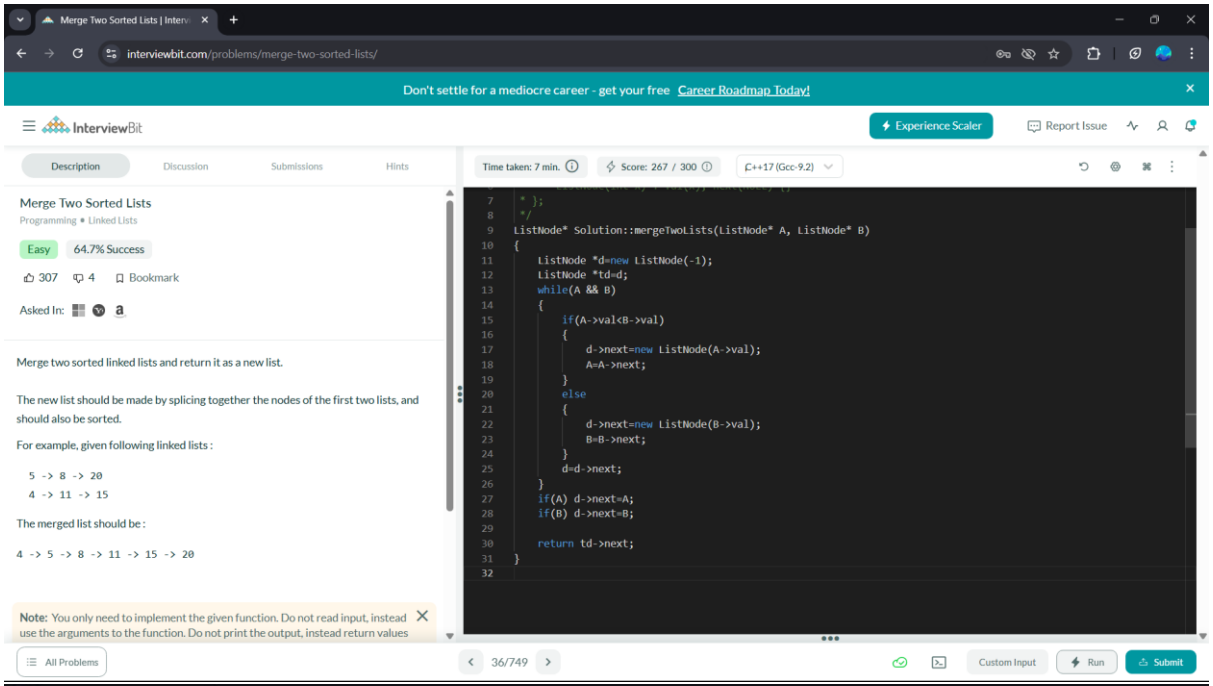
Reverse Alternate K Nodes | In
interviewbit.com/problems/reverse-alternate-k-nodes/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 3 min.
Score: 300 / 300
C++17 (Gcc-9.2)
Reverse Alternate K Nodes
Programming • Linked Lists
Medium
71.3% Success
244
11
Bookmark
Asked In:
Problem Description
Given a linked list A of length N and an integer B.
You need to reverse every alternate B nodes in the linked list A.
Problem Constraints
1 <= N <= 10^5
1<= Value in Each Link List Node <= 10^3
1 <= B <= N
N is divisible by B
*/
ListNode* Solution::solve(ListNode* A, int k)
{
vector<int> v;
ListNode *head=A;
while(head)
{
v.push_back(head->val);
head=head->next;
}
for(int i=0; i<v.size(); i=i+k+k)
reverse(v.begin()+i,v.begin()+i+k);
head=A;
for(int i=0; i<v.size(); i++)
{
head->val=v[i];
head=head->next;
}
return A;
}
All Problems
399/749
Custom Input
Run
Submit

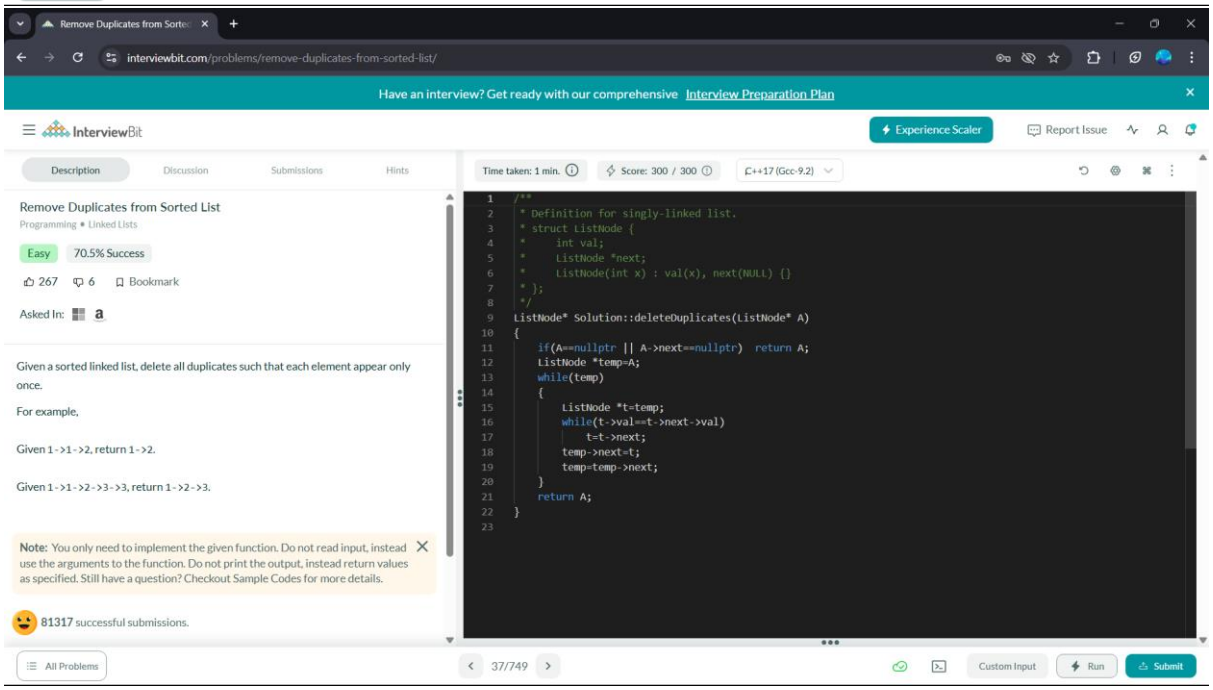

Reverse Link List II | Interviewbit
interviewbit.com/problems/reverse-link-list-ii/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 6 hours +
Score: 135 / 450
C++17 (Gcc-9.2)
Reverse Link List II
Programming • Linked Lists
Medium
44.0% Success
286
6
Bookmark
Asked In:
Reverse a linked list from position m to n. Do it in-place and in one-pass.
For example:
Given 1->2->3->4->5->NULL, m = 2 and n = 4,
return 1->4->3->2->5->NULL.
Note:
Given m, n satisfy the following condition:
1 ≤ m ≤ n ≤ length of list.
Note 2:
Usually the version often seen in the interviews is reversing the whole
ListNode *cur=h,*pre=nullptr,*next=nullptr;
while(cur)
{
next=cur->next;
cur->next=pre;
pre=cur;
cur=next;
}
return pre;
}
ListNode* Solution::reverseBetween(ListNode* head, int left, int right)
{
if(head==nullptr || head->next==nullptr) return head;
ListNode *temp=head;
ListNode *start=nullptr,*end=nullptr;
ListNode *ans=new ListNode(-1);
ListNode *td=ans;
for(int i=1;i<left;i++)
temp=temp->next;
start=temp;
ListNode *ds=start;
for(int i=left;i<right;i++)
ds=ds->next;
if(ds->next)
{
end=ds->next;
ds->next=nullptr;
}
start=reverse(start);
temp=head;
All Problems
45/749
Custom Input
Run
Submit

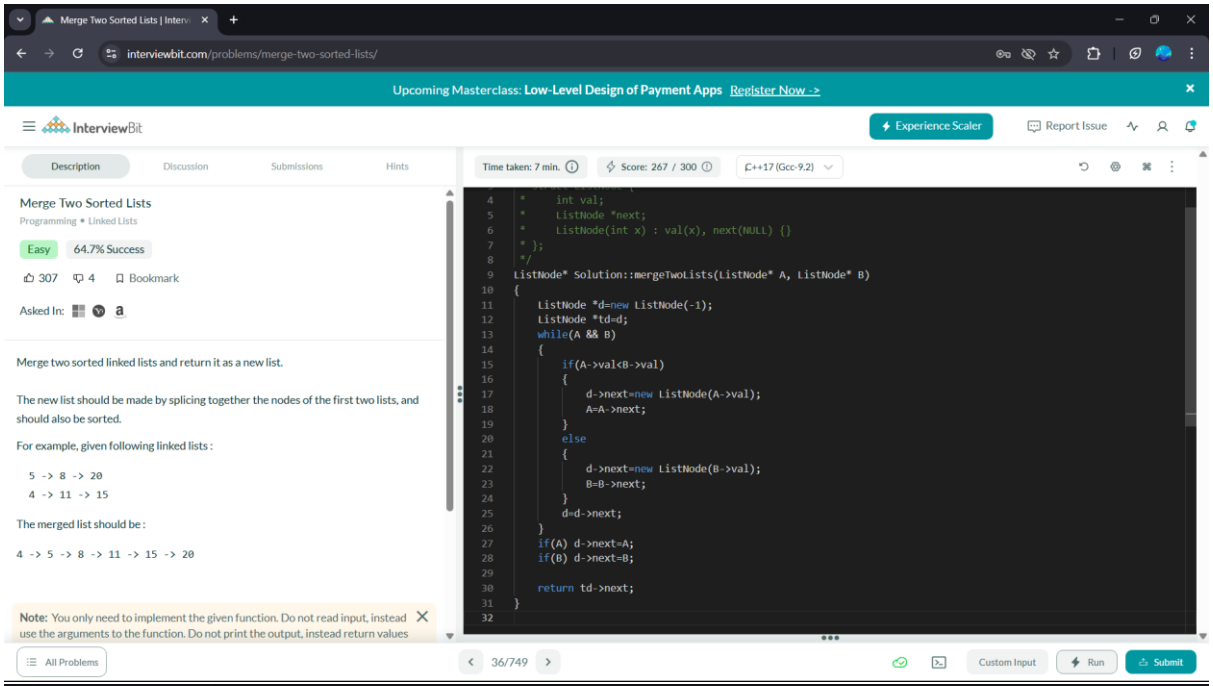

Reorder List | Interviewbit
interviewbit.com/problems/reorder-list/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 7 min.
Score: 593 / 600
C++17 (Gcc-9.2)
Reorder List
Programming • Linked Lists
Hard
56.4% Success
330
10
Bookmark
Asked In:
Given a singly linked list
L: L0 → L1 → … → Ln-1 → Ln,
reorder it to:
L0 → Ln → L1 → Ln-1 → L2 → Ln-2 → …
You must do this in-place without altering the nodes' values.
For example,
Given {1,2,3,4}, reorder it to {1,4,2,3}.
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
*/
ListNode *find_mid(ListNode *h)
{
ListNode *s=h,*f=h;
while(f->next && f->next->next)
{
s=s->next;
f=f->next->next;
}
return s;
}
ListNode* reverse(ListNode *h)
{
ListNode *pre=nullptr,*cur=h,*next=nullptr;
while(cur)
{
next=cur->next;
cur->next=pre;
pre=cur;
cur=next;
}
return pre;
}
ListNode* Solution::reorderList(ListNode* A)
{
if(A==nullptr || A->next==nullptr) return A;
All Problems
33/749
Custom Input
Run
Submit

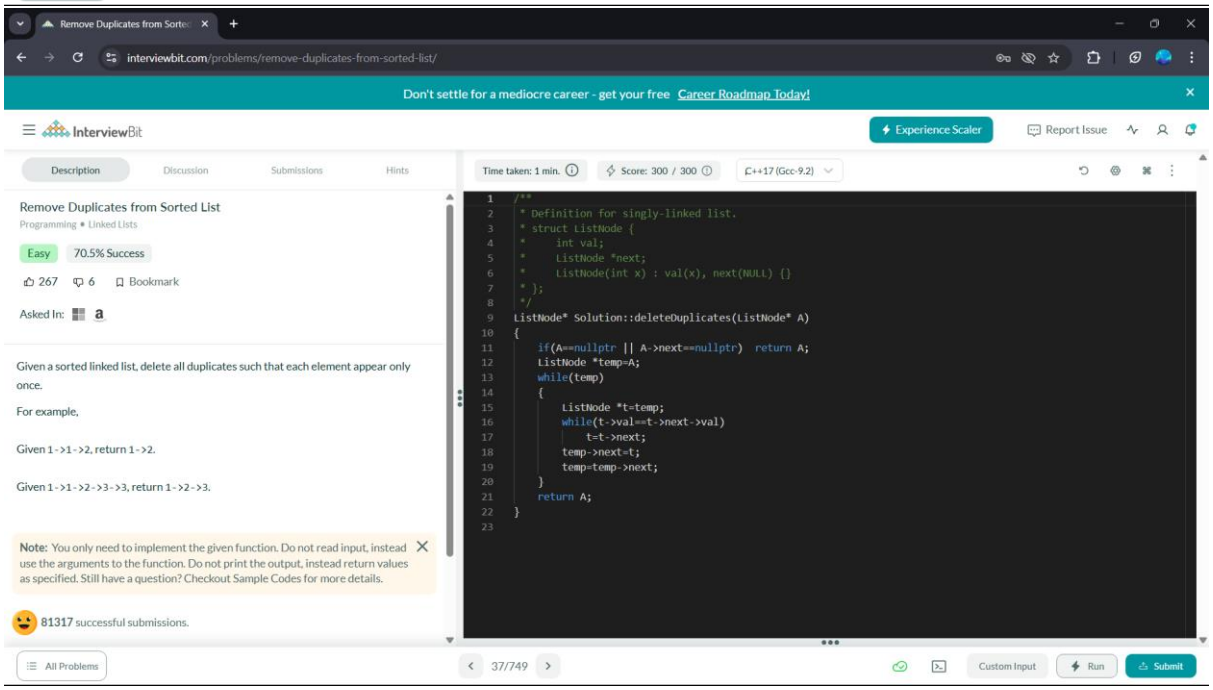

Palindrome List | Interviewbit
interviewbit.com/problems/palindrome-list/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 2 min.
Score: 200 / 200
C++17 (Gcc-9.2)
Palindrome List
Programming • Linked Lists
Medium
42.0% Success
320
10
Bookmark
Asked In:
Problem Description
Given a singly linked list A, determine if it's a palindrome. Return 1 or 0, denoting if it's a palindrome or not, respectively.
Problem Constraints
1 <= |A| <= 10^5
Input Format
The first and the only argument of input contains a pointer to the head of the
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
int Solution::lPalin(ListNode* A)
{
vector<int> a;
while(A)
{
a.push_back(A->val);
A=A->next;
}
int i,n=a.size()-1;
for(i=0;i<=n/2;i++)
{
if(a[i]!=a[n-i]) return false;
}
return true;
}
All Problems
311/749
Custom Input
Run
Submit

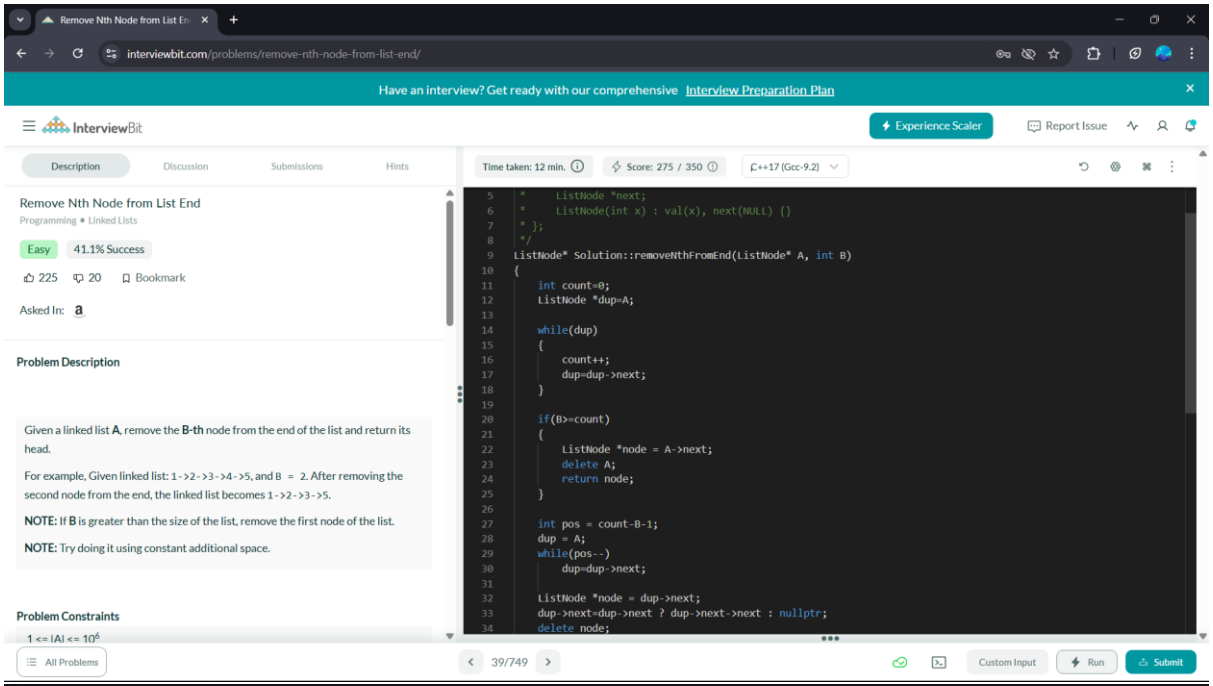

Remove Duplicates from Sorte
interviewbit.com/problems/remove-duplicates-from-sorted-list-ii/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 6 hours +
Score: 90 / 300
C++17 (Gcc-9.2)
Remove Duplicates from Sorted List II
Programming • Linked Lists
Medium
37.3% Success
309
5
Bookmark
Asked In:
Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list.
For example,
Given 1->2->3->3->4->4->5, return 1->2->5.
Given 1->1->1->2->3, return 2->3.
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
53062 successful submissions.
ListNode* Solution::deleteDuplicates(ListNode* A)
{
map<int,int> map;
ListNode *t=A;
ListNode *ans=new ListNode(-1);
ListNode *td=ans;
while(t)
{
map[t->val]++;
t=t->next;
}
for(auto [key,value]:map)
{
if(value==1)
{
td->next=new ListNode(key);
td=td->next;
}
}
return ans->next;
}
All Problems
35/749
Custom Input
Run
Submit

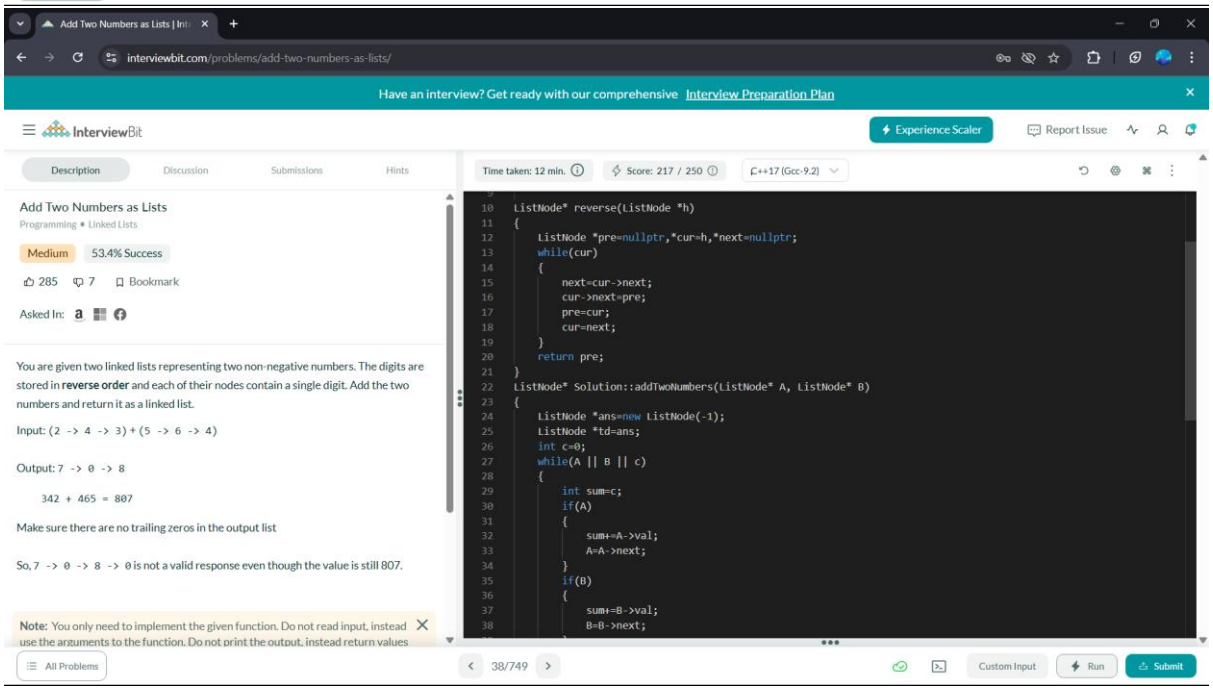

Merge Two Sorted Lists | Interv
interviewbit.com/problems/merge-two-sorted-lists/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 7 min.
Score: 267 / 300
C++17 (Gcc-9.2)
Merge Two Sorted Lists
Programming • Linked Lists
Easy
64.7% Success
307
4
Bookmark
Asked In:
Merge two sorted linked lists and return it as a new list.
The new list should be made by splicing together the nodes of the first two lists, and should also be sorted.
For example, given following linked lists :
5 -> 8 -> 20
4 -> 11 -> 15
The merged list should be :
4 -> 5 -> 8 -> 11 -> 15 -> 20
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values
* };
*/
ListNode* Solution::mergeTwoLists(ListNode* A, ListNode* B)
{
ListNode *d=new ListNode(-1);
ListNode *td=d;
while(A && B)
{
if(A->val<B->val)
{
d->next=new ListNode(A->val);
A=A->next;
}
else
{
d->next=new ListNode(B->val);
B=B->next;
}
d=d->next;
}
if(A) d->next=A;
if(B) d->next=B;
return td->next;
}
All Problems
36/749
Custom Input
Run
Submit

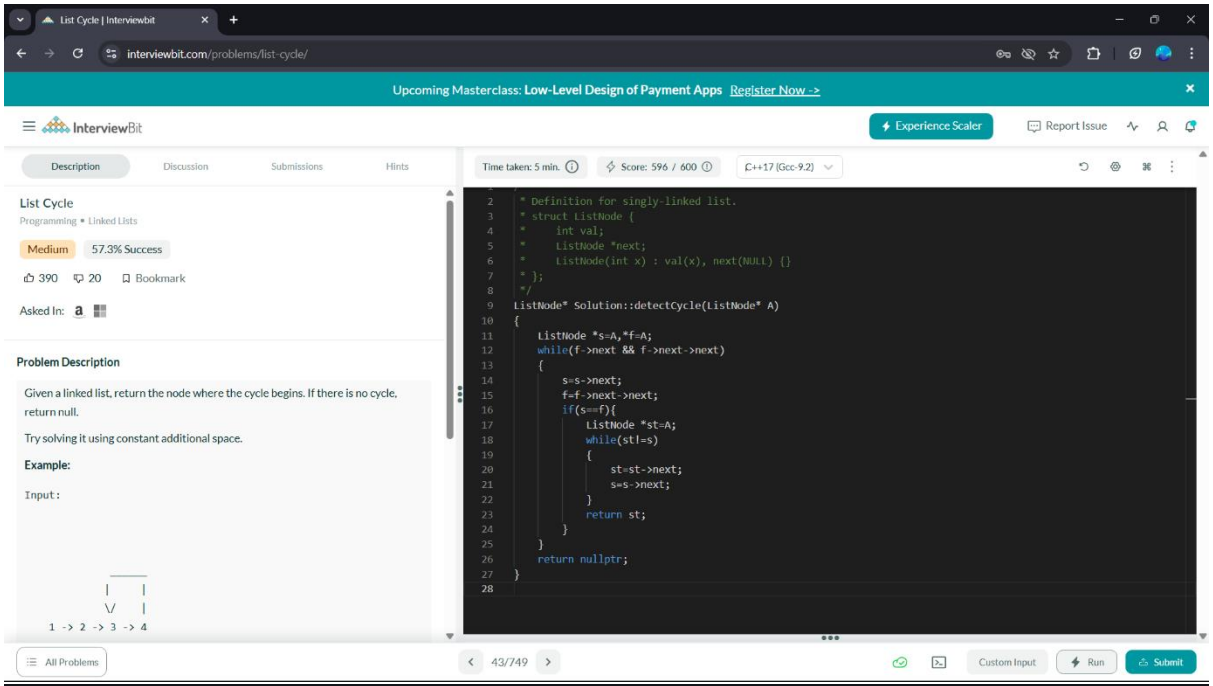

Remove Duplicates from Sorte
interviewbit.com/problems/remove-duplicates-from-sorted-list/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 1 min.
Score: 300 / 300
C++17 (Gcc-9.2)
Remove Duplicates from Sorted List
Programming • Linked Lists
Easy
70.5% Success
267
6
Bookmark
Asked In:
Given a sorted linked list, delete all duplicates such that each element appear only once.
For example,
Given 1->1->2, return 1->2.
Given 1->1->2->3->3, return 1->2->3.
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
81317 successful submissions.
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::deleteDuplicates(ListNode* A)
{
if(A==nullptr || A->next==nullptr) return A;
ListNode *temp=A;
while(temp)
{
ListNode *t=temp;
while(t->val==t->next->val)
t=t->next;
temp->next=t;
temp=temp->next;
}
return A;
}
All Problems
37/749
Custom Input
Run
Submit

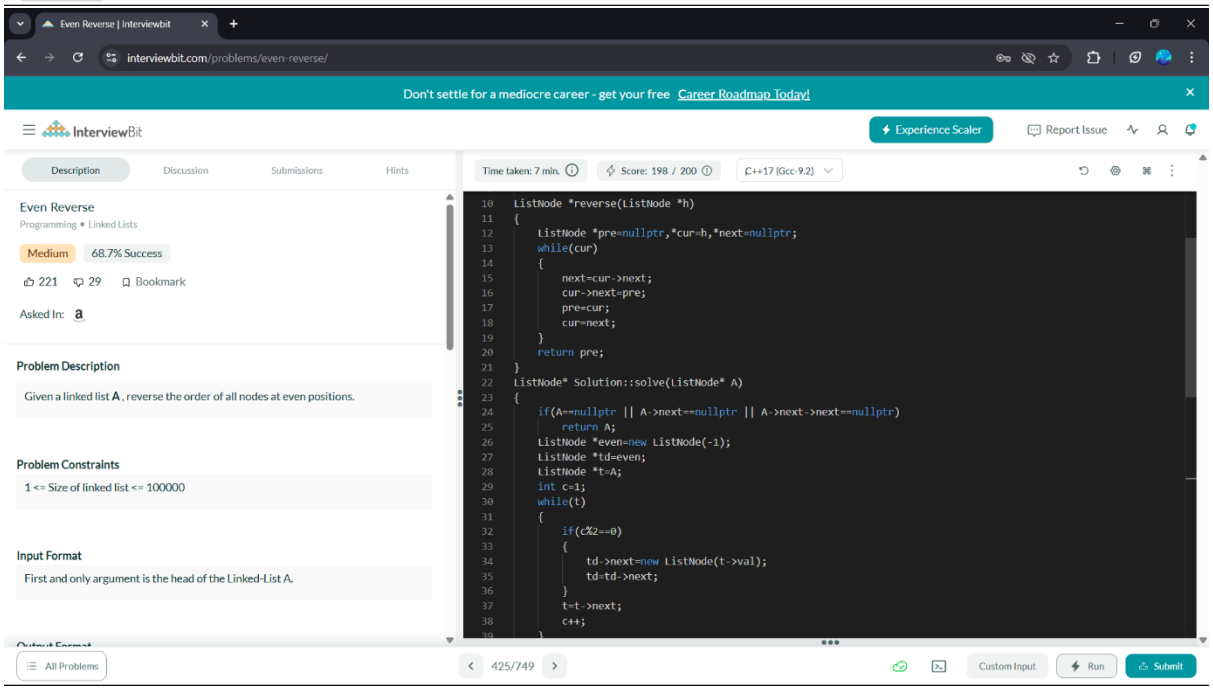

Merge Two Sorted Lists | Interv
interviewbit.com/problems/merge-two-sorted-lists/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 7 min.
Score: 267 / 300
C++17 (Gcc-9.2)
Merge Two Sorted Lists
Programming • Linked Lists
Easy
64.7% Success
307
4
Bookmark
Asked In:
Merge two sorted linked lists and return it as a new list.
The new list should be made by splicing together the nodes of the first two lists, and should also be sorted.
For example, given following linked lists :
5 -> 8 -> 20
4 -> 11 -> 15
The merged list should be :
4 -> 5 -> 8 -> 11 -> 15 -> 20
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::mergeTwoLists(ListNode* A, ListNode* B)
{
ListNode *d=new ListNode(-1);
ListNode *td=d;
while(A && B)
{
if(A->val<B->val)
{
d->next=new ListNode(A->val);
A=A->next;
}
else
{
d->next=new ListNode(B->val);
B=B->next;
}
d=d->next;
}
if(A) d->next=A;
if(B) d->next=B;
return td->next;
}
All Problems
36/749
Custom Input
Run
Submit

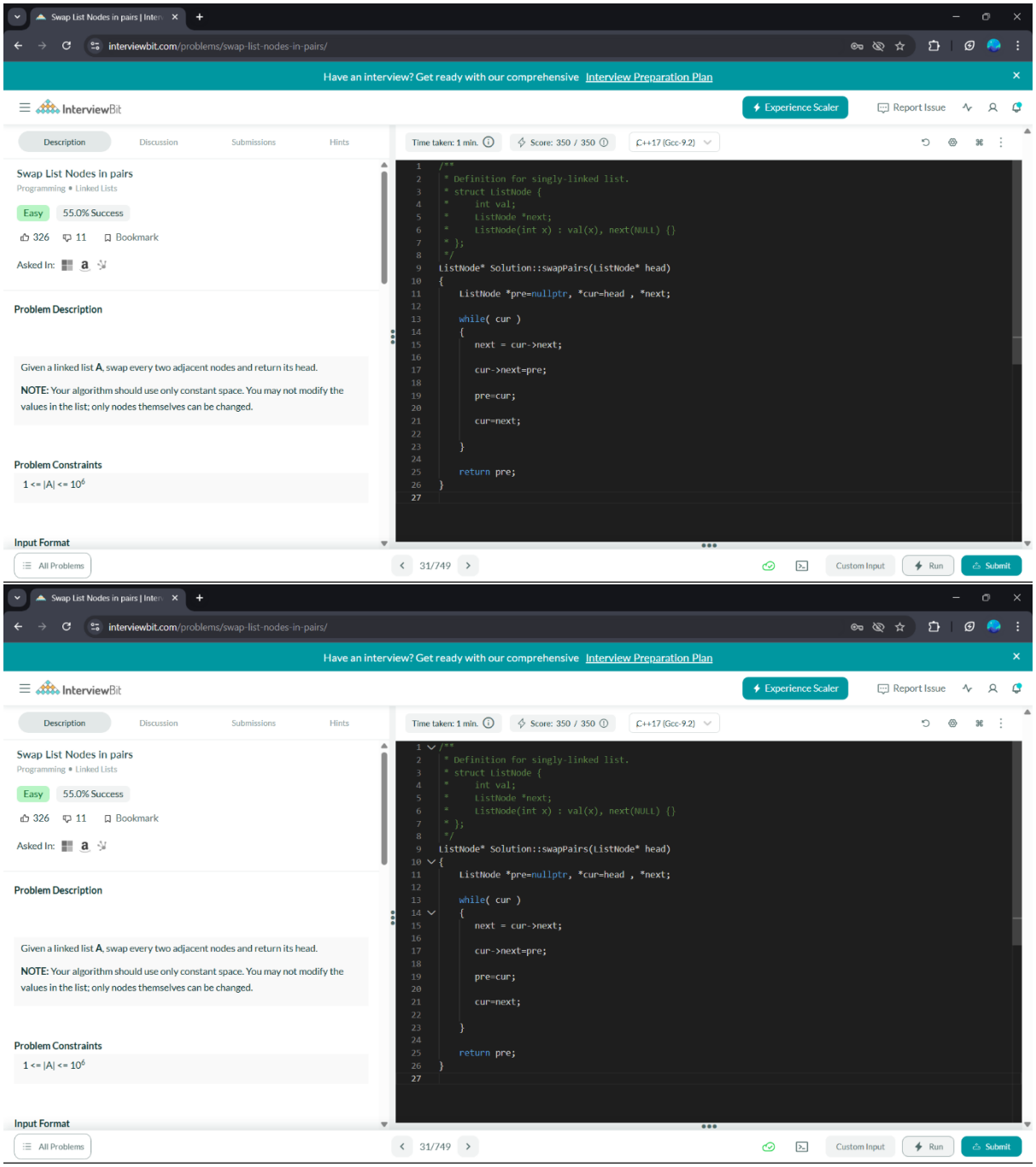

Remove Duplicates from Sorte
interviewbit.com/problems/remove-duplicates-from-sorted-list/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 1 min.
Score: 300 / 300
C++17 (Gcc-9.2)
Remove Duplicates from Sorted List
Programming • Linked Lists
Easy
70.5% Success
267
6
Bookmark
Asked In:
Given a sorted linked list, delete all duplicates such that each element appear only once.
For example,
Given 1->1->2, return 1->2.
Given 1->1->2->3->3, return 1->2->3.
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
81317 successful submissions.
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::deleteDuplicates(ListNode* A)
{
if(A==nullptr || A->next==nullptr) return A;
ListNode *temp=A;
while(temp)
{
ListNode *t=temp;
while(t->val==t->next->val)
t=t->next;
temp->next=t;
temp=temp->next;
}
return A;
}
All Problems
37/749
Custom Input
Run
Submit


Remove Nth Node from List En
interviewbit.com/problems/remove-nth-node-from-list-end/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 12 min.
Score: 275 / 350
C++17 (Gcc-9.2)
Remove Nth Node from List End
Programming • Linked Lists
Easy
41.1% Success
225
20
Bookmark
Asked In:
Problem Description
Given a linked list A, remove the B-th node from the end of the list and return its head.
For example, Given linked list: 1->2->3->4->5, and B = 2. After removing the second node from the end, the linked list becomes 1->2->3->5.
NOTE: If B is greater than the size of the list, remove the first node of the list.
NOTE: Try doing it using constant additional space.
Problem Constraints
1 <= |A| <= 10^6
ListNode* Solution::removeNthFromEnd(ListNode* A, int B)
{
int count=0;
ListNode *dup=A;
while(dup)
{
count++;
dup=dup->next;
}
if(B>=count)
{
ListNode *node = A->next;
delete A;
return node;
}
int pos = count-B-1;
dup = A;
while(pos--)
dup=dup->next;
ListNode *node = dup->next;
dup->next=dup->next ? dup->next->next : nullptr;
delete node;
All Problems
39/749
Custom Input
Run
Submit

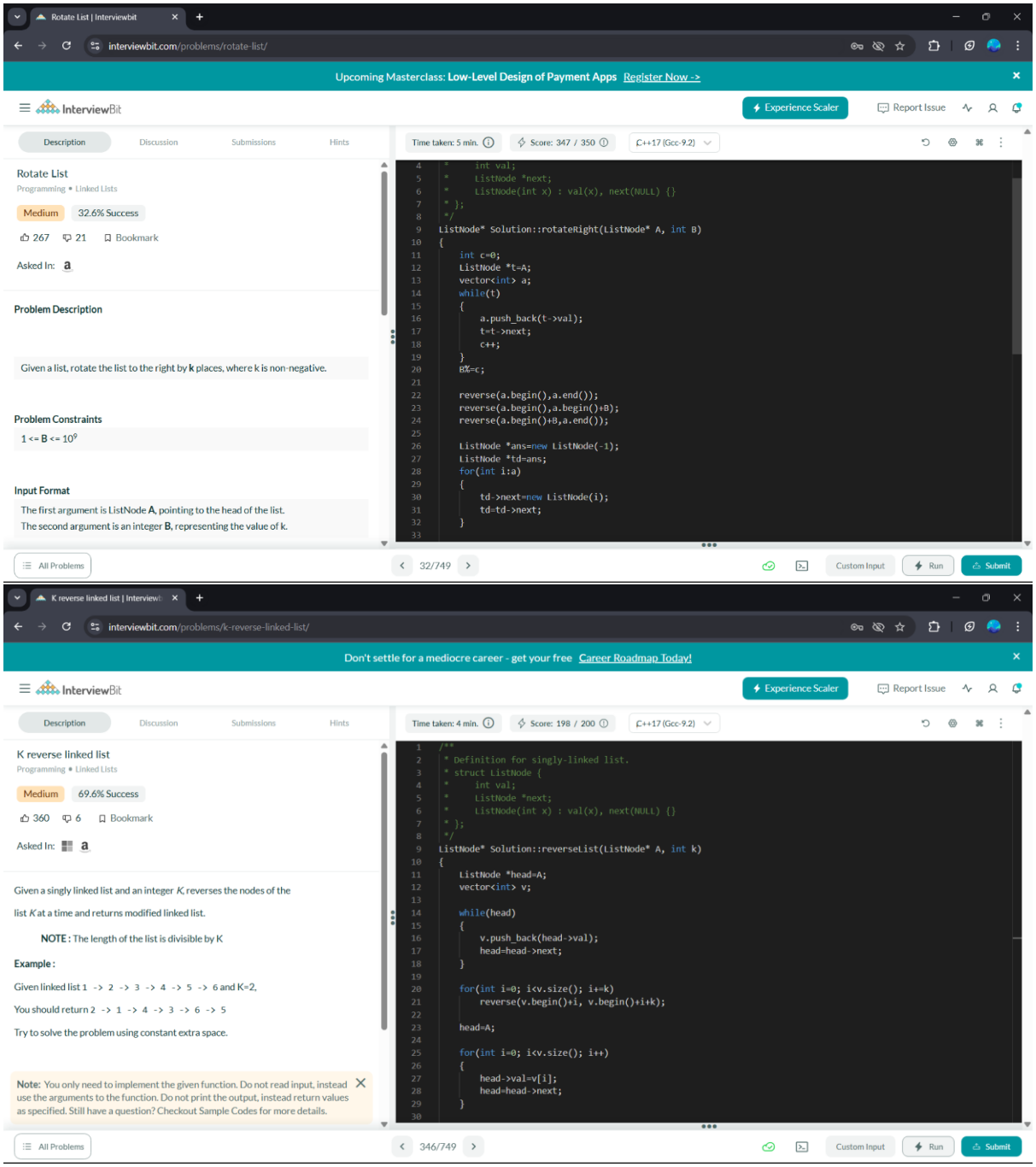

Add Two Numbers as Lists | Int
interviewbit.com/problems/add-two-numbers-as-lists/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 12 min.
Score: 217 / 250
C++17 (Gcc-9.2)
Add Two Numbers as Lists
Programming • Linked Lists
Medium
53.4% Success
285
7
Bookmark
Asked In:
You are given two linked lists representing two non-negative numbers. The digits are stored in reverse order and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.
Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 0 -> 8
342 + 465 = 807
Make sure there are no trailing zeros in the output list
So, 7 -> 0 -> 8 -> 0 is not a valid response even though the value is still 807.
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values
ListNode* reverse(ListNode *h)
{
ListNode *pre=nullptr,*cur=h,*next=nullptr;
while(cur)
{
next=cur->next;
cur->next=pre;
pre=cur;
cur=next;
}
return pre;
}
ListNode* Solution::addTwoNumbers(ListNode* A, ListNode* B)
{
ListNode *ans=new ListNode(-1);
ListNode *td=ans;
int c=0;
while(A || B || c)
{
int sum=c;
if(A)
{
sum+=A->val;
A=A->next;
}
if(B)
{
sum+=B->val;
B=B->next;
All Problems
38/749
Custom Input
Run
Submit


List Cycle | Interviewbit
interviewbit.com/problems/list-cycle/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 5 min.
Score: 596 / 600
C++17 (Gcc-9.2)
List Cycle
Programming • Linked Lists
Medium
57.3% Success
390
20
Bookmark
Asked In:
Problem Description
Given a linked list, return the node where the cycle begins. If there is no cycle, return null.
Try solving it using constant additional space.
Example:
Input :
1 -> 2 -> 3 -> 4
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
ListNode* Solution::detectCycle(ListNode* A)
{
ListNode *s=A,*f=A;
while(f->next && f->next->next)
{
s=s->next;
f=f->next->next;
if(s==f){
ListNode *st=A;
while(st!=s)
{
st=st->next;
s=s->next;
}
return st;
}
}
return nullptr;
}
All Problems
43/749
Custom Input
Run
Submit


Even Reverse | Interviewbit
interviewbit.com/problems/even-reverse/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 7 min.
Score: 198 / 200
C++17 (Gcc-9.2)
Even Reverse
Programming • Linked Lists
Medium
68.7% Success
221
29
Bookmark
Asked In:
Problem Description
Given a linked list A , reverse the order of all nodes at even positions.
Problem Constraints
1 <= Size of linked list <= 100000
Input Format
First and only argument is the head of the Linked-List A.
ListNode *reverse(ListNode *h)
{
ListNode *pre=nullptr,*cur=h,*next=nullptr;
while(cur)
{
next=cur->next;
cur->next=pre;
pre=cur;
cur=next;
}
return pre;
}
ListNode* Solution::solve(ListNode* A)
{
if(A==nullptr || A->next==nullptr || A->next->next==nullptr)
return A;
ListNode *even=new ListNode(-1);
ListNode *td=even;
ListNode *t=A;
int c=1;
while(t)
{
if(c%2==0)
{
td->next=new ListNode(t->val);
td=td->next;
}
t=t->next;
c++;
}
All Problems
425/749
Custom Input
Run
Submit


Swap List Nodes in pairs | Inter
interviewbit.com/problems/swap-list-nodes-in-pairs/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 1 min.
Score: 350 / 350
C++17 (Gcc-9.2)
Swap List Nodes in pairs
Programming • Linked Lists
Easy
55.0% Success
326
11
Bookmark
Asked In:
Problem Description
Given a linked list A, swap every two adjacent nodes and return its head.
NOTE: Your algorithm should use only constant space. You may not modify the values in the list; only nodes themselves can be changed.
Problem Constraints
1 <= |A| <= 10^6
Input Format
All Problems
31/749
Custom Input
Run
Submit

Swap List Nodes in pairs | Inter
interviewbit.com/problems/swap-list-nodes-in-pairs/
Have an interview? Get ready with our comprehensive Interview Preparation Plan
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 1 min.
Score: 350 / 350
C++17 (Gcc-9.2)
Swap List Nodes in pairs
Programming • Linked Lists
Easy
55.0% Success
326
11
Bookmark
Asked In:
Problem Description
Given a linked list A, swap every two adjacent nodes and return its head.
NOTE: Your algorithm should use only constant space. You may not modify the values in the list; only nodes themselves can be changed.
Problem Constraints
1 <= |A| <= 10^6
Input Format
All Problems
31/749
Custom Input
Run
Submit


Rotate List | Interviewbit
interviewbit.com/problems/rotate-list/
Upcoming Masterclass: Low-Level Design of Payment Apps Register Now ->
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 5 min.
Score: 347 / 350
C++17 (Gcc-9.2)
Rotate List
Programming • Linked Lists
Medium
32.6% Success
267
21
Bookmark
Asked In:
Problem Description
Given a list, rotate the list to the right by k places, where k is non-negative.
Problem Constraints
1 <= B <= 10^9
Input Format
The first argument is ListNode A, pointing to the head of the list.
The second argument is an integer B, representing the value of k.
All Problems
32/749
Custom Input
Run
Submit

K reverse linked list | Interviewbit
interviewbit.com/problems/k-reverse-linked-list/
Don't settle for a mediocre career - get your free Career Roadmap Today!
InterviewBit
Experience Scaler
Report Issue
Description
Discussion
Submissions
Hints
Time taken: 4 min.
Score: 198 / 200
C++17 (Gcc-9.2)
K reverse linked list
Programming • Linked Lists
Medium
69.6% Success
360
6
Bookmark
Asked In:
Given a singly linked list and an integer K, reverses the nodes of the
list K at a time and returns modified linked list.
NOTE : The length of the list is divisible by K
Example :
Given linked list 1 -> 2 -> 3 -> 4 -> 5 -> 6 and K=2,
You should return 2 -> 1 -> 4 -> 3 -> 6 -> 5
Try to solve the problem using constant extra space.
Note: You only need to implement the given function. Do not read input, instead use the arguments to the function. Do not print the output, instead return values as specified. Still have a question? Checkout Sample Codes for more details.
All Problems
346/749
Custom Input
Run
Submit