There are 1 questions for a total of 20 points.

---

(20) 1. The egg drop problem: There is a building with n floors. You have identical eggs with the property that any egg will break if and only if thrown from floor B and above. If an egg breaks, you cannot fix and reuse it. You want to determine the value of B. Consider the case where you only have one egg. In this case, the worst-case number of trials needed is $n-1$ since the value of B may be equal to n and the only strategy is to start dropping the egg from floor $1, 2, 3, ..., n-1$. Now, suppose you have two eggs. Consider the strategy that minimizes the number of trials in the worst case. Let this worst-case number of trials be denoted by $T(n, 2)$. Similarly, we can define $T(n, 3), T(n, 4), ...$ (for 3 eggs, 4 eggs etc.). Design an algorithm to compute the value of $T(n, k)$ for any given $k(1 \leq k \leq 100)$ and $n(1 \leq n \leq 10000)$. You may assume that $1 \leq B \leq n$.

Your programs should take input from a file named `input.txt` and should write the output in a file named `output.txt`. Your programs should produce an output within 2 minutes for this assignment. The format for input and output files is as follows.

INPUT: The first line of the input file gives n, k. Below is an example of an input file.

```
5,1
```

OUTPUT: The output should be $T(n, k)$. For example, consider the output file corresponding to the input file above:

```
4
```

Note: There are **no** whitespaces or newline characters at the end of the output file. Having such whitespaces might lead to the autograder marking the output as incorrect.

SUBMISSION INSTRUCTIONS: All your program files should be in a directory `hw5_prog`. You will be asked to create a zip of this directory and submit this zip file. In this directory, there should be a makefile that will compile your code (read about makefile on the net in case you do not know what it is). After running `make`, the directory should have an executable called `eggdrop`. This, when executed, should read the input file (`input.txt`) and write the answer in the output file (`output.txt`).