Sudo Code

Given a number of rows, print out a right triangle made of \*s.

3:

\*

\*\*

\*\*\*

-----

**numRows = given input**

**r = current row**

**c = current column**

Suggestion: make a table of values to help see patterns

R # of \*s

1 1

2 2

3 3

n n

Start at r = 1 ---> r = numRows {

Start at c = 1 → c = r {

print “\*”

}

print “\n”

// (\ is called the escape sequence)

}

Given a number of rows, rpint out a right justified right triangle made of \*s.

3:

--\*

-\*\*

\*\*\*

**numRows = given input**

**r = current row**

**c = current column**

Suggestion: make a table of values to help see patterns

R | # of spaces | # of \*s

1 2 1

2 1 2

3 0 3

Start at r = 1 → r = numRows {

Start at c = numRows - r → c = 1 {

print “\*”

}

print “\n”

// (\ is called the escape sequence)

}

Given a number of rows,print out a pyramid of \*s 4:

\*

\*\*

\*\*\*

\*\*\*\*

|  |  |  |
| --- | --- | --- |
| Rows (r) | Number of spaces on the left | Number of Stars “\*\_” |
| 1 | 3 | 1 |
| 2 | 2 | 2 |
| 3 | 1 | 3 |
| 4 | 0 | 4 |

**numRows = given input**

**r = current row**

**c = current column**

Start at r = 1 → r = numRows {

Start at c = 1 → c = r {

print “\*”

}

print “\n”

}

Create a tile pattern composed of black and white tiles with a fringe of black tiles all around and 2 or 3 black tiles in 1 or 2 rows in the center, equally spaced from the boundary. The inputs to your algorithmn are the total number of rows and columns in the pattern.

**Variables**

**Input width & height**

**centerWidth = 2 + width % 2**

**centerHeight = 2 + height % 2**

**horizontalPadding = (width - 2 - centerWidth) / 2**

**verticalPaddig = (height - 2 - centerHeight) / 2**

**Repeat width times:**

**print black**

**print \n (\n breaks to the next line)**

**Repeat verticalPadding times:**

**Print black**

**Repeat width - 2 times:**

**Print white**

**Print black + \n**

**Repeat centerHeight times:**

**Print black**

**repeat horizontalPadding times:**

**white**

**Repeat centerWidth times:**

**black**

**repeat horizontalPadding times:**

**white**

**print black + \n**

**Repeat verticalPadding times:**

**Print black**

**Repeat width - 2 times:**

**Print white**

**Print black + \n**

**repeat width times:**

**print black**

Start at r = 1 → r = numOfBlackTiles {

Start at c = 1 → c = 1 {

print “X”

}

print “\n”

}

XXXXXX

X----X

X-XX-X  
X----X  
XXXXXX