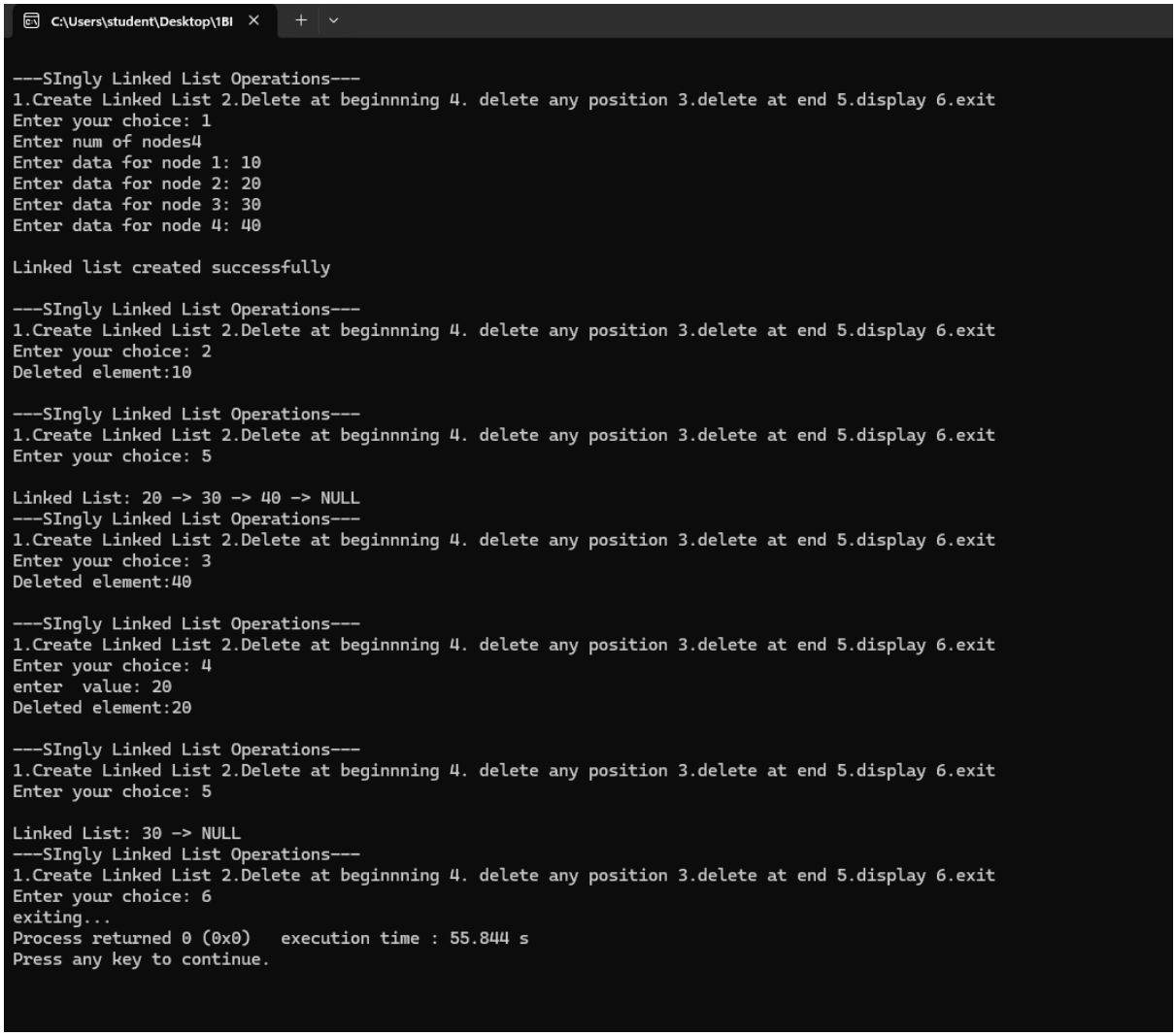
C:\Users\student\Desktop\1BI
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 1
Enter num of nodes4
Enter data for node 1: 10
Enter data for node 2: 20
Enter data for node 3: 30
Enter data for node 4: 40
Linked list created successfully
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 2
Deleted element:10
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 5
Linked List: 20 -> 30 -> 40 -> NULL
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 3
Deleted element:40
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 4
enter value: 20
Deleted element:20
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 5
Linked List: 30 -> NULL
---SIngly Linked List Operations---
1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit
Enter your choice: 6
exiting...
Process returned 0 (0x0) execution time : 55.844 s
Press any key to continue.

<global> deleteSpecific(int value) : void

art here | linkedlistinsertion.c | lab4b.c

```c
#include<stdio.h>
#include<stdlib.h>
struct Node{
int data;
struct Node *next;
};
struct Node *head=NULL;
void createList(int n){
struct Node *newNode,*temp;
int data,i;
if(n<=0){printf("Number of nodes should be greater than 0\n");
return ;}
for(i=1;i<=n;i++)
{
    newNode=(struct Node *)malloc(sizeof(struct Node));
    if(newNode==NULL){
        printf("Memory allocation failed\n");
        return;
    }
    printf("Enter data for node %d: ",i);
    scanf("%d",&data);
    newNode ->data=data;
    newNode->next=NULL;
    if(head==NULL){
        head=newNode;
    }
    else{
        temp->next=newNode;
    }
    temp=newNode;
    }
    printf("\nLinked list created successfully\n");
}
void deleteFirst(){
struct Node *temp;
if(head==NULL){
```

Start here | linkedlistinsertion.c | lab4b.c

```c
    head=head->next;
    printf("Deleted element:%d\n",temp->data);
    free(temp);
    return;
}
while(temp!=NULL && temp->data!=value){
    prev=temp;
    temp=temp->next;
    printf("Deleted element:%d\n,temp->data");
    free(temp);
}
}
void displayList(){
struct Node *temp=head;
if(head==NULL){
    printf("List is empty\n");
}
printf("\nLinked List: ");
while (temp!=NULL){
    printf("%d -> ",temp->data);
    temp=temp->next;
}
printf("NULL");
}
int main(){
int choice,n,value;
while(1){
    printf("\n---SIngly Linked List Operations---\n");
    printf("1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit\n");

    printf("Enter your choice: ");
    scanf("%d",&choice);
    switch(choice){
    case 1:printf("Enter num of nodes");
           scanf("%d" ,&n);
           createList(n);
```

linkedlistinsertion.c | lab4b.c

```c
void deleteFirst(){
struct Node *temp;
if(head==NULL){
    printf("List is empty.Nothing to delete.\n");
    return;
}
temp=head;
head=head->next;
printf("Deleted element:%d\n",temp->data);
free(temp);
}
void deleteLast(){
struct Node *temp,*prev;
if(head==NULL){
    printf("List is empty .Nothing to delete\n");
    return;
}
temp=head;
while(temp->next!=NULL){
    prev=temp;
    temp=temp->next;
}
printf("Deleted element:%d\n",temp->data);
prev->next=NULL;
free(temp);
}
void deleteSpecific(int value){
struct Node *temp=head,*prev=NULL;
if(head==NULL){
    printf("List is empty .Nothing to delete\n");
    return;
}
if(head->data==value){
    head=head->next;
    printf("Deleted element:%d\n",temp->data);
    free(temp);
```

Start here | linkedlistinsertion.c | lab4b.c

```
67      head=head->next;
68      printf("Deleted element:%d\n",temp->data);
69      free(temp);
70      return;
71  }
72  while(temp!=NULL && temp->data!=value){
73      prev=temp;
74      temp=temp->next;
75      printf("Deleted element:%d\n,temp->data");
76      free(temp);
77  }
78  }
79  void displayList(){
80  struct Node *temp=head;
81  if(head==NULL){
82      printf("List is empty\n");
83  }
84  printf("\nLinked List: ");
85  while (temp!=NULL){
86      printf("%d -> ",temp->data);
87      temp=temp->next;
88  }
89  printf("NULL");
90  }
91  int main(){
92  int choice,n,value;
93  while(1){
94      printf("\n---SIngly Linked List Operations---\n");
95      printf("1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.display 6.exit\n");
96
97      printf("Enter your choice: ");
98      scanf("%d",&choice);
99      switch(choice){
100     case 1:printf("Enter num of nodes");
101           scanf("%d" ,&n);
102           createList(n);
```

ere | linkedlistinsertion.c | lab4b.c

```
92  int choice,n,value;
93  while(1){
94      printf("\n---SIngly Linked List Operations---\n");
95      printf("1.Create Linked List 2.Delete at beginnning 4. delete any position 3.delete at end 5.disp
96
97      printf("Enter your choice: ");
98      scanf("%d",&choice);
99      switch(choice){
00      case 1:printf("Enter num of nodes");
01             scanf("%d" ,&n);
02             createList(n);
03             break;
04      case 2:
05              deleteFirst();
06              break;
07      case 4:printf("enter  value: ");
08               scanf("%d",&value);
09              deleteSpecific(value);
10               break;
11      case 3:
12
13         deleteLast();break;
14      case 5:
15          displayList();
16          break;
17      case 6:printf("exiting...");
18              exit(0);
19
20      default:printf("Invalid choice Try again\n");
21      }
22  }
23  return 0;
24  }
25
26
```