1- Create a Double Circular Linked List and add nodes to it.

قم بإنشاء Double Circular Linked List وأضف node إليها.

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
2 3 1
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
    struct gammal* 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 t;
    }
    g->next = add(g->next, head);
    return g;
}

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

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

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

    do {
        printf("1) Add\n"
               "2) Show\n"
               "Please, enter a number: ");
        scanf("%d", &a);
        if (a == 1)
            head = add(head, head);
        else if (a == 2)
            show(head, head);
    } while (a);

    return 0;
}
```

2- Create a Double Circular Linked List and add nodes to it and Find the length of the doubly linked list.

قم بإنشاء Double Circular Linked List وأضف node إليها وابحث عن طول Double Circular Linked List.

Input & Output

```
1) Add
2) Find Length
3) Show
Please, enter a number: 1
Enter num: 1
1) Add
2) Find Length
3) Show
Please, enter a number: 1
Enter num: 2
1) Add
2) Find Length
3) Show
Please, enter a number: 1
Enter num: 4
1) Add
2) Find Length
3) Show
Please, enter a number: 2
Length of the doubly linked list: 3
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
    struct gammal* 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 t;
    }
    g->next = add(g->next, head);
    return g;
}

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

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

    do {
        length++;
        current = current->prev;
    } while (current != head);

    return length;
}

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

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

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

    do {
        printf("1) Add\n"
               "2) Find Length\n"
               "3) Show\n"
               "Please, enter a number: ");
        scanf("%d", &a);
        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            int length = findLength(head);
            printf("Length of the doubly linked list: %d\n", length);
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

3- Create a Double Circular Linked List and add nodes to it and Delete all nodes from the doubly linked list.

إنشاء Double Circular Linked List وإضافة node إليها وحذف كافة node من Double Circular Linked List.

Input & Output

```
1) Add
2) Delete All Nodes
3) Show
Please, enter a number: 1
Enter num: 1
1) Add
2) Delete All Nodes
3) Show
Please, enter a number: 1
Enter num: 2
1) Add
2) Delete All Nodes
3) Show
Please, enter a number: 1
Enter num: 3
1) Add
2) Delete All Nodes
3) Show
Please, enter a number: 2
All nodes deleted.
1) Add
2) Delete All Nodes
3) Show
Please, enter a number: 3
List is empty.
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
    struct gammal* 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 t;
    }
    g->next = add(g->next, head);
    return g;
}

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

    struct gammal *current = head, *temp;

    do {
        temp = current;
        current = current->next;
        free(temp);
    } while (current != head);

    printf("All nodes deleted.\n");
}

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

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

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

    do {
        printf("1) Add\n"
               "2) Delete All Nodes\n"
               "3) Show\n"
               "Please, enter a number: ");
        scanf("%d", &a);
        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            deleteAllNodes(head);
            head = NULL; // Reset the head after deleting all nodes
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

4- Create a Double Circular Linked List and add nodes to it and Calculate the sum of all nodes in the doubly linked list.

قم بإنشاء Double Circular Linked List وأضف node إليها واحسب مجموع جميع node في القائمة المرتبطة بشكل مزدوج.

Input & Output

```
1) Add
2) Calculate Sum
3) Show
Please, enter a number: 1
Enter num: 1
1) Add
2) Calculate Sum
3) Show
Please, enter a number: 1
Enter num: 2
1) Add
2) Calculate Sum
3) Show
Please, enter a number: 1
Enter num: 3
1) Add
2) Calculate Sum
3) Show
Please, enter a number: 1
Enter num: 5
1) Add
2) Calculate Sum
3) Show
Please, enter a number: 2
Sum of nodes in the doubly linked list: 11
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
    struct gammal* 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 t;
    }
    g->next = add(g->next, head);
    return g;
}

int calculateSum(struct gammal* head) {
    if (head == NULL) {
        printf("List is empty.\n");
        return 0;
    }

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

    do {
        sum += current->num;
        current = current->prev;
    } 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->prev;
    } while (g != head);
    printf("\n------\n");
}

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

    do {
        printf("1) Add\n"
               "2) Calculate Sum\n"
               "3) Show\n"
               "Please, enter a number: ");
        scanf("%d", &a);
        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            int sum = calculateSum(head);
            printf("Sum of nodes in the doubly linked list: %d\n", sum);
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

5- Create a Double Circular Linked List and add nodes to it. Find the maximum element in the doubly linked list.

قم بإنشاء Double Circular Linked List وأضف node إليها ابحث عن الحد الأقصى للعنصر في القائمة المرتبطة بشكل مزدوج.

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: 5
1) Add
2) Find Max Element
3) Show
Please, enter a number: 1
Enter num: 7
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 doubly linked list: 7
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
    struct gammal* 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 t;
    }
    g->next = add(g->next, head);
    return g;
}

int findMaxElement(struct gammal* head) {
    if (head == NULL) {
        printf("List is empty.\n");
        return 0; // Assuming 0 as a default value for an empty list
    }

    int maxElement = head->num;
    struct gammal* current = head->prev; // Start from the last node

    do {
        if (current->num > maxElement) {
            maxElement = current->num;
        }
        current = current->prev;
    } while (current != head);

    return maxElement;
}

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

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

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

    do {
        printf("1) 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);
            printf("Maximum element in the doubly linked list: %d\n", maxElement);
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

6- Create a Double Circular Linked List and add nodes to it and Find the minimum element in the doubly linked list.

قم بإنشاء Double Circular Linked List وأضف node إليها وابحث عن الحد الأدنى من العناصر في القائمة المرتبطة بشكل مزدوج.

Input & Output

```
1) Add
2) Find Min Element
3) Show
Please, enter a number: 1
Enter num: 1
1) Add
2) Find Min Element
3) Show
Please, enter a number: 1
Enter num: -5
1) Add
2) Find Min Element
3) Show
Please, enter a number: 1
Enter num: 2
1) Add
2) Find Min Element
3) Show
Please, enter a number: 2
Minimum element in the doubly linked list: -5
```

Solution

```c
// www.gammal.tech

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

struct gammal {
    int num;
    struct gammal* next;
    struct gammal* 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 t;
    }
    g->next = add(g->next, head);
    return g;
}

int findMinElement(struct gammal* head) {
    if (head == NULL) {
        printf("List is empty.\n");
        return 0; // Assuming 0 as a default value for an empty list
    }

    int minElement = head->num;
    struct gammal* current = head->prev; // Start from the last node

    do {
        if (current->num < minElement) {
            minElement = current->num;
        }
        current = current->prev;
    } while (current != head);

    return minElement;
}

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

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

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

    do {
        printf("1) Add\n"
               "2) Find Min Element\n"
               "3) Show\n"
               "Please, enter a number: ");
        scanf("%d", &a);
        if (a == 1)
            head = add(head, head);
        else if (a == 2) {
            int minElement = findMinElement(head);
            printf("Minimum element in the doubly linked list: %d\n", minElement);
        } else if (a == 3)
            show(head, head);
    } while (a);

    return 0;
}
```

7- Write a program to count the number of occurrences of a specific element in a Double Circular Linked List.

اكتب برنامجاً لحساب عدد مرات ظهور عنصر معين في Double Circular Linked List.

Input & Output

```
1) Add
2) Count occurrences of an element
0) Exit
Please, enter a number: 1
Enter num: 2
1) Add
2) Count occurrences of an element
0) Exit
Please, enter a number: 1
Enter num: 3
1) Add
2) Count occurrences of an element
0) Exit
Please, enter a number: 1
Enter num: 3
1) Add
2) Count occurrences of an element
0) Exit
Please, enter a number: 1
Enter num: 2
1) Add
2) Count occurrences of an element
0) Exit
Please, enter a number: 2
Enter the element to count occurrences: 3
Occurrences of 3 in the doubly circular linked list: 2
```

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 countOccurrences(struct gammal* head, int target) {
    if (head == NULL) {
        printf("Doubly circular linked list is empty. Occurrences: 0\n");
        return 0;
    }

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

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

    printf("Occurrences of %d in the doubly circular linked list: %d\n", target, occurrences);
    return occurrences;
}

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

    do {
        printf("1) Add\n"
               "2) Count occurrences of 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 count occurrences: ");
            scanf("%d", &target);
            countOccurrences(head, target);
        }

    } while (choice != 0);

    return 0;
}
```

8- Write a program to count the number of even elements in a Double Circular Linked List.

اكتب برنامجًا لحساب عدد العناصر الزوجية في Double Circular Linked List.

Input & Output

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

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 countEven(struct gammal* head) {
    if (head == NULL) {
        printf("Doubly circular linked list is empty. No even elements.\n");
        return 0;
    }

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

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

    printf("Number of even elements in the doubly circular linked list: %d\n", count);
    return count;
}

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

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

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

    } while (choice != 0);

    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;
}
```