

1- Write a program to create a double linked list by taking user input and display its elements

اكتب برنامجًا لإنشاء double linked list عن طريق أخذ مدخلات المستخدم وعرض عناصرها

Input

```
Enter the number of elements: 4
Enter the elements:
1 2 3 4
```

Output

```
Double Linked List: 1 <-> 2 <-> 3 <-> 4 <-> NULL
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

void displayList(struct Node* head) {
    if (head == NULL) {
        printf("NULL\n");
        return;
    }
    printf("%d <-> ", head->data);
    displayList(head->next);
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    printf("Double Linked List: ");
    displayList(head);

    return 0;
}
```

2- Write a program to calculate the sum of elements in a double linked list

اكتب برنامجًا لحساب مجموع العناصر في double linked list

Input

```
Enter the number of elements: 4
Enter the elements:
1 2 3 4
```

Output

```
Sum of elements: 10
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int calculateSum(struct Node* head) {
    if (head == NULL) {
        return 0;
    }
    return head->data + calculateSum(head->next);
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    int sum = calculateSum(head);
    printf("Sum of elements: %d\n", sum);

    return 0;
}
```

3- Write a program to find the sum of even numbers in a double linked list

اكتب برنامجًا لإيجاد مجموع الأعداد الزوجية في double linked list

Input

```
Enter the number of elements: 4
Enter the elements:
1 2 5 8
```

Output

```
Sum of even numbers: 10
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int sumOfEven(struct Node* head) {
    if (head == NULL) {
        return 0;
    }
    int currentSum = (head->data % 2 == 0) ? head->data : 0;
    return currentSum + sumOfEven(head->next);
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    int evenSum = sumOfEven(head);
    printf("Sum of even numbers: %d\n", evenSum);

    return 0;
}
```

4- Write a program to search for a specific element in a double linked list

اكتب برنامجًا للبحث عن عنصر محدد في double linked list

Input

```
Enter the number of elements: 4
Enter the elements:
1 5 8 2
Enter the element to search: 8
```

Output

```
8 is found in the list.
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int searchElement(struct Node* head, int key) {
    if (head == NULL) {
        return 0; // Not found
    }
    if (head->data == key) {
        return 1; // Found
    }
    return searchElement(head->next, key);
}

int main() {
    struct Node* head = NULL;
    int n, value, key;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    printf("Enter the element to search: ");
    scanf("%d", &key);

    if (searchElement(head, key)) {
        printf("%d is found in the list.\n", key);
    } else {
        printf("%d is not found in the list.\n", key);
    }

    return 0;
}
```

5- Write a program to reverse a double linked list

اكتب برنامجًا لعكس double linked list

Input

```
Enter the number of elements: 5
Enter the elements:
1 2 3 4 5
```

Output

```
Double Linked List before reversal: 1 <-> 2 <-> 3 <-> 4 <-> 5 <-> NULL
Double Linked List after reversal: 5 <-> 4 <-> 3 <-> 2 <-> 1 <-> NULL
```


Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

struct Node* reverseList(struct Node* head) {
    if (head == NULL || head->next == NULL) {
        return head;
    }
    struct Node* restReversed = reverseList(head->next);
    head->next->next = head;
    head->next = NULL;
    return restReversed;
}

void displayList(struct Node* head) {
    if (head == NULL) {
        printf("NULL\n");
        return;
    }
    printf("%d <-> ", head->data);
    displayList(head->next);
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    printf("Double Linked List before reversal: ");
    displayList(head);

    head = reverseList(head);

    printf("Double Linked List after reversal: ");
    displayList(head);

    return 0;
}
```

6- Write a program to count the occurrences of a specific element in a double linked list

اكتب برنامجاً لحساب تكرارات عنصر معين في double linked list

Input

```
Enter the number of elements: 5
Enter the elements:
1 2 2 3 5
Enter the element to count occurrences: 2
```

Output

```
Occurrences of 2: 2
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int countOccurrences(struct Node* head, int key) {
    if (head == NULL) {
        return 0;
    }
    int count = (head->data == key) ? 1 : 0;
    return count + countOccurrences(head->next, key);
}

int main() {
    struct Node* head = NULL;
    int n, value, key;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    printf("Enter the element to count occurrences: ");
    scanf("%d", &key);

    int occurrences = countOccurrences(head, key);
    printf("Occurrences of %d: %d\n", key, occurrences);

    return 0;
}
```

7- Write a program to find the maximum element in a double linked list

اكتب برنامجًا للعثور على الحد الأقصى للعنصر في double linked list

Input

```
Enter the number of elements: 5
Enter the elements:
1 2 3 5 4
```

Output

```
Maximum element: 5
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int findMax(struct Node* head) {
    if (head == NULL) {
        return -1; // Assuming all elements are non-negative
    }
    int maxRest = findMax(head->next);
    return (head->data > maxRest) ? head->data : maxRest;
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    int maxElement = findMax(head);
    printf("Maximum element: %d\n", maxElement);

    return 0;
}
```

8- Write a program to find the minimum element in a double linked list

اكتب برنامجًا للعثور على أقل عنصر في double linked list

Input

```
Enter the number of elements: 5
Enter the elements:
5 1 2 3 4
```

Output

```
Minimum element: 1
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int findMin(struct Node* head, int min) {
    if (head == NULL) {
        return min;
    }
    min = (head->data < min) ? head->data : min;
    return findMin(head->next, min);
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    int minElement = findMin(head, head->data);
    printf("Minimum element: %d\n", minElement);

    return 0;
}
```

9- Write a program to insert an element at a specific position in a double linked list

اكتب برنامجًا لإدخال عنصر في موضع محدد في double linked list

Input

```
Enter the number of elements: 5
Enter the elements:
2 3 4 5 6
Enter the position to insert at: 3
Enter the new element: 1
```

Output

```
Double Linked List after insertion: 2 <-> 3 <-> 4 <-> 1 <-> 5 <-> 6 <-> NULL
```

Solution


```

// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

struct Node* insertAtPosition(struct Node* head, int position, int data) {
    if (position < 0) {
        printf("Invalid position.\n");
        return head;
    }
    if (position == 0 || head == NULL) {
        struct Node* newNode = createNode(data);
        newNode->next = head;
        if (head != NULL) {
            head->prev = newNode;
        }
        return newNode;
    }
    head->next = insertAtPosition(head->next, position - 1, data);
    if (head->next != NULL) {
        head->next->prev = head;
    }
    return head;
}

void displayList(struct Node* head) {
    if (head == NULL) {
        printf("NULL\n");
        return;
    }
    printf("%d <-> ", head->data);
    displayList(head->next);
}

int main() {
    struct Node* head = NULL;
    int n, value, position, newData;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    printf("Enter the position to insert at: ");
    scanf("%d", &position);

    printf("Enter the new element: ");
    scanf("%d", &newData);

    head = insertAtPosition(head, position, newData);

    printf("Double Linked List after insertion: ");
    displayList(head);

    return 0;
}

```

10- Write a program to count the number of even elements in a double linked list

اكتب برنامجًا لحساب عدد العناصر الزوجية في double linked list

Input

```
Enter the number of elements: 5
Enter the elements:
1 2 3 4 5
```

Output

```
Number of even elements: 2
```

Solution

```
// www.gammal.tech

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

struct Node* insertEnd(struct Node* head, int data) {
    if (head == NULL) {
        return createNode(data);
    }
    head->next = insertEnd(head->next, data);
    struct Node* temp = head->next;
    temp->prev = head;
    return head;
}

int countEven(struct Node* head) {
    if (head == NULL) {
        return 0;
    }
    int count = (head->data % 2 == 0) ? 1 : 0;
    return count + countEven(head->next);
}

int main() {
    struct Node* head = NULL;
    int n, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = insertEnd(head, value);
    }

    int evenCount = countEven(head);
    printf("Number of even elements: %d\n", evenCount);

    return 0;
}
```
