

Assignn_28

```
#include <stdio.h>
```

```
// Structure to represent a complex number
```

```
typedef struct {
```

```
    float real;
```

```
    float imag;
```

```
} Complex;
```

```
// Function to add two complex numbers
```

```
Complex addComplex(Complex c1, Complex c2) {
```

```
    Complex result;
```

```
    result.real = c1.real + c2.real;
```

```
    result.imag = c1.imag + c2.imag;
```

```
    return result;
```

```
}
```

```
// Function to subtract two complex numbers
```

```
Complex subtractComplex(Complex c1, Complex c2) {
```

```
    Complex result;
```

```
    result.real = c1.real - c2.real;
```

```
    result.imag = c1.imag - c2.imag;
```

```
    return result;
```

```
}
```

```
int main() {
```

```
    Complex c1, c2, result;
```

```
    int choice;
```

```
    // Input two complex numbers from the user
```

```
    printf("Enter the first complex number (real and imaginary parts): ");
```

```
    scanf("%f %f", &c1.real, &c1.imag);
```

```
    printf("Enter the second complex number (real and imaginary parts): ");
```

```
    scanf("%f %f", &c2.real, &c2.imag);
```

```
    // Menu-driven loop
```

```
    do {
```

```
        printf("\nMenu:\n");
```

```
        printf("1. Add\n");
```

```
        printf("2. Subtract\n");
```

```
        printf("3. Exit\n");
```

```
        printf("Enter your choice: ");
```

```
        scanf("%d", &choice);
```

```

switch (choice) {
    case 1:
        result = addComplex(c1, c2);
        printf("Result of addition: %.2f + %.2fi\n", result.real, result.imag);
        break;
    case 2:
        result = subtractComplex(c1, c2);
        printf("Result of subtraction: %.2f + %.2fi\n", result.real, result.imag);
        break;
    case 3:
        printf("Exiting the program.\n");
        break;
    default:
        printf("Invalid choice. Please try again.\n");
}
} while (choice != 3);

return 0;
}

```

Assign_29

```

#include <stdio.h>

// Function to check if a number is even
int isEven(int num) {
    // Check if the number is divisible by 2
    if (num % 2 == 0) {
        return 1; // Return 1 if even
    } else {
        return 0; // Return 0 if odd
    }
}

int main() {
    int number;

    // Accept a number from the user
    printf("Enter an integer: ");
    scanf("%d", &number);

```

```

// Check if the number is even or odd using isEven function
if (isEven(number)) {
    printf("%d is even.\n", number);
} else {
    printf("%d is odd.\n", number);
}

return 0;
}

```

Assign_30

```

#include <stdio.h>

// Function to display the next n characters
void displayNextNChars(char ch, int n) {
    printf("Next %d characters from '%c':\n", n, ch);
    for (int i = 1; i <= n; i++) {
        printf("%c ", ch + i); // Display the next character
    }
    printf("\n");
}

int main() {
    char character;
    int n;

    // Accept a character from the user
    printf("Enter a character: ");
    scanf(" %c", &character);

    // Accept the number of next characters to display
    printf("Enter the number of next characters to display: ");
    scanf("%d", &n);

    // Call the function to display the next n characters
    displayNextNChars(character, n);

    return 0;
}

```

Power of X to Y

```

#include <stdio.h>
int main() {
    int base, exp;
    long double result = 1.0;
    printf("Enter a base number: ");
    scanf("%d", &base);
    printf("Enter an exponent: ");
    scanf("%d", &exp);

    while (exp != 0) {
        result *= base;
        --exp;
    }
    printf("Answer = %.0Lf", result);
    return 0;
}

```

4. C Program to Check Whether a Number is Prime or Not

```

#include <stdio.h>

int main() {

    int num, i, flag = 0;

    printf("Enter a number: ");

    scanf("%d", &num);

    for (i = 2; i <= num / 2; ++i) {

        if (num % i == 0) {

            flag = 1;

            break;

        }

    }

    if (flag == 0)

        printf("%d is a prime number.\n", num);

```

```
else

printf("%d is not a prime number.\n", num);

return 0;

}
```

Sum of n numbers using recursion:

```
#include <stdio.h>

int addNumbers(int n);

int main() {

    int num;
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    printf("Sum = %d", addNumbers(num));
    return 0;
}

int addNumbers(int n) {
    if (n != 0)
        return n + addNumbers(n - 1);
    else
        return n;
}
```

Reverse the number:

```
#include <stdio.h>

int reverseNumber(int num) {
    int reversed = 0;
    while (num != 0) {
        reversed = reversed * 10 + num % 10;
        num /= 10;
    }
    return reversed;
}
```

```

int main() {
    int num;

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

    int reversed = reverseNumber(num);

    printf("Reversed number: %d\n", reversed);

    return 0;
}

```

Fibonacci series:

```

#include <stdio.h>
int main() {

    int i, n;

    // initialize first and second terms
    int t1 = 0, t2 = 1;

    // initialize the next term (3rd term)
    int nextTerm = t1 + t2;

    // get no. of terms from user
    printf("Enter the number of terms: ");
    scanf("%d", &n);

    // print the first two terms t1 and t2
    printf("Fibonacci Series: %d, %d, ", t1, t2);

    // print 3rd to nth terms
    for (i = 3; i <= n; ++i) {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }

    return 0;
}

```

// Program to check for leap year using ternary operator

```
#include<stdio.h>
int main()
{
    int y;
    printf("enter year for checking leap year or not\n");
    scanf("%d",&y);
    //Using ternary operator
    (((y%100!=0)&&(y%4==0))||(y%400==0))?printf("%d is leap
year",y):printf("%d is not a leap year",y);
    return 0;
}
```

//Write a recursive function to calculate the sum of first n natural numbers.

```
#include<stdio.h>

// Function prototype
int sumOfNaturalNumbers(int n);

// Function definition
int sumOfNaturalNumbers(int n) {
    if (n == 0) {
        return 0;
    } else {
        // Recursive call to calculate the sum
        return n + sumOfNaturalNumbers(n - 1);
    }
}

int main() {
    int n;

    // Input the value of n
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Check if the input is non-negative
    if (n < 0) {
        printf("Invalid input. Please enter a non-negative integer.\n");
    }
}
```

```

        return 1; // Exit with an error code
    }

    // Call the recursive function to calculate the sum
    int result = sumOfNaturalNumbers(n);

    // Display the result
    printf("The sum of the first %d natural numbers is: %d\n", n, result);

    return 0;
}

```

Max in array

```

#include <stdio.h>

int findMax(int arr[], int n) {
    int max = arr[0]; // Initialize max to the first element

    for (int i = 1; i < n; i++) {
        if (arr[i] > max) {
            max = arr[i]; // Update max if current element is greater
        }
    }

    return max;
}

int main() {
    int n;

    // Input array size
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input array elements
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Find and display the maximum element
    int max = findMax(arr, n);
    printf("The maximum element in the array is: %d\n", max);
}

```



```

        return 0;
    }

#include <stdio.h>

void findMaxMin(int arr[], int n, int *max, int *min) {
    *max = arr[0];
    *min = arr[0];

    for (int i = 1; i < n; i++) {
        if (arr[i] > *max) {
            *max = arr[i];
        }
        if (arr[i] < *min) {
            *min = arr[i];
        }
    }
}

int main() {
    int n;

    // Input array size
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input array elements
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int max, min;

    // Function to find max and min
    findMaxMin(arr, n, &max, &min);

    // Output results
    printf("Maximum element: %d\n", max);
    printf("Minimum element: %d\n", min);

    return 0;
}

```

// Online C compiler to run C program online

write a c program by taking 3d matrix from user and making spiral printing it

```
#include <stdio.h>

void spiralPrint(int mat[][10], int R, int C) {
    int top = 0, bottom = R - 1, left = 0, right = C - 1;

    while (top <= bottom && left <= right) {
        // Print the top row from left to right
        for (int i = left; i <= right; i++)
            printf("%d ", mat[top][i]);
        top++;

        // Print the right column from top to bottom
        for (int i = top; i <= bottom; i++)
            printf("%d ", mat[i][right]);
        right--;

        // Print the bottom row from right to left
        if (top <= bottom) {
            for (int i = right; i >= left; i--)
                printf("%d ", mat[bottom][i]);
            bottom--;
        }

        // Print the left column from bottom to top
        if (left <= right) {
            for (int i = bottom; i >= top; i--)
                printf("%d ", mat[i][left]);
            left++;
        }
    }
}

int main() {
    int mat[10][10], R, C;

    printf("Enter the number of rows and columns: ");
    scanf("%d %d", &R, &C);

    printf("Enter the elements of the matrix:\n");
```

```
    for (int i = 0; i < R; i++) {
        for (int j = 0; j < C; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    printf("Spiral order printing of the matrix:\n");
    spiralPrint(mat, R, C);

    return 0;
}
```