

Data Structure Linked list Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Data Structure Linked list will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Data Structure Linked list Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Data Structure Linked list category. To ensure quality, each submission is checked by our team, before it becomes live. This [Data Structure Linked list Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



Data Structure Linked list Interview Questions And Answers Guide.

Question - 1:

Explain the most efficient method to reverse a linked list?

Ans:

To call the function:

```
newHead = reverse(head);
struct Node *reverse(Node *curp)
{
    static struct Node *head = curp;
    static struct Node *revHead = NULL;
    if (curp == NULL)
        return NULL;
    if (curp->next == NULL)
        revHead = curp;
    else
        reverse(curp->next)->next = curp;
    if (curp == head) {
        curp->next = NULL;
        return revHead;
    }
    else
        return curp;
}
```

[View All Answers](#)

Question - 2:

Do you know how to reverse String in Java?

Ans:

This is one of my favorite question. Since String is one of the most important type in programming, you expect lot of question related to String any data structure interview. There are many ways to reverse Sting in Java or any other programming language, and interviewer will force you to solve this problem by using without API i.e. without using reverse() method of StringBuffer. In follow-up he may ask to reverse String using recursion as well. See 3 ways to reverse String in Java to learn reversing String using both loops and recursion in Java.

[View All Answers](#)

Question - 3:

Suppose In an integer array, there is 1 to 100 number, out of one is duplicate, how to find?

Ans:

This is a rather simple data structures question, especially for this kind of. In this case you can simply add all numbers stored in array, and total sum should be equal to $n(n+1)/2$. Now just subtract actual sum to expected sum, and that is your duplicate number. Of course there is a brute force way of checking each number against all other numbers, but that will result in performance of $O(n^2)$ which is not good. By the way this trick will not work if array have multiple duplicates or its not numbers forming arithmetic progression. Here is example of one way to find duplicate number in array.

[View All Answers](#)

Question - 4:

Explain how to find 3rd element from end in a linked list in one pass?

Ans:

This is another frequently asked linked list interview question. This question is exactly similar to finding middle element of linked list in single pass. If we apply same trick of maintaining two pointers and increment other pointer, when first has moved upto 3rd element, then when first pointer reaches to the end of linked list, second pointer will be pointing to the 3rd element from last in a linked list.



[View All Answers](#)

Question - 5:

Do you know how to find if linked list has loop?

Ans:

This question has bit of similarity with earlier algorithm and data structure interview question. I mean we can use two pointer approach to solve this problem. If we maintain two pointers, and we increment one pointer after processing two nodes and other after processing every node, we are likely to find a situation where both the pointers will be pointing to same node. This will only happen if linked list has loop.

[View All Answers](#)

Question - 6:

Tell me how to find middle element of linked list in one pass?

Ans:

One of the most popular question from data structures and algorithm, mostly asked on telephonic interview. Since many programmer know that, in order to find length of linked list we need to first traverse through linkedlist till we find last node, which is pointing to null, and then in second pass we can find middle element by traversing only half of length. They get confused when interviewer ask him to do same job in one pass. In order to find middle element of linked list in one pass you need to maintain two pointer, one increment at each node while other increments after two nodes at a time, by having this arrangement, when first pointer reaches end, second pointer will point to middle element of linked list. See this trick to find middle element of linked list in single pass.

[View All Answers](#)

Question - 7:

Do you know what does the following function do for a given Linked List?

Ans:

```
void fun1(struct node* head)
{
    if(head == NULL)
        return;

    fun1(head->next);
    printf("%d ", head->data);
}
```

fun1() prints the given Linked List in reverse manner. For Linked List 1->2->3->4->5, fun1() prints 5->4->3->2->1.

[View All Answers](#)

Question - 8:

By using C++ with an example describe linked list?

Ans:

To use a linked list in C++ the following structure is to be declared:

```
typedef struct List
{long Data;
List* Next;
List ()
{Next=NULL;
Data=0;
}
};
```

```
typedef List* ListPtr.
```

The following code snippet is used to add a node.

```
void SLList::AddANode()
{ListPtr->Next = new List;
ListPtr=ListPtr->Next;
}
```

The following code snippet is used to traverse the list

```
void showList(ListPtr listPtr)
{
while(listPtr!=NULL) {
cout<<listPtr->Data;
}
return temp;
}
```

[View All Answers](#)

Question - 9:

What is linked list?

Ans:

A linked list is a chain of nodes. Each and every node has at least two nodes, one is the data and other is the link to the next node. These linked lists are known as single linked lists as they have only one pointer to point to the next node. If a linked list has two nodes, one is to point to the previous node and the other is to point to the next node, then the linked list is known as doubly linked list.

[View All Answers](#)

**Question - 10:**

Tell me about circular linked list?

Ans:

An array of pointers is an array consisting of pointers. Here, each pointer points to a row of the matrix or an element. E.g char *array [] = {"a", "b"}. This is an array of pointers to characters.

[View All Answers](#)

Question - 11:

Explain reverse a linked list recursive Java solution?

Ans:

```
public void recursiveReverse(Node currentNode )
{
    //check for empty list
    if(currentNode == NULL)
        return;
    /* if we are at the TAIL node:
       recursive base case:
    */
    if(currentNode.next == NULL)
    {
        //set HEAD to current TAIL since we are reversing list
        head = currentNode;
        return; //since this is the base case
    }
    recursiveReverse(currentNode.next);
    currentNode.next.next = currentNode;
    currentNode.next = null; //set "old" next pointer to NULL
}
```

[View All Answers](#)

Question - 12:

Explain Java code for recursive solution's base case?

Ans:

```
/* if we are at the TAIL node:
*/
if(currentNode.next == NULL)
{
    //set HEAD to TAIL since we are reversing list
    head = currentNode;
    return; //since this is the base case
}
```

[View All Answers](#)

Question - 13:

Tell me what should be done in the base case for this recursive problem?

Ans:

Now, we know what the base case is and how to check for it, but what exactly should we be doing in this case? Well, we obviously want to have a return statement so that we can put a stop to the recursion. But, is there anything else that we should be doing here as well? It turns out that there is something else we need to do in the base case, because in the base case we are at the very end of the linked list. This means that because we are trying to reverse the list, we need to set the head pointer to point to the very last node.

[View All Answers](#)

Question - 14:

Explain reverse a linked list iterative solution in Java?

Ans:

```
public reverseListIteratively (Node head)
{
    if (head == NULL || head.next == NULL)
        return; //empty or just one node in list
    Node Second = head.next;
    //store third node before we change
    Node Third = Second.next;
    //Second's next pointer
    Second.next = head; //second now points to head
    head.next = NULL; //change head pointer to NULL
    //only two nodes, which we already reversed
    if (Third == NULL)
        return;
    Node CurrentNode = Third;
    Node PreviousNode = Second;
    while (CurrentNode != NULL)
    {
```



```
Node NextNode = CurrentNode.next;
CurrentNode.next = PreviousNode;
/* repeat the process, but have to reset
   the PreviousNode and CurrentNode
*/
PreviousNode = CurrentNode;
CurrentNode = NextNode;
}
head = PreviousNode; //reset the head node
}
```

[View All Answers](#)

Question - 15:

How to reverse a linked list iterative algorithm?

Ans:

- * The head node's next pointer should be set to NULL since the head will become the tail. This is an exception for the head node, and can be done outside the while loop. But, before we do this we will need a temp variable to point to the 2nd node (the node after the head node), because the only way to reference the 2nd node is through the head node's next pointer.
- * The 2nd node (the node after the head node) should have its own next pointer changed to point to the head node. This will reverse the order of the nodes. But, remember that the 2nd node's next pointer will at first be pointing to the 3rd node. This means that before we change the 2nd node's next pointer, we have to save a reference to the 3rd node otherwise we will have no way of referencing the 3rd node. So, we simply store a reference to the 3rd node in a variable before we change the 2nd node's next pointer.
- * The 3rd node then becomes the "first" node in the while loop and we repeat the process of changing pointers described in step 2.
- * Continue step 3 until we come across a node that has a next pointer set to NULL. When we do come across a NULL next pointer we just set the head node to point to the node that has the NULL next pointer. This node was previously the tail node, but is now the head node because we are reversing the linked list.

[View All Answers](#)

Question - 16:

How to reverse a singly linked list?

Ans:

First off, in case you don't already know, the word 'iterative' when used in problems like these basically means that we use some sort of loop to solve the problem - whether it's a while loop, a for loop, or whatever type of loop you desire to use. We choose to use a while loop to come up with a solution. Let's assume that we are going to start reversing the linked list starting from the very first node - the head node. What it basically comes down to is changing pointers from one node to the next so that the entire linked list becomes reversed. There is definitely a process - an algorithm - that we will want to follow in order to do that.

[View All Answers](#)

Question - 17:

Given an unsorted linked list, and without using a temporary buffer, write a method that will delete any duplicates from the linked list?

Ans:

Complexity: $O(n^2)$ If you can sort the list in $O(n \log n)$ then it will take $O(n \log n)$.

[View All Answers](#)

Question - 18:

Explain linked list using C++ with an example?

Ans:

A linked list is a chain of nodes. Each and every node has at least two nodes, one is the data and other is the link to the next node. These linked lists are known as single linked lists as they have only one pointer to point to the next node. If a linked list has two nodes, one is to point to the previous node and the other is to point to the next node, then the linked list is known as doubly linked list.

To use a linked list in C++ the following structure is to be declared:

```
typedef struct List
```

```
{long Data;
```

```
List* Next;
```

```
List ()
```

```
{Next=NULL;
```

```
Data=0;
```

```
}
```

```
};
```

```
typedef List* ListPtr.
```

The following code snippet is used to add a node.

```
void SLList::AddANode()
```

```
{ListPtr->Next = new List;
```

```
ListPtr=ListPtr->Next;
```

```
}
```

The following code snippet is used to traverse the list

```
void showList(ListPtr listPtr)
```

```
{
```

```
while(listPtr!=NULL) {
```

```
cout<<listPtr->Data;
```

```
}
```

```
return temp;
```

```
}
```

[View All Answers](#)

**Question - 19:**

Explain circular linked list?

Ans:

In a circular linked list the first and the last node are linked. This means that the last node points to the first node in the list. Circular linked list allow quick access to first and last record using a single pointer.

[View All Answers](#)

Question - 20:

What is circular linked list?

Ans:

An array of pointers is an array consisting of pointers. Here, each pointer points to a row of the matrix or an element. E.g char *array [] = {"a", "b"}. This is an array of pointers to characters.

[View All Answers](#)

Question - 21:

How to reverse singly link list?

Ans:

Reverse a singly link list using iteration:-

1. First, set a pointer (*current) to point to the first node i.e. current=base
 2. Move ahead until current!=null (till the end)
 3. set another pointer (* next) to point to the next node i.e. next=current->next
 4. store reference of *next in a temporary variable (*result) i.e current->next=result
 5. swap the result value with current i.e. result=current
 6. and now swap the current value with next. i.e. current=next
 7. return result and repeat from step 2
- A linked list can also be reversed using recursion which eliminates the use of a temporary variable.

[View All Answers](#)

Question - 22:

Explain the steps to insert data into a singly linked list?

Ans:

Steps to insert data into a singly linked list:-

Insert in middle

Data: 1, 2, 3, 5

1. Locate the node after which a new node (say 4) needs to be inserted.(say after 3)
2. create the new node i.e. 4
3. Point the new node 4 to 5 (new_node->next = node->next(5))
4. Point the node 3 to node 4 (node->next =newnode)

Insert in the beginning

Data: 2, 3, 5

1. Locate the first node in the list (2)
2. Point the new node (say 1) to the first node. (new_node->next=first) Insert in the end

Insert in the end

Data: 2, 3, 5

1. Locate the last node in the list (5)
2. Point the last node (5) to the new node (say 6). (node->next=new_node)

[View All Answers](#)

Data Structure Most Popular & Related Interview Guides

- 1 : [Data Center Manager Interview Questions and Answers.](#)
- 2 : [Creative UI/UX Designers Interview Questions and Answers.](#)
- 3 : [Sort And Searching Interview Questions and Answers.](#)
- 4 : [Data Structure Arrays Interview Questions and Answers.](#)
- 5 : [Data Architect Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

<https://InterviewQuestionsAnswers.ORG/>
support@InterviewQuestionsAnswers.ORG