

A screenshot of a C programming environment, likely an IDE or terminal window. The title bar reads "Programming with C". The left sidebar contains icons for file operations like New, Open, Save, Find, Replace, and Help. The main area shows a C program named "dsa4.c".

```
File Edit Selection View Go Run Terminal Help ← → ⌂ Programming with C ⌂ X
```

```
C dsa4.c X
```

```
C dsa4.c > main()
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node
5 {
6     int data;
7     struct Node* next;
8 };
9
10 int main()
11 {
12     struct Node *head, *first, *second;
13
14     head = (struct Node*)malloc(sizeof(struct Node));
15     first = (struct Node*)malloc(sizeof(struct Node));
16     second = (struct Node*)malloc(sizeof(struct Node));
17
18     head->data = 10;
19     first->data = 64;
20     second->data = 97;
21
22     head->next = first;
23     first->next = second;
24     second->next = NULL;
25
26     struct Node *third = (struct Node*)malloc(sizeof(struct Node));
27     third->data = 72;
28     third->next = NULL;
29     second->next = third;
30
31     struct Node* temp = head;
32     while (temp != NULL)
33     {
34         printf("%d->", temp->data);
35         temp = temp->next;
36     }
37     printf("NULL\n");
38
39     free(head);
40     free(first);
41     free(second);
42     free(third);
43
44     return 0;
45 }
```

The code defines a singly linked list structure where each node contains an integer data field and a pointer to the next node. The `main` function creates four nodes (head, first, second, third) and links them together. It then prints the data of each node followed by a separator and the address of the next node until it reaches the end of the list (NULL). Finally, it frees all the memory allocated for the nodes.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell + ⌂ ⌂ ⌂ ⌂ | ×

```
PS C:\Programming with C> gcc dsa4.c
PS C:\Programming with C> ./a.exe
10->64->97->72->NULL
PS C:\Programming with C> █
```