

1- Create a circular linked list with user-specified nodes and display the elements.

قم بإنشاء circular linked مع node المحددة من قبل المستخدم وعرض العناصر.

Input & Output

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

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

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

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

Solution

```

// 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;
}

void show(struct gammal* g, struct gammal* head) {
    do {
        printf("%d ", g->num);
        g = g->next;
    } while (g != head);
}

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

    do {
        printf("\n1) Add\n2) Show\nPlease enter a number: ");
        scanf("%d", &choice);

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

    } while (choice);

    return 0;
}
```

2- Write a program Count the number of nodes in the circular linked list and display the count.

كتابة برنامج لحساب عدد nodes في circular linked وعرض العدد.

Input & Output

```
Enter num: 1
1) Add
2) Show
3) Count Nodes
Please enter a number: 1
Enter num: 2
1) Add
2) Show
3) Count Nodes
Please enter a number: 1
Enter num: 3
1) Add
2) Show
3) Count Nodes
Please enter a number: 3
Number of nodes in the circular linked list: 3
1) Add
2) Show
3) Count Nodes
Please enter a number: 2
1 2 3
```

Solution

```

// 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;
}

void show(struct gammal* g, struct gammal* head) {
    do {
        printf("%d ", g->num);
        g = g->next;
    } while (g != head);
}

int countNodes(struct gammal* head) {
    if (head == NULL)
        return 0;

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

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

    return count;
}

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

    do {
        printf("\n1) Add\n2) Show\n3) Count Nodes\nPlease enter a number: ");
        scanf("%d", &choice);

        if (choice == 1)
            head = add(head, head);
        else if (choice == 2)
            show(head, head);
        else if (choice == 3) {
            int count = countNodes(head);
            printf("Number of nodes in the circular linked list: %d\n", count);
        }

    } while (choice);

    return 0;
}
```

3- Create a circular linked list with user-specified nodes and display the elements and Search for a specific element in a circular linked list.

إنشاء circular linked مع العقد المحددة من قبل المستخدم وعرض العناصر والبحث عن عنصر معين في circular linked.

Input & Output

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

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

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

1) Add
2) Search Element
3) Show
Please enter a number: 2
Enter the element to search: 3
3 found in the list.
```

Solution

```

// 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 searchElement(struct gammal* head, int target) {
    if (head == NULL) {
        return 0; // Empty list, element not found
    }

    struct gammal* current = head;

    do {
        if (current->num == target) {
            return 1; // Element found
        }
        current = current->next;
    } while (current != head);

    return 0; // Element not found
}

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, target;
    struct gammal* head = NULL;

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

        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            printf("Enter the element to search: ");
            scanf("%d", &target);
            if (searchElement(head, target)) {
                printf("%d found in the list.\n", target);
            } else {
                printf("%d not found in the list.\n", target);
            }
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

4- Create a circular linked list with user-specified nodes and display the elements and Reverse a circular linked list.

قم بإنشاء circular linked مع العقد المحددة من قبل المستخدم وعرض العناصر وعكس circular linked.

Input & Output

```
1) Add
2) Reverse List
3) Show
Please enter a number: 1
Enter num: 1
```

```
1) Add
2) Reverse List
3) Show
Please enter a number: 1
Enter num: 2
```

```
1) Add
2) Reverse List
3) Show
Please enter a number: 1
Enter num: 3
```

```
1) Add
2) Reverse List
3) Show
Please enter a number: 2
List reversed.
```

```
1) Add
2) Reverse List
3) Show
Please enter a number: 3
3 2 1
```

Solution

```

// 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;
}

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

    struct gammal *current = head, *prev = NULL, *nextNode = NULL;

    do {
        nextNode = current->next;
        current->next = prev;
        prev = current;
        current = nextNode;
    } while (current != head);

    head->next = prev;
    head = prev;

    printf("List reversed.\n");

    return head;
}

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) Reverse List\n"
            "3) Show\n"
            "Please enter a number: ");
        scanf("%d", &a);

        if (a == 1)
            head = add(head, head);
        else if (a == 2)
            head = reverseList(head);
        else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```


5- Create a circular linked list with user-specified nodes and display the elements and Insert a new node at the end of a circular linked list.

قم بإنشاء circular linked مع العقد المحددة من قبل المستخدم وعرض العناصر وإدراج عقدة جديدة في نهاية circular linked.

Input & Output

```
1) Add
2) Insert at End
3) Show
Please enter a number: 1
Enter num: 1

1) Add
2) Insert at End
3) Show
Please enter a number: 1
Enter num: 2

1) Add
2) Insert at End
3) Show
Please enter a number: 1
Enter num: 3

1) Add
2) Insert at End
3) Show
Please enter a number: 2
Enter data for the new node to insert at the end: 8
Node with data 8 inserted at the end.

1) Add
2) Insert at End
3) Show
Please enter a number: 3
1 2 3 8
```

Activate Windows

Solution

```

// 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;
}

struct gammal* insertAtEnd(struct gammal* head, int newData) {
    struct gammal* newNode = (struct gammal*)malloc(sizeof(struct gammal));
    newNode->num = newData;
    newNode->next = head;

    struct gammal* last = head;
    while (last->next != head) {
        last = last->next;
    }

    last->next = newNode;
    printf("Node with data %d inserted at the end.\n", newData);

    return head;
}

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, newData;
    struct gammal* head = NULL;

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

        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            printf("Enter data for the new node to insert at the end: ");
            scanf("%d", &newData);
            head = insertAtEnd(head, newData);
        } else if (a == 3)
            show(head, head);
    } while (a);

    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 واعرص 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

```

// 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

```
// 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

```

// 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- Create a circular linked list with user-specified nodes and display the elements and Calculate the average of all nodes in a circular linked list.

قم بإنشاء circular linked مع العقد المحددة من قبل المستخدم وعرض العناصر وحساب متوسط جميع العقد في circular linked.

Input & Output

```
1) Add
2) Calculate Average
3) Show
Please enter a number: 1
Enter num: 1

1) Add
2) Calculate Average
3) Show
Please enter a number: 1
Enter num: 2

1) Add
2) Calculate Average
3) Show
Please enter a number: 1
Enter num: 3

1) Add
2) Calculate Average
3) Show
Please enter a number: 2
Average of nodes in the list: 2.00

1) Add
2) Calculate Average
3) Show
Please enter a number: 3
1 2 3
```



```

// 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;
}

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

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

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

    return (float)sum / count;
}

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) Calculate Average\n"
            "3) Show\n"
            "Please enter a number: ");
        scanf("%d", &a);

        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            float average = calculateAverage(head);
            if (average != -1) {
                printf("Average of nodes in the list: %.2f\n", average);
            }
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}

```

10- Create a circular linked list with user-specified nodes and display the elements and Copy a circular linked list into another circular linked list.

قم بإنشاء circular linked مع العقد المحددة من قبل المستخدم وعرض العناصر ونسخ circular linked إلى circular linked أخرى.

Input & Output

```
1) Add to Original List
2) Copy List
3) Show Original List
4) Show Copied List
Please enter a number: 1
Enter num: 1
```

```
1) Add to Original List
2) Copy List
3) Show Original List
4) Show Copied List
Please enter a number: 1
Enter num: 2
```

```
1) Add to Original List
2) Copy List
3) Show Original List
4) Show Copied List
Please enter a number: 1
Enter num: 3
```

```
1) Add to Original List
2) Copy List
3) Show Original List
4) Show Copied List
Please enter a number: 1
Enter num: 4
```

```
1) Add to Original List
2) Copy List
3) Show Original List
4) Show Copied List
Please enter a number: 2
List copied successfully.
```

```
1) Add to Original List
2) Copy List
3) Show Original List
4) Show Copied List
Please enter a number: 4
1 2 3 4
```

Activate Windows
Go to Settings to activate Windows.

```

// 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;
}

struct gammal* copyList(struct gammal* originalHead) {
    if (originalHead == NULL) {
        printf("Original list is empty.\n");
        return NULL;
    }

    struct gammal *originalCurrent = originalHead, *newHead = NULL, *newCurrent = NULL;

    do {
        struct gammal* newNode = (struct gammal*)malloc(sizeof(struct gammal));
        newNode->num = originalCurrent->num;

        if (newHead == NULL) {
            newHead = newNode;
            newCurrent = newNode;
        } else {
            newCurrent->next = newNode;
            newCurrent = newNode;
        }

        originalCurrent = originalCurrent->next;
    } while (originalCurrent != originalHead);

    // Close the circular link
    newCurrent->next = newHead;

    printf("List copied successfully.\n");

    return newHead;
}

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* originalHead = NULL;
    struct gammal* copiedHead = NULL;

    do {
        printf("\n1) Add to Original List\n"
            "2) Copy List\n"
            "3) Show Original List\n"
            "4) Show Copied List\n"
            "Please enter a number: ");
        scanf("%d", &a);

        if (a == 1)
            originalHead = add(originalHead, originalHead);
        else if (a == 2)
            copiedHead = copyList(originalHead);
        else if (a == 3)
            show(originalHead, originalHead);
        else if (a == 4)
            show(copiedHead, copiedHead);
    } while (a);

    return 0;
}

```