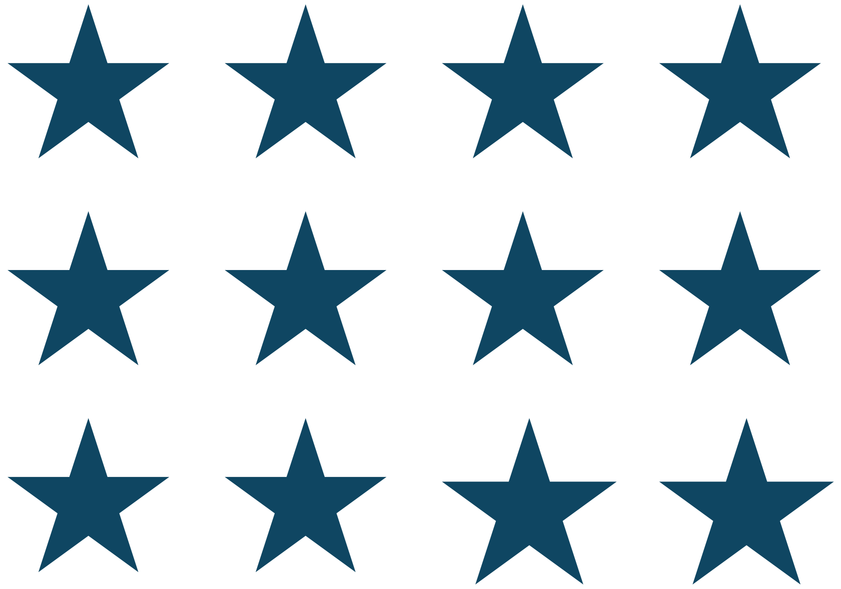
Math Series

C Programming

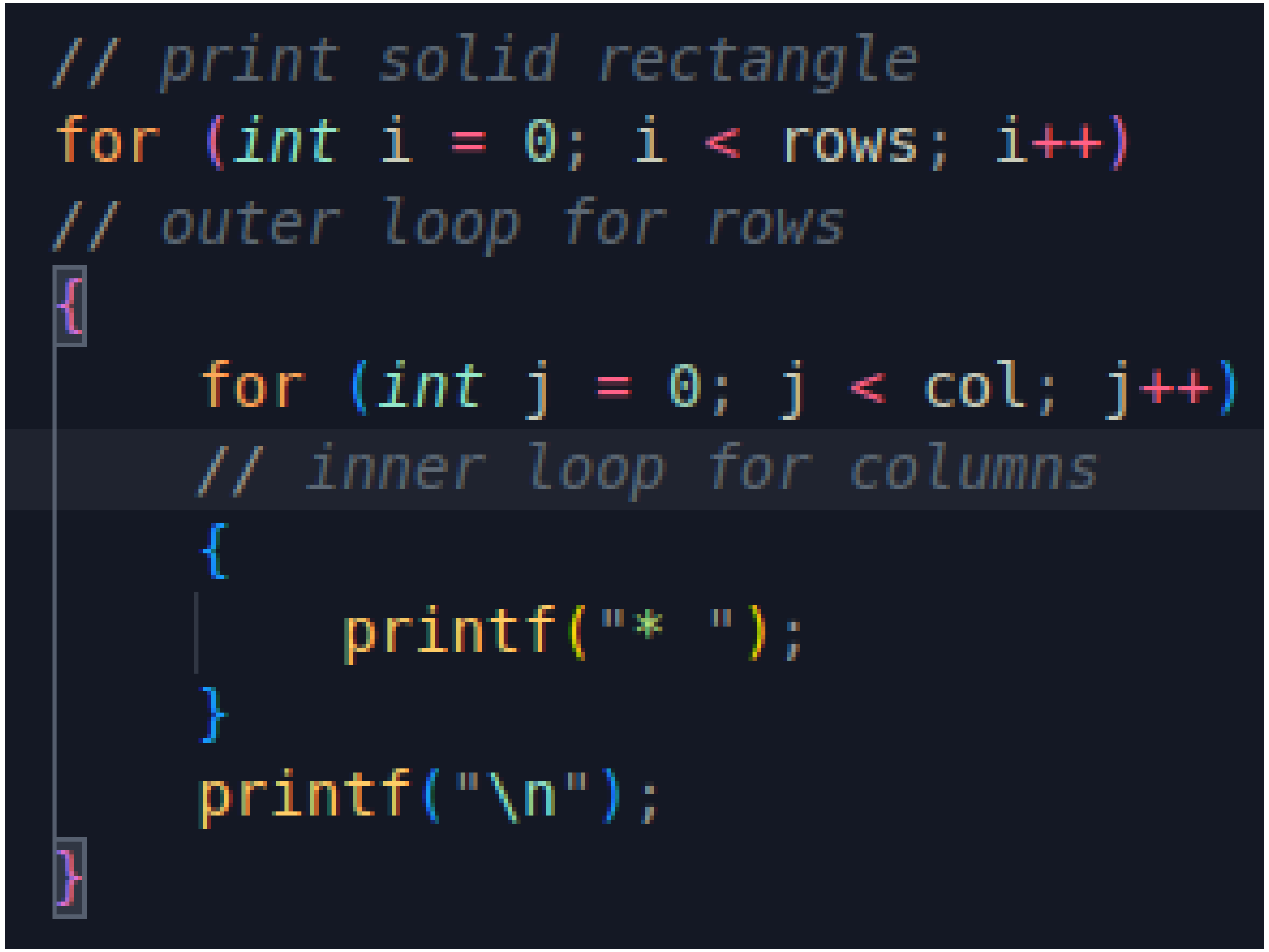
UG Sem-3 Major (Kalyani University)

Day - 06

Solid Rectangle Pattern

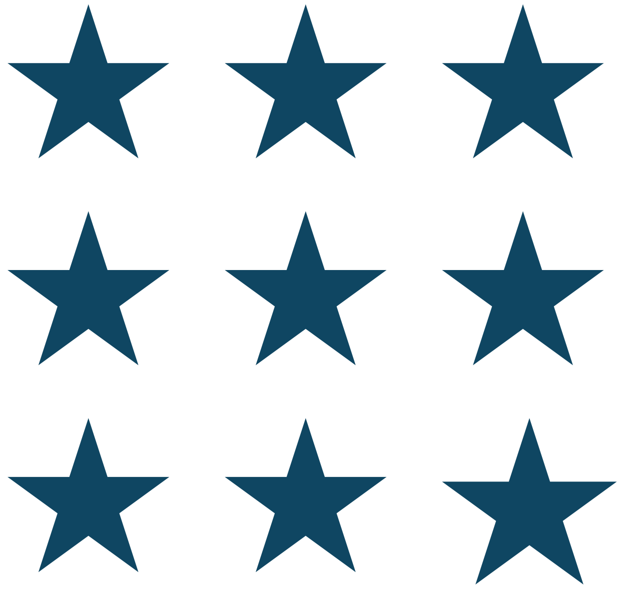


(Rows=3, Columns=4)

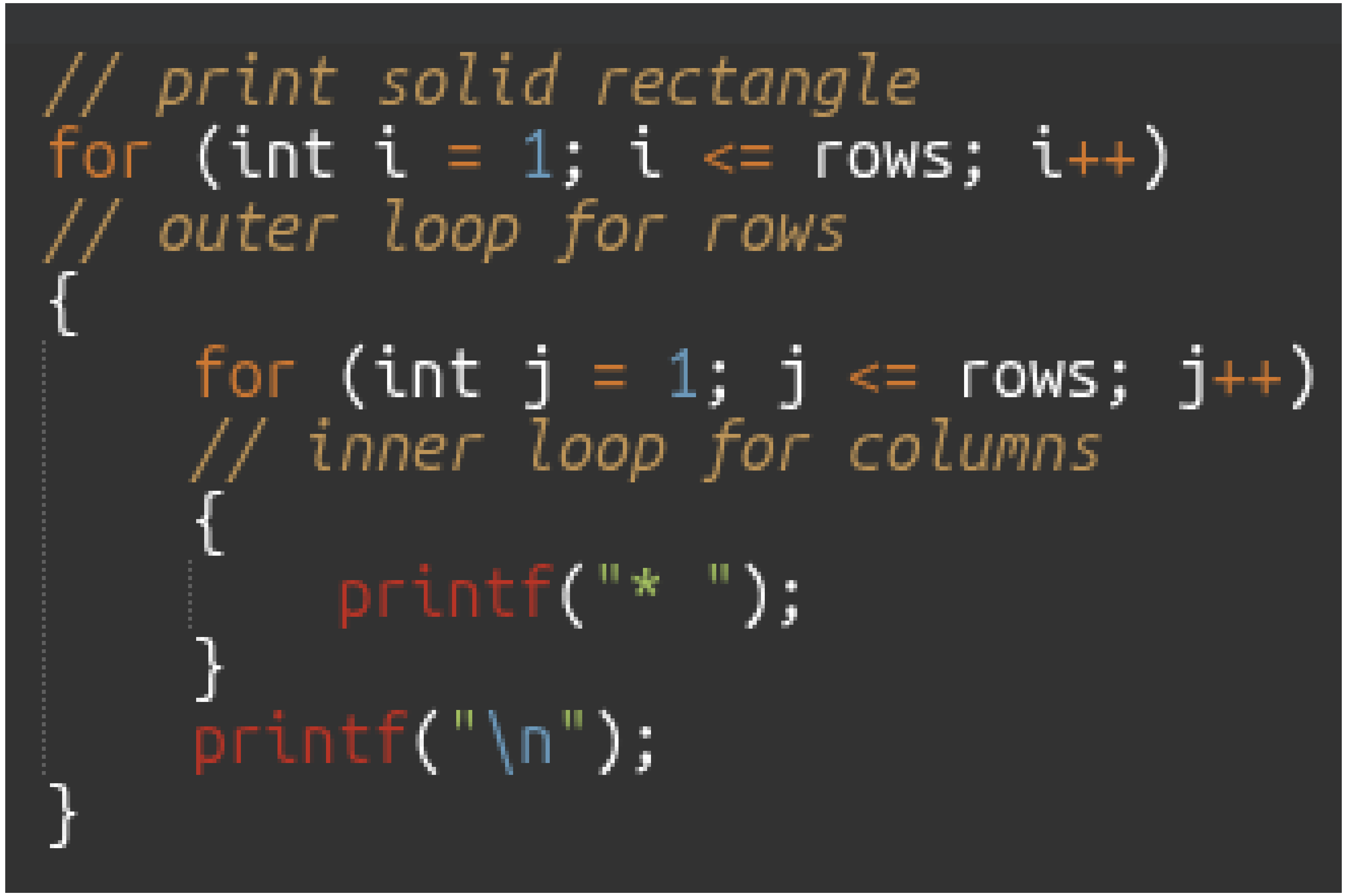


```
// print solid rectangle
for (int i = 0; i < rows; i++)
// outer loop for rows
{
    for (int j = 0; j < col; j++)
    // inner loop for columns
    {
        printf("* ");
    }
    printf("\n");
}
```

Solid Square Pattern

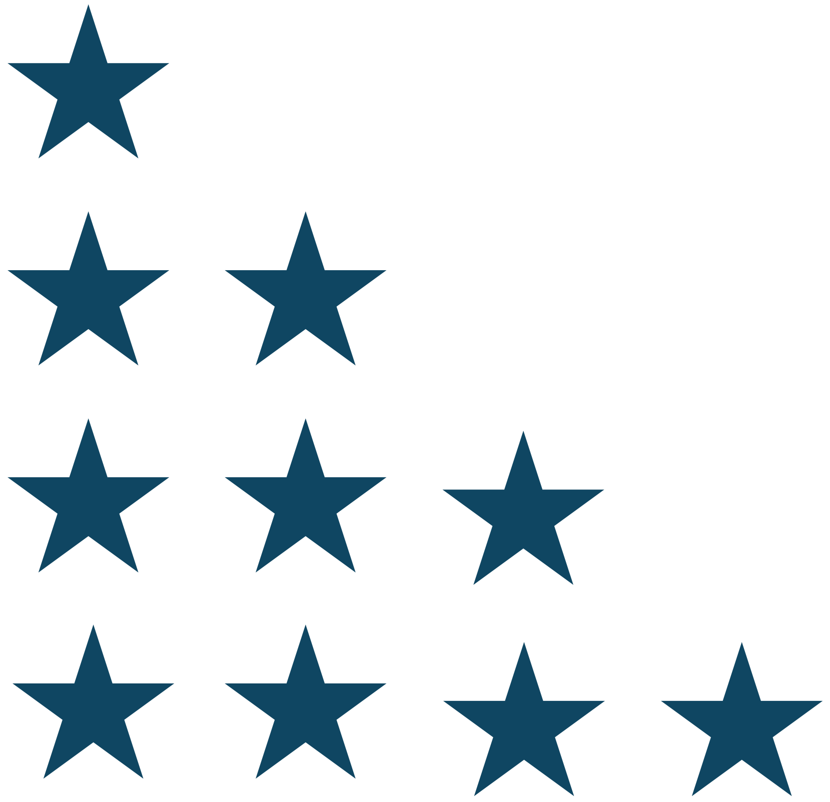


(Rows=3, Columns=3)

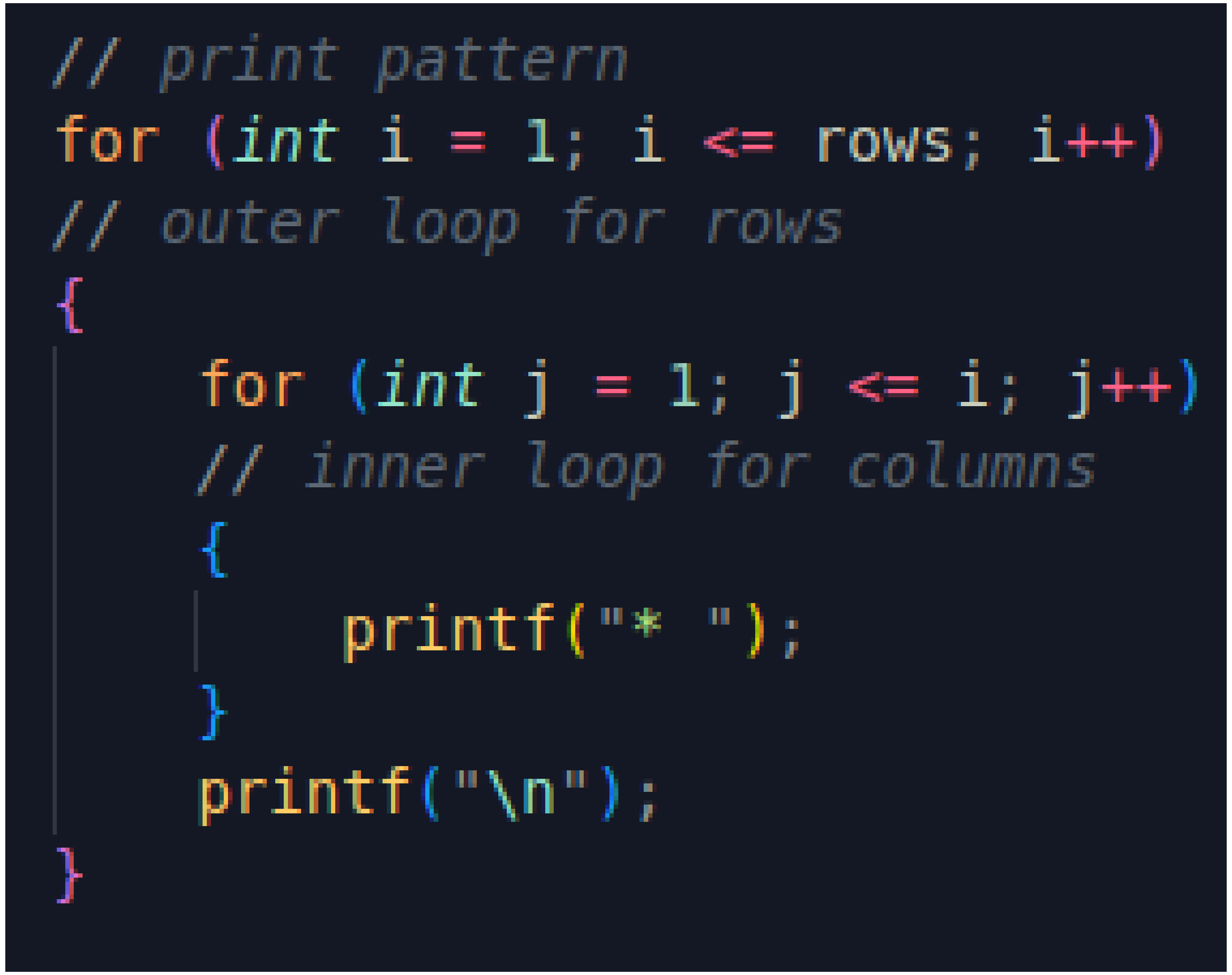


```c
// print solid rectangle
for (int i = 1; i <= rows; i++)
// outer loop for rows
{
    for (int j = 1; j <= rows; j++)
    // inner loop for columns
    {
        printf("* ");
    }
    printf("\n");
}
```

Right Half Pyramid Pattern

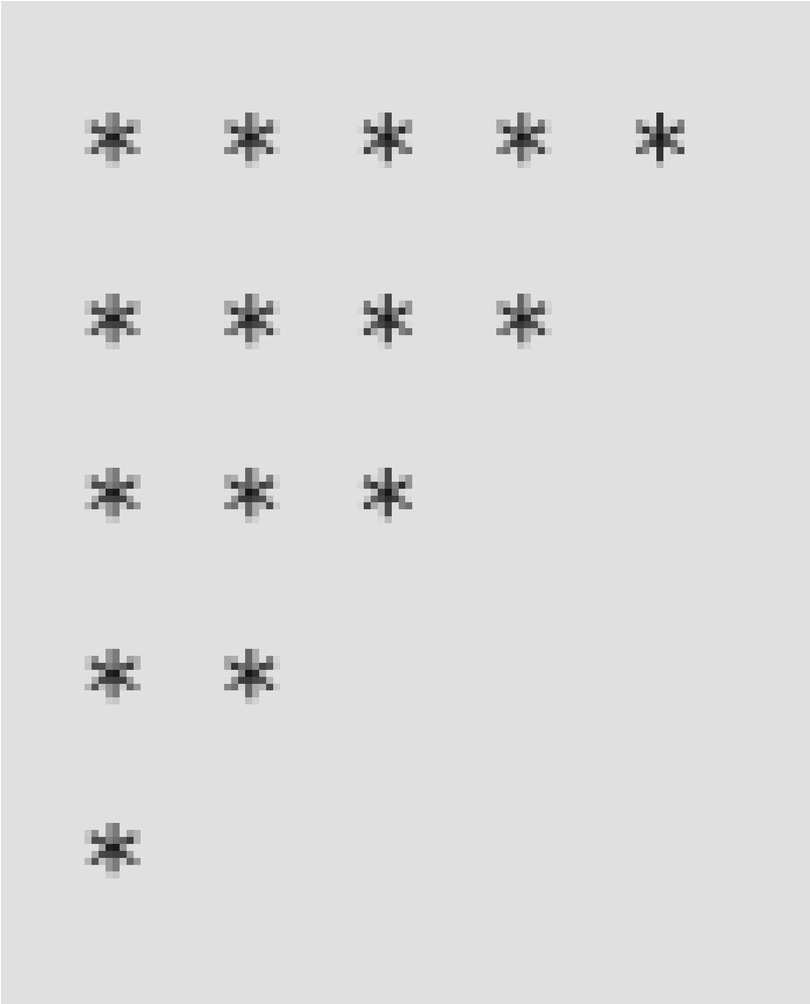


Here No. of Rows = 4



```
// 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");
}
```

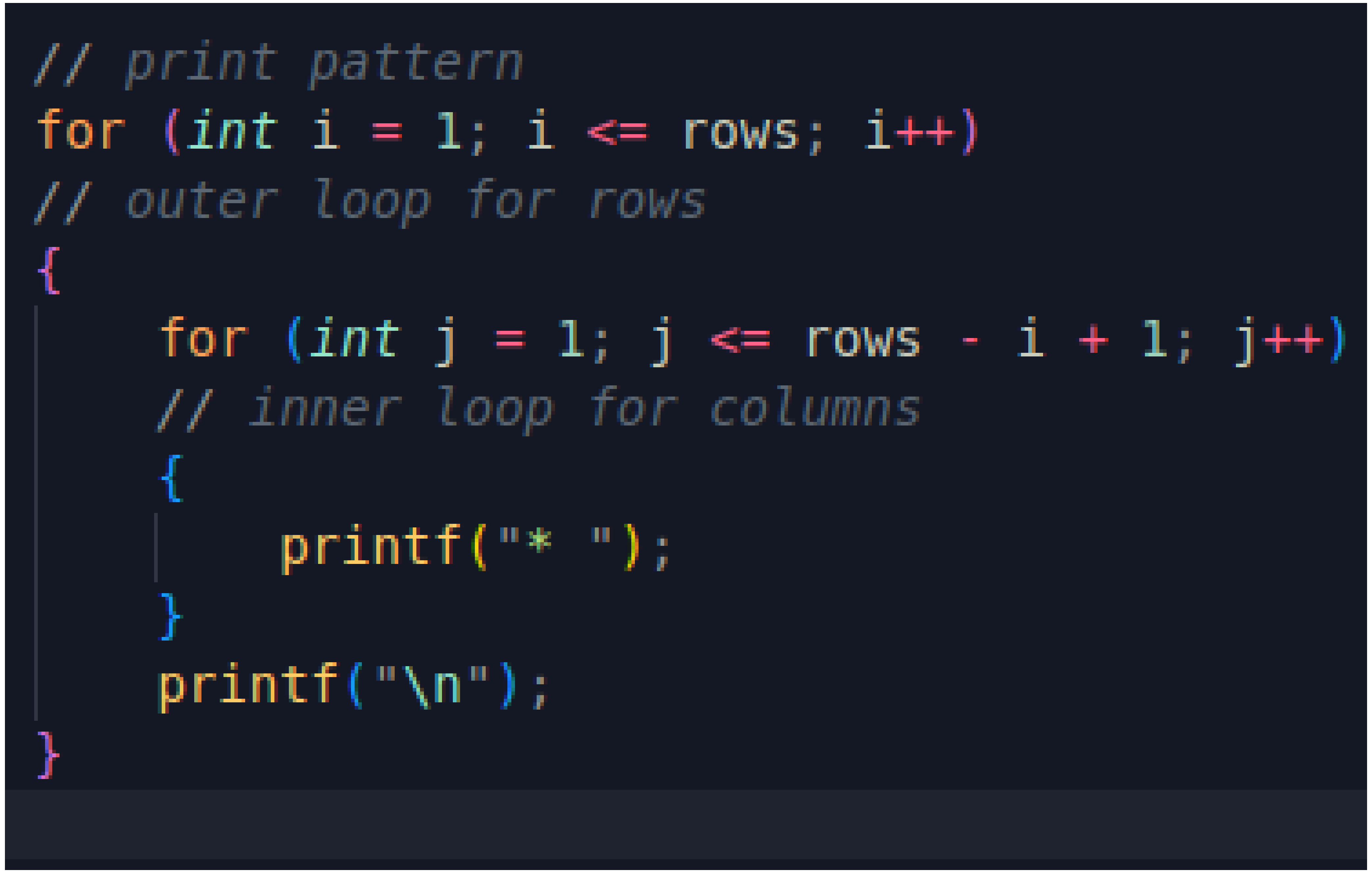
Inverted Right Half Pyramid Pattern



```
*  *  *  *  *

*  *  *  *

*  *  *

*  *

*
```

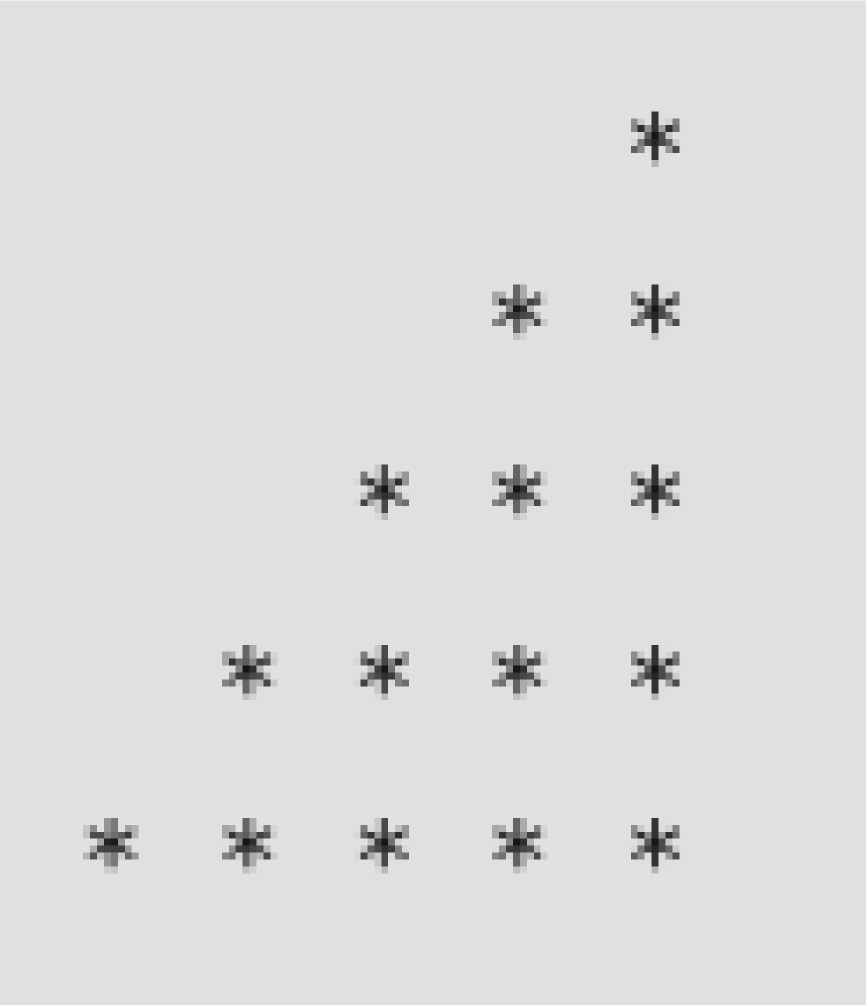
Here No. of Rows = 5

i + j = No. of Rows



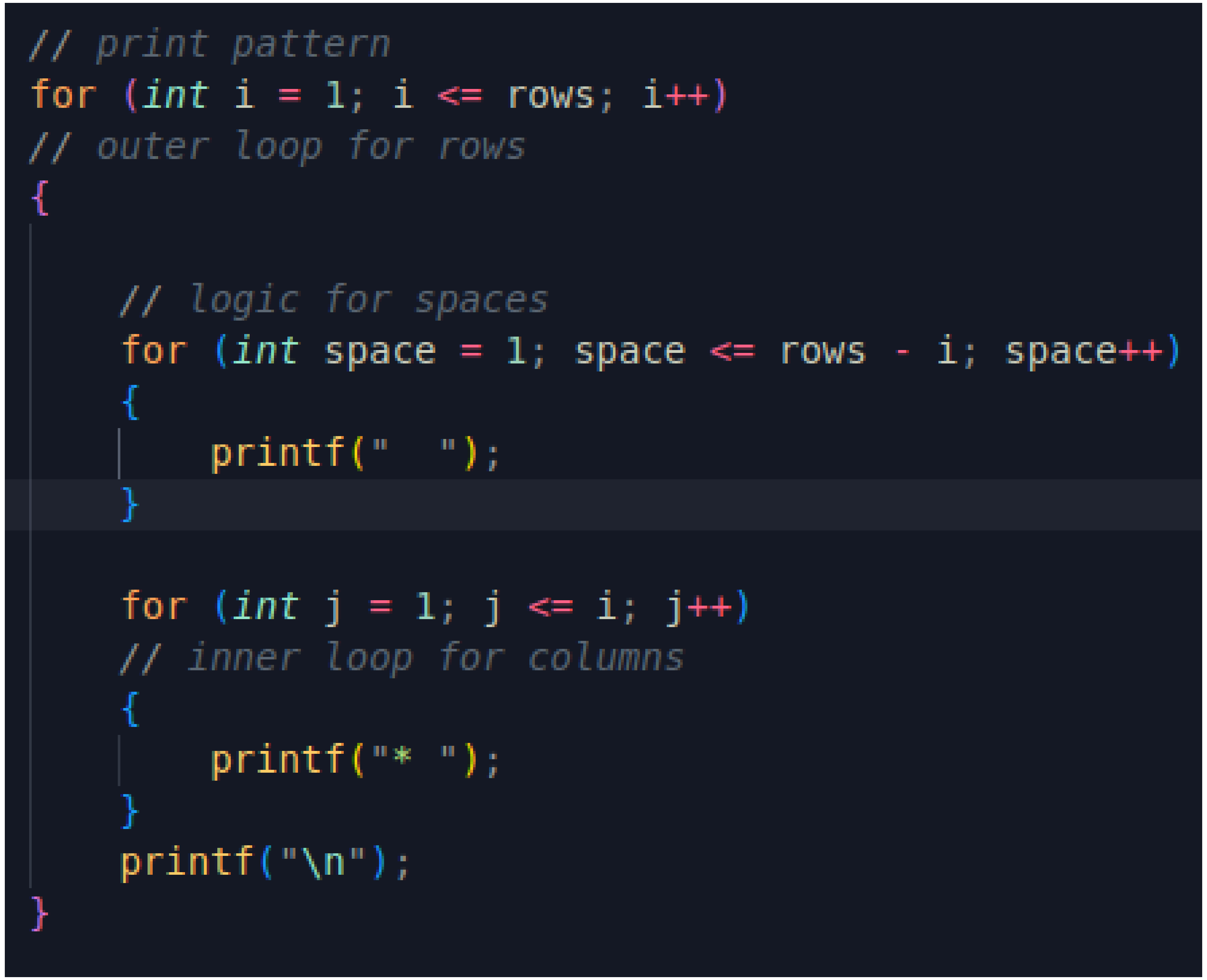
```
// print pattern
for (int i = 1; i <= rows; i++)
// outer loop for rows
{
    for (int j = 1; j <= rows - i + 1; j++)
    // inner loop for columns
    {
        printf("* ");
    }
    printf("\n");
}
```

Left Half Pyramid Pattern



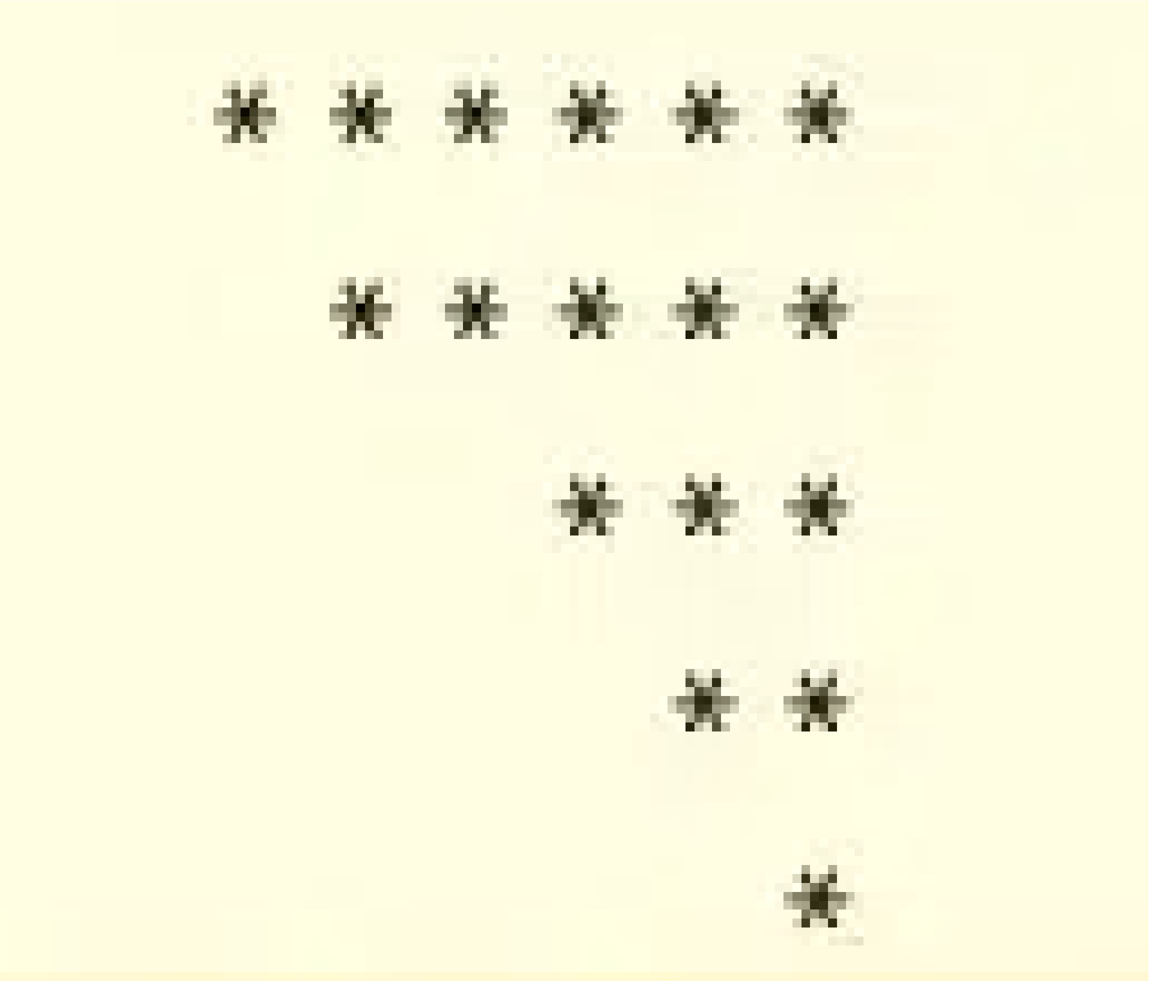
```
        *
      * *
    * * *
  * * * *
* * * * *
```

Here No. of Rows = 5



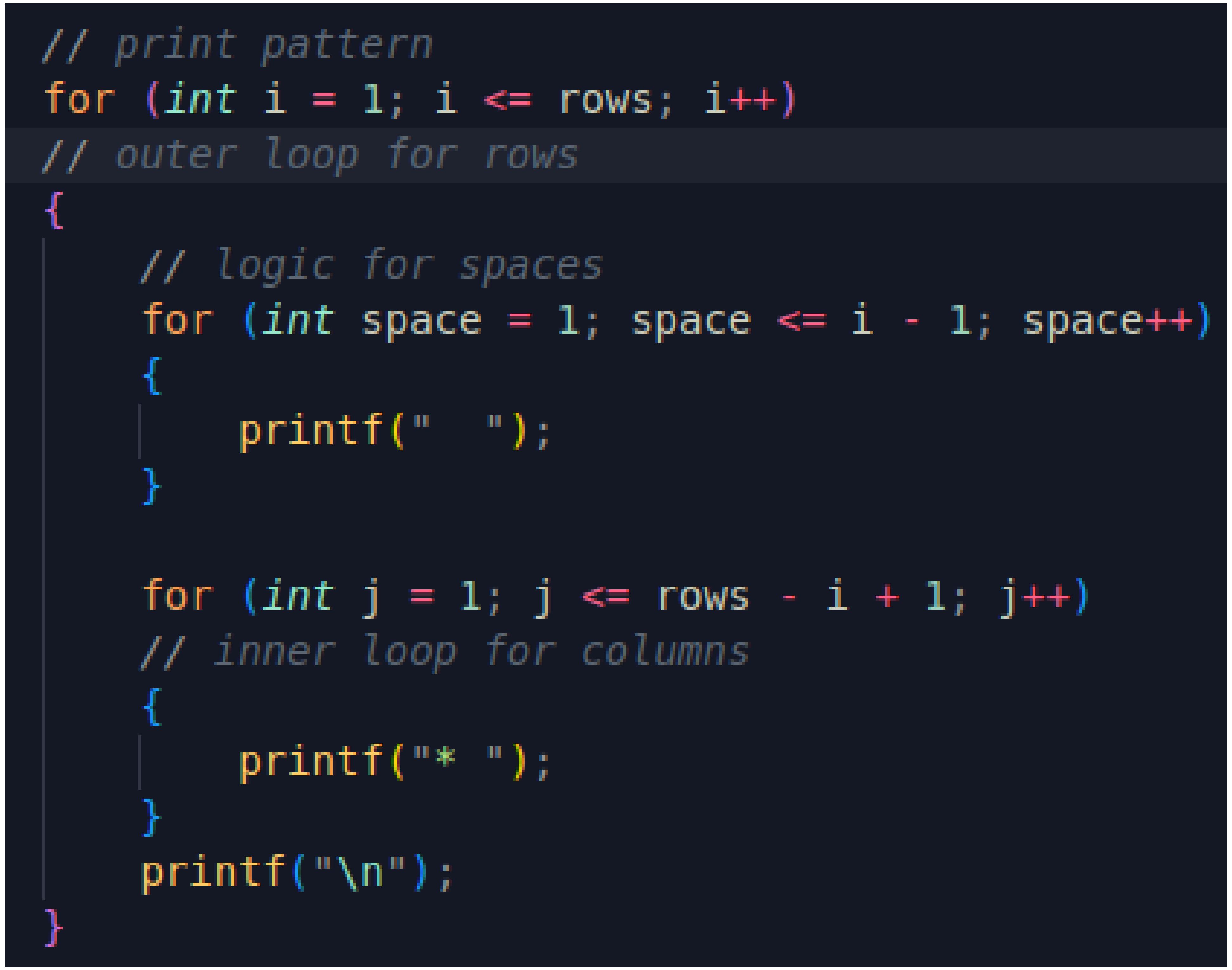
```c
// print pattern
for (int i = 1; i <= rows; i++)
// outer loop for rows
{

    // 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");
}
```

Inverted Left Half Pyramid Pattern



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

Here No. of Rows = 5

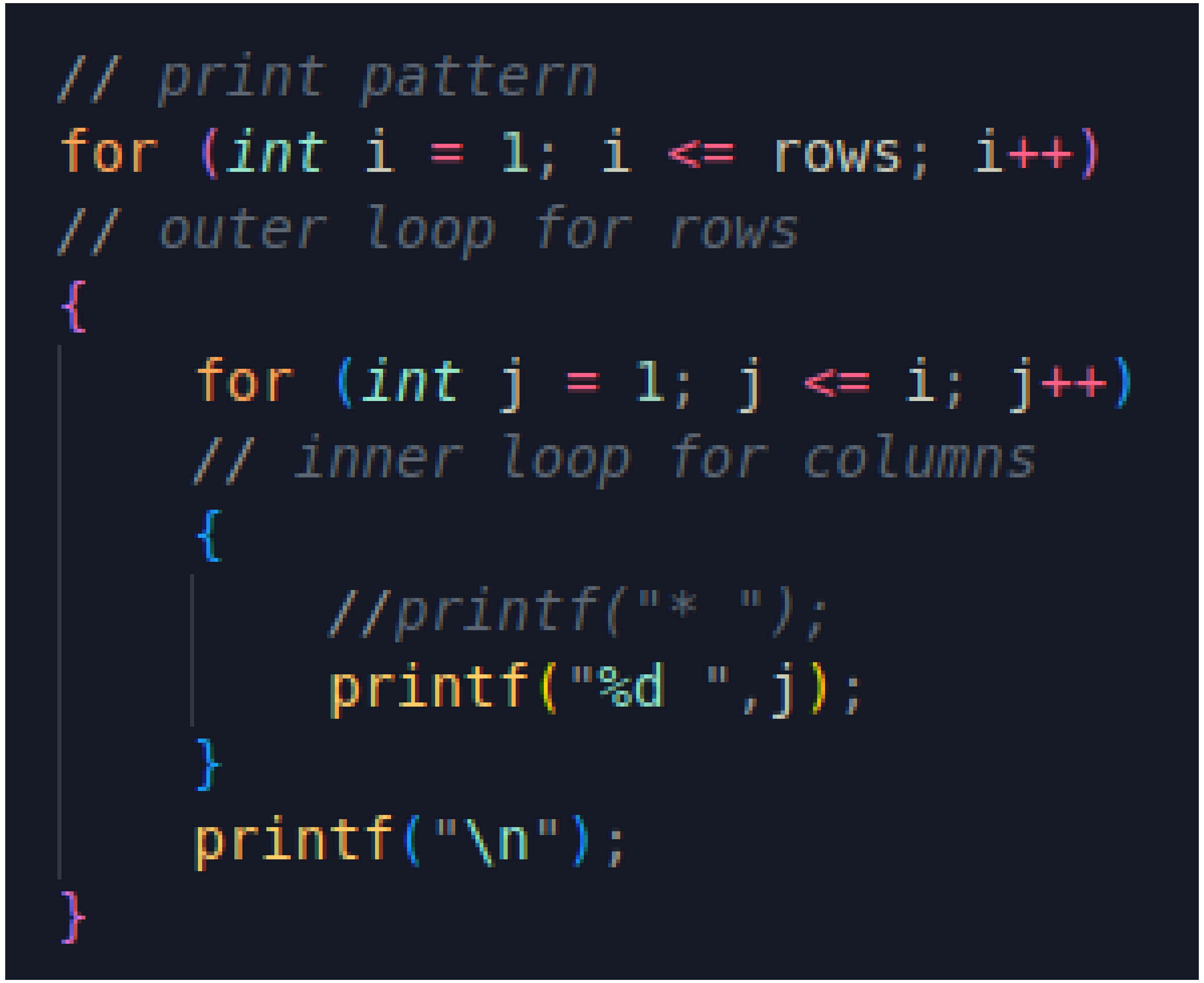


```
// print pattern
for (int i = 1; i <= rows; i++)
// outer loop for rows
{
    // logic for spaces
    for (int space = 1; space <= i - 1; space++)
    {
        printf("  ");
    }

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

Number Pattern-1

```
1
1 2
1 2 3
1 2 3 4
```

Here No. of Rows = 4

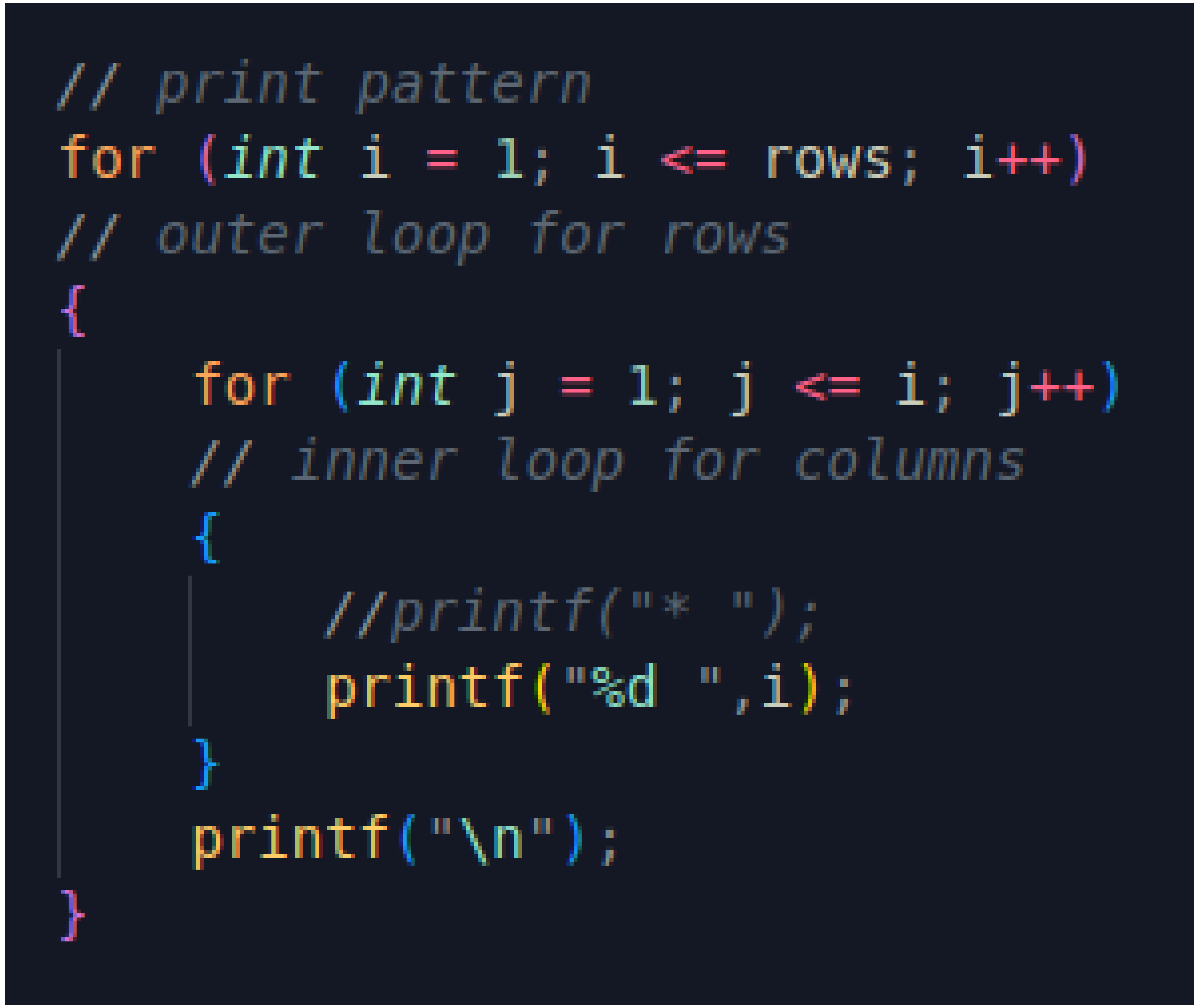


```
// print pattern
for (int i = 1; i <= rows; i++)
// outer loop for rows
{
    for (int j = 1; j <= i; j++)
    // inner loop for columns
    {
        //printf("* ");
        printf("%d ",j);
    }
    printf("\n");
}
```

Number Pattern-2

```
1
2 2
3 3 3
4 4 4 4
```

Here No. of Rows = 4



```
// 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");
}
```

math.h

math.h is a header file in the standard library of the C programming language designed for basic mathematical operations.

Problem based on this:

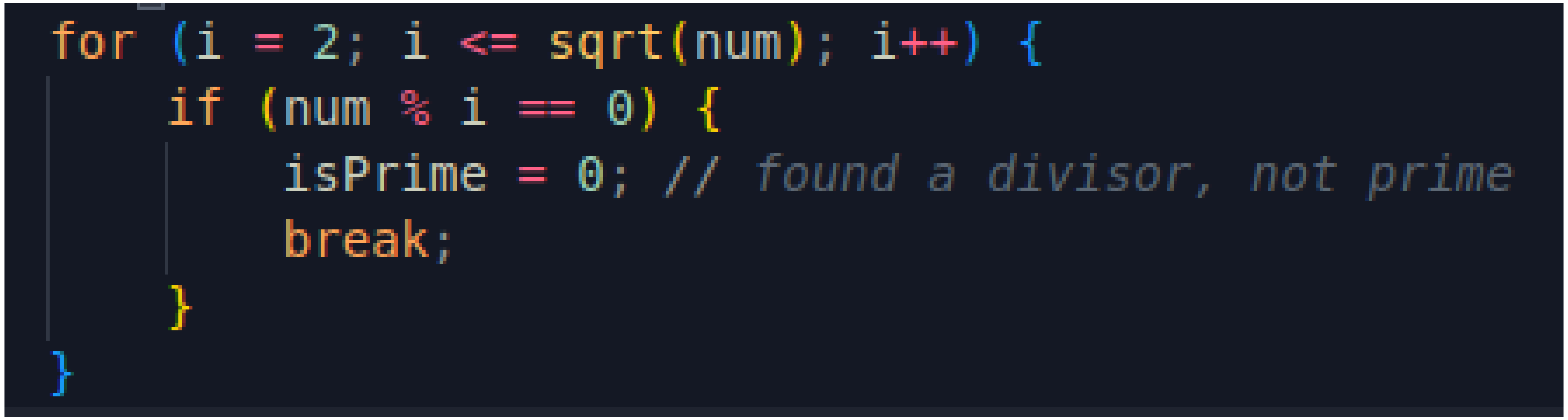
1) Find square root of a number

2) Check prime number

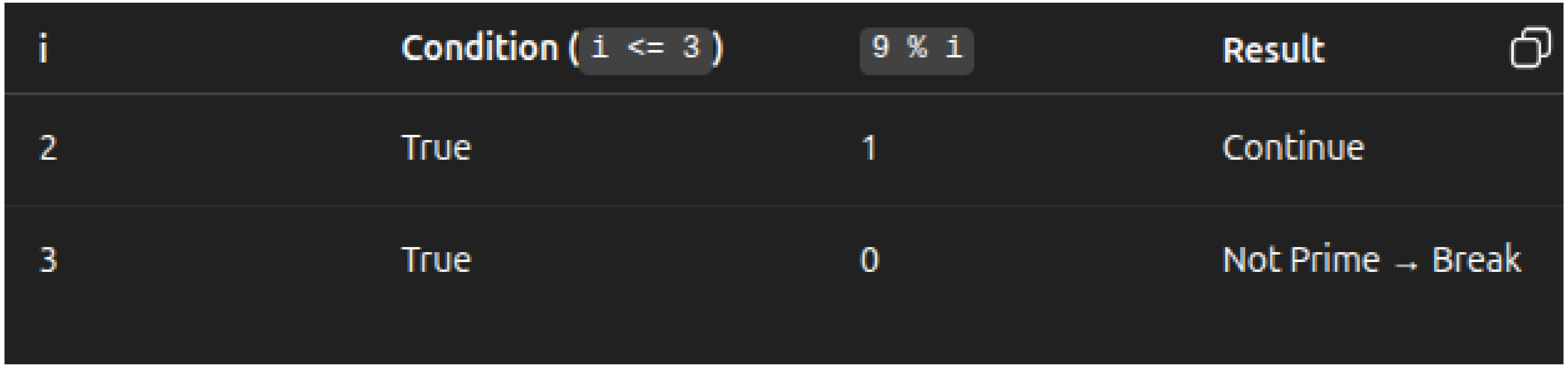
Math Functions in C Standard Library

Function Name	Math Name	Value	Example		
`abs(x)`	absolute value	$\|x\|$	`abs(-1)`	returns	1
`fabs(x)`	absolute value	$\|x\|$	`fabs(-3.2)`	returns	3.2
`pow(x,y)`	raise to the power	x^y	`pow(2.0, 3.0)`	returns	8.0
`sqrt(x)`	square root	$x^{0.5}$	`sqrt(2.0)`	returns	1.414…
`exp(x)`	exponential	e^x	`exp(1.0)`	returns	2.718…
`log(x)`	natural logarithm	$\ln x$	`log(2.718…)`	returns	1.0
`log10(x)`	common logarithm	$\log x$	`log10(100.0)`	returns	2.0
`sin(x)`	sine	$\sin x$	`sin(3.14…)`	returns	0.0
`cos(x)`	cosine	$\cos x$	`cos(3.14…)`	returns	-1.0
`tan(x)`	tangent	$\tan x$	`tan(3.14…)`	returns	0.0
`ceil(x)`	ceiling	$\lceil x \rceil$	`ceil(2.5)`	returns	3.0
`floor(x)`	floor	$\lfloor x \rfloor$	`floor(2.5)`	returns	2.0

Check for a Prime Number



```
for (i = 2; i <= sqrt(num); i++) {
    if (num % i == 0) {
        isPrime = 0; // found a divisor, not prime
        break;
    }
}
```



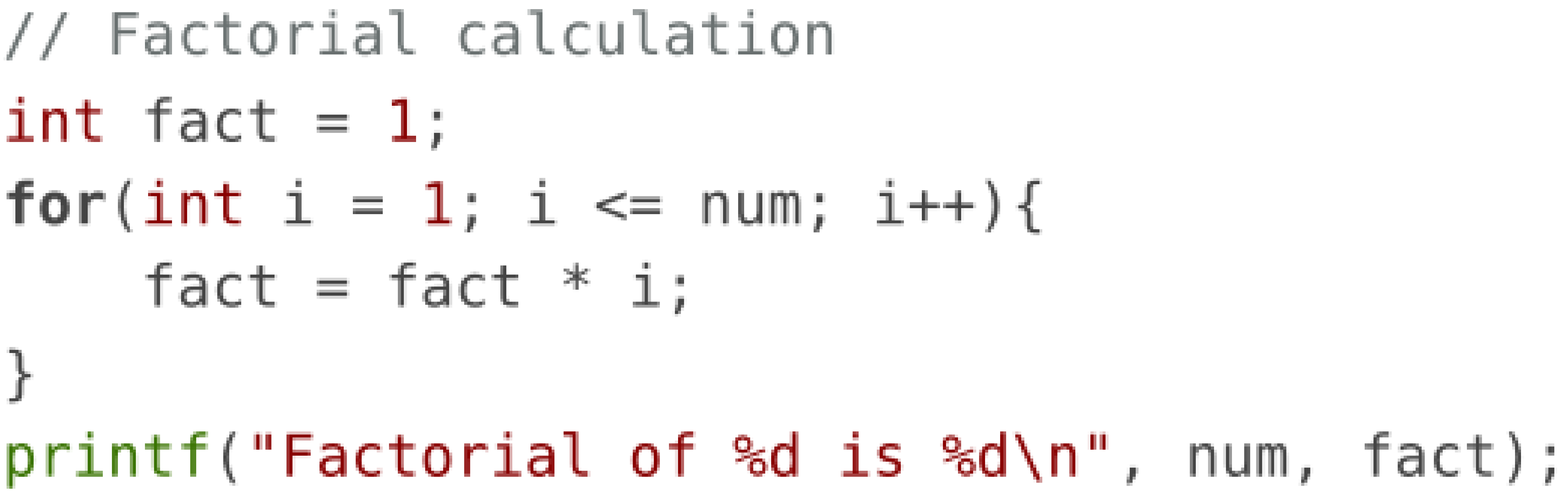
i	Condition (`i <= 3`)	`9 % i`	Result
2	True	1	Continue
3	True	0	Not Prime → Break

Sum of 100 Natural numbers

- Find using formula : n(n+1)/2
- Find using for loop
- Find using while loop

Factorial of a number

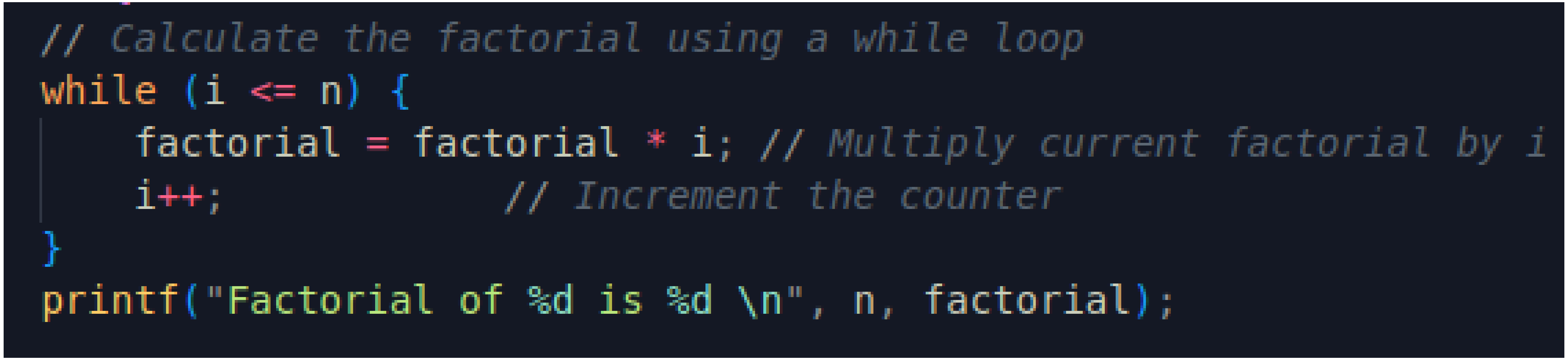
- Find using for loop



```c
// Factorial calculation
int fact = 1;
for(int i = 1; i <= num; i++){
    fact = fact * i;
}
printf("Factorial of %d is %d\n", num, fact);
```

Factorial of a number

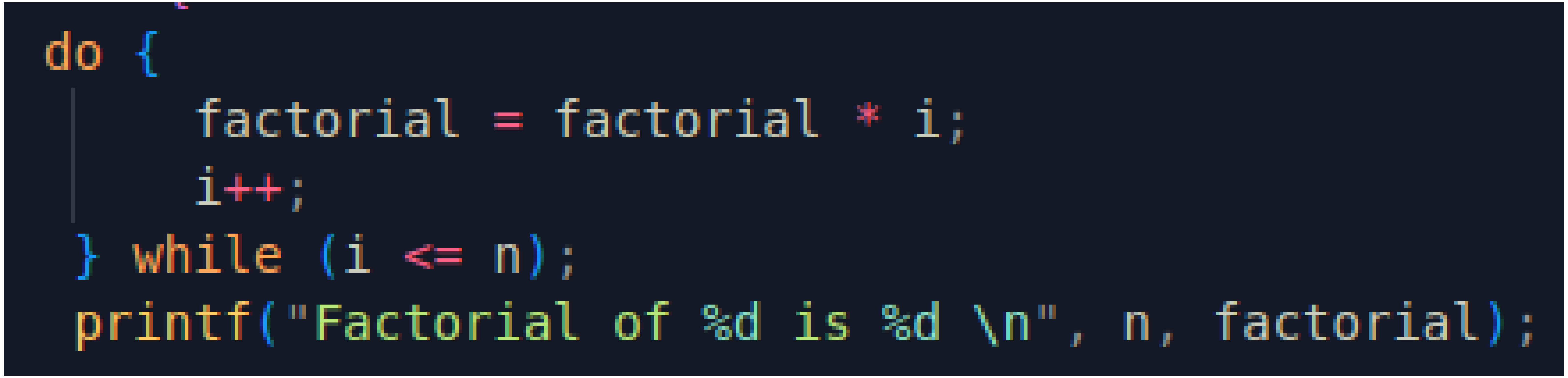
- Find using while loop



```c
// 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);
```

Factorial of a number

- Find using do-while loop



```
do {
     factorial = factorial * i;
     i++;
 } while (i <= n);
 printf("Factorial of %d is %d \n", n, factorial);
```