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

// Define the structure for a doubly linked list node
typedef struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
} Node;

// Function to create a new node with given data
Node* createNode(int value) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = value;



    newNode->prev = NULL;
    newNode->next = NULL;
    return newNode;
}

// Function to display the doubly linked list with pointer addresses
void displayList(Node* head) {
    Node* temp = head;
    printf("\nDoubly Linked List Structure:\n");
    printf("-----------------------------------------------------------------\n");
    printf("| %-10s | %-10s | %-5s | %-10s |\n", "Node Addr", "Prev Addr", "Data", "Next Addr");
    printf("-----------------------------------------------------------------\n");

    while (temp != NULL) {
        printf("| %-10p | %-10p | %-5d | %-10p |\n",
               (void*)temp,
               (void*)temp->prev,
               temp->data,
               (void*)temp->next);
        temp = temp->next;
    }

    printf("-----------------------------------------------------------------\n");
}

int main() {
    // Step 1: Create 5 independent nodes
```

```
    Node* N1 = createNode(1);
    Node* N2 = createNode(2);
    Node* N3 = createNode(3);
    Node* N4 = createNode(4);
    Node* N5 = createNode(5);

    // Step 2: Link nodes to form the doubly linked list
    N1->next = N2;

    N2->prev = N1;
    N2->next = N3;

    N3->prev = N2;
    N3->next = N4;

    N4->prev = N3;
    N4->next = N5;

    N5->prev = N4;

    // Step 3: Display the list with pointer addresses
    displayList(N1);

    // Cleanup: Free allocated memory
    free(N1);



    free(N2);
    free(N3);
    free(N4);
    free(N5);

    return 0;
}
```

OUTPUT

```
Doubly Linked List Structure:
------------------------------------------------------------
| Node Addr  | Prev Addr  | Data  | Next Addr  |
------------------------------------------------------------
| 0x38e482a0 | (nil)      | 1     | 0x38e482c0 |
| 0x38e482c0 | 0x38e482a0 | 2     | 0x38e482e0 |
| 0x38e482e0 | 0x38e482c0 | 3     | 0x38e48300 |
| 0x38e48300 | 0x38e482e0 | 4     | 0x38e48320 |
| 0x38e48320 | 0x38e48300 | 5     | (nil)      |
------------------------------------------------------------


=== Code Execution Successful ===
```