

1- Write a program that counts the number of nodes in a linked list.

اكتب برنامجًا يحسب عدد nodes في linked list.

Input & Output

```
1) Add
2) Show
3) Count Nodes
4) Exit
Enter a number: 1
Enter the number: 2
1) Add
2) Show
3) Count Nodes
4) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Count Nodes
4) Exit
Enter a number: 1
Enter the number: 4
1) Add
2) Show
3) Count Nodes
4) Exit
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the number of nodes in the linked list
int countNodes(struct gammal* head) {
    int count = 0;

    // Traverse the list and count nodes
    while (head != NULL) {
        count++;
        head = head->next;
    }

    return count;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, nodeCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Nodes\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                nodeCount = countNodes(head);
                printf("Number of nodes in the list: %d\n", nodeCount);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}
```

2- Write a program that counts the number of nodes containing odd numbers in a linked list.

اكتب برنامجًا يحسب عدد nodes التي تحتوي على أرقام فردية في linked list

Input & Output

```
1) Add
2) Show
3) Count Odd Nodes
4) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Count Odd Nodes
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Count Odd Nodes
4) Exit
Enter a number: 1
Enter the number: 6
1) Add
2) Show
3) Count Odd Nodes
4) Exit
Enter a number: 3
Number of nodes with odd numbers: 2
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the number of nodes with odd numbers
int countOddNodes(struct gammal* head) {
    int count = 0;

    // Traverse the list and count nodes with odd numbers
    while (head != NULL) {
        if (head->num % 2 != 0) {
            count++;
        }
        head = head->next;
    }

    return count;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, oddCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Odd Nodes\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                oddCount = countOddNodes(head);
                printf("Number of nodes with odd numbers: %d\n", oddCount);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}
```

3- Write a program that counts the number of nodes containing even numbers in a linked list.

اكتب برنامجًا يحسب عدد nodes التي تحتوي على أرقام زوجية في linked list

Input & Output

```
1) Add
2) Show
3) Count Even Nodes
4) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Count Even Nodes
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Count Even Nodes
4) Exit
Enter a number: 1
Enter the number: 2
1) Add
2) Show
3) Count Even Nodes
4) Exit
Enter a number: 3
Number of nodes with even numbers: 1
1) Add
2) Show
3) Count Even Nodes
4) Exit
Enter a number: 4
```

Activate Windows
Go to Settings to activate Windows.

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the number of nodes with even numbers
int countEvenNodes(struct gammal* head) {
    int count = 0;

    // Traverse the list and count nodes with even numbers
    while (head != NULL) {
        if (head->num % 2 == 0 && head->num != 0) {
            count++;
        }
        head = head->next;
    }

    return count;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, evenCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Even Nodes\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                evenCount = countEvenNodes(head);
                printf("Number of nodes with even numbers: %d\n", evenCount);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}
```

4- Write a program that counts the number of nodes containing even numbers and counts the total number of nodes in a linked list.

اكتب برنامجًا يحصي عدد nodes التي تحتوي على أرقام زوجية ويحصي العدد الإجمالي للعقد في القائمة المرتبطة.

Input & Output

```
1) Add
2) Show
3) Count Even Nodes
4) Count All Nodes
5) Exit
Enter a number: 1
Enter the number: 2
1) Add
2) Show
3) Count Even Nodes
4) Count All Nodes
5) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Count Even Nodes
4) Count All Nodes
5) Exit
Enter a number: 1
Enter the number: 4
1) Add
2) Show
3) Count Even Nodes
4) Count All Nodes
5) Exit
Enter a number: 3
Number of nodes with even numbers: 2
1) Add
2) Show
3) Count Even Nodes
4) Count All Nodes
5) Exit
Enter a number: 4
Total number of nodes in the list: 4
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the number of nodes with even numbers
int countEvenNodes(struct gammal* head) {
    int evenCount = 0;

    // Traverse the list and count nodes with even numbers
    while (head != NULL) {
        if (head->num % 2 == 0 && head->num != 0) {
            evenCount++;
        }
        head = head->next;
    }

    return evenCount;
}

// Function to count the total number of nodes in the linked list
int countAllNodes(struct gammal* head) {
    int totalCount = 0;

    // Traverse the list and count all nodes
    while (head != NULL) {
        totalCount++;
        head = head->next;
    }

    return totalCount;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, evenCount, totalCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Even Nodes\n4) Count All Nodes\n5) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                evenCount = countEvenNodes(head);
                printf("Number of nodes with even numbers: %d\n", evenCount);
                break;
            case 4:
                totalCount = countAllNodes(head);
                printf("Total number of nodes in the list: %d\n", totalCount);
                break;
            case 5:
                exit(0);
        }
    } while (choice != 5);

    return 0;
}

```


5- Write a program that calculates the sum of all nodes containing odd numbers in a linked list.

اكتب برنامجًا يحسب مجموع كل node التي تحتوي على أرقام فردية في
linked list

Input & Output

```
1) Add
2) Show
3) Sum of Odd Numbers
4) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Sum of Odd Numbers
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Sum of Odd Numbers
4) Exit
Enter a number: 1
Enter the number: 6
1) Add
2) Show
3) Sum of Odd Numbers
4) Exit
Enter a number: 3
Sum of nodes with odd numbers: 8
1) Add
2) Show
3) Sum of Odd Numbers
4) Exit
Enter a number: 4
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to calculate the sum of nodes with odd numbers
int sumOfOddNodes(struct gammal* head) {
    int oddSum = 0;

    // Traverse the list and sum nodes with odd numbers
    while (head != NULL) {
        if (head->num % 2 != 0) {
            oddSum += head->num;
        }
        head = head->next;
    }

    return oddSum;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, oddSum;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Sum of Odd Numbers\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                oddSum = sumOfOddNodes(head);
                printf("Sum of nodes with odd numbers: %d\n", oddSum);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}

```

6- Write a program that calculates the sum of all nodes in a linked list.

اكتب برنامجًا يحسب مجموع كل node في linked list.

Input & Output

```
1) Add
2) Show
3) Sum of All Numbers
4) Exit
Enter a number: 1
Enter the number: 2
1) Add
2) Show
3) Sum of All Numbers
4) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Sum of All Numbers
4) Exit
Enter a number: 3
Sum of all nodes: 5
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to calculate the sum of all nodes
int sumOfAllNodes(struct gammal* head) {
    int totalSum = 0;

    // Traverse the list and sum all nodes
    while (head != NULL) {
        totalSum += head->num;
        head = head->next;
    }

    return totalSum;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, totalSum;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Sum of All Numbers\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                totalSum = sumOfAllNodes(head);
                printf("Sum of all nodes: %d\n", totalSum);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}

```

7- Write a program that counts the number of nodes containing negative numbers in a linked list.

اكتب برنامجًا يحسب عدد node التي تحتوي على أرقام سالبة في linked list

Input & Output

```
1) Add
2) Show
3) Count Negative Numbers
4) Exit
Enter a number: 1
Enter the number: -2
1) Add
2) Show
3) Count Negative Numbers
4) Exit
Enter a number: 1
Enter the number: -8
1) Add
2) Show
3) Count Negative Numbers
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Count Negative Numbers
4) Exit
Enter a number: 1
Enter the number: -1
1) Add
2) Show
3) Count Negative Numbers
4) Exit
Enter a number: 3
Number of nodes with negative numbers: 3
```

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the number of nodes with negative numbers
int countNegativeNodes(struct gammal* head) {
    int negativeCount = 0;

    // Traverse the list and count nodes with negative numbers
    while (head != NULL) {
        if (head->num < 0) {
            negativeCount++;
        }
        head = head->next;
    }

    return negativeCount;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, negativeCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Negative Numbers\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                negativeCount = countNegativeNodes(head);
                printf("Number of nodes with negative numbers: %d\n", negativeCount);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}

```

8- Write a program that counts the occurrences of a specific number (e.g., 5) in a linked list.

اكتب برنامج يقوم بعد تكرارات رقم معين (مثلاً 5) في linked list.

Input & Output

```
1) Add
2) Show
3) Count Occurrences of a Number
4) Exit
Enter a number: 1
Enter the number: 2
1) Add
2) Show
3) Count Occurrences of a Number
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Count Occurrences of a Number
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Count Occurrences of a Number
4) Exit
Enter a number: 1
Enter the number: 6
1) Add
2) Show
3) Count Occurrences of a Number
4) Exit
Enter a number: 3
Enter the target number: 5
Number of occurrences of 5: 2
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the occurrences of a specific number in the linked list
int countOccurrences(struct gammal* head, int target) {
    int count = 0;

    // Traverse the list and count occurrences
    while (head != NULL) {
        if (head->num == target) {
            count++;
        }
        head = head->next;
    }

    return count;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, target, occurrenceCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Occurrences of a Number\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                printf("Enter the target number: ");
                scanf("%d", &target);
                occurrenceCount = countOccurrences(head, target);
                printf("Number of occurrences of %d: %d\n", target, occurrenceCount);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}

```

9- Write a program that counts the number of nodes containing positive numbers in a linked list.

اكتب برنامج يحسب عدد node التي تحتوي على أرقام موجبة في linked list.

Input & Output

```
1) Add
2) Show
3) Count Positive Numbers
4) Exit
Enter a number: 1
Enter the number: 3
1) Add
2) Show
3) Count Positive Numbers
4) Exit
Enter a number: 1
Enter the number: 5
1) Add
2) Show
3) Count Positive Numbers
4) Exit
Enter a number: 1
Enter the number: -2
1) Add
2) Show
3) Count Positive Numbers
4) Exit
Enter a number: 3
Number of nodes with positive numbers: 2
```

Solution

```

// www.gammal.tech

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

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

// Function to add a new node to the end of the linked list
void add(struct gammal* g) {
    int num;

    printf("Enter the number: ");
    scanf("%d", &num);

    if (g->num == -1) {
        g->num = num;
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;

        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->num = num;
        g->next = NULL;
    }
}

// Function to count the number of nodes with positive numbers
int countPositiveNodes(struct gammal* head) {
    int positiveCount = 0;

    // Traverse the list and count nodes with positive numbers
    while (head != NULL) {
        if (head->num > 0) {
            positiveCount++;
        }
        head = head->next;
    }

    return positiveCount;
}

// Function to display the linked list
void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%d\n", g->num);
        g = g->next;
    }
}

int main() {
    int choice, positiveCount;
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));

    do {
        printf("1) Add\n2) Show\n3) Count Positive Numbers\n4) Exit\n");
        printf("Enter a number: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add(head);
                break;
            case 2:
                show(head);
                break;
            case 3:
                positiveCount = countPositiveNodes(head);
                printf("Number of nodes with positive numbers: %d\n", positiveCount);
                break;
            case 4:
                exit(0);
        }
    } while (choice != 4);

    return 0;
}

```

10- Write a program to add a person with a payment of 100 and the name "John" to the linked list. Print the updated linked list.

اكتب برنامجًا لإضافة شخص بدفعة 100 واسم "جون" إلى linked list طباعة linked list المحدثة.

Output

```
-----  
0  
-----  
John 100
```

Solution

```

// www.gammal.tech

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

struct gammal {
    int pyment;
    char name[20];
    struct gammal* next;
};

void add(struct gammal* g) {
    if (g->pyment == -1) {
        g->pyment = 100;
        strcpy(g->name, "John");
        g->next = NULL;
    } else {
        while (g->next != NULL)
            g = g->next;
        g->next = (struct gammal*)malloc(sizeof(struct gammal));
        g = g->next;
        g->pyment = 100;
        strcpy(g->name, "John");
        g->next = NULL;
    }
}

void show(struct gammal* g) {
    while (g != NULL) {
        printf("-----\n");
        printf("%s ", g->name);
        printf("%d\n", g->pyment);
        g = g->next;
    }
}

int main() {
    struct gammal* head = (struct gammal*)malloc(sizeof(struct gammal));
    add(head);
    show(head);
    return 0;
}
```
