

4.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Node {
```

```
    int data;
```

```
    struct Node *next;
```

```
};
```

```
int main() {
```

```
    struct Node *head, *first, *second;
```

```
    head = (struct Node *) malloc (size of (struct Node));
```

```
    first = (struct Node *) malloc (size of (struct Node));
```

```
    second = (struct Node *) malloc (size of (struct Node));
```

```
    head → data = 100;
```

```
    head → next = first;
```

```
    first → next = second;
```

```
    second → data = 300;
```

```
    second → next = NULL;
```

```
    struct Node *temp = head;
```

```
    printf ("Linked List = ");
```

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

```
        printf ("%d " → , temp → data);
```

```
        temp = temp → next;
```

```
    }
```

```
    printf ("NULL");
```

```
    return 0;
```

```
}
```

OUTPUT :- Linked list : 100 → 200 → 300 → NULL

```
main.c
10 struct Node *head, *first, *second;
11
12 head = (struct Node*)malloc(sizeof(struct Node));
13 first = (struct Node*)malloc(sizeof(struct Node));
14 second = (struct Node*)malloc(sizeof(struct Node));
15
16 head->data = 100;
17 head->next = first;
18
19 first->data = 200;
20 first->next = second;
21
22 second->data = 300;
23 second->next = NULL;
24
25 struct Node* temp = head;
26
27 printf("Linked List:");
28 while (temp != NULL) {
29     printf("%d->", temp->data);
30     temp = temp->next;
31 }
32 printf("NULL");
33
34 return 0;
35 }
```

Run

Output

Clear

```
* Linked List:100->200->300->NULL
=== Code Execution Successful ===
```

Programiz PRO

Premium
Courses by
Programiz

Learn More

