

1. What will be the output of the following code?

```
#include <bits/stdc++.h>
using namespace std;
void fun(int *p){
    *p = 20;
}
int main(){
    int val = 10;
    int *ptr = &val;
    fun(ptr);
    cout << *ptr << endl;
    return 0;
}
```

- a. 10
- b. 20**
- c. Segmentation Fault
- d. Compiler error

Explanation: কোডে, fun ফাংশনে একটি পয়েন্টার পাস করা হয়েছে। পয়েন্টারটি val এর এড্রেস নির্দেশ করছে।

*p = 20; এই লাইনটি val এর ভ্যালু পরিবর্তন করে 20 করে।

cout << *ptr প্রিন্ট করার সময় ptr পয়েন্ট করছে val এর ঠিকানায়। তাই আউটপুট হবে 20।

2. What will be the output of the following code?

```
#include <bits/stdc++.h>
using namespace std;
void fun(int *p){
    p = NULL;
}
int main(){
    int val = 10;
    int *ptr = &val;
    fun(ptr);
    cout << *ptr << endl;
    return 0;
}
```

```
}
```

a. 10

b. 20

c. Segmentation Fault

d. Compiler error

Explanation: কোডে, fun ফাংশনের মধ্যে p = NULL করলে শুধু local pointer(fun ফাংশনের মধ্যে) p NULL হয়ে যায়। ptr এর সাথে এর কোনো সম্পর্ক নেই। তাই, ptr এখনো val এর ঠিকানায় পয়েন্ট করছে, যার ভ্যালু 10।

3. What will be the output of the following code?

```
#include <bits/stdc++.h>
using namespace std;
void fun(int *&p){
    p = NULL;
}
int main() {
    int val = 10;
    int *ptr = &val;
    fun(ptr);
    cout << ptr << endl;
    return 0;
}
```

a. 10

b. 20

c. 0

d. Compiler error

Explanation: সি++-এ NULL মেমরি ঠিকানা হিসেবে বোঝানো হয়। যখন এটি cout দিয়ে প্রিন্ট করা হয়, তখন 0 আউটপুট হয়। int*& ব্যবহার করার ফলে ফাংশনের ভেতরে করা পরিবর্তন সরাসরি মূল পয়েন্টারে প্রতিফলিত হয়।

4. What will the following code snippet do?

```
Node *tmp = head;
```

```
while (tmp->next != NULL)
{
    tmp = tmp->next;
}
tmp->next = newNode;
```

- a. Insert a node at head
- b. Insert a node at any position
- c. Delete the head node
- d. Insert a node only at tail**

Explanation: লুপটি লিংকড লিস্টের শেষ পর্যন্ত চলে এবং tmp কে last node এ সেট করে। এরপর tmp->next = newNode করলে নতুন নোডটি লিস্টের টেইলে যোগ হয়।

5. What will the following code snippet do?

```
newNode->next = head;
head = newNode;
```

- a. Insert a node at head**
- b. Insert a node at any position
- c. Delete the head node
- d. Insert a node only at tail

Explanation: কোডটি একটি নতুন নোডকে লিস্টের হেড হিসাবে সেট করে। প্রথমে newNode->next = head, বর্তমান হেড নোডকে নতুন নোডের পরবর্তী হিসেবে সেট করে। তারপর head = newNode, নতুন নোডকে হেড হিসাবে আপডেট করে।

6. What will the following code snippet do?

```
Node *tmp = head;
for (int i = 1; i <= pos - 1; i++)
{
    tmp = tmp->next;
}
newNode->next = tmp->next;
tmp->next = newNode;
```

- a. Insert a node at head

b. Insert a node at any position

- c. Delete a node from any position
- d. Insert a node only at tail

Explanation: কোডটি নির্দিষ্ট অবস্থানে একটি নোড যোগ করার জন্য ব্যবহৃত হয়।

লুপটি tmp কে পজিশনের আগের নোডে নিয়ে যায়।

এরপর newNode->next = tmp->next এবং tmp->next = newNode ব্যবহার করে নতুন নোড যোগ করা হয়।

7. How can we print a singly linked list in reverse order?

- a. By traversing the list twice
- b. By traversing the list once
- c. Using a loop

d. Using recursion

Explanation: In a singly linked list, প্রতিটি নোডের শুধুমাত্র পরবর্তী নোডের ঠিকানা থাকে। তাই, লিস্টটি reverse order এ প্রিন্ট করতে হলে, রিকার্সিভ পদ্ধতি সবচেয়ে কার্যকর।

প্রথমে রিকার্সিভ ফাংশনের মাধ্যমে শেষ নোডে পৌঁছানো হয়। এরপর, রিকার্সিভ কল থেকে ফিরে আসার সময় একে একে নোড প্রিন্ট করা হয়।

8. How can you find the length of a singly linked list?

- a. Use a recursive function to traverse the list and count the nodes
- b. Traverse the list using a loop and count the nodes

c. Both a and b

- d. It's not possible to find the length of a singly linked list

Explanation: আমরা singly linkedlist এ দুই ভাবেই লেন্থ কাউন্ট করতে পারি, রিকার্সিভ ভাবে কল করার সময় এবং লুপ চালিয়ে ট্রাভার্স করার সময়।

9. What is the time complexity of traversing a singly linkedlist?

- a. $O(1)$
- b. $O(N)$**
- c. $O(N*N)$
- d. $O(\log N)$

Explanation: লুপ চালিয়ে লিস্টের প্রতিটি নোড একবার করে ভিজিট করতে হয়। যদি N নোড থাকে, তাহলে ট্রাভার্সের জন্য N বার অপারেশন লাগে।

10. What is the time complexity of inserting an element at head in a singly linkedlist?

- a. $O(N)$
- b. $O(1)$**

- c. $O(N*N)$
- d. $O(\log N)$

Explanation: হেডে নতুন ভ্যালু যোগ করতে কোনো লুপ চালাতে হয় না। নতুন নোডের next হেডে সেট করা এবং নতুন নোডকে হেডে সেট করা। এটি কনস্ট্যান্ট টাইমে সম্পন্ন হয়।