main.c

Share Run

```c
#include <stdio.h>

int main() {
    int n = 5;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= n - i; ++j) {
            printf(" ");
        }
        for (int k = n - i + 1; k <= n; ++k) {
            printf("%d", k);
        }
        printf("\n");
    }
    return 0;
}
```

Output Clear

```
    5
   45
  345
 2345
12345


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

void printStars(int n) {
    for (int i = 0; i < n; i++) {
        printf("*\n");
    }
}

int main() {
    printStars(1);
    printf("\n");
    printStars(3);
    printf("\n");
    printStars(5);
    printf("\n");
    printStars(3);
    printf("\n");
    printStars(1);

    return 0;
}
```

Output

Clear

```
*

*
*
*

*
*
*
*
*

*
*
*

*


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

void printStarLine(int n) {
    for (int i = 0; i < n; i++) {
        printf("*");
    }
    printf("\n");
}

int main() {
    printStarLine(1);
    printStarLine(3);
    printStarLine(5);
    printStarLine(7);
    printStarLine(9);
    printStarLine(7);
    printStarLine(5);
    printStarLine(3);
    printStarLine(1);

    return 0;
}
```

Output Clear

```
*
***
*****
*******
*********
*******
*****
***
*

=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

void printCenteredStarLine(int n, int max) {
    int spaces = (max - n) / 2;
    for (int i = 0; i < spaces; i++) {
        printf(" ");
    }
    for (int i = 0; i < n; i++) {
        printf("*");
    }
    printf("\n");
}

int main() {
    int maxWidth = 7;
    printCenteredStarLine(1, maxWidth);
    printCenteredStarLine(3, maxWidth);
    printCenteredStarLine(5, maxWidth);
    printCenteredStarLine(7, maxWidth);
    printCenteredStarLine(5, maxWidth);
    printCenteredStarLine(3, maxWidth);
    printCenteredStarLine(1, maxWidth);

    return 0;
}
```

Output

Clear

```
   *
  ***
 *****
*******
 *****
  ***
   *


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

int isPrime(int num) {
    if (num <= 1) {
        return 0;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}

int main() {
    int n;

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

    printf("Prime numbers from 1 to %d are:\n", n);

    for (int i = 1; i <= n; i++) {
        if (isPrime(i)) {
            printf("%d ", i);
        }
```

Output Clear

```
Enter the value of n: 2 3 5 7 11 13 17 19 23 29
Prime numbers from 1 to 2 are:
2


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int arr[100];
    int n, i;

    printf("Enter the number of elements (up to 100): ");
    scanf("%d", &n);

    if (n > 100 || n <= 0) {
        printf("Invalid size. Please enter a number between 1 and
            100.\n");
        return 1;
    }

    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    printf("\nThe elements you entered are:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
```

Output Clear

```
Enter the number of elements (up to 100): 5
Enter 5 elements:
Element 1: 10
Element 2: -3
Element 3: 0
Element 4: 88
Element 5: 123

The elements you entered are:
10 -3 0 88 123


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int n, i, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);

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

    printf("Sum of array elements = %d", sum);
    return 0;
}

```

Output Clear

```
Enter number of elements: 5
Enter array elements: 2 4 6 8 10
Sum of array elements = 30

=== Code Execution Successful ===
```

main.c

Share Run

```c
    int n, i;
    printf("Enter number of elements: ");
    scanf("%d", &n);

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

    int max = arr[0];
    int min = arr[0];

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

    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d\n", min);

    return 0;
}
```

Output Clear

```
Enter number of elements: 5
Enter array elements: 10 25 3 40 18
Maximum element = 40
Minimum element = 3


=== Code Execution Successful ===
```

main.c

Share Run

```c

int main() {
    int n, i, even = 0, odd = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

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

    for (i = 0; i < n; i++) {
        if (arr[i] % 2 == 0)
            even++;
        else
            odd++;
    }

    printf("\nCount of even numbers = %d", even);
    printf("\nCount of odd numbers = %d", odd);

    return 0;
}
```

Output Clear

```
Enter number of elements: 6
Enter array elements: 12 7 9 14 22 5

Count of even numbers = 3
Count of odd numbers = 3

=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int n, i, positive = 0, negative = 0, zero = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

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

    for (i = 0; i < n; i++) {
        if (arr[i] > 0)
            positive++;
        else if (arr[i] < 0)
            negative++;
        else
            zero++;
    }

    printf("\nCount of positive numbers = %d", positive);
    printf("\nCount of negative numbers = %d", negative);
    printf("\nCount of zeros = %d", zero);
```

Output

Clear

```
Enter number of elements: 6
Enter array elements: 5 -3 0 8 -1 0

Count of positive numbers = 2
Count of negative numbers = 2
Count of zeros = 2

=== Code Execution Successful ===
```