New Mexico State University
Programming Competition
Fall 2022

Problem Set

Problem 1: Factorize

Point Value: 3

Bounty: $30

Description:
Given a positive integer `n`, output all of its integer factors greater than 1. If a factor appears multiple times, output it as many times as it is a factor for n. Output factors in order from smallest to largest.

```
2 <= n <= 1000000
```

Sample Input:
```
2
100
7919
```

Sample Output:
```
2
2 2 5 5
7919
```

Problem 2: Morse Code

Point Value: 3

Bounty: $30

Description:
Morse code is a signaling protocol used to communicate using patterns of signals of alternating lengths. There are two signal lengths: short and long, which are typically represented as the characters '.' and '-' respectively. Below is a table of the morse code patterns for all 26 characters of the English alphabet.

CHARACTER	CODE	CHARACTER	CODE
A	.-	B	-...
C	-.-.	D	-..
E	.	F	..-.
G	--.	H	....
I	..	J	.---
K	-.-	L	.-..
M	--	N	-.
O	---	P	.--.
Q	--.-	R	.-.
S	...	T	-
U	..-	V	...-
W	.--	X	-..-
Y	-.--	Z	--..

Given a sequence of characters in morse code, compute the equivalent sequence in the English alphabet. Each morse code character will be separated by a space. There will be an arbitrary number of input morse code words, each appearing on their own line. Output each equivalent English word on its own line. Read input until EOF is encountered.

Sample Input:

```
.... . .-.. .-.. ---
..- -. -.-. --- .--. -.-- .-. .. --. .... - .- -... .-.. .
```

Sample Output:

```
hello
uncopyrightable
```

Problem 3: Data Cleaning

Point Value: 3 **Bounty:** $30

Description:
You will be given an n by m matrix representing a data table. Values in a row are separated by spaces (0x20). Rows are separated by newlines (0x0A). Unfortunately, some of the values in the table have been corrupted. Apply the following data cleaning rules to the data table and then print out a cleaned version of the table:

1. Any entry containing any character other than a letter or a number must be replaced with a -1
2. Any row with more or less than m columns in it must be deleted, and the size of n updated.
3. All letters must be set to upper case letters.

You can assume the input will be encoded in ASCII. The first two lines of input will tell you n and m. The first two lines of your output must be the final value of n and m. If you filter out all rows, return an n and m of 0.

Input Format:
```
n
m
# # # # # ... #
# # # # # ... #
# # # # # ... #
# # # # # ... #
# # # # # ... #
...
# # # # # ... #
```

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

Sample Output:
```
2
3
0 APPLE 1
1 0 0
```

Sample Input:

```
2
5
[NaN] ??? 05 0a 1soda
pizza 0 a33 1 1 34
```

Sample Output:

```
1
5
-1 -1 05 0A 1SODA
```

Problem 4: Rhyming Numbers

Point Value: 4 **Bounty:** $30

Description:
Given two sets of numbers, return the number of unique "rhyming pairs" of numbers that contain one number from each set. One number can be used in multiple rhyming pairs.

Two numbers "rhyme", if their last 3 digits are identical. For example, 13567 rhymes with 567. All numbers will be positive. Numbers less than three digits long should be zero-padded when testing whether two numbers "rhyme".

Sample Input:
```
[4134,5136,3166,1100]
[444,536,5100,3166]
```

Sample Output:
```
2
```

Sample Input:
```
[4134,5136,3100,1100]
[444,3134,5100,3100]
```

Sample Output:
```
5
```

Sample Input:
```
[4134,3100,3100,1100]
[444,3134,5100,3100]
```

Sample Output:
```
5
```

Sample Input:
```
[2001,1]
[1001]
```

Sample Output:
```
2
```

Problem 5: Deep Palindromes

Point Value: 6 **Bounty:** $75

Description:
A string is a palindrome if it is equal to its reverse, i.e. it reads the same forward and backward, e.g. "racecar". In this problem, we will consider deep palindromes. A deep palindrome is a palindrome which when split in half renders two identical palindromes. If a string has an odd number of characters, we ignore the middle character when checking if it is a deep palindrome. As an example, "abacaba" is a deep palindrome.

We can have deep palindromes several levels deep. For a second level deep palindrome, each half of the initial string is a first level deep palindrome. For example, "abacabadabacaba" is a third level deep palindrome as shown below

`abacabadabacaba`	deep palindrome level 3
`abacaba`	deep palindrome level 2
`aba`	deep palindrome level 1
`a`	palindrome

Given a list of strings, each one appearing on its own line, report whether the string is a) "not a palindrome", a "palindrome", or a "deep palindrome level <n>". Input strings will not contain spaces and will be delineated by return lines. Input deep palindromes may be as deep as 10 levels.

Sample Input:
```
Hello!
racecar
abacaba
abacabadabacaba
```

Sample Output:
```
not a palindrome
palindrome
deep palindrome level 2
deep palindrome level 3
```

Problem 6: Pathfinding

Point Value: 7 **Bounty:** $75

Description:
You are given an n by n matrix G of ones and zeros representing connections between a graph of n nodes. If G[i,j] is a one, then there exists a directional edge from node i to j. The syntax is G[row, column]. Given two nodes a and b, return the minimum number of edges on a path between a and b. If no such path exists, return -1.

Input Format:

```
a
b
n
# # # # # ... #
# # # # # ... #
# # # # # ... #
# # # # # ... #
# # # # # ... #
...
# # # # # ... #
```

Sample Input:

```
0
1
3
0 1 1
1 0 0
1 1 0
```

0
1
2

Sample Output:

```
1
```

Sample Input:

```
0
1
5
0 0 0 0 1
0 0 1 1 1
0 1 0 0 1
0 1 0 1 0
1 1 1 0 0
```

a
b

Sample Output:

```
2
```

Sample Input:

```
0
1
3
0 0 1
0 0 0
0 0 0
```

Sample Output:

```
-1
```

Problem 7: Burger Margin Maximization

Point Value: 9 **Bounty:** $75

Required Runtime: < 1 minute

Description:
Bill is opening up a burger joint. His plan is to specialize in big burgers. However, as it is notoriously difficult to be profitable in the restaurant business, Bill wants to make sure to maximize his revenue everyday.

Bill's burger joint sells hamburgers with varying numbers of ¼ pound patties, each with its own price. However, the butcher shop where Bill gets his hamburger meat sells him an unspecified amount of meat everyday.

Given a listing of all of the burgers (by number of patties) that Bill sells, their prices, and a whole number of pounds of hamburger meat available, compute the maximum gross sales that Bill can possibly make that day.

The first line of input will be the whole number of pounds **w** of hamburger meat sold to Bill. Note this value is in pounds, and each patty is ¼ pound. So with 7 pounds of hamburger, 28 ¼ pound patties can be made.

The second line of input is how many different sizes of burger, **b**, that are on the menu.

The following lines will list the burger prices. Burgers prices will be listed as an integer number of patties followed by the floating point price. Each patty number price pair will appear on their own line. There will always be a burger with just one patty on the menu. You must output the maximum possible gross sales to two decimal places.

Input bounds:
$0 <= w <= 10000$ $1 <= b <= 100$

Sample Input:
```
5
4
1 5.50
2 11.50
3 17.30
8 41.10
```

Sample Output:
```
115.30
```

Problem 8: Finding Escaped Cows

Point Value: 17 **Bounty:** $300

Bounty: $300

Description:
A cattle rancher has purchased some newfangled GPS trackers for each of her n cows. She has also placed one on each of the m fence posts of her pasture. The locations of cows and fence posts are given as integer coordinate pairs in R_2 separated by spaces. Return the coordinate pairs of any cows outside the pasture fence. You are guaranteed that the shape of the pasture is convex. You are guaranteed no cows will be exactly on any fenceline. You are not guaranteed to be given the list of fence posts in the order they are connected. There will be no duplicate points.

Sort your list of lost cows from largest to smallest x, with the largest y acting as tiebreaker.

Input Format:

n

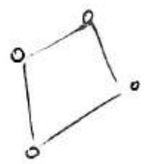
m

(x_1, y_1) $(x_2, y_2) \dots (x_n, y_n)$

(x_1, y_1) $(x_2, y_2) \dots (x_m, y_m)$

Sample Input:

```
3
4
(0,0) (2,2) (300,300)
(-1,1) (-1,-1) (1,1) (1,-1)
```



Sample Output:

```
2
(2,2) (300,300)
```

Sample Input:

```
10
9
(10,-7) (1,-8) (6,-3) (2,-1) (0,0) (-3,3) (-3,7) (1,3) (8,3) (-1,-1)
(11,0) (-2,3) (1,4) (5,-9) (0,-5) (-3,2) (11,-10) (4,3) (7,2)
```

Sample Output:

```
4
(-3,3) (-3,7) (1,-8) (8,3)
```