Stanford ProCo

May 26, 2012

Problem Packet
Novice Division

Sponsored by:

Overview: Print a string.

Description: Life as Tom Riddle is tough. Not only are you destined to spend your days stuck in a book until your infallible schemes succeed, your only means of escape is through the mind of some silly eleven year old who's stumbled across your diary. Sadly, she seems resistant to your well-meant instructions, but you've developed a fool-proof strategy.

To wreak havoc in the castle, you've already had her petrify a rather annoying cat who's hissed at you one time too many. Now, you need to send her a message to write on the wall and send the populace into a frenzied panic. The message is below. Remember, if you don't have her write the message exactly as given, she won't succeed and your quest for escape will fail!

Filename: nov21.{java, cpp, c, cc, py}

Input: There is no input for this problem.

Output: Print the following string exactly as shown, on two lines, including the quotes:

```
"The Chamber of Secrets has been opened.
Enemies of the heir, beware."
```

Assumptions: No assumptions.

Sample Input #1: There is no input for this problem.

Sample Output #1:
```
"The Chamber of Secrets has been opened.
Enemies of the heir, beware."
```

Overview: Find the range of a sequence of integers.

Description: Draco Malfoy is celebrating! He just received the highest score on a recent charms exam, besting all those brainy Ravenclaws, not to mention that pesky Hermione Granger. Now, he is desperate to know by now much he beat the worst person (either Crabbe or Goyle). Luckily, he has sneakily obtained a list of test scores, expressed as a sequence of integers. Help Draco find the difference between the maximum and minimum integers so he can know how much he gets to gloat!

Filename: nov22.{java, cpp, c, cc, py}

Input: The first line contains an integer n representing the number of test scores. The next line contains the n test scores.

Output: Print a single integer indicating the range of the given sequence.

Assumptions: $1 \leq n \leq 1000$
The test scores will be integers between -100,000 and 100,000, inclusive.
All input will be valid.

Sample Input #1:
```
5
1 20 -5 6 8
```

Sample Output #1:
```
25
```

Sample Input #2:
```
2
100 99
```

Sample Output #2:
```
1
```

Overview: Find the penultimate occurence of a character in string

Description: Harry knows that the last Horcrux is himself, so he needs your help in finding the penultimate Horcrux. He has lined up all the objects that could be Horcruxes. Each object is represented by a character and Horcruxes have a special character representing them. Write a program to find the second to last occurrence of a Horcrux. Good luck!

Filename: nov23.{java, cpp, c, cc, py}

Input: The first line contains a single lowercase (a - z) character *ch* indicating the character of the Horcrux to be found. The second line contains a string *s*.

Output: Print the position of the penultimate occurrence of *ch* in *s*, where the first character of *s* is at position 0. If *ch* only occurs once or not at all, output `-1`.

Assumptions: The first line will have exactly one character, and the first line will be a lower case letter.
s will have between 1 and 1000 characters, inclusive.
s will contain only lowercase (a - z) characters and space characters.
All input will be valid.

Sample Input #1:
```
h
the quick brown dementor jumps over the quaffle
```

Sample Output #1:
```
1
```

Sample Input #2:
```
q
hufflepuffs like hippogriffs
```

Sample Output #2:
```
-1
```

Overview: Figure out if a sequence is arithmetic, geometric, both, or neither.

Description: Harry Potter is duelling Voldemort to save the wizarding world. The best duellers surprise their opponents, casting spells that are hard to predict. After all, if you can predict your opponent's next move, you can defend against it more effectively. You notice that Voldemort is not always casting unpredictable spells; sometimes his spell sequences form patterns.

In an *arithmetic sequence* of numbers, each term is the previous term plus a constant (i.e., a, $a+b$, $a+2b$, $a+3b$, …). In a *geometric sequence* of numbers, each term is the previous term multiplied by a constant (i.e., a, ar, ar^2, ar^3, …). Help Harry save the entire wizarding community by analyzing Voldemort's sequences.

2, 4, 6, 8, who do we appreciate? Harry will appreciate you!

Filename: nov24.{java, cpp, c, cc, py}

Input: The first line contains an integer n representing the number of terms in the sequence. The next line contains n space-separated integers $x_1 \ldots x_n$ representing the given sequence.

Output: Print the type of the sequence $x_1 \ldots x_n$, which will be `ARITHMETIC` if the sequence is arithmetic but not geometric, `GEOMETRIC` if the sequence is geometric but not arithmetic, `BOTH` if the sequence is both arithmetic and geometric, or `NEITHER` if the sequence is neither.

Assumptions: $1 \le n \le 1{,}000$
Each x_i will be between -30,000 and 30,000, inclusive.
All input will be valid.

Sample Input #1:
```
1
5
```

Sample Output #1:
```
BOTH
```

Sample Input #2:
```
5
-1 1 3 5 7
```

Sample Output #2:
```
ARITHMETIC
```

Overview: Find the length of the longest run in a string.

Description: A group of Gryffindors are playing a variant of Exploding Snap. Each player gets to play a sequence of cards. They can continue playing until one of their cards explodes, at which point the next person gets a turn. The winner is the person who plays the most cards in a row.

A single string can be used to represent this game: each player is represented by a distinct letter or digit, and the string represents the sequence in which players played their cards. For example, "`aaabba`" means that player "a" played three cards in a row, and then player "b" played two cards and then "a" played one card.

Hermione is looking for a strategy in this game. In particular, she's interested in how many cards in a row it takes to win. Write a program that takes in a game string and outputs the number of cards that it took to win in that game. In the above example, the number of cards to win is three.

Filename: nov51.{java, cpp, c, cc, py}

Input: The first line contains an integer *n* representing the length of the string *s*. The second line contains *s*.

Output: Print a single integer representing the highest number of cards in a row played by any player.

Assumptions: $1 \le n \le 100$
s will only contain lowercase alphanumeric (`a` - `z`, `0` - `9`) characters.

Sample Input #1:
```
3
aab
```

Sample Output #1:
```
2
```

Sample Input #2:
```
10
abbb123333
```

Sample Output #2:
```
4
```

Overview: Find the smallest integer that can be constructed from a given set of digits.

Description: You are in transfiguration class with Professor McGonagall. Transfiguration is the study of transforming one object into another object, a bullfrog into a broomstick. However, before you can transform bullfrogs to broomsticks, you must first practice on nonliving things, such as strings of digits.

Professor McGonagall has given you your first assignment: given a set of digits `0` - `9`, find the smallest integer that can be formed from those digits. She has promised you that once you have mastered this transfiguration assignment, you can move on to transforming bullfrogs to broomsticks!

Filename: nov52.{java, cpp, c, cc, py}

Input: The first line contains an integer *n* representing the number of digits in the string. The second line contains a string *s* of *n* digits `0` - `9`.

Output: Print the smallest integer that can be formed from the given digits, with no leading zeros.

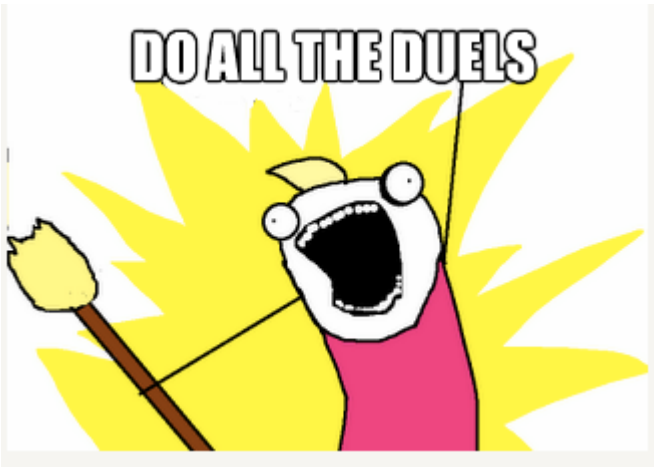
Assumptions: $1 \le n \le 1{,}000$
There will be at least one non-zero digit in *s*.

Sample Input #1:
```
11
87654332211
```

Sample Output #1:
```
11223345678
```

Sample Input #2:
```
3
065
```

Sample Output #2:
```
506
```



Overview: Report the final score of a two-player duel.

Description: You just joined the Hogwarts dueling club! You are so excited for your first practice duel the next day that you decide to read up on duel scoring. At each round of the duel, one wizard is designated as the attacker and the other is the defender. When an attacker wins a round, s/he gets a point and becomes the attacker for the next round. When a defender wins a round, s/he gains no points but becomes the attacker the next round. Practice your new dueling knowledge by calculating the scores of the given duels!

Filename: nov53.{java, c, cpp, cc, py}

Input: The first line contains an integer *n* representing the number of rounds in the duel. The second line contains exactly *n* digits, where each digit is either a `0` or a `1`. A `1` corresponds to your winning the round, and a `0` corresponds to your losing the round.

Output: Print your score, followed by a colon, followed by your opponent's score.

Assumptions: You start the first round as the attacker.
$1 \le n \le 1{,}000{,}000$

Sample Input #1:
```
16
0001111010110011
```

Sample Output #1:
```
5:3
```

Sample Input #2:
```
2
10
```

Sample Output #2:
```
1:0
```

Overview: Convert a number from one base to another.

Description: Hermione is trying to send a number to Harry telling him her whereabouts, but she doesn't want anyone else, especially the Death Eaters, to be able to read it. So she decides to foil any eavesdroppers by transforming the number prior to transmission.

Traditionally, numbers are represented in base 10, but numbers can also be represented in other bases. A number $a_d a_{d-1} ... a_0$ base b, where $a_d, a_{d-1}, \ldots, a_0$ are the digits, has value $a_d b^d + a_{d-1} b^{d-1} + \ldots + a_0 b^0$. Furthermore, each digit a_i is between 0 and b-1 inclusive. For example, 10 base 7 is the same as 7 base 10 and 123 base 7 is the same as 66 base 10. Hermione decides to transform her number from one base to another. To further confuse eavesdroppers, she uses lowercase alphabetic characters instead of numeric digits. That is, a represents 0, b represents 1, etc. In base h (7), ca (20) would be equal to be (14) base k (10).

Filename: nov54.{java, cpp, c, cc, py}

Input: The first line contains an integer n representing the number of digits in the input string s. The second line contains s, and the third line contains two characters b_s b_e representing the starting and ending bases, respectively.

Output: Print a string of characters representing s in base b_e, with no leading zeroes (a's).

Assumptions: s will represent a valid integer in b_s with no leading zeroes (a's).
s base b_s will have value $< 2^{31}$.
Neither b_s nor b_e will be a or b.

Sample Input #1:
```
4
ccac
kc
```

Sample Output #1:
```
baaabaabbaba
```

Sample Input #2:
```
1
h
kh
```

Sample Output #2:
```
ba
```

Overview: Find the minimal total King distance from a point to all of the other points

Description: Hermione has finally convinced Harry to hold meetings for Dumbledore's Army again, yay! Sadly, the Room of Requirement is not an option, as it has been compromised. Instead, Harry must determine where the best new meeting place is, considering all *n* members of the D.A. are at different locations on a grid. Out of sheer laziness, Harry plans to have the D.A. meet at one of these *n* locations.

The best meeting point is defined as the one that minimizes the sum of King distances between the chosen location and the other *n*-1 points. The King distance between two points is defined as the number of steps it takes to go from one point to the other, at each step travelling to one of the 8 surrounding lattice points (much like a King on a chessboard). For example, it takes one step to go from (0, 0) to the points (0,1), (1,1), (1,0), (1,-1), (0,-1), (-1,-1), (-1,0), and (-1,1).

Math was never Harry's strong point, so he's having trouble figuring out where to meet. Can you help him?

Filename: nov91.{java, cpp, c, cc, py}

Input: The first line contains an integer *n* representing the number of meeting points. The next *n* lines each contains two integers *x y* representing the coordinates of the possible meeting points.

Output: Print the minimum total King distance of the best meeting point.

Assumptions: $1 \le n \le 1{,}000$
Each coordinate will have absolute value ≤ 1,000,000,000

Sample Input #1:
```
2
1 1
1 2
```

Sample Output #1:
```
1
```

Sample Input #2:
```
3
2 1
1 2
0 3
```

Sample Output #2:
```
2
```

Overview: Sort a list of strings from shortest to longest.

Description: "*Detention! And twenty points from Slytherin!*" - Professor McGonagall

Gah! You have just been assigned detention with Professor Snape - for *walking disrespectfully*. Tonight your task is to sort the numerous ingredients in his potions cupboard. Unfortunately, Professor Snape is a particular man; instead of sorting the ingredients in alphabetic order, you must sort the string on each ingredient's label from shortest to longest. If two strings are the same length, then they must be sorted in alphabetical order. To prove to Snape that you have properly sorted everything, you must also show him the ingredient labels found at particular indices. Here's to hoping that you'll finish before curfew!

Filename: nov92.{java, cpp, c, cc, py}

Input: The first line contains an integer m representing the number of strings in the list. In the next m lines, the i-th line ($1 \le i \le m$) contains a single string s_i representing the label of ingredient i. The next line contains a single integer n representing the number of indices to print out. In the next n lines, the j-th line ($1 \le j \le n$) contains a single integer n_j indicating an index to print out.

Output: Print exactly n lines, with the j-th line ($1 \le j \le n$) containing the string at index n_j after sorting.

Assumptions:
$1 \le m \le 20{,}000$
$1 \le n \le m$
$0 \le n_j < m$, for all j
All strings s_i will contain only uppercase (A - Z) characters.
All strings s_i will have length between 1 and 10, inclusive.
All strings s_i will be unique.
All indices are 0-based.

Sample Input #1:
```
2
AAA
AA
1
1
```

Sample Output #1:
```
AAA
```

Sample Input #2:
```
3
C
PQ
RT
2
2
0
```

Sample Output #2:
```
RT
C
```

Overview: Find the minimum number of steps to reach a destination.

Description: Although ubiquitous in proper Wizarding society, magical transportation is a complex field of research that many Unspeakables delight in studying. Before his arrest in 1996, Augustus Rookwood had developed a novel mode of transportation called *apparition tokens* that may very well have turned the tide in the Death Eaters' favor, were it not for certain limitations.

Apparition tokens are simply a set of n magically interlinked tokens, each infused with a number and spread throughout the land. Based on the number on the token, a person who knows of the token's existence can instantly travel that many tokens forward or backward, as long as the new token is valid. For example if token 5 had been infused with "2", a Death Eater at that token could travel to token 3 or token 7.

Death Eaters could use these secret tokens to travel about and bypass any pesky anti-apparition wards. But Rookwood faced a practical challenge: apparition tokens are magically taxing, and it is therefore optimal to take as few steps as possible to travel from one token to another. Rookwood never took a programming class so he had no clue how to solve this problem, but fortunately, you have! Assuming you start at tile 1, what is the minimum number of steps you must take to reach tile k?

Filename: nov93.{java, cpp, c, cc, py}

Input: The first line of input contains two integers n k as defined above. In the next n lines, the i-th line ($1 \le i \le n$) contains an integer s_i representing the number infused with token i.

Output: Print the minimum number of steps to move from tile 1 to tile k, or `-1` if no such path exists.

Assumptions: $1 \le k \le n \le 1{,}000$
$0 \le s_i \le 1{,}000$ for all i

Sample Input #1:
```
6 2
2
1
2
1
1
4
```

Sample Output #1:
```
9
```

Sample Input #2:
```
6 2
2
1
2
1
1
5
```

Sample Output #2:
```
-1
```

Overview: Does the index of any number in a sorted array equal its value?

Description: Harry, Ron, and Hermione are attempting to reach the sorcerer's stone before Voldemort can get there! Unfortunately, they have reached a seemingly unsolvable puzzle. In front of them is a line of n numbered boxes, from 0 at the very left to n-1 at the far right. Each box can be opened to reveal a number inside. Behind the boxes are two doors. A note next to the first box explains the puzzle:

Each of these boxes contains a number, and the numbers are in sorted order from left to right. If one of the boxes contains a number that matches the number on the outside, behind the right door is a tiger that will devour you, and behind the left is the sorcerer's stone. If there is no such box, the doors are reversed. However, be warned! You may open at most 42 boxes before all the boxes explode and you are trapped forever!

Help Harry, Ron, and Hermione choose the safe door! You are given the size n of a hidden sorted array. By querying the values in the array at most *42* times, determine whether the value of some element in the array equals its index.

Filename: nov94.{java, cpp, c, cc, py}

Input / Ouput: This is an interactive problem. This means that your program will receive input based on the output your program produces. When your program starts, you will be provided an integer n corresponding to the size of the array A. Your program must then do one of the following:

1. Output an integer i ($0 \le i < n$), after which your program will be provided the integer $A[i]$. This counts as 1 query.
2. Output -1, indicating that there is some i such that $A[i] = i$
3. Output -2, indicating that for all i, $A[i] \ne i$

You MUST output a new line character and flush the output stream after each output:

In C, use `printf("\n"); fflush(stdout);`
In C++, use `cout << endl << flush;`
In Java, use `System.out.println(); System.out.flush();`
In Python, use `sys.stdout.write("\n"); sys.stdout.flush();`

Assumptions: $1 \le n \le 10{,}000$
All values in A are distinct.
All values in A are between -1,000,000,000 and 1,000,000,000, inclusive.
All indices are 0-based.

Sample Sequence #1: Hidden sequence: 0 1 2 3 4 5 6 7 8 10

```
COMPUTER  10
YOU       0
COMPUTER  0
YOU       -1
```

Sample Sequence #2: Hidden sequence: 100000 100001

```
COMPUTER  2
YOU       0
COMPUTER  100000
YOU       1
COMPUTER  100001
YOU       -2
```