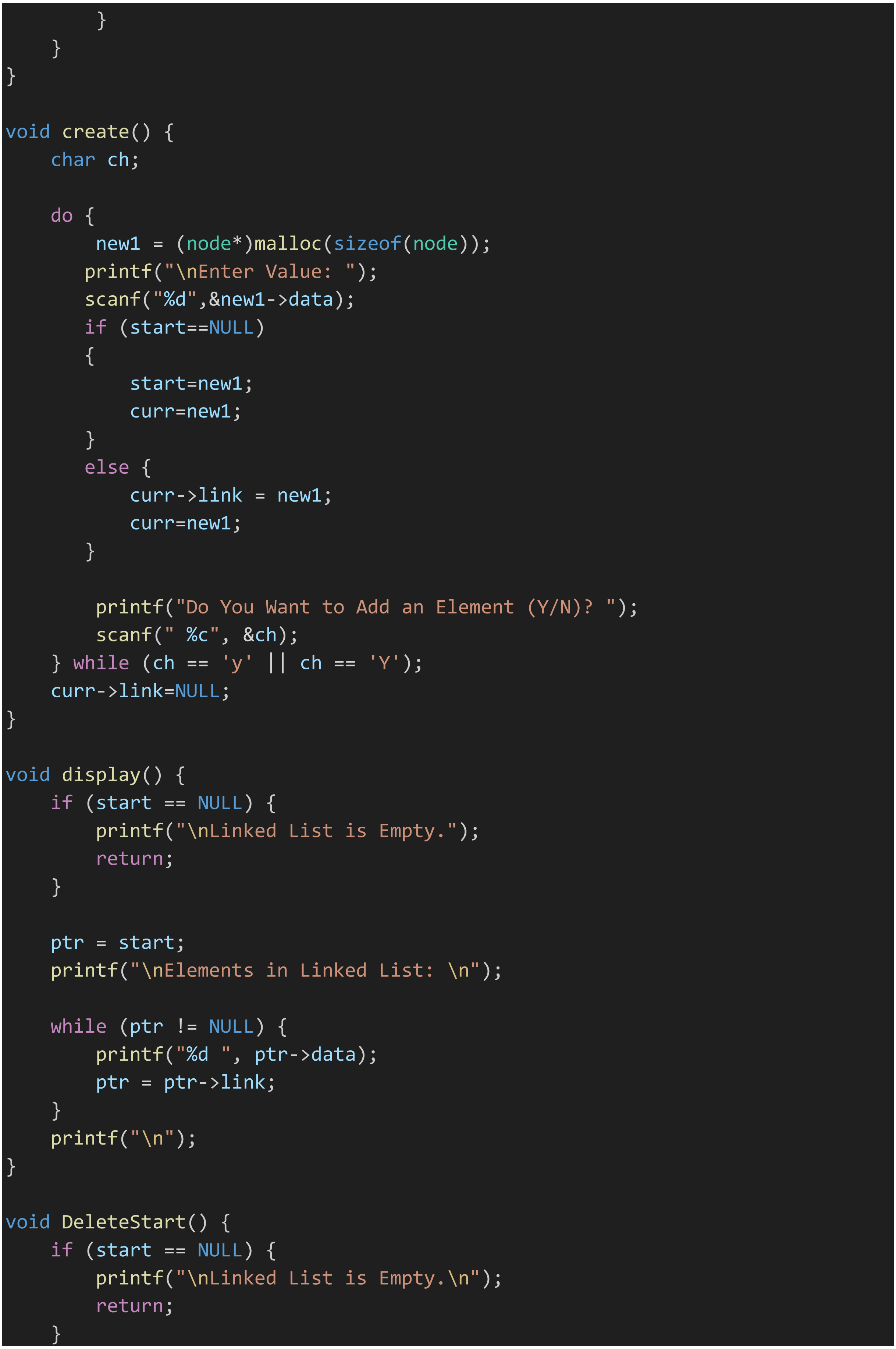
Lab -6 Linked List Deletion

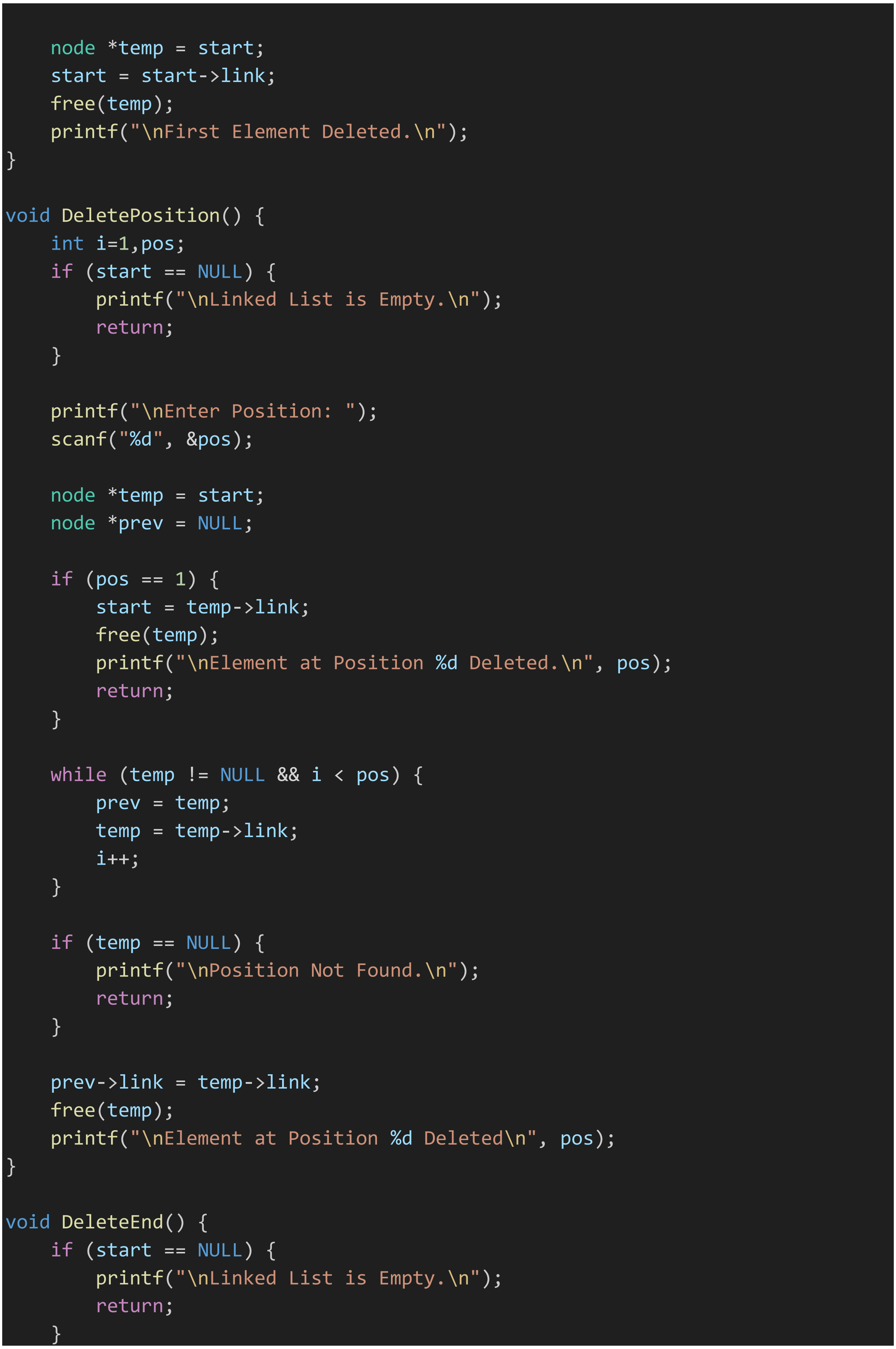
NAME: Afreen Anz

USN: 1BM23CS016

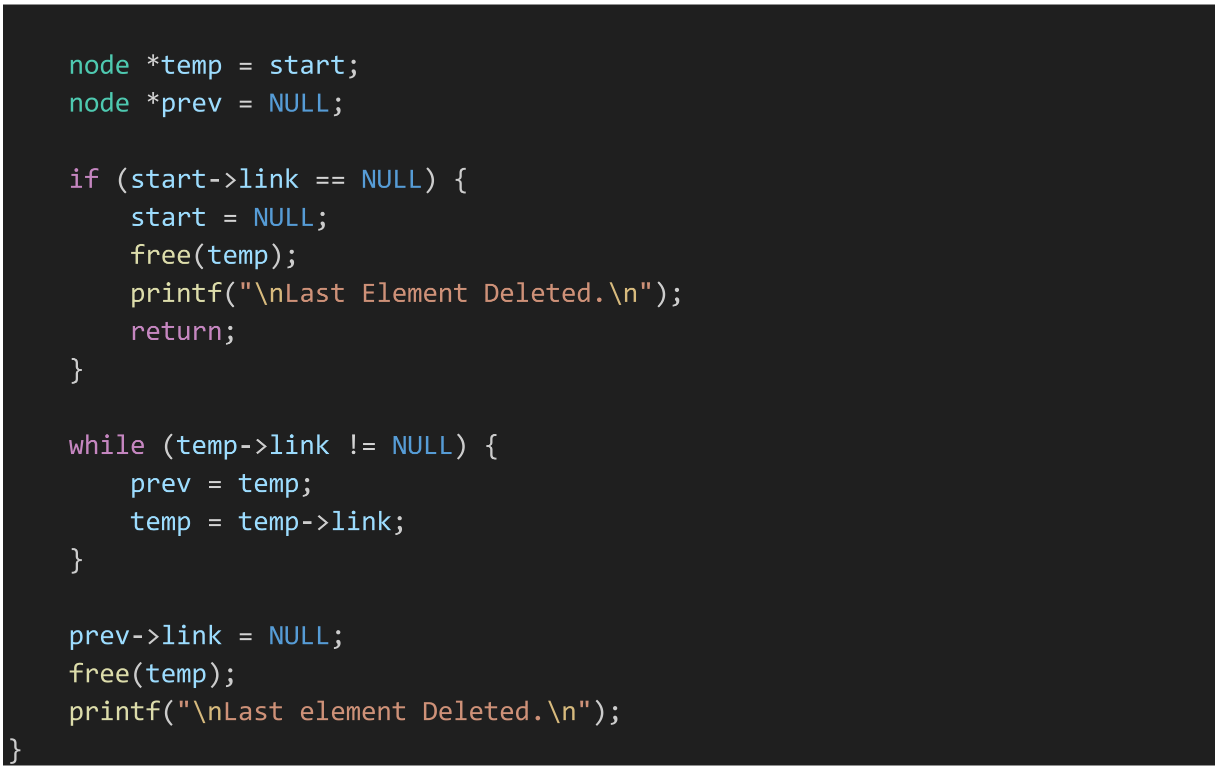
```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node *link;
};

typedef struct Node node;
node *start = NULL;
node *new1, *curr, *ptr;


void create();
void display();
void DeleteStart();
void DeletePosition();
void DeleteEnd();


void main() {
    int ch;
    while (1) {
        printf("\n1. Create \n2. Display \n3. Delete from Beginning \n4.
Delete at Position \n5. Delete at End \n6. Exit");
        printf("\nEnter Your Choice: ");
        scanf("%d", &ch);

        switch (ch) {
            case 1: create();
                break;
            case 2: display();
                break;
            case 3: DeleteStart();
                break;
            case 4: DeletePosition();
                break;
            case 5: DeleteEnd();
                break;
            case 6: exit(0);
```



```
        }
    }
}

void create() {
    char ch;

    do {
        new1 = (node*)malloc(sizeof(node));
       printf("\nEnter Value: ");
       scanf("%d",&new1->data);
       if (start==NULL)
       {
           start=new1;
           curr=new1;
       }
       else {
           curr->link = new1;
           curr=new1;
       }

        printf("Do You Want to Add an Element (Y/N)? ");
        scanf(" %c", &ch);
    } while (ch == 'y' || ch == 'Y');
    curr->link=NULL;
}

void display() {
    if (start == NULL) {
        printf("\nLinked List is Empty.");
        return;
    }

    ptr = start;
    printf("\nElements in Linked List: \n");

    while (ptr != NULL) {
        printf("%d ", ptr->data);
        ptr = ptr->link;
    }
    printf("\n");
}

void DeleteStart() {
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }
```

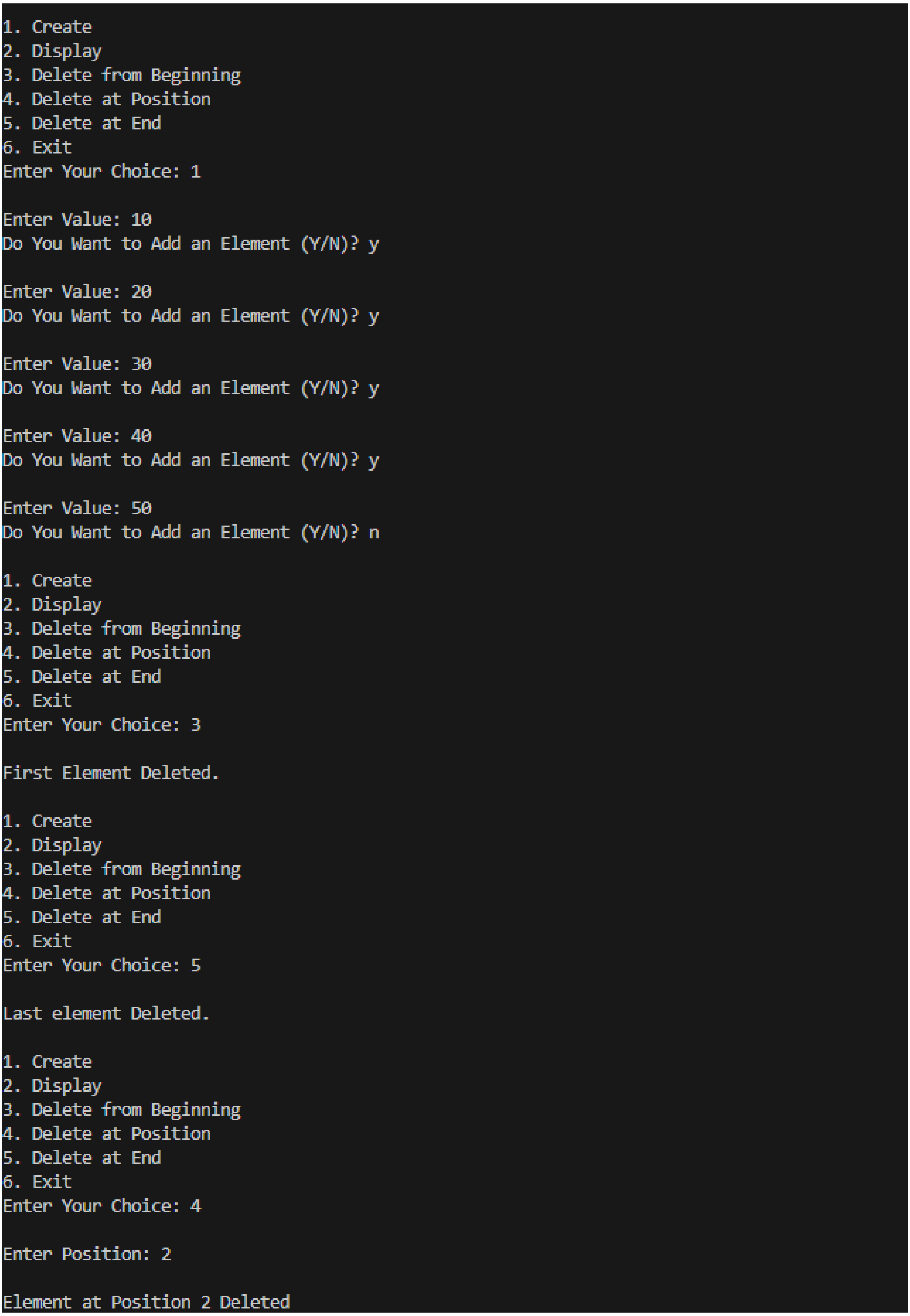


```
    node *temp = start;
    start = start->link;
    free(temp);
    printf("\nFirst Element Deleted.\n");
}

void DeletePosition() {
    int i=1,pos;
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }

    printf("\nEnter Position: ");
    scanf("%d", &pos);

    node *temp = start;
    node *prev = NULL;

    if (pos == 1) {
        start = temp->link;
        free(temp);
        printf("\nElement at Position %d Deleted.\n", pos);
        return;
    }

    while (temp != NULL && i < pos) {
        prev = temp;
        temp = temp->link;
        i++;
    }

    if (temp == NULL) {
        printf("\nPosition Not Found.\n");
        return;
    }

    prev->link = temp->link;
    free(temp);
    printf("\nElement at Position %d Deleted\n", pos);
}

void DeleteEnd() {
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }
```

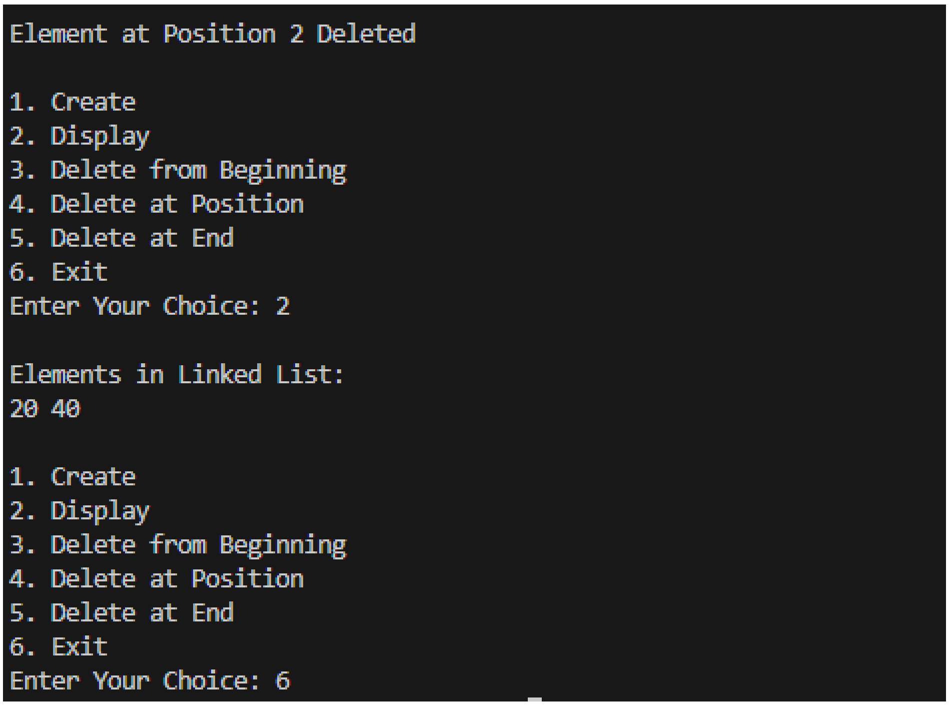


```c
    node *temp = start;
    node *prev = NULL;

    if (start->link == NULL) {
        start = NULL;
        free(temp);
        printf("\nLast Element Deleted.\n");
        return;
    }

    while (temp->link != NULL) {
        prev = temp;
        temp = temp->link;
    }

    prev->link = NULL;
    free(temp);
    printf("\nLast element Deleted.\n");
}
```

Output:



```
1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 1

Enter Value: 10
Do You Want to Add an Element (Y/N)? y

Enter Value: 20
Do You Want to Add an Element (Y/N)? y

Enter Value: 30
Do You Want to Add an Element (Y/N)? y

Enter Value: 40
Do You Want to Add an Element (Y/N)? y

Enter Value: 50
Do You Want to Add an Element (Y/N)? n

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 3

First Element Deleted.

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 5

Last element Deleted.

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 4

Enter Position: 2

Element at Position 2 Deleted
```



```
Element at Position 2 Deleted

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
20 40

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 6
```