Work Till You Drop (due 30 Oct 2020)

Vyasa has to complete a programming assignment overnight. He has to write n lines of code before morning. He is dead tired and he tries drinking some black coffee to keep him awake. But each time he drinks a cup of coffee he stays awake for a short amount of time but his productivity goes down by a constant factor *k*

This is how he plans to write the program. He will write the first *v* lines of code, then drink his first cup of coffee. Since his productivity has gone down by a factor of *k* he will write *v // k* lines of code. He will have another cup of coffee and then write *v // k**2* lines of code. He will have another cup of coffee and write *v // k**3* lines of code and so on. He will collapse and fall asleep when *v // k ** p* becomes 0.

Now Vyasa does want to complete his assignment and maximize on his sleep. So he wants to figure out the minimum allowable value of *v* for a given productivity factor that will allow him to write at least n lines of code before he falls asleep.

Input: You will read your input from standard input as given in the following format work.in:

```
2
300 2
59 9
```

The first line is *T* the number of test cases. This will be followed by *T* lines of input. Each line of input will have two numbers *n* and *k*. *n* is the number of lines of code to write and *k* is the productivity factor, where $1 \leq n \leq 10^6$ and $2 \leq k \leq 10$.

Output: For each test case write your result to standard out as shown in work.out. In your output there will be *v* lines of code the Vyasa has to write, as well as the time it took for each function. For the above two test cases, the output will be:

```
Binary Search: 152
Time: 9.512901306152344e-05

Linear Search: 152
Time: 0.0005910396575927734


Binary Search: 54
Time: 4.696846008300781e-05

Linear Search: 54
Time: 9.012222290039062e-05
```

Do not worry if your times don't match exactly. For this assignment, main has been written completely for you, and nothing needs to be changed in it.

You will be solving this problem in 2 ways. First, you will write a function that uses a linear search to solve the problem. Then you will write a function that uses a modified binary search algorithm to solve it again. Both functions will return the same answer, but the binary search method will usually be faster.

It is recommended that you write a helper function, which given a value v representing the number of lines Vyasa writes before his first cup of coffee and a value k, the productivity factor, will calculate the number of lines Vyasa will write before falling asleep. This can be called in both the linear and binary functions to make the computations easier.

You must complete this assignment on your own. You may not acquire from another person or the internet a partial or complete solution to this assignment. You may not show another student your solution to an assignment. You may not have another person (other than the instructor, TAs, or proctors)

walk you through the solution. You may only discuss general ideas and approaches with other students but you may not develop code together.

Here is the template of the file Work.py that you will be submitting. You may **NOT** change the names of the functions but you may add as many helper functions as needed. You will follow the standard coding conventions in Python. Your file will have the following header:

```
#  File: Work.py

#  Description:

#  Student Name:

#  Student UT EID:

#  Course Name: CS 313E

#  Unique Number:

#  Date Created:

#  Date Last Modified:
```

You can always add more function than those listed. Mac users will run their program on the command line as follows:

```
python3 Work.py < work.in
```

Windows users will run their code on the command line as follows:

```
python Work.py < work.in
```

Use the Canvas system to submit your **Work.py** file. We should receive your work by 11 PM on Friday, 30 Oct 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

- Your Python program should have the proper header.
- Your code must run before submission.
- You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.