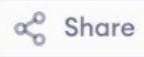


main.c



Share

Run

Output

Clear

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct Node {
4      int data ;
5      struct Node *next;
6  };
7
8
9  int main (){
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
17     head->data=106;
18     head->next=first;
19
20     first->data=223;
21     first->next=second;
22
23     second->data=300;
```

98  
106  
223  
300

=== Code Execution Successful ===

A diagram of a linked list. It consists of four nodes connected by arrows. The first node contains the value 22, the second contains 4, the third contains 43, and the fourth contains the text 'none'. The node with '4' is highlighted with a green border.

main.c



Share

Run

Output

Clear

```
19
20     first->data=223;
21     first->next=second;
22
23     second->data=300;
24     second->next=NULL;
25
26     struct Node *newNode=(struct Node*)malloc(sizeof(struct Node));
27     newNode->data=98;
28     newNode->next=head;
29     head=newNode;
30
31
32     struct Node*temp=head;
33     while(temp!=NULL){
34         printf("%d\n", temp-> data);
35         temp = temp->next;
36     }
37     free (second);
38     free (first);
39     free (head);
40     return 0;
41 }
```

98

106

223

300

=== Code Execution Successful ===

```
#include <stdio.h>
#include <stdlib.h>
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 -> data = 200;
    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;
}
```