Doubly Linked List 05-02-2024

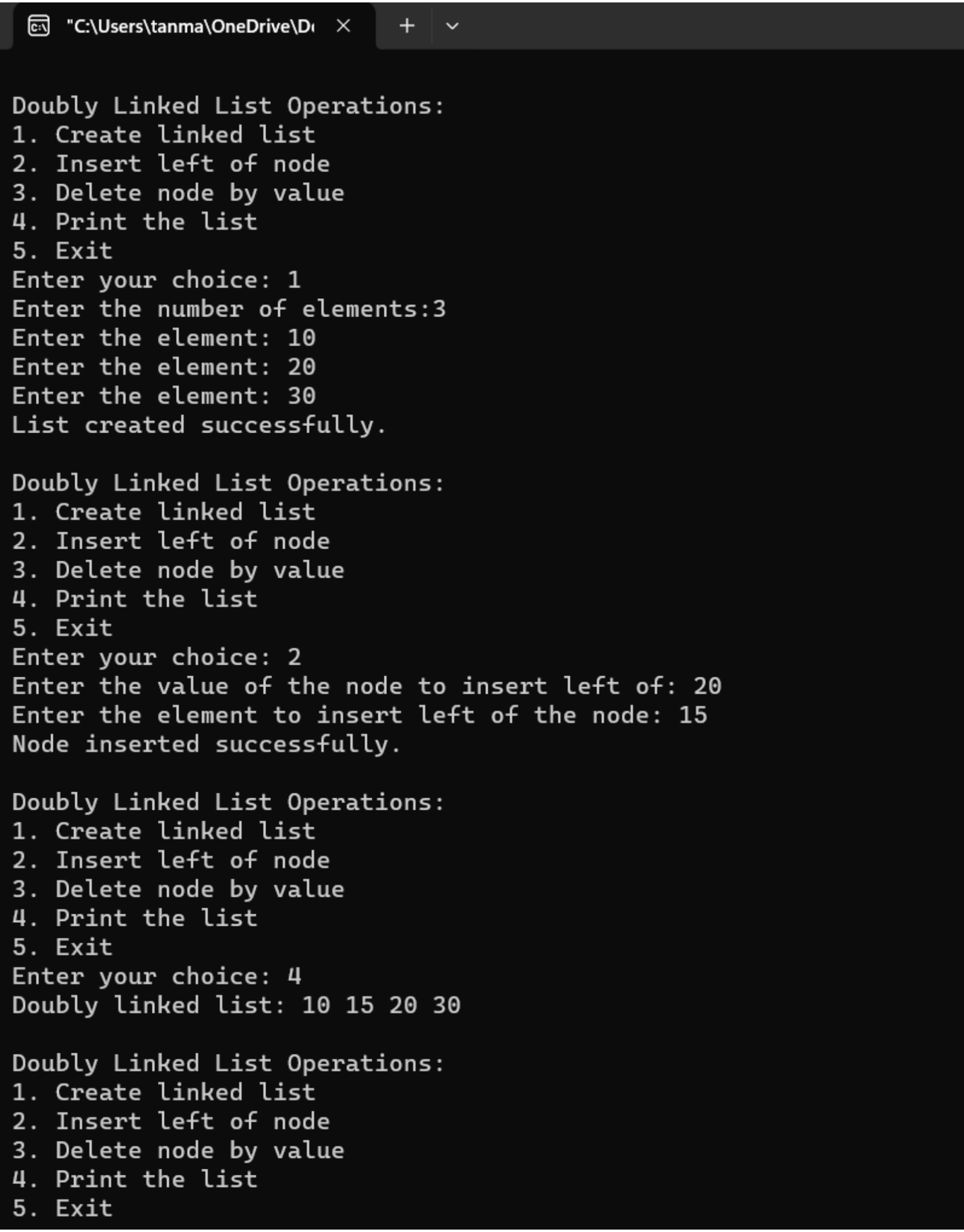
```
#include <stdio.h>
#include <stdlib.h>
    struct node {
    int data;
    struct node *prev;
    struct node * next; };

    Struct node *head = NULL;
    void createlist()
    { int i, n;
     struct node *newnode;
     Struct node * temp;
     printf (" Enter the number of elements :");
     Scanf ("%d", &n);
     for(i=0; i<n; i++) {
        newnode = (struct node *) malloc(sizeof (struct node));
        printf (" Enter the element: ");
        Scanf (" %d", &newnode -> data);
        if (head == NULL){
            head = temp = newnode;
            head -> prev = NULL;
            temp -> next = NULL; }
      else{ temp -> next = newnode;
            newnode -> prev = temp;
            temp = newnode;
            temp -> next = NULL; }
      } printf (" List created Successfully \n");
}
void Insertleft ( struct node * temp, int data)
 { struct node* newnode;
   if (temp == NULL)
     { printf (" Target node not exist \n");
       return; }
   newnode = (Struct node*) malloc(sizeof (struct node));
   newnode -> data = data;
   new node -> next = temp;
   new node -> prev = temp -> prev;
```

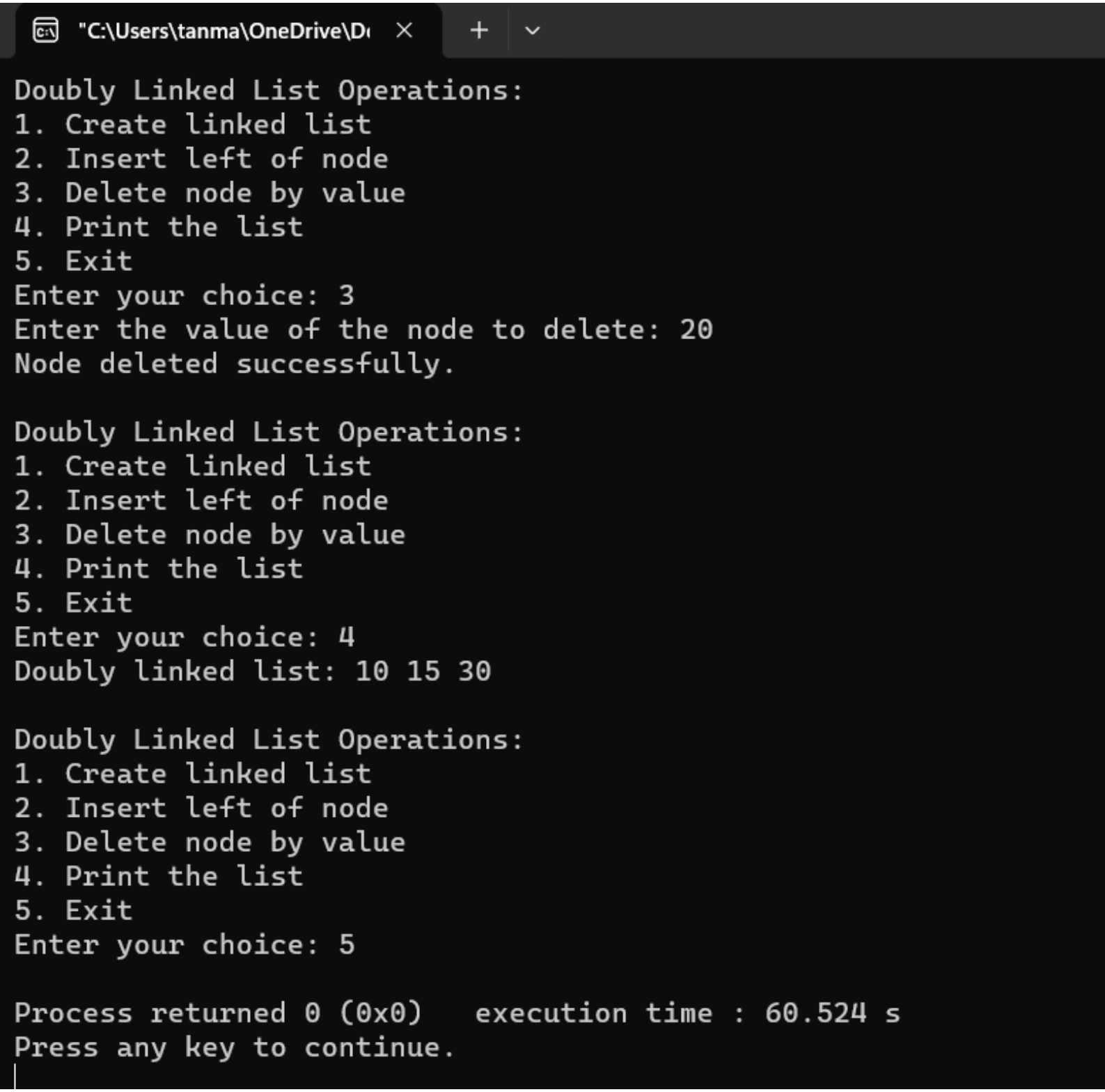
```
if (temp -> prev != NULL)
{ temp -> prev -> next = newnode; }
temp -> prev = newnode;
if (head == temp)
  { head = newnode; }
printf (" Node inserted successfully \n"); }

void deletenode (int key) {
  struct node * current = head;
  while (current != NULL) {
    if (current -> data == key) {
        if (current -> prev != NULL) {
   current -> prev -> next = current -> next; }
  if (current -> next != NULL) {
    current -> next -> prev = current -> prev; }
  if (current == head)
    { head = current -> next; }
   free (current);
    printf (" Node deleted successfully \n");
    return; }
  current = current -> next; }
   printf (" Node with value %d not found! \n", key); }

void printlist() {
    struct node * temp = head;
    if (temp != NULL) {
    printf (" %d", temp -> data);
    temp = temp -> next; }
  printf (" \n"); }

int main() {
   int choice, data, targetvalue, deletevalue;
  while(1) { printf (" Doubly Linked List Operations: \n");
   printf (" 1. Create \n 2. Insert Left of node \n"
           3. Delete node by value \n", 4. Print \n,
           5. Exit \n");
  printf (" Enter your choice : ");
  scanf (" %d", &choice);
```

```
switch (choice)
{ case 1: createlist();
          break;
  case 2: printf ("Enter value of node inserted to left");
          scanf ("%d", &target value);
          printf ("Enter the element to insert);
          scant ("%d", &data);
          struct node * temp = head;
          while (temp != NULL) {
            if (temp -> data == target value)
          {  insertleft (temp, data);
             break; }
            temp = temp -> next; }
          break;
Case 3 : printf ("Enter the value of node to delete");
          scant ("%d", &delete value);
          deletenode (delete value);
          break;
Case 4 : printlist();
          break;
case 5 : exit(0);
          break;
default: printf ("Invalid choice \n"); }
{   return 0;
  }
```

OUTPUT:

```
Doubly Linked List Operations.
1. Create
2. Insert left of Node
3. Delete node by value
4. Print
5. Exit.
Enter your choice: 1
Enter number of elements: 3
Enter the element : 2
Enter the element : 4
Enter the element: 6
List created successfully.
```


"C:\Users\tanma\OneDrive\D
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit
Enter your choice: 1
Enter the number of elements:3
Enter the element: 10
Enter the element: 20
Enter the element: 30
List created successfully.
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit
Enter your choice: 2
Enter the value of the node to insert left of: 20
Enter the element to insert left of the node: 15
Node inserted successfully.
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit
Enter your choice: 4
Doubly linked list: 10 15 20 30
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit


"C:\Users\tanma\OneDrive\D
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit
Enter your choice: 3
Enter the value of the node to delete: 20
Node deleted successfully.
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit
Enter your choice: 4
Doubly linked list: 10 15 30
Doubly Linked List Operations:
1. Create linked list
2. Insert left of node
3. Delete node by value
4. Print the list
5. Exit
Enter your choice: 5
Process returned 0 (0x0) execution time : 60.524 s
Press any key to continue.