Practical Exercise 5
(Week 6)
(2 Marks)

Task 5.1

Write a recursive function, `power`, that takes two integers x and y as parameters such that x is nonzero and returns x^y. You can use the following recursive definition to calculate x^y:

If $y \geq 0$,

$$power(x,y) = \begin{cases} 1, & y = 0; \\ x, & y = 1; \\ x \times power(x, y-1), & y > 1. \end{cases}$$

If $y<0$,

$$power(x,y) = \frac{1}{power(x,-y)}.$$

Also, write a main function to test your function.

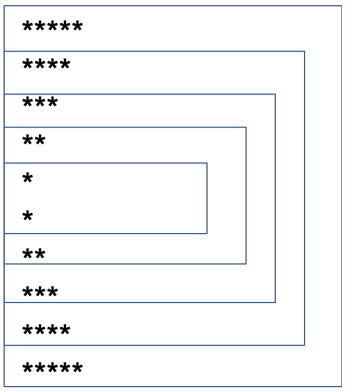
Task 5.2

Write two nested recursive functions to generate the following pattern of stars with n stars in the first line (here `n=4`):

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

Hint:

1. *For these two recursive functions, the first one, say* `printStars(int n)`*, is used to print a line of* `n` *stars; the second one,* `printLines(int m)`*, is to print* `m` *pairs of lines.*
2. *Think in the way that the whole picture is generated in the following pattern:*



Task 5.3

Download the code *Task5_3.zip* under Practical 5. The file contains a simplified State class (*state.h*) and a driver (*task5_3.cpp*). Add code into the following function

```
bool search(stack<State> states, State target)
```

in *Task5_3.cpp* to implement search as a recursive function so that it checks if the *target* state in the stack *states*.

Task 5.4 (Assignment 1 pre-check)

Based on the State class you created, implement the following three functions into the class:

1. `int removeBlockFrom(int column)`: remove the top block from the specified column. If succeeded, return the block value; otherwise, return -1.
2. `bool insertBlockTo(int column, int value)`: insert a new block with the value to the specified column. Return true if succeeded; otherwise, return false.
3. `bool moveBlock(int source, int destination)`: combine the above two functions to move a block from the source column to the destination column.

Test your code with a class driver.