Mini Exam: Recursion and Backtracking

Problems for exercises and homework for the "Data Structures and Algorithms Basics" course from the official "Applied Programmer" curriculum.

You can check your solutions here: https://judge.softuni.bg/Contests/2727/Mini-Exam-Recursion-and-Backtracking

1. Calculate Complexity: Sum Rectangle Areas

Calculate the expected running time **`O(f(n))`** in the **worst case** for the following C# function:

```
static int SumRectAreas(int width, int height)
{
  int sum = 0;
  for (int x = 1; x <= width; x++)
  {
    for (int y = 1; y <= height; y++)
      sum = sum + x * y;
  }
  return sum;
}
```

Send in the judge system one of the following values, corresponding to the correct complexity:

constant	logarithmic	sqrt(n)	linear	n*sqrt(n)
quadratic	n*log(n)	cubic	2^n	exponential

2. Calculate Complexity: Special Sum

Calculate the expected running time **`O(f(n))`** in the **worst case** for the following C# function:

```
static int SpecialSum(int start, int end)
{
    int sum = 0;
    for (int p = start; p <= end; p++)
    {
      for (int q = end; q <= start; q++)
      {
         for (int k = 1; k <= end; k++)
           sum++;
      }
      sum = sum + 50;
    }
    return sum;
}
```

Assume **`start < end`**.

3. Draw Reversed "K"

Write a program that draws **reversed "K" figure** like the example below depending on **n**. Use **recursion**.

Examples

Input	Output
2	## # * **
5	##### #### ### ## # * ** *** **** *****

4. Reversed Vectors

Generate all **n-bit** vectors of **zeroes** and **ones** in lexicographic order. Then **reverse** each vector (e. g. 100 -> 001). Print each vector on a separate line. Use **recursion**.

Examples

Input	Output	Comment
3	000 100 010 110 001 101 011 111	The generated vectors: 000 001 010 011 100 101 110 111 The reversed vectors: **000** **100** **010** **110** **001** **101** **011** **111**
5	00000 10000 01000 … 01111 11111	