sumN3.c

```c
#include <stdio.h>

int main() {
    int n, i = 1, sum = 0;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    if (n <= 0) {
        printf("Please enter a positive integer.\n");
        return 1;
    }

    // using while loop
    while (i <= n) {
        sum = sum + i;
        i++;
    }

    // using do-while loop
    /*
    do {
        sum = sum + i;
        i++;
    } while (i <= n);
    */

    printf("Sum of the first %d natural numbers is %d\n", n, sum);

    return 0;
}
```

sumN2.c

```c
#include <stdio.h>

int main(){

    int n, sum;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    if (n <= 0) {
        printf("Please enter a positive integer.\n");
        return 1;
    }

    sum = n * (n + 1) / 2;

    printf("Sum of the first %d natural numbers is %d\n", n, sum);

    return 0;
}
```

sumN.c

```c
#include <stdio.h>

int main() {
    int n, i, sum = 0;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    // Input validation for natural numbers (which start from 1)
    if (n <= 0) {
        printf("Please enter a positive integer.\n");
        return 1; // Exit with an error code
    }

    for (i = 1; i <= n; i++) {
        sum = sum + i;
    }

    printf("Sum of the first %d natural numbers is %d\n", n, sum);

    return 0;
}

```

examples/squareRoot.c

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

// Function to calculate the square root of a number

int main() {
    double number, result;

    // Prompt the user for input
    printf("Enter a number to find its square root: ");
    scanf("%lf", &number);

    // Check if the number is non-negative
    if (number < 0) {
        printf("Error: Cannot compute the square root of a negative number.\n");
        return 1;
    }

    // Calculate the square root
    result = sqrt(number);

    // Display the result
    printf("The square root of %.3f is %.3f\n", number, result);

    return 0;
}
```

solidSquare.c

```c
#include <stdio.h>

// solid square

int main()
{

    int rows;

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


    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        for (int j = 1; j <= rows; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}


    /**
    * Example Output:
    * Enter number of rows: 5
    * * * * *
    * * * * *
    * * * * *
    * * * * *
    * * * * *
    */
```

solidRectangle.c

```c
#include <stdio.h>

// Solid Rectangle Program

int main(){

    // define variables for rows and columns
    int rows,col;
   // get user input
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    printf("Enter number of columns: ");
    scanf("%d",&col);

    // print pattern
    for (int i = 0; i < rows; i++)
    // outer loop for rows
    {
        for (int j = 0; j < col; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }

    /**
     * Alternative method
     */

    /*
    // print solid rectangle
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        for (int j = 1; j <= col; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }
    */




    return 0;
}


```

```c
/**
* Example Output:
* Enter number of rows: 4
* Enter number of columns: 5
* * * * *
* * * * *
* * * * *
* * * * *
*/
```

rightHalfPyramid.c

```c
#include <stdio.h>

// Right Half Pyramid Pattern

int main()
{

    int rows;

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


    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        for (int j = 1; j <= i; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}

```

leftHalfPyramid.c

```c
#include <stdio.h>

// Left Half Pyramid Pattern

int main()
{
    int rows;

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

    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {

        // logic for spaces
        for (int space = 1; space <= rows - i; space++)
        {
            printf("  ");
        }

        for (int j = 1; j <= i; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}


/*
 * Example Output:
 * Enter number of rows: 5
 *         *
 *       * *
 *     * * *
 *   * * * *
 * * * * * *
 */
```

InvertedRHP.c

```c
#include <stdio.h>

// Inverted Right Half Pyramid Pattern

int main()
{

    int rows;

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


    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        for (int j = 1; j <= rows - i + 1; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}

 /**
 * Example Output:
 * Enter number of rows: 5
 * + + + + +
 * + + + +
 * + + +
 * + +
 * +
 */
```

ILHP.c

```c
#include <stdio.h>

// Inverted Left Half Pyramid Pattern

int main()
{
    int rows;

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

    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        // logic for spaces
        for (int space = 1; space <= i - 1; space++)
        {
            printf("  ");
        }

        for (int j = 1; j <= rows - i + 1; j++)
        // inner loop for columns
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}


/*
 * Example Output:
 * Enter number of rows: 5
 * * * * *
 *   * * *
 *     * *
 *       *
 */

```

fact2.c

```c
#include <stdio.h>

int main() {
    int n, i = 1;
    int factorial = 1;
    printf("Enter a non-negative integer: ");
    scanf("%d", &n);

    // Check if the number is negative
    if (n < 0) {
        printf("Error: Factorial is not defined for negative numbers.\\n");
    } else {
       do {
            factorial = factorial * i;
            i++;
        } while (i <= n);
        printf("Factorial of %d is %d \n", n, factorial);
    }

    return 0;
}

```

fact.c

```c
#include <stdio.h>

int main() {
    int n, i = 1;
    int factorial = 1;
    printf("Enter a non-negative integer: ");
    scanf("%d", &n);

    // Check if the number is negative
    if (n < 0) {
        printf("Error: Factorial is not defined for negative numbers.\\n");
    } else {
        // Calculate the factorial using a while loop
        while (i <= n) {
            factorial = factorial * i; // Multiply current factorial by i
            i++;            // Increment the counter
        }
        printf("Factorial of %d is %d \n", n, factorial);
    }

    return 0;
}

```

checkPrime.c

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

// check if a number is prime

int main(){

    int num, i, isPrime = 1;

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

    if (num <= 1) {
        isPrime = 0; // numbers less than or equal to 1 are not prime
    } else {
        for (i = 2; i <= sqrt(num); i++) {
            if (num % i == 0) {
                isPrime = 0; // found a divisor, not prime
                break;
            }
        }
    }

    if (isPrime) {
        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }

    return 0;
}
```

p2.c

```c
#include <stdio.h>
/*
 * Print the following pattern for given number of rows.
 * Enter number of rows: 5
 * 1
 * 2 2
 * 3 3 3
 * 4 4 4 4
 * 5 5 5 5 5
 */

int main()
{

    int rows;

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


    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        for (int j = 1; j <= i; j++)
        // inner loop for columns
        {
            //printf("* ");
            printf("%d ",i);
        }
        printf("\n");
    }

    return 0;
}

```

p1.c

```c
#include <stdio.h>
/*
 * Print the following pattern for given number of rows.
 * Enter number of rows: 5
 * 1
 * 1 2
 * 1 2 3
 * 1 2 3 4
 * 1 2 3 4 5
 */

int main()
{

    int rows;

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


    // print pattern
    for (int i = 1; i <= rows; i++)
    // outer loop for rows
    {
        for (int j = 1; j <= i; j++)
        // inner loop for columns
        {
            //printf("* ");
            printf("%d ",j);
        }
        printf("\n");
    }

    return 0;
}

```