Nested for loop

for (int r = 1; r <= 5; r++) {
    for (int c = 1; c <= 10; c++) {
        cout << "*";
    }
    cout << endl;
}

Whenever we’d like to go over a 2-D pattern, a nested loop is a good set up.
- outer loop go through the rows 1-height for (int r = 1; r <= height; r++)
- inner loop go through the columns 1-width for (int c = 1; c <= width; c++)
- body of inner loop specify what to print for each position
- print a new line at the end of each row (inner loop)
Print a Triangle of

/** given n print a n X n triangle of *
 * Ex: if n is 5, then it looks like following
 r1*
 r2**
 r3***
 r4****
 r5*****
 */

int n;
for (int r = 1; r <= n; r++) {
    // width of each row is same as row number
    for (int c = 1; c <= r; c++) { 
        cout << "*" ;
    }
    cout << endl;
}

Print a Triangle of *
/* print an upside down n X n triangle of * * Ex: if n is 5, then it looks like following */

r5*****
r4****
r3***
r2**
r1*
*/

int n;
for (int r = n; r >= 1; r--) {
    // width of each row is same as row number
    for (int c = 1; c <= r; c++){
        cout << "*";
    } cout << endl;
}
Print a back slash of

/* given n print a n X n back slash of */
* Ex: if n is 5, it looks like following

coordinate:
t 12345
r 1*
r 2 **
r 3 ***
r 4 ****
r 5 *****

each position specified by a row# and column#
*/

int n;
for (int r = 1; r <= n; r++) {
    //width of each row is same as row number
    for (int c = 1; c <= r; c++) {
        if (r == c) cout << "*";
        else cout << " ";
    }
    cout << endl;
}

Print an arrow of *

/* given width, print an arrow of *
* Ex: if width is 3, it looks like following
*
* *
* *
* *
* *
* *

each position specifies by a row# and column#
star on down diagonal: r == c
star on up diagonal: r + c == 6 or height + 1 */

int width, height;
height = 2 * width - 1;
for (int r = 1; r <= height; r++) {
    for (int c = 1; c <= width; c++) {
        if (r == c || r + c == height + 1) cout << "*";
        else cout << " ";
    }
    cout << endl;
}
Print a side triangle of *
/* given width, print a side triangle of *
 * Ex: if width is 3, it looks like following *
  *
  **
  ***
  **
  *

each position specifies by a row# and column#
star on down diagonal: r == c
star on up diagonal: r + c == height + 1
star bounded by these lines:

r >= c && r + c <= height + 1 */

int height = 2 * width - 1;
for (int r = 1; r <= height; r++) {
    for (int c = 1; c <= width; c++) {
        if (r >= c && r + c <= height + 1) cout << "*";
        else cout << " ";
    }
    cout << endl;
}