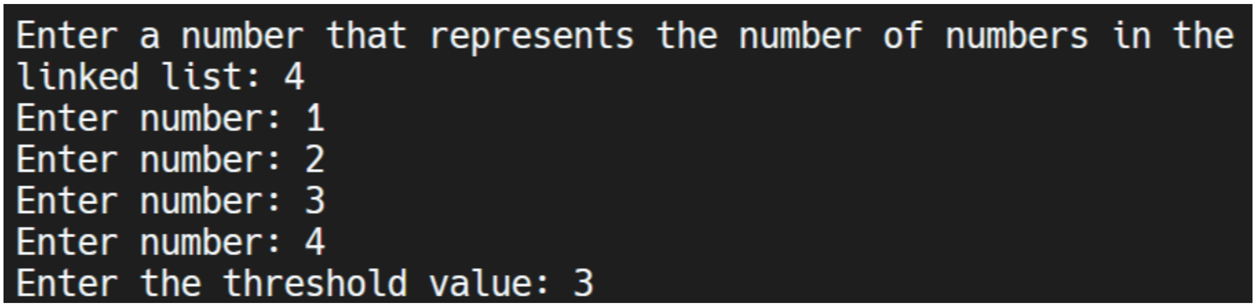
1- Write a program to find and print the product of all numbers less than a given value in the linked list using recursion.

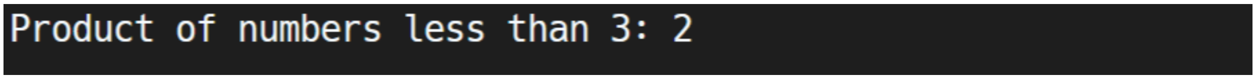
اكتب برنامجًا للعثور على ضرب جميع الأرقام الأقل من قيمة معينة في linked
list وطباعته .

Input



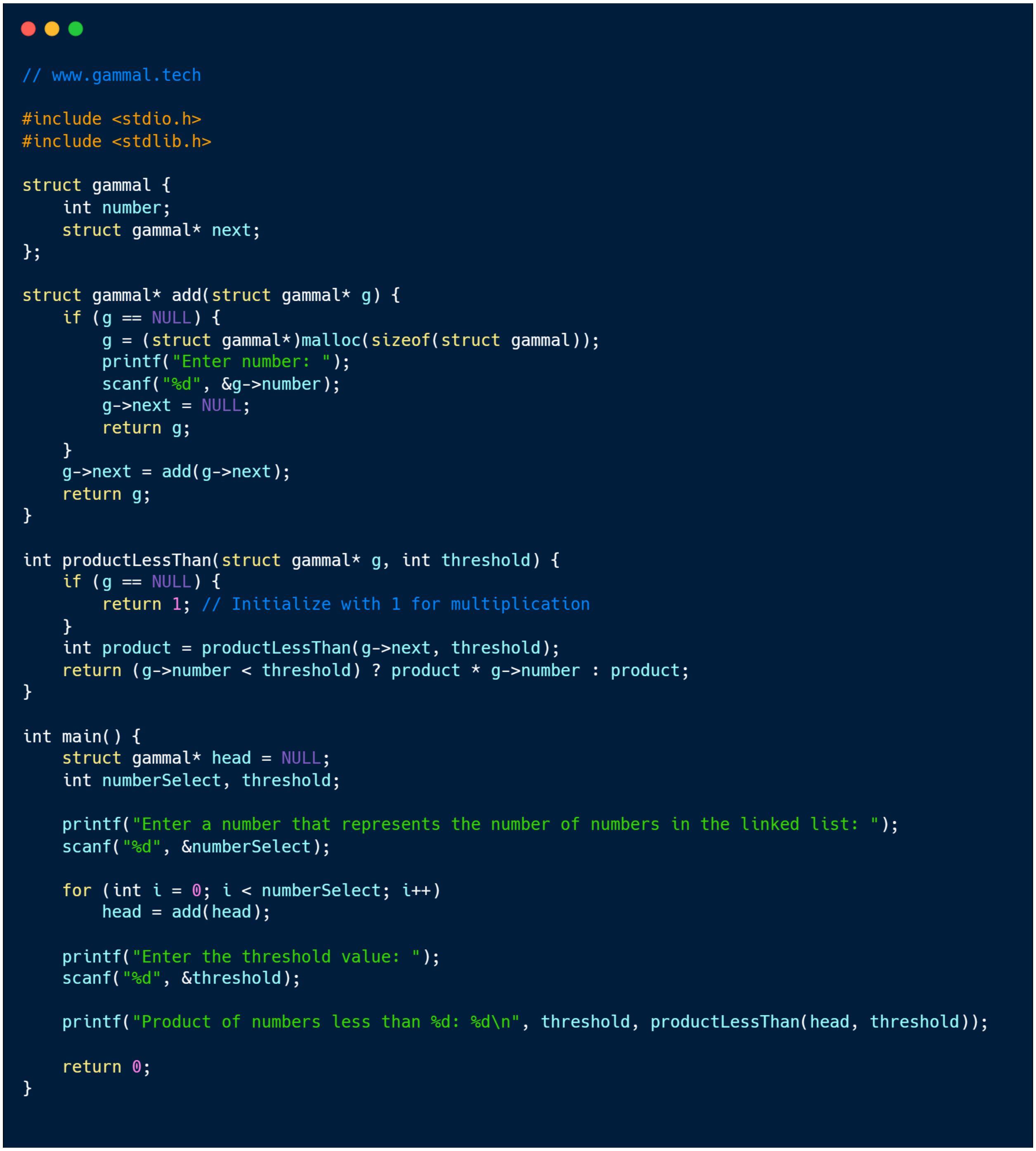
```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
Enter the threshold value: 3
```

Output



```
Product of numbers less than 3: 2
```

## Solution

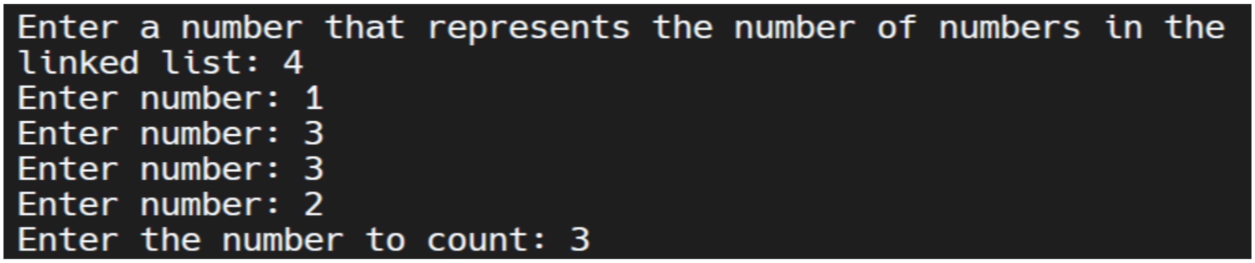


```c
// www.gammal.tech

#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 productLessThan(struct gammal* g, int threshold) {
    if (g == NULL) {
        return 1; // Initialize with 1 for multiplication
    }
    int product = productLessThan(g->next, threshold);
    return (g->number < threshold) ? product * g->number : product;
}

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

    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("Enter the threshold value: ");
    scanf("%d", &threshold);

    printf("Product of numbers less than %d: %d\n", threshold, productLessThan(head, threshold));

    return 0;
}
```

2- Write a program to find and print the count of numbers equal to a given value in the linked list using recursion.

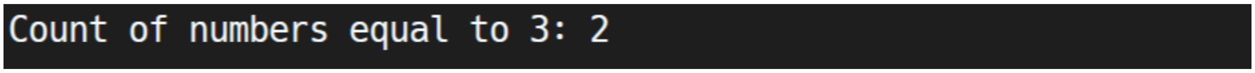
اكتب برنامجًا للعثور على عدد الأرقام المساوية لقيمة معينة في linked list
وطباعته .

## Input



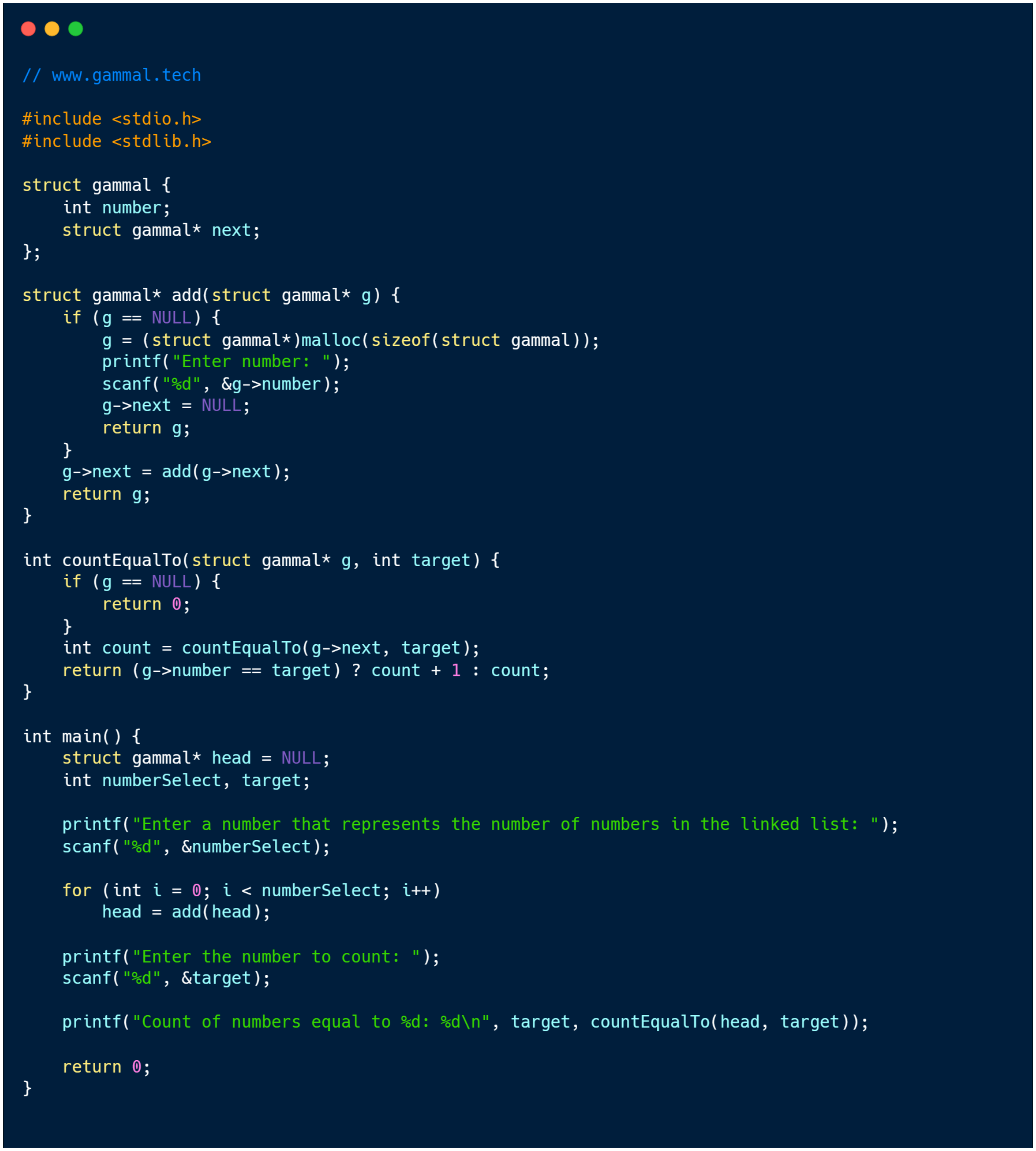
```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 1
Enter number: 3
Enter number: 3
Enter number: 2
Enter the number to count: 3
```

## Output



```
Count of numbers equal to 3: 2
```

## Solution

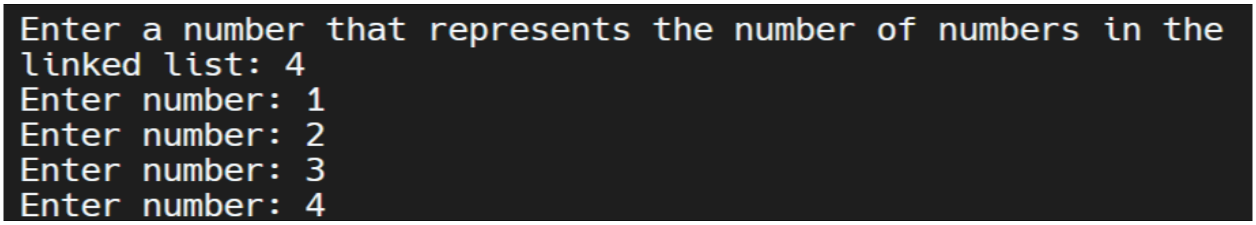


```c
// www.gammal.tech

#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 countEqualTo(struct gammal* g, int target) {
    if (g == NULL) {
        return 0;
    }
    int count = countEqualTo(g->next, target);
    return (g->number == target) ? count + 1 : count;
}

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

    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("Enter the number to count: ");
    scanf("%d", &target);

    printf("Count of numbers equal to %d: %d\n", target, countEqualTo(head, target));

    return 0;
}
```

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

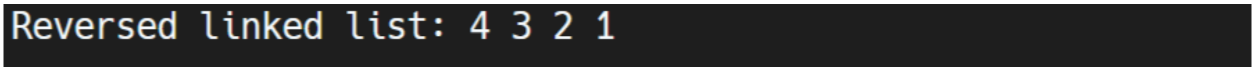
اكتب برنامجًا لعكس linked list وطباعتها .

## Input



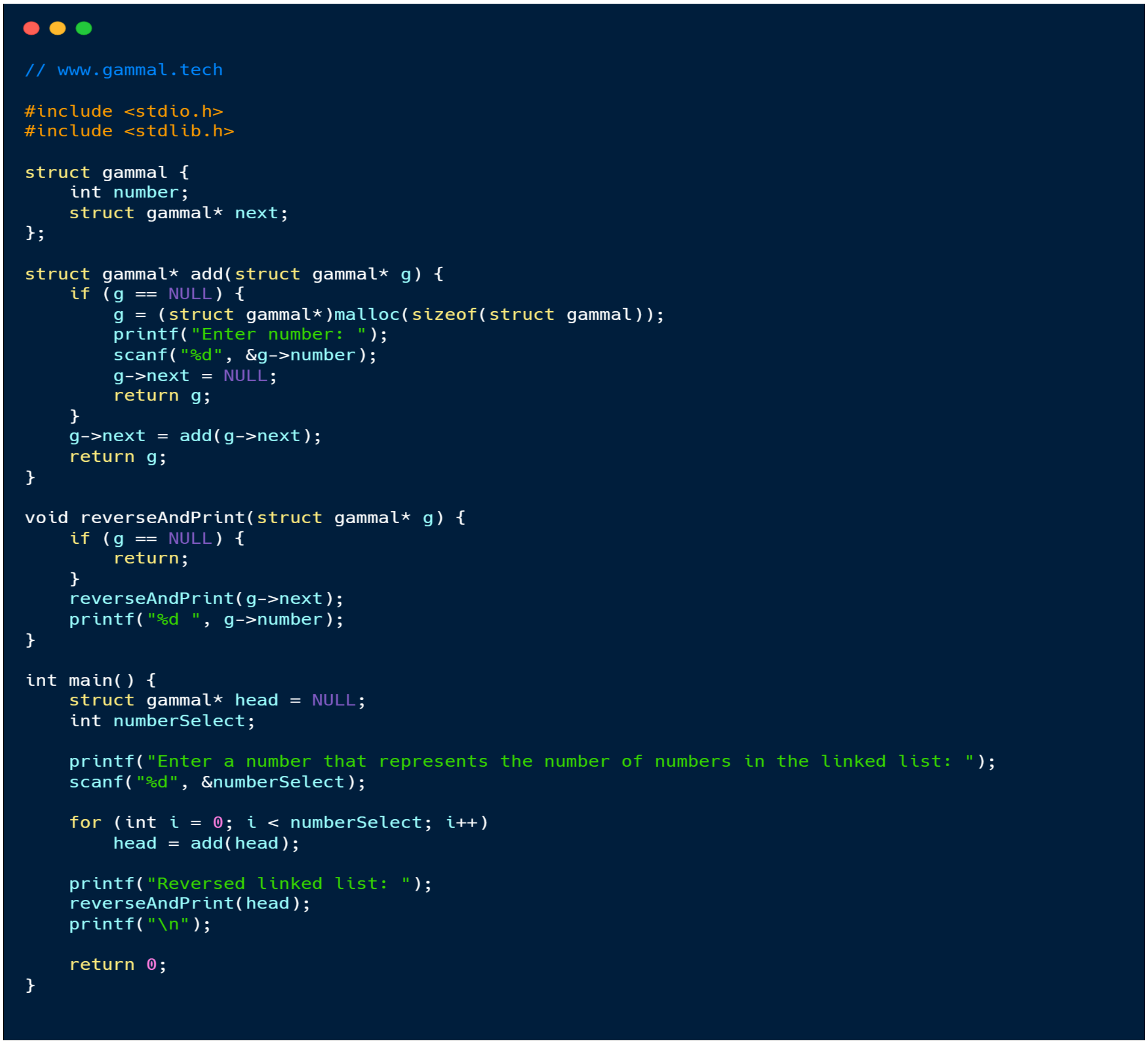
```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
```

## Output



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

## Solution

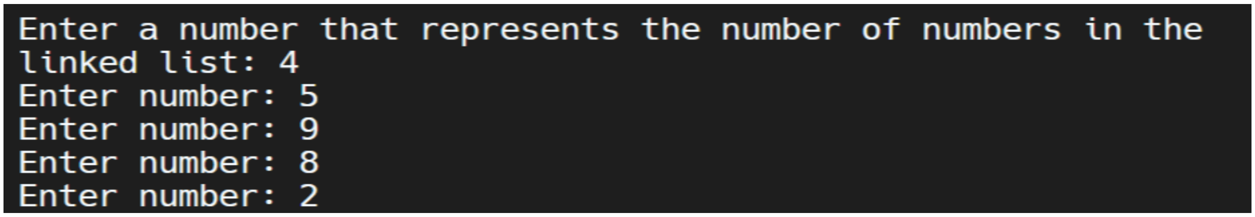


```c
// www.gammal.tech

#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;
}
```

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

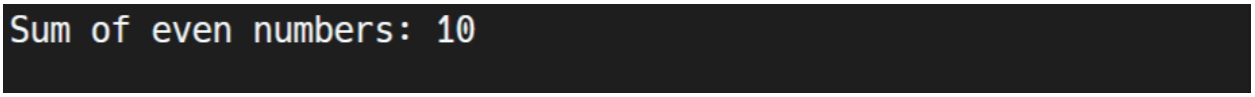
اكتب برنامجًا للعثور على مجموع الأعداد الزوجية في linked list وطباعته

### Input



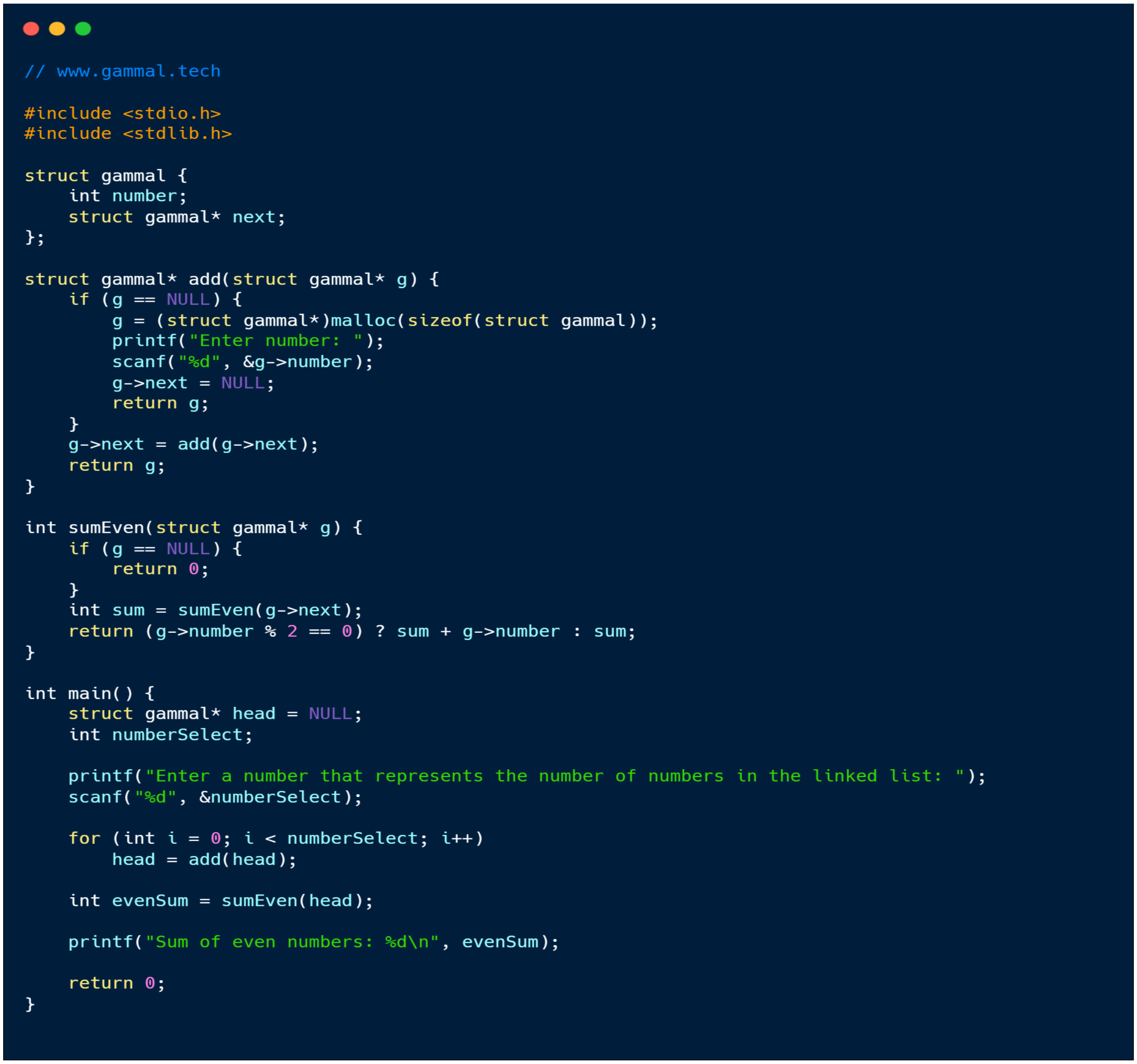
```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 5
Enter number: 9
Enter number: 8
Enter number: 2
```

### Output



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

### Solution

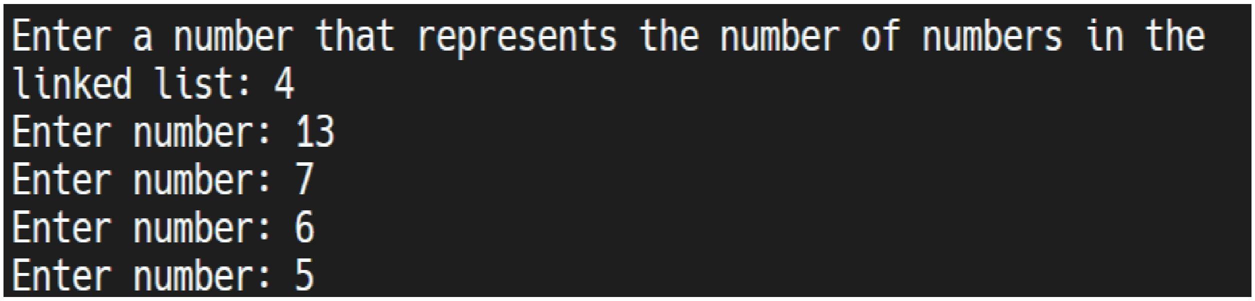


```c
// www.gammal.tech

#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;
}
```

5- Write a program to find and print the count of prime numbers in the linked list

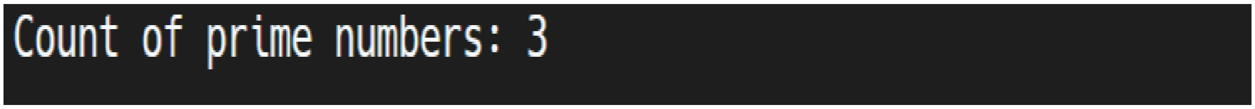
اكتب برنامجًا للعثور على عدد الأعداد الأولية في linked list وطباعته

Input



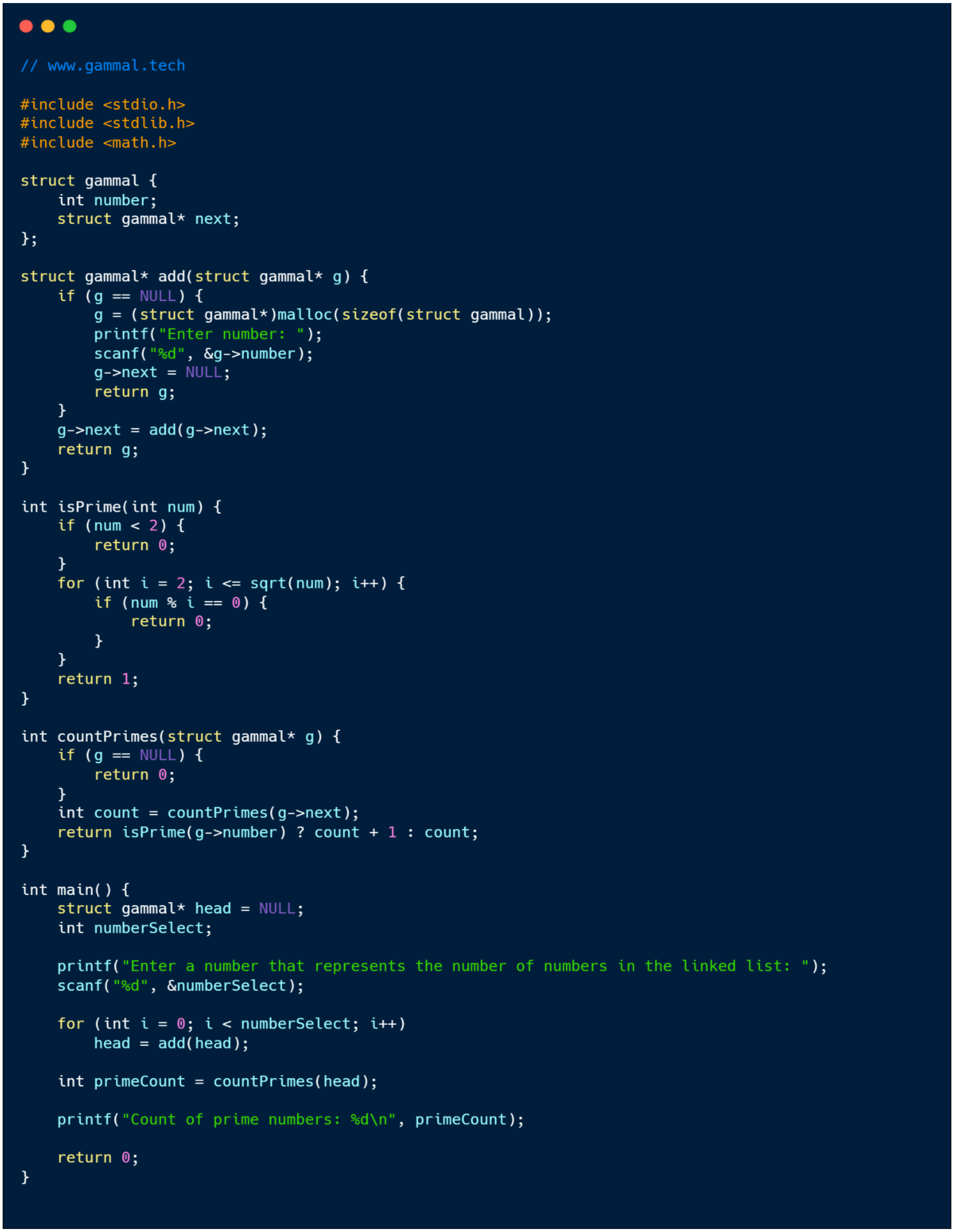
```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 13
Enter number: 7
Enter number: 6
Enter number: 5
```

Output



```
Count of prime numbers: 3
```

## Solution



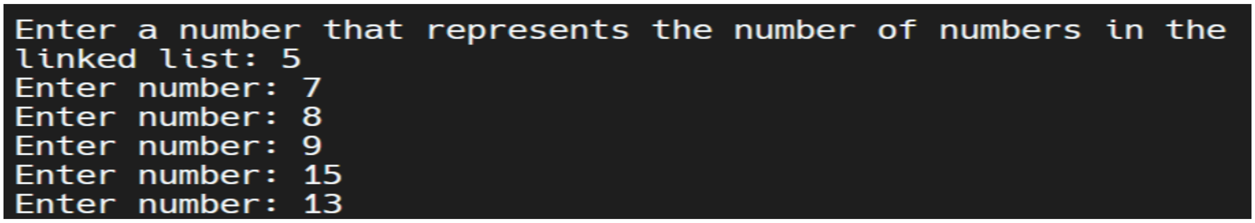
```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <math.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 isPrime(int num) {
    if (num < 2) {
        return 0;
    }
    for (int i = 2; i <= sqrt(num); i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}

int countPrimes(struct gammal* g) {
    if (g == NULL) {
        return 0;
    }
    int count = countPrimes(g->next);
    return isPrime(g->number) ? count + 1 : count;
}

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 primeCount = countPrimes(head);

    printf("Count of prime numbers: %d\n", primeCount);

    return 0;
}
```

---

6- Write a program to find and print the count of numbers divisible by 3 in the linked list

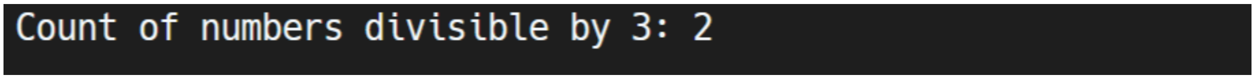
اكتب برنامجًا للعثور على عدد الأعداد القابلة للقسمة على 3 وطباعته في linked list

## Input



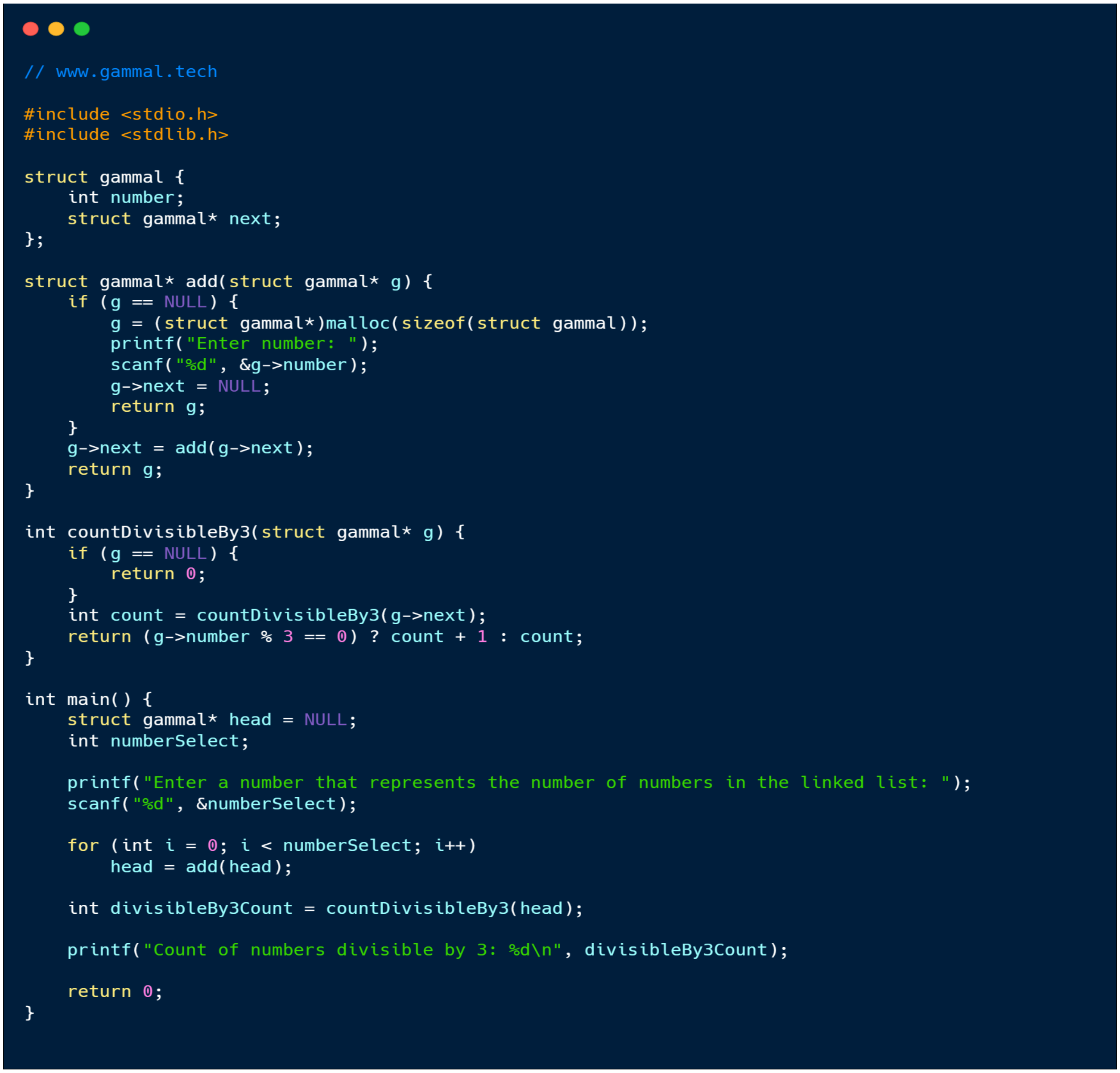
```
Enter a number that represents the number of numbers in the
linked list: 5
Enter number: 7
Enter number: 8
Enter number: 9
Enter number: 15
Enter number: 13
```

## Output



```
Count of numbers divisible by 3: 2
```

## Solution

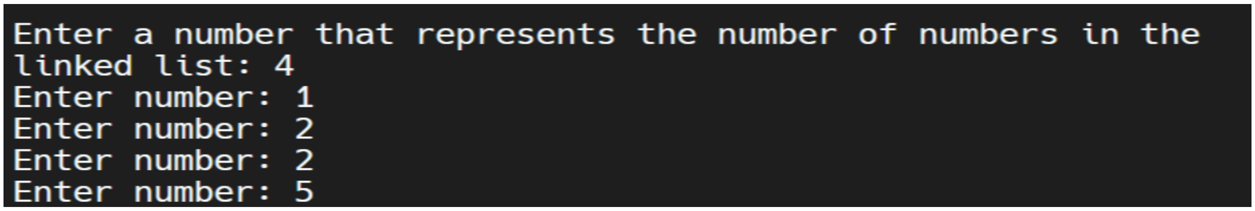


```c
// www.gammal.tech

#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 countDivisibleBy3(struct gammal* g) {
    if (g == NULL) {
        return 0;
    }
    int count = countDivisibleBy3(g->next);
    return (g->number % 3 == 0) ? count + 1 : count;
}

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 divisibleBy3Count = countDivisibleBy3(head);

    printf("Count of numbers divisible by 3: %d\n", divisibleBy3Count);

    return 0;
}
```

## 7- Write a program to find and print the sum of all odd numbers in the linked list

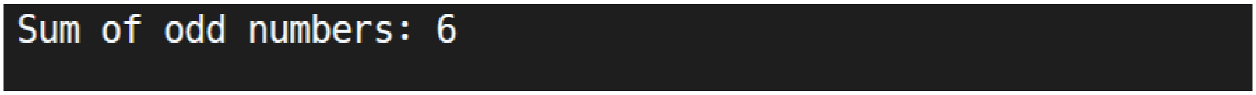
اكتب برنامجًا للعثور على مجموع الأعداد الفردية في linked list وطباعته

## Input



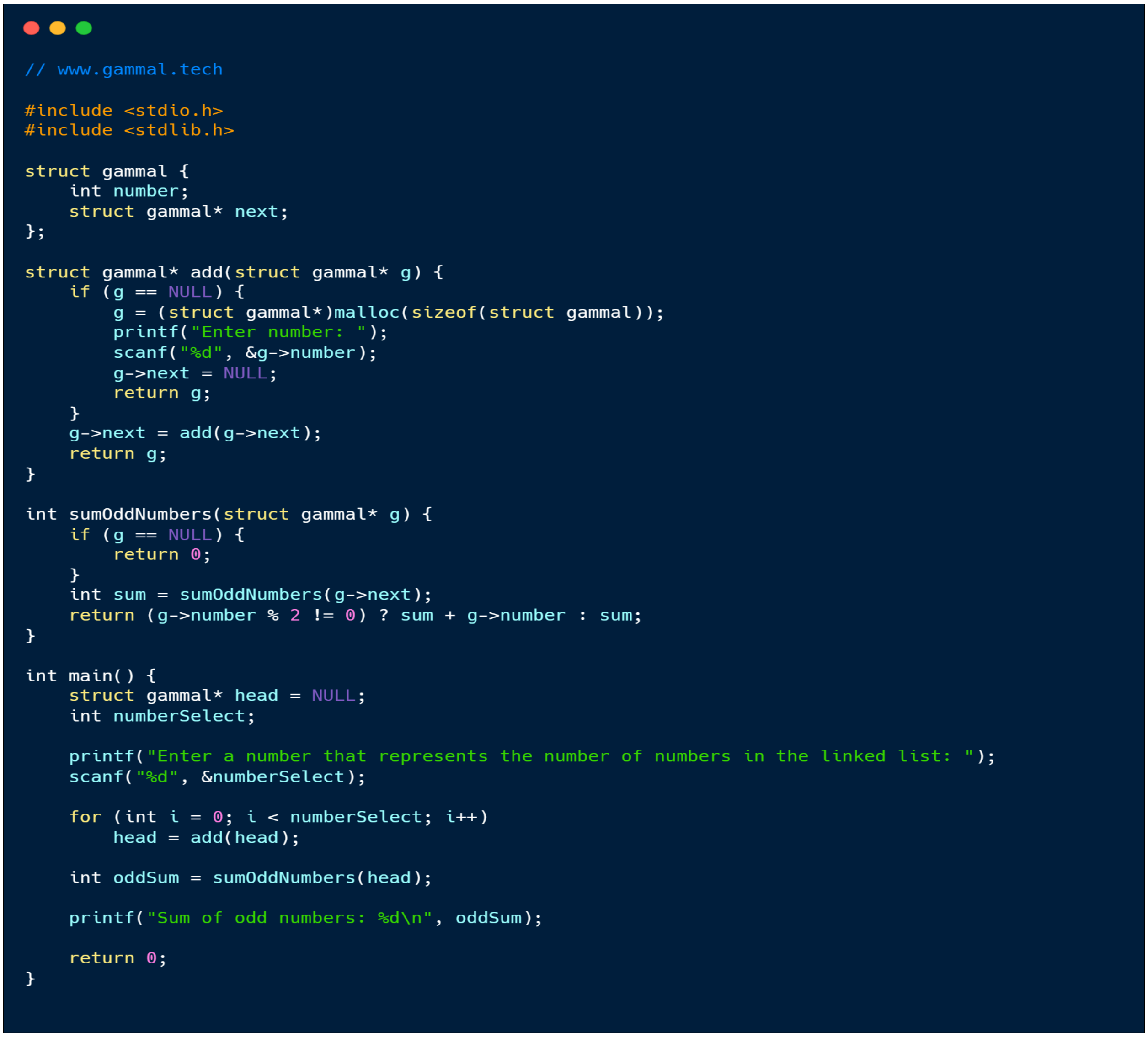
```
Enter a number that represents the number of numbers in the
linked list: 4
Enter number: 1
Enter number: 2
Enter number: 2
Enter number: 5
```

## Output



```
Sum of odd numbers: 6
```

## Solution

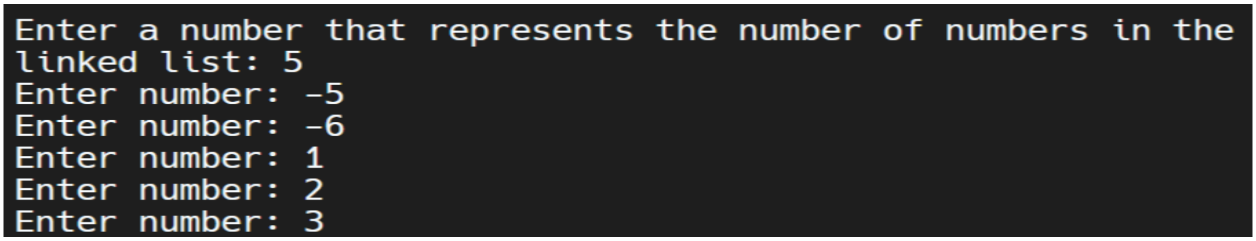


```c
// www.gammal.tech

#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 sumOddNumbers(struct gammal* g) {
    if (g == NULL) {
        return 0;
    }
    int sum = sumOddNumbers(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 oddSum = sumOddNumbers(head);

    printf("Sum of odd numbers: %d\n", oddSum);

    return 0;
}
```

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

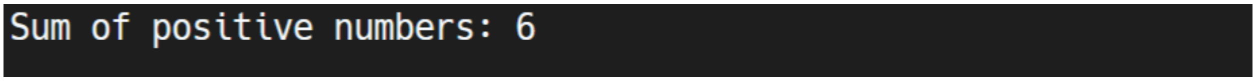
اكتب برنامجًا للعثور على مجموع جميع الأرقام الموجبة في linked list
وطباعتها

## Input



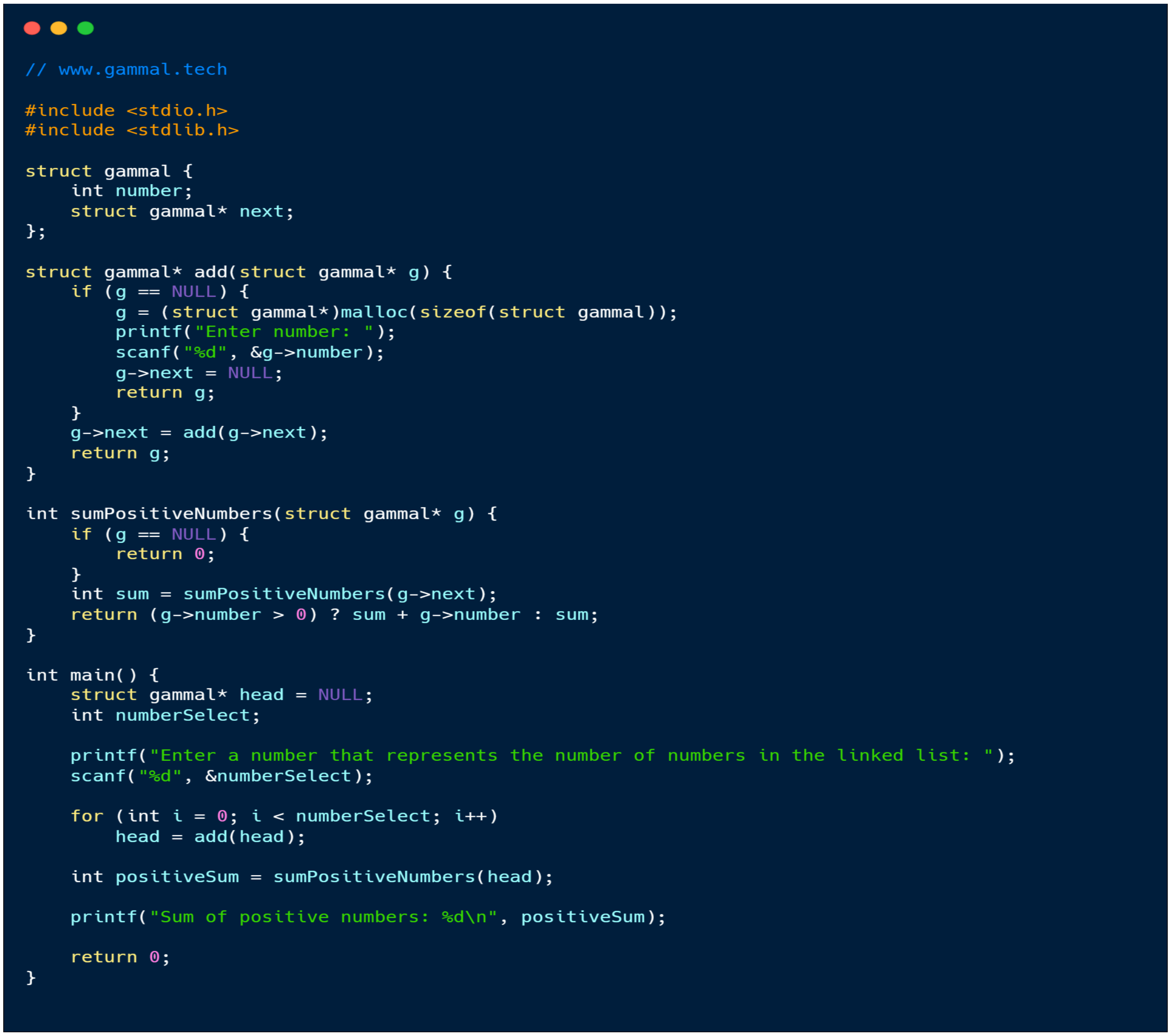
```
Enter a number that represents the number of numbers in the
linked list: 5
Enter number: -5
Enter number: -6
Enter number: 1
Enter number: 2
Enter number: 3
```

## Output



```
Sum of positive numbers: 6
```

## Solution

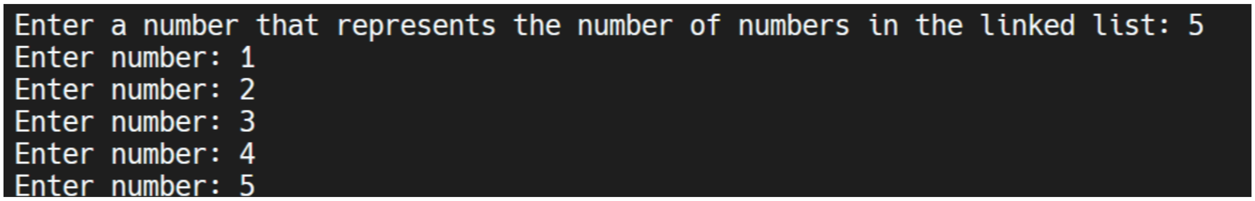


```c
// www.gammal.tech

#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 sumPositiveNumbers(struct gammal* g) {
    if (g == NULL) {
        return 0;
    }
    int sum = sumPositiveNumbers(g->next);
    return (g->number > 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 positiveSum = sumPositiveNumbers(head);

    printf("Sum of positive numbers: %d\n", positiveSum);

    return 0;
}
```

9- Write a program to find and print the difference between the sum of even numbers and the sum of odd numbers in the linked list

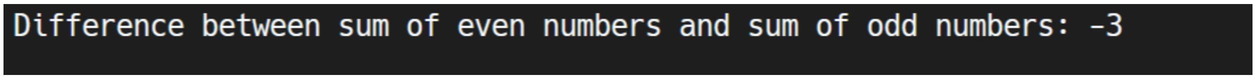
اكتب برنامجًا لإيجاد وطباعة الفرق بين مجموع الأعداد الزوجية ومجموع الأعداد الفردية في linked list

Input



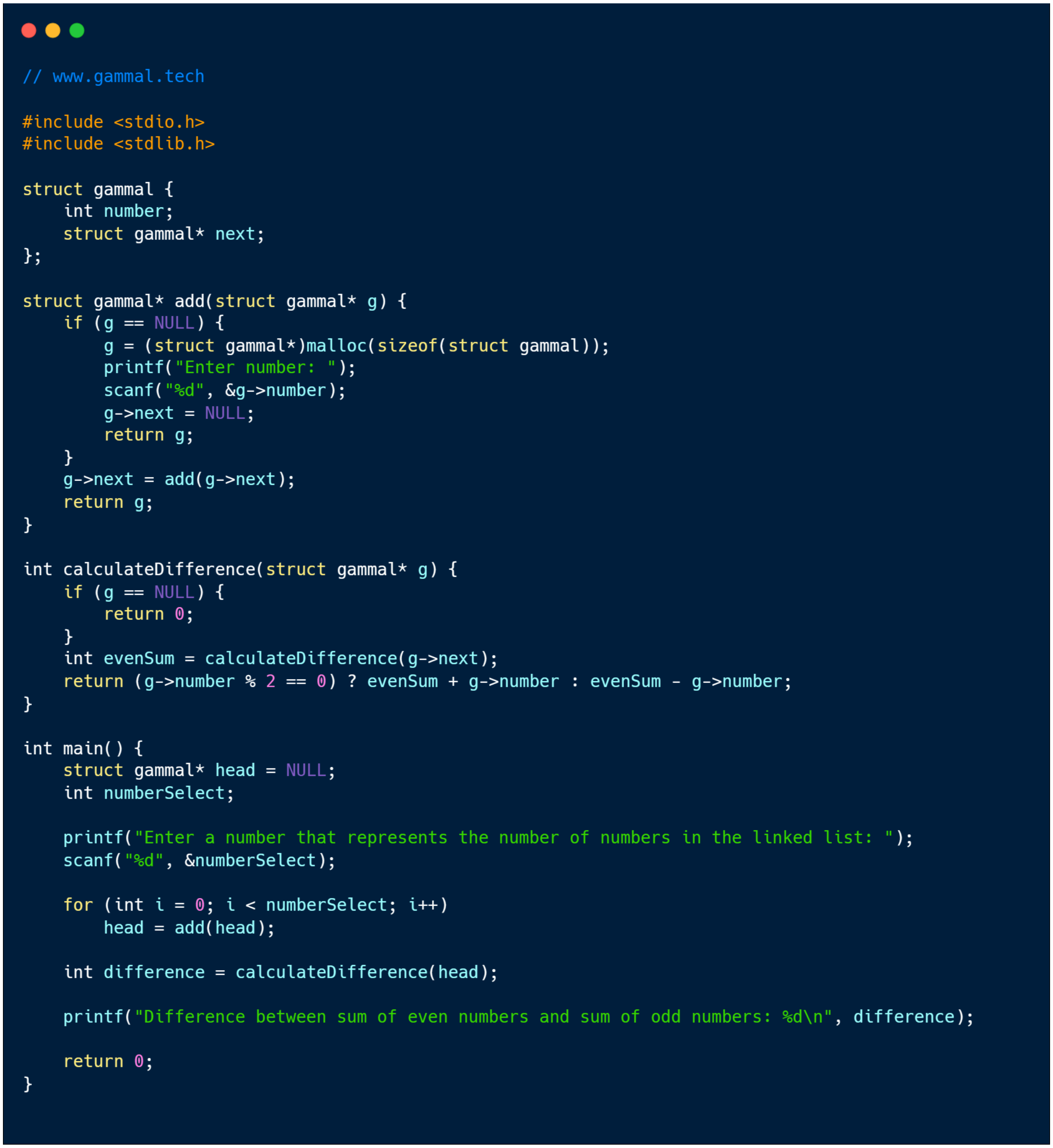
```
Enter a number that represents the number of numbers in the linked list: 5
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
Enter number: 5
```

Output



```
Difference between sum of even numbers and sum of odd numbers: -3
```

Solution



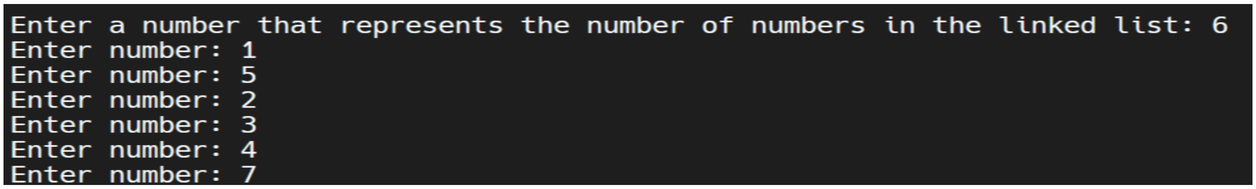
```c
// www.gammal.tech

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

    int difference = calculateDifference(head);

    printf("Difference between sum of even numbers and sum of odd numbers: %d\n", difference);

    return 0;
}
```

---

## 10- Write a program to find and print the minimum and maximum numbers in the linked list

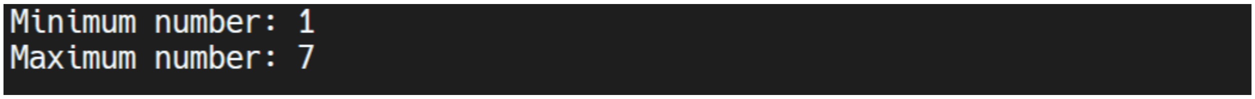
اكتب برنامجًا للعثور على الحد الأدنى والحد الأقصى للأرقام في linked list
وطباعتها

## Input



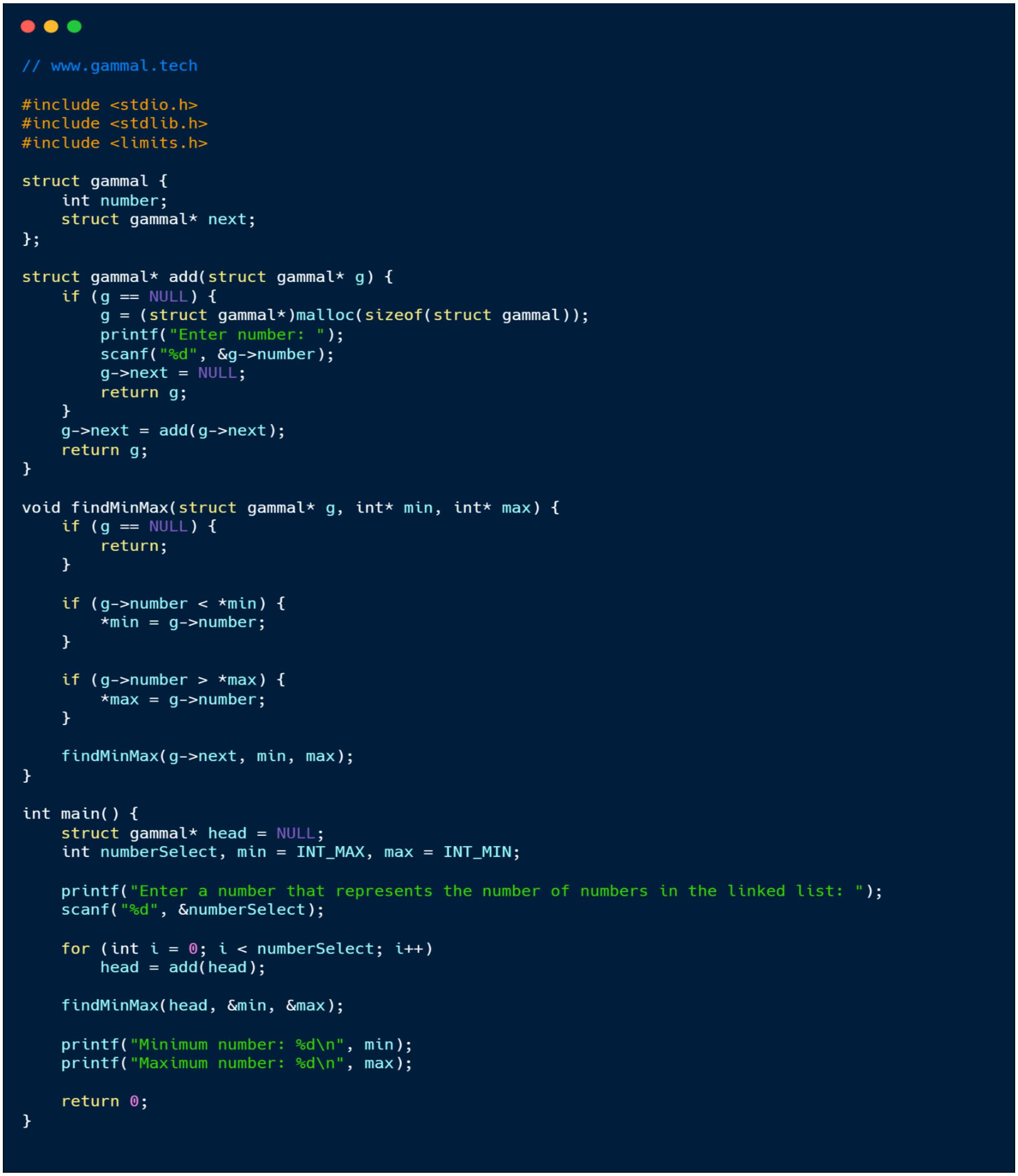
```
Enter a number that represents the number of numbers in the linked list: 6
Enter number: 1
Enter number: 5
Enter number: 2
Enter number: 3
Enter number: 4
Enter number: 7
```

## Output



```
Minimum number: 1
Maximum number: 7
```

## Solution



```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <limits.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 findMinMax(struct gammal* g, int* min, int* max) {
    if (g == NULL) {
        return;
    }

    if (g->number < *min) {
        *min = g->number;
    }

    if (g->number > *max) {
        *max = g->number;
    }

    findMinMax(g->next, min, max);
}

int main() {
    struct gammal* head = NULL;
    int numberSelect, min = INT_MAX, max = INT_MIN;

    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);

    findMinMax(head, &min, &max);

    printf("Minimum number: %d\n", min);
    printf("Maximum number: %d\n", max);

    return 0;
}
```