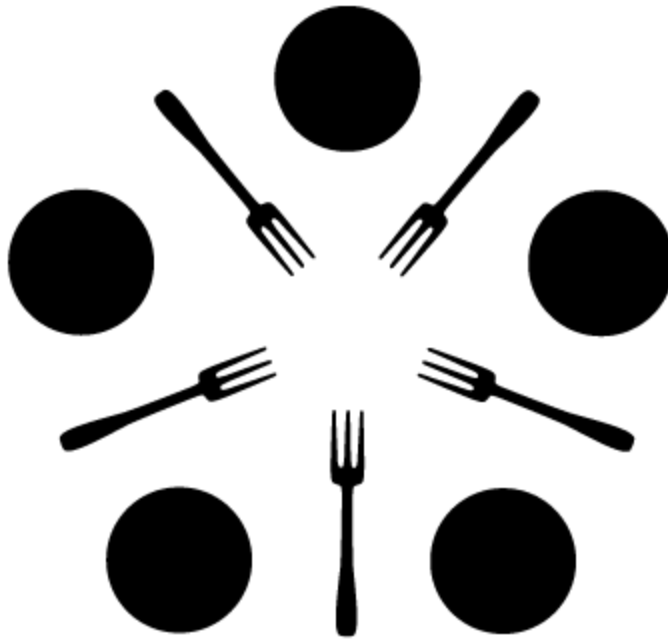


Philadelphia Classic 2013
Hosted by the Dining Philosophers
University of Pennsylvania



dining philosophers
UNIVERSITY OF PENNSYLVANIA COMPUTER SCIENCE CLUB

Basic rules:

4 hours, 9 problems, 1 computer per team

You can only use the internet for accessing the Javadocs, and for PC² (to submit your solutions)

Do not modify any of the given methods! They are all specified in the desired format.

There is no penalty for incorrect submissions. You will receive 1 point per problem solved.

Number of incorrect submissions will be used only as a tiebreaker.

1. Anagrams

In the game Anagrams, you start with a random 3 letter word. To generate longer words, you take the 3-letter word, potentially scramble the letters in the word, and add a letter. For example, CAT could generate CATS, TACK, or PACT. Define a Sequence to be a list of words, $[n_1, n_2, \dots, n_k]$ such that 1) the first word (n_1) is 3 letters long, and 2) n_1 generates n_2 , n_2 generates n_3 , etc. Define an “ n -Sequence” to be a sequence that ends with an n -letter word; i.e. [CAT, PACT, PATCH] is a 5-Sequence.

Your inputs will be an int, n , a string, k , and a sorted list of words representing the dictionary. You can assume that the words in the dictionary are all between length 3 and 8, and consist only of uppercase letters (no hyphens, etc). The dictionary we will be testing with is included in your files as dictionary.txt, and in the main method, we read it into memory.

k will be a 3-letter word from the dictionary, and n will be between 4 and 8. Return the number of n -Sequences that start with k .

Sample Input:

(1): 4, CAT
(2): 5, DOG

Sample Output:

(1): 18 (the 18 words generated by CAT are ACTA, ACTS, CANT, CART, CAST, CATE, CATS, CHAT, COAT, FACT, PACT, SCAT, TACE, TACH, TACK, TACO, TACT, TALC)
(2): 58

2. Palindrome Search

Write a program which finds the longest palindromic substring of a given string. A substring is considered palindromic if reversing the order of the characters in the string results in the same string. (As a note, “a” “ab” are two of the substrings of “abc” but “ac” is not)

Sample Input:

The input will be a string containing at least one palindromic substring of length > 1. You can assume that will consist only of lowercase letters.

(1): “youdontwanttomissracecarsspeedingaroundthetrack”

(2): “ababbabababaaababaabbaa”

Sample Output:

Output the longest palindromic substring of the input string. If there are multiple such substrings of equal length, output the first one.

(1): “ssracecarss”

(2): “ababaaababa”

3. Circular Primes

Write a program which determines whether a given number is a “circular prime”. A number is considered a circular prime if and only if each number created by rotating its base 10 digits is prime. For example, 113 is considered a circular prime because 113, 131, and 311 are all prime.

Sample Input:

The input will be a number of type long. You can assume that numbers given as inputs are greater than 1.

(1): 8675309

(2): 2

(3): 11

Sample Output:

Output the Boolean value true if the number is a circular prime; false if not.

(1): false

(2): true

(3): true

4. Sum of Digits in Bases

You may be familiar with representation of numbers in different “bases”. The standard numeric system is the “decimal” system, or base 10. Another well-known system is binary, or base 2. When counting in base N , each digit can have a maximum value of $N-1$ (so in base 10, the maximum digit value is 9; in base 2, each digit can only be 0 or 1). For bases greater than 10, such as base 16, the digits A, B, C, D, E, and F represent values from 10 to 15.

The concept of sum of digits of a number can easily be extended to any base. If you have the number “AC3” in base 14, for example, the sum of digits should be treated as $10 (A) + 12 (C) + 3 = 25$.

Define a “P-Q-overlap” to be a number which has the same sum of digits in base P and base Q . For example, 75 can be represented as 2210 base 3, and 203 base 6. Since it has a sum of digits of 5 in both bases, it is a 3-6-overlap.

In this problem, you will be given an input, N . Your task is to determine, which *pair* of bases P and Q , between 2 and 16 (inclusive), have the maximum P-Q overlaps over the set of integers from 1 and N , inclusive. If there are more than 1 such pairs P, Q , choose the pair with the highest value of P (where $P < Q$).

Sample Input

(1): 3
(2): 37

Sample Output

If your pair of bases are P and Q , with $P < Q$, return $100 * Q + P$.
(1): 1615
(2): 1508

5. Texas Hold 'Em

Texas Hold 'Em is a common form of poker. From a standard 52 card deck, players are each dealt two cards face down (called their "pocket"), and then five cards are placed face up in the "community". Of the seven cards that each player has to choose from (2 pocket + 5 community), he picks the strongest five-card combination: these five cards make up his "hand"; the player with the best five-card hand wins that game. Any other cards not in the hand do not affect its ranking.

The following rules determine the ranking of hands:

- Ace is the highest-ranking individual card, and 2 is the lowest.
 - Suits do not have an associated value/ranking. They are only used to determine whether a hand forms a flush.
 - A flush is a hand which contains five cards of the same suit.
 - A straight is a hand which contains five cards in sequential rank, e.g. 3-4-5-6-7. For straights, Ace can serve as the lowest (A-2-3-4-5) or highest card (10-J-Q-K-A), but you cannot wrap around (K-A-2-3-4 is not a straight).
 - The ranking of hands is done in categories - any hand in one category beats any hand in a lower category. The categories are ordered below, and include the method of ranking hands in the same category.
 - Some categories (such as four of a kind, pair, etc) are specified by fewer than five cards (whereas a straight and full house are specified by all five cards in the hand); the additional cards in the hand are called "kickers".
1. Straight flush: A hand with five cards of the same suit in sequence. Two such hands are compared by their card which is ranked highest (a A-2-3-4-5 straight flush is lower than a 2-3-4-5-6 straight flush).
 2. Four of a kind: A hand with four cards of the same rank, and any other card (kicker), e.g. 3-3-3-3-7. Two such hands are compared by the rank of the four of a kind card; if that is the same, they are compared by the rank of the kicker.
 3. Full House: A hand with three cards of one rank and two cards of another rank, e.g. 3-3-3-2-2. Two such hands are compared by the rank that has three cards; if that is the same, they are compared by the rank that has two cards. So 3-3-3-2-2 defeats 2-2-2-K-K.
 4. Flush: Five cards of the same suit. Flushes are compared by the rank of their highest card; if that is the same, then by their second-highest card; if that is the same, third-highest, etc.
 5. Straight: Five cards in sequence. Like straight flushes, compared by the highest-ranked card.
 6. Three of a kind: A hand with three cards of one rank, and two cards of different ranks (each different from each other), e.g. 7-7-7-9-2. Compared by the rank of the three of a kind card; if that is the same, then by the highest kicker; if that is the same, then by the second kicker.
 7. Two Pair: A hand with two cards of one rank, two cards of another rank, and another card of a third rank. Compared by the rank of the highest pair, else the rank of the second pair,

else by the kicker.

8. One Pair: A hand with two cards of one rank, and three cards of different ranks, each different from each other. Compared by the rank of the pair, and then by the kickers in descending order.
9. High Card: A hand meeting none of the above categories. Compared by the rank of the highest-ranked card; if those are equal, then by the second-highest, etc.

You will be given two players' pocket cards, and the five cards in the community (you can assume that the cards are all distinct). Each card will be represented as a concatenation of two values, where the first value represents the card rank (2 to 10, J, Q, K, and A) and the last value represents the suit (H, S, C, D). All letters will be in uppercase. For each player and the community, each card will be separated from the other cards by a space, and there will be no leading or trailing spaces.

Your task is to determine whether player 1 has a better hand than player 2.

Sample Input:

(1):

Player 1: AH AS

Player 2: AD AC

Community: 3H 5H 7H 9H JD

(2):

Player 1: 7H 4C

Player 2: AD 7C

Community: AH QS 5H QD JH

Sample Output:

(1): true

(2): false

6. Spiral Diagonal

Starting with the number 1 and moving to the right in a clockwise direction a 5 by 5 spiral is formed as follows:

```
 21 22 23 24 25
 20  7  8  9 10
 19  6  1  2 11
 18  5  4  3 12
 17 16 15 14 13
```

It can be verified that the sum of the numbers on the diagonal from the upper left to the bottom right is 45.

Determine the sum of the numbers on the diagonal from the upper left to the bottom right of a N by N spiral formed in the same way.

Sample Input

The input will be the number N. N is guaranteed to be odd.

(1): 5

(2): 3

Sample Output

(1): 45

(2): 11

7. Lucky Numbers

In number theory, a lucky number is a natural number generated by a “sieve”, a type of sequence-generating operation. You start with the set of odd positive integers, and remove every n -th number, where n is the **next surviving number** in the sequence. This process is repeated over and over ad infinitum. 1 is skipped, for obvious reasons, so n begins with $n = 3$.

Start with the set of set of odd numbers: 1,3,5,7,9,11,13,15,17,19,21...

The next surviving number is 3, so $n = 3$, and every third remaining number is eliminated.

Set after removing every third number 1,3,7,9,13,15,19,21...

The next surviving number is 7, so $n = 7$, and every seventh remaining number is eliminated.

Set after removing every seventh number: 1,3,7,9,13,15,21...

...and so on.

A number is lucky if it is never removed from the set.

Sample Input

The input will be a number of type int, and will be less than 2 million.

(1): 1

(2): 19

(3): 21

(4): 808342

Sample Output

Return a boolean indicating if the given integer is considered lucky.

(1): true

(2): false

(3): true

(4): false

8. Equation Solver

In this problem, we will ask you to implement a basic equation solver. Don't worry - it's only of one variable, and the maximum degree of the equation is two.

Your input will be a string which contains the equation. Terms will be separated from operators (+, -, or =) by spaces, but each term will not have any spaces in it. Each term can have an integer coefficient, and an x term, which can be to the first power (no exponentiation) or the second power. The "^" (carat) symbol will be used to denote exponentiation of the x term. A possible input is:

" $3x^2 + x + 3 = 3 + 5x^2 + -7x$ ". You must be able to solve for x for all equations of this form. There may be multiple terms of the same degree on one side of the equation.

You can assume that the equation will have a solution; if there is more than one solution, output the greater of the two solutions, rounded to the nearest integer.

Sample Input

(1): " $3x^2 + x + 3 = 3 + 5x^2 + -7x$ "

(2): " $5x + x - 37 = 0$ "

Sample Output

(1): 4

(2): 6

9. Next!

Given a set of digits $\{1, 2, \dots, n\}$, we can generate a list of all n -digit numbers that use each of those digits exactly once. If $n = 3$, the set of such numbers is $\{123, 132, 213, 231, 312, 321\}$. It is straightforward to order these numbers numerically. For this problem, your challenge is, given one number in this list, to generate the next largest element in the list. You can assume that you will not be given the largest element in the list.

Sample Input:

You will be given a string representing a “number”, up to 30 digits long (for $n > 10$, use A = 10, B = 11, C = 12, D = 13, E = 14, F = 15, G = 16, H = 17, I = 18, J = 19, K = 20, L = 21, M = 22, N = 23, O = 24, P = 25, Q = 26, R = 27, S = 28, T = 29, U = 30). All digits in the number will be distinct. All letters will be in uppercase.

(1): 132
(2): 123456789ABCDEFGHIJK
(3): 35241

Sample Output:

(1): 213
(2): 123456789ABCDEFGHIJK
(3): 35412

PClassic 2013

Please give us your input about the event, so we can make it better next year!

Please rate the following:	1 - Poor	2 - Fair	3 - Okay	4 - Good	5 - Awesome
Overall competition					
Registration process					
Android Tutorial					
Competition Instructions					
Competition Room					
Questions					
Lunch					

What was your favorite part of this event?

What would you change/improve about the event?

I am a Student _ Teacher _ representing _____ High school