
1- Write a program to reverse and print the linked list using recursion.

اكتب برنامجًا لعكس linked list وطباعتها .

Input

```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
```

Output

```
Reversed linked list: 4 3 2 1
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int number;
    struct gammal* next;
};

struct gammal* add(struct gammal* g) {
    if (g == NULL) {
        g = (struct gammal*)malloc(sizeof(struct gammal));
        printf("Enter number: ");
        scanf("%d", &g->number);
        g->next = NULL;
        return g;
    }
    g->next = add(g->next);
    return g;
}

void reverseAndPrint(struct gammal* g) {
    if (g == NULL) {
        return;
    }
    reverseAndPrint(g->next);
    printf("%d ", g->number);
}

int main() {
    struct gammal* head = NULL;
    int numberSelect;

    printf("Enter a number that represents the number of numbers in the linked list: ");
    scanf("%d", &numberSelect);

    for (int i = 0; i < numberSelect; i++)
        head = add(head);

    printf("Reversed linked list: ");
    reverseAndPrint(head);
    printf("\n");

    return 0;
}
```

2- Write a program to find and print the sum of all even numbers in the linked list

اكتب برنامجًا للعثور على مجموع الأعداد الزوجية في linked list وطباعته

Input

```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 5
Enter number: 9
Enter number: 8
Enter number: 2
```

Output

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

Solution

```c
// www.gammal.tech

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

struct gammal {
    int number;
    struct gammal* next;
};

struct gammal* add(struct gammal* g) {
    if (g == NULL) {
        g = (struct gammal*)malloc(sizeof(struct gammal));
        printf("Enter number: ");
        scanf("%d", &g->number);
        g->next = NULL;
        return g;
    }
    g->next = add(g->next);
    return g;
}

int sumEven(struct gammal* g) {
    if (g == NULL) {
        return 0;
    }
    int sum = sumEven(g->next);
    return (g->number % 2 == 0) ? sum + g->number : sum;
}

int main() {
    struct gammal* head = NULL;
    int numberSelect;

    printf("Enter a number that represents the number of numbers in the linked list: ");
    scanf("%d", &numberSelect);

    for (int i = 0; i < numberSelect; i++)
        head = add(head);

    int evenSum = sumEven(head);

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

    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

```c
// 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

```c
// 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

```c
// 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- Create a circular linked list with user-specified nodes and display the elements and Find the sum of all nodes in a circular linked list.

أنشئ circular linked تحتوي على العقد المحددة من قبل المستخدم واعرض العناصر وابحث عن مجموع كل العقد في circular linked.

Input & Output

```
1) Add
2) Sum Nodes
3) Show
Please enter a number: 1
Enter num: 1

1) Add
2) Sum Nodes
3) Show
Please enter a number: 1
Enter num: 2

1) Add
2) Sum Nodes
3) Show
Please enter a number: 1
Enter num: 3

1) Add
2) Sum Nodes
3) Show
Please enter a number: 2
Sum of nodes: 6

1) Add
2) Sum Nodes
3) Show
Please enter a number: 3
1 2 3
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
};

struct gammal* add(struct gammal* g, struct gammal* head) {
    if (head == NULL || g->next == head) {
        struct gammal* t = (struct gammal*)malloc(sizeof(struct gammal));
        printf("Enter num: ");
        scanf("%d", &t->num);
        if (head == NULL) {
            t->next = t;
            return t;
        }
        t->next = head;
        g->next = t;
        return g;
    }
    g->next = add(g->next, head);

    return g;
}

int sumNodes(struct gammal* head) {
    if (head == NULL) {
        return 0; // Empty list, sum is 0
    }

    int sum = 0;
    struct gammal* current = head;

    do {
        sum += current->num;
        current = current->next;
    } while (current != head);

    return sum;
}

void show(struct gammal* g, struct gammal* head) {
    if (g == NULL) {
        printf("List is empty.\n");
        return;
    }

    do {
        printf("%d ", g->num);
        g = g->next;
    } while (g != head);
    printf("\n");
}

int main() {
    int a;
    struct gammal* head = NULL;

    do {
        printf("\n1) Add\n"
               "2) Sum Nodes\n"
               "3) Show\n"
               "Please enter a number: ");
        scanf("%d", &a);

        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            int nodeSum = sumNodes(head);
            printf("Sum of nodes: %d\n", nodeSum);
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

7- Create a circular linked list Check if a circular linked list is empty.

قم بإنشاء circular linked وتحقق مما إذا كانت circular linked فارغة.

Input & Output

```
1) Check if Empty
2) Show
Please enter a number: 1
Circular linked list is empty.
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
};

int isEmpty(struct gammal* head) {
    return (head == NULL);
}

void show(struct gammal* g, struct gammal* head) {
    if (g == NULL) {
        printf("List is empty.\n");
        return;
    }

    do {
        printf("%d ", g->num);
        g = g->next;
    } while (g != head);
    printf("\n");
}

int main() {
    int a;
    struct gammal* head = NULL;

    do {
        printf("\n1) Check if Empty\n"
               "2) Show\n"
               "Please enter a number: ");
        scanf("%d", &a);

        if (a == 1) {
            if (isEmpty(head)) {
                printf("Circular linked list is empty.\n");
            } else {
                printf("Circular linked list is not empty.\n");
            }
        } else if (a == 2)
            show(head, head);
    } while (a);

    return 0;
}
```

8- Create a circular linked list with user-specified nodes and display the elements and Find the maximum element in a circular linked list.

قم بإنشاء circular linked مع العقد المحددة من قبل المستخدم وعرض العناصر وابحث عن الحد الأقصى للعنصر في circular linked.

Input & Output

```
1) Add
2) Find Max Element
3) Show
Please enter a number: 1
Enter num: 1

1) Add
2) Find Max Element
3) Show
Please enter a number: 1
Enter num: 2

1) Add
2) Find Max Element
3) Show
Please enter a number: 1
Enter num: 3

1) Add
2) Find Max Element
3) Show
Please enter a number: 2
Maximum element in the list: 3

1) Add
2) Find Max Element
3) Show
Please enter a number: 3
1 2 3
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
};

struct gammal* add(struct gammal* g, struct gammal* head) {
    if (head == NULL || g->next == head) {
        struct gammal* t = (struct gammal*)malloc(sizeof(struct gammal));
        printf("Enter num: ");
        scanf("%d", &t->num);
        if (head == NULL) {
            t->next = t;
            return t;
        }
        t->next = head;
        g->next = t;
        return g;
    }
    g->next = add(g->next, head);

    return g;
}

int findMaxElement(struct gammal* head) {
    if (head == NULL) {
        printf("List is empty.\n");
        return -1; // Assuming -1 as an indicator for an empty list
    }

    int max = head->num;
    struct gammal* current = head->next;

    do {
        if (current->num > max) {
            max = current->num;
        }
        current = current->next;
    } while (current != head);

    return max;
}

void show(struct gammal* g, struct gammal* head) {
    if (g == NULL) {
        printf("List is empty.\n");
        return;
    }

    do {
        printf("%d ", g->num);
        g = g->next;
    } while (g != head);
    printf("\n");
}

int main() {
    int a;
    struct gammal* head = NULL;

    do {
        printf("\n1) Add\n"
               "2) Find Max Element\n"
               "3) Show\n"
               "Please enter a number: ");
        scanf("%d", &a);

        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            int maxElement = findMaxElement(head);
            if (maxElement != -1) {
                printf("Maximum element in the list: %d\n", maxElement);
            }
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

9- Write a program to find the sum of even numbers in a Double Circular Linked List.

اكتب برنامجًا لإيجاد مجموع الأعداد الزوجية في Double Circular Linked List.

Input & Output

```
1) Add
2) Sum of even numbers
0) Exit
Please, enter a number: 1
Enter num: 1
1) Add
2) Sum of even numbers
0) Exit
Please, enter a number: 1
Enter num: 2
1) Add
2) Sum of even numbers
0) Exit
Please, enter a number: 1
Enter num: 3
1) Add
2) Sum of even numbers
0) Exit
Please, enter a number: 1
Enter num: 4
1) Add
2) Sum of even numbers
0) Exit
Please, enter a number: 2
Sum of even numbers in the doubly circular linked list: 6
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next, * prev;
};

struct gammal* add(struct gammal* g, struct gammal* head) {
    if (head == NULL || g->next == head) {
        struct gammal* t = (struct gammal*)malloc(sizeof(struct gammal));

        printf("Enter num: ");
        scanf("%d", &t->num);
        if (head == NULL) {
            t->next = t;
            t->prev = t;
            return t;
        }
        t->next = head;
        head->prev = t;
        g->next = t;
        t->prev = g;
        return g;
    }
    g->next = add(g->next, head);
    return g;
}

int sumOfEvenNumbers(struct gammal* head) {
    if (head == NULL) {
        printf("Doubly circular linked list is empty. Sum of even numbers: 0\n");
        return 0;
    }

    struct gammal* current = head->next;
    int sum = 0;

    do {
        if (current->num % 2 == 0) {
            sum += current->num;
        }
        current = current->next;
    } while (current != head->next);

    printf("Sum of even numbers in the doubly circular linked list: %d\n", sum);
    return sum;
}

int main() {
    int choice;
    struct gammal* head = NULL;

    do {
        printf("1) Add\n"
               "2) Sum of even numbers\n"
               "0) Exit\n"
               "Please, enter a number: ");
        scanf("%d", &choice);

        if (choice == 1)
            head = add(head, head);
        else if (choice == 2)
            sumOfEvenNumbers(head);

    } while (choice != 0);

    return 0;
}
```

10- Write a program to search for a specific element in a Double Circular Linked List.

اكتب برنامجًا للبحث عن عنصر محدد في Double Circular Linked List.

Input & Output

```
1) Add
2) Search for an element
0) Exit
Please, enter a number: 1
Enter num: 3
1) Add
2) Search for an element
0) Exit
Please, enter a number: 1
Enter num: 5
1) Add
2) Search for an element
0) Exit
Please, enter a number: 1
Enter num: 7
1) Add
2) Search for an element
0) Exit
Please, enter a number: 2
Enter the element to search: 5
Element 5 found at position 1.
1) Add
2) Search for an element
0) Exit
Please, enter a number: 2
Enter the element to search: 9
Element 9 not found in the doubly circular linked list.
1) Add
2) Search for an element
0) Exit
Please, enter a number: 0
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next, * prev;
};

struct gammal* add(struct gammal* g, struct gammal* head) {
    if (head == NULL || g->next == head) {
        struct gammal* t = (struct gammal*)malloc(sizeof(struct gammal));

        printf("Enter num: ");
        scanf("%d", &t->num);
        if (head == NULL) {
            t->next = t;
            t->prev = t;
            return t;
        }
        t->next = head;
        head->prev = t;
        g->next = t;
        t->prev = g;
        return g;
    }
    g->next = add(g->next, head);
    return g;
}

int searchElement(struct gammal* head, int target) {
    if (head == NULL) {
        printf("Doubly circular linked list is empty. Cannot search for the element.\n");
        return 0;
    }

    struct gammal* current = head->next;
    int position = 1;

    do {
        if (current->num == target) {
            printf("Element %d found at position %d.\n", target, position);
            return 1;
        }
        current = current->next;
        position++;
    } while (current != head->next);

    printf("Element %d not found in the doubly circular linked list.\n", target);
    return 0;
}

int main() {
    int choice, target;
    struct gammal* head = NULL;

    do {
        printf("1) Add\n"
               "2) Search for an element\n"
               "0) Exit\n"
               "Please, enter a number: ");
        scanf("%d", &choice);

        if (choice == 1)
            head = add(head, head);
        else if (choice == 2) {
            printf("Enter the element to search: ");
            scanf("%d", &target);
            searchElement(head, target);
        }

    } while (choice != 0);

    return 0;
}
```