1- Write a program that takes an integer input 'num' from the user and prints the numbers from 'num' to -2 in decreasing order, excluding zero. using recursion

اكتب برنامجًا يأخذ عددًا صحيحًا "num" من المستخدم ويطبع الأرقام من "num" إلى -2 بترتيب تنازلي، باستثناء الصفر. using recursion

Input

```
Enter a number: 3
```

Output

```
Numbers from 3 to -2 (excluding zero): 3 2 1 -1 -2
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to print numbers from n to -2 excluding zero recursively
void printNumbers(int n) {
    if (n > -3) {
        if (n != 0)
            cout << n << " ";
        printNumbers(n - 1);
    }
}

int main() {
    int num;

    // Input: Get an integer from the user
    cout << "Enter a number: ";
    cin >> num;

    // Output: Display the numbers from N to -2 excluding zero
    cout << "Numbers from " << num << " to -2 (excluding zero): ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

2- Create a program that prompts the user to enter an integer 'num' and then prints the even numbers from 'num' to 1 in decreasing order. using recursion

قم بإنشاء برنامج يطلب من المستخدم إدخال عدد صحيح "num" ثم طباعة الأرقام الزوجية من "num" إلى 1 بترتيب تنازلي.using recursion

Input

```
Enter a number: 7
```

Output

```
Even numbers from 7 to 1: 6 4 2
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to print even numbers from n to 1 recursively
void printNumbers(int n) {
    if (n > 0) {
        if (n % 2 == 0)
            cout << n << " ";
        printNumbers(n - 1);
    }
}

int main() {
    int num;

    // Input: Get an integer from the user
    cout << "Enter a number: ";
    cin >> num;

    // Output: Display even numbers from N to 1
    cout << "Even numbers from " << num << " to 1: ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

3- Write a program that takes an integer input 'num' from the user and prints the numbers from -2 to 'num' in ascending order. using recursion

اكتب برنامجًا يأخذ عددًا صحيحًا "num" من المستخدم ويطبع الأرقام من -2 إلى "num" بترتيب تصاعدي. using recursion

Input

```
Enter a number: 3
```

Output

```
Numbers from -2 to 3: -2 -1 0 1 2 3
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to print numbers from -2 to n recursively
void printNumbers(int n) {
    if (n > -3) {
        printNumbers(n - 1);
        cout << n << " ";
    }
}

int main() {
    int num;

    // Input: Get an integer from the user
    cout << "Enter a number: ";
    cin >> num;

    // Output: Display numbers from -2 to N
    cout << "Numbers from -2 to " << num << ": ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

4- Write a program that prompts the user to enter a positive integer 'num'. The program should count and display the number of odd digits in 'num'

اكتب برنامجًا يطلب من المستخدم إدخال عدد صحيح موجب "num". يجب أن يقوم البرنامج بحساب وعرض عدد الأرقام الفردية في "num"

Input

```
Enter a positive integer: 25635
```

Output

```
Number of odd digits: 3
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to count the number of odd digits in a positive integer recursively
int countOddDigits(int n) {
    if (n == 0)
        return 0;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is odd
        if (lastDigit % 2 != 0)
            return 1 + countOddDigits(n / 10);
        else
            return countOddDigits(n / 10);
    }
}

int main() {
    int num;

    // Input: Get a positive integer from the user
    cout << "Enter a positive integer: ";
    cin >> num;

    // Output: Display the number of odd digits
    cout << "Number of odd digits: " << countOddDigits(num) << endl;

    return 0;
}
```

5- Write a program that prompts the user to enter a positive integer 'num'. The program should count and display the number of odd digits in 'num' using a recursive function named countOddDigits. Additionally, the program should calculate and display the sum of odd digits in 'num' using a recursive function named sumOddDigits.

اكتب برنامجًا يطلب من المستخدم إدخال عدد صحيح موجب "num". يجب أن يقوم البرنامج بحساب وعرض عدد الأرقام الفردية في "num" باستخدام دالة متكررة تسمى countOddDigits. بالإضافة إلى ذلك، يجب على البرنامج حساب وعرض مجموع الأرقام الفردية في "num" باستخدام دالة متكررة تسمى sumOddDigits.

Input

```
Enter a positive integer: 216545
```

Output

```
Number of odd digits: 3
Sum of odd digits: 11
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to count the number of odd digits in a positive integer recursively
int countOddDigits(int n) {
    if (n == 0)
        return 0;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is odd
        if (lastDigit % 2 != 0)
            return 1 + countOddDigits(n / 10);
        else
            return countOddDigits(n / 10);
    }
}

// Function to calculate the sum of odd digits in a positive integer recursively
int sumOddDigits(int n) {
    if (n == 0)
        return 0;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is odd
        if (lastDigit % 2 != 0)
            return lastDigit + sumOddDigits(n / 10);
        else
            return sumOddDigits(n / 10);
    }
}

int main() {
    int num;

    // Input: Get a positive integer from the user
    cout << "Enter a positive integer: ";
    cin >> num;

    // Output: Display the number of odd digits
    cout << "Number of odd digits: " << countOddDigits(num) << endl;

    // Output: Display the sum of odd digits
    cout << "Sum of odd digits: " << sumOddDigits(num) << endl;

    return 0;
}
```

6- Write a program that prompts the user to enter a positive integer 'num'. The program should count and display the number of odd digits in 'num' using a recursive function named countOddDigits. Additionally, the program should calculate and display the sum of odd digits in 'num' using a recursive function named sumOddDigits, and find the product of odd digits using a recursive function named productOddDigits.

اكتب برنامجًا يطلب من المستخدم إدخال عدد صحيح موجب "num". يجب أن يقوم البرنامج بحساب وعرض عدد الأرقام الفردية في "num" باستخدام دالة متكررة تسمى countOddDigits. بالإضافة إلى ذلك، يجب على البرنامج حساب وعرض مجموع الأرقام الفردية في "num" باستخدام دالة recursive تسمى sumOddDigits، والعثور على product الأرقام الفردية باستخدام دالة recursive تسمى ProductOddDigits.

Input

```
Enter a positive integer: 12345
```

Output

```
Number of odd digits: 3
Sum of odd digits: 9
Product of odd digits: 15
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to count the number of odd digits in a positive integer recursively
int countOddDigits(int n) {
    if (n == 0)
        return 0;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is odd
        if (lastDigit % 2 != 0)
            return 1 + countOddDigits(n / 10);
        else
            return countOddDigits(n / 10);
    }
}

// Function to calculate the sum of odd digits in a positive integer recursively
int sumOddDigits(int n) {
    if (n == 0)
        return 0;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is odd
        if (lastDigit % 2 != 0)
            return lastDigit + sumOddDigits(n / 10);
        else
            return sumOddDigits(n / 10);
    }
}

// Function to calculate the product of odd digits in a positive integer recursively
int productOddDigits(int n) {
    if (n == 0)
        return 1;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is odd
        if (lastDigit % 2 != 0)
            return lastDigit * productOddDigits(n / 10);
        else
            return productOddDigits(n / 10);
    }
}

int main() {
    int num;

    // Input: Get a positive integer from the user
    cout << "Enter a positive integer: ";
    cin >> num;

    // Output: Display the number of odd digits
    cout << "Number of odd digits: " << countOddDigits(num) << endl;

    // Output: Display the sum of odd digits
    cout << "Sum of odd digits: " << sumOddDigits(num) << endl;

    // Output: Display the product of odd digits
    cout << "Product of odd digits: " << productOddDigits(num) << endl;

    return 0;
}
```

7- Write a program to find and print the length (number of nodes) of the linked list.

اكتب برنامجًا للعثور على طول (عدد node) لlinked list وطباعتها.

Output

```
Length of the linked list: 3
```

Solution

```c
// www.gammal.tech

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

struct node {
    int data;
    struct node* next;
};

int main() {
    struct node *head, *temp;
    head = (struct node*)malloc(sizeof(struct node));
    head->data = 2;
    head->next = (struct node*)malloc(sizeof(struct node));
    head->next->data = 3;
    head->next->next = (struct node*)malloc(sizeof(struct node));
    head->next->next->data = 4;
    head->next->next->next = NULL;

    // Find the length of the linked list
    int length = 0;
    temp = head;
    while (temp != NULL) {
        length++;
        temp = temp->next;
    }

    // Print the length
    printf("Length of the linked list: %d\n", length);

    // Free allocated memory
    temp = head;
    while (temp != NULL) {
        struct node* nextNode = temp->next;
        free(temp);
        temp = nextNode;
    }

    return 0;
}
```

8- Write a program to check if the linked list contains a node with data value 5. Print "Found" if it exists, otherwise print "Not Found".

اكتب برنامجًا للتحقق مما إذا كانت linked list تحتوي على nodeبقيمة البيانات 5. اطبع "تم العثور عليه" إذا كان موجودًا، وإلا فاطبع "لم يتم العثور عليه".

Output

```
Not Found
```

Solution

```c
// www.gammal.tech

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

struct node {
    int data;
    struct node* next;
};

int main() {
    struct node *head, *temp;
    head = (struct node*)malloc(sizeof(struct node));
    head->data = 2;
    head->next = (struct node*)malloc(sizeof(struct node));
    head->next->data = 3;
    head->next->next = (struct node*)malloc(sizeof(struct node));
    head->next->next->data = 4;
    head->next->next->next = NULL;

    // Check if the linked list contains a node with data value 5
    int found = 0;
    temp = head;
    while (temp != NULL) {
        if (temp->data == 5) {
            found = 1;
            break;
        }
        temp = temp->next;
    }

    // Print the result
    if (found) {
        printf("Found\n");
    } else {
        printf("Not Found\n");
    }

    // Free allocated memory
    temp = head;
    while (temp != NULL) {
        struct node* nextNode = temp->next;
        free(temp);
        temp = nextNode;
    }

    return 0;
}
```

9- 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

```c
// 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;
}
```

10- 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

```c
// 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;
}
```