Meet the Word Lock!

m letters per tumbler
- *Chosen from [A-Z]*
- *No lowercase / specials*
- *No duplicates*
- *No blanks*

n tumblers

The Word Lock is a padlock that allows a password to be chosen as the combination. That word is exactly n letters long.

Mike is an employee of the Word Lock Manufacturing Company. He is a Product Manager, responsible for the customer's experience of choosing a password. He knows that customers expect their lock to offer the greatest possible choice of English-language passwords, would only ever use an English-language word as their password, and expect the lock itself not to be cumbersome (i.e. too large an n or m). He observes that competing locks have $4 \leq n \leq 5$, and $6 \leq m \leq 10$.

Mike faces some secondary constraints, too. The legal department has told him that when advertising, he can claim only the number of valid English-language password choices – not the m^n possible lock combinations – which the advertising team wants to maximize. On the other hand, the manufacturing team urges him to keep n and m as low as possible to contain production cost and improve product reliability.

Mike calls Rachel from the Product Security team and says, "The physical security team has built us an indestructible lock! We won't need to worry about someone picking or cutting it. But I'm concerned about the password: how many tumblers will our lock have, how many letters on each, and what letters will those letters be? We have a lot of stakeholders to please, but if these locks are easy to crack it'll be the end of our company. We have to get it right."

PROBLEM 1: CRACK IT

- Define and implement an algorithm to crack a Word Lock password conforming to the constraints outlined above. I will type my password into your program securely, and your code will crack it and provide it back to me in cleartext.

PROBLEM 2: COMPETITIVE ANALYSIS

Your leading competitor's lock has $n = 4$ and $m = 10$, with tumblers $T_1 = \{B, D, G, H, L, M, P, R, S, T\}$, $T_2 = \{A, E, H, I, L, N, O, R, U, Y\}$, $T_3 = \{A, C, E, L, N, O, R, S, T, U\}$, and $T_4 = \{A, D, E, K, L, N, S, T, Y, [blank]\}$. Clearly this design allows for three- or four-letter password choices.

- This competitor advertises "10,000 Possible Choices". Do you agree? On what basis might they have made this claim?
- Assume it takes an attacker 1 second to try a three-letter guess on average, and 1.5 seconds to try a four-letter guess on average. What is the average time it will take to crack the password on this lock?

PROBLEM 3: BUILD A BETTER LOCK

- Define and implement an algorithm that takes integers $n > 0$ and $2 \leq m \leq 26$ as input, and outputs the letter choices for tumblers T_1 through T_n that maximize the number of password choices for those inputs, subject to the product constraints above (in particular, no blanks). If there are multiple sets of tumblers T_1 through T_n that produce the same (*maximum*) number of password choices, only one must be output.
- Create a simple plot of choices for (n, m) with the corresponding maximum number of producible passwords for Product Manager Mike. (You'll need an optimum set of letter choices for each (n, m) pairing, too!). Which do you suggest he choose so as to best satisfy customers, legal, advertising, manufacturing, and security? Why?