March 18. 2022

Table of Contents

Guess the passcode **2**
The Martian Clock **4**
PacManFun **7**
POWAGRAMs **10**
Smart Penalty System (Saher2) **12**
AI powered factory **16**
Selling my 3D Game **19**
TROJENA1 ..... **21**
Extrinsic Integers **23**
Saudi Arabia Railways Company **26**
Which candies can I get? **29**
My autonomous car **32**

Guess the passcode

Saleh and his friends discovered a large cache of text files. Apparently, these contain information about various bitcoins. The text files have alphanumeric filenames of length 20 and are passcode protected. After many attempts to recover the contents of the text files they were about to give up when they discovered one of the text files containing instructions to determine the passcode. Unfortunately, this file is corrupted. Only the following information could be retrieved.

The passcode is determined by reading the file name. The filename may contain up to 3 patterns as follows:

1. Pattern **"ah"**: Letters 'a' and 'h' should occur together in the string. Letter 'a' should occur first followed by letter 'h', with zero or more letters in between.
2. Pattern **"ej"**: Letter 'e' should occur first followed by letter 'j', with zero or more letters in between.
3. Pattern **"ly"**: Letter 'l' should occur first followed by letter 'y', with zero or more letters in between.

If any of the patterns above are found in the filename, the passcode can be determined easily using the pattern number as a guide.

Example: For a file name **"a0Bdh332gl"**, the passcode would be **1** as only one Pattern *"ah"* is found.

Nested patterns, for example, **"abdeeealjjhjtu"**, are reported only once, i.e. Passcode would be **12**. Only Pattern 1 letters *'a'* and *'h'*; and Pattern 2 letters *'e'* and *'j'* are detected.

For any invalid input, the corresponding Passcode would be **0**.

Help Saleh write a program to read data from the console and determine the passcode for the data.

Input Format

The first line consists of an integer N followed by N number of lines containing alphanumeric strings of various lengths. These are the filenames.

Constraints

```
N > 1
0 < Filename_length <= 20
```

Output Format

For each filename, a corresponding passcode is generated.

Sample Input 0

```
4
abcdehmnj
abdeeealjjjhtuy
aaahhhlll
aaaleeeejlyll
```

Sample Output 0

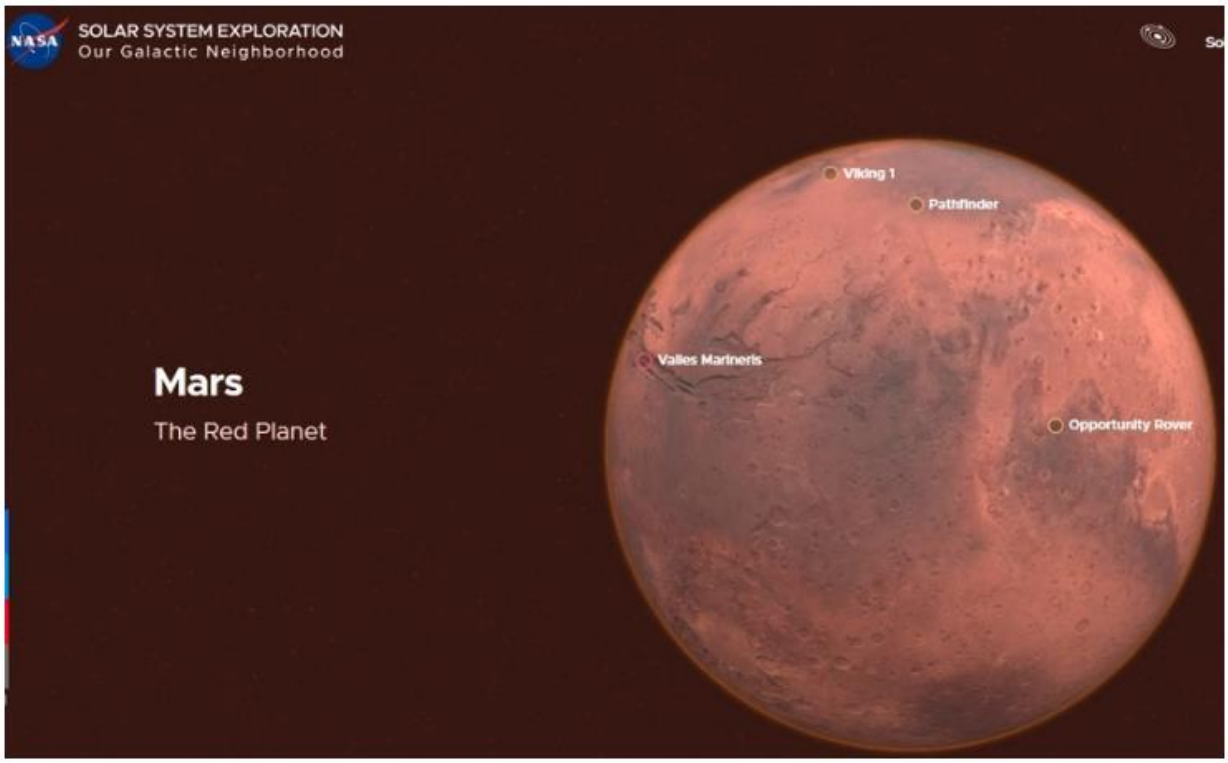
```
12
123
1
32
```

Explanation 0

The input consists of 4 lines, each is a filename. The first file name consists of Pattern 1 "ah" and Pattern 2 "ej". Therefore the passcode is 12. The second file name consists of Pattern 1 "ah"; Pattern 2 "ej" and Pattern 3 "ly". The passcode is 123. The third file name consists of none of the patterns; Therefore 0 is the passcode. The fourth file name consists of Pattern 3 "ly" and Pattern 2 "ej"; therefore the passcode is 32.

The Martian Clock

Mars is one of Earth's two closest planetary neighbors (Venus is the other). Despite being inhospitable to humans, robotic explorers – like NASA's new Perseverance rover – are serving as pathfinders to eventually get humans to the surface of the Red Planet.



With a radius of 2,106 miles (3,390 kilometers), Mars is about half the size of Earth. As Mars orbits the Sun, it completes one rotation every 24.6 hours, which is very similar to one day on Earth (23.9 hours). Martian days are called sols – short for "solar day." A year on Mars lasts 669.6 sols, which is the same as 687 Earth days.

Like Earth, Mars has distinct seasons, but they last longer than seasons here on Earth since Mars takes longer to orbit the Sun (because it's farther away). And while here on Earth the seasons are evenly spread over the year, lasting 3 months (or one quarter of a year), on Mars the seasons vary in length because of Mars' elliptical, egg-shaped orbit around the Sun. Spring is the longest season at 194 sols. Summer is 154 sols. Autumn is the shortest at 142 days, and winter is 178 sols.

Sara is a Space travel enthusiast. She is challenging her friends by giving them a sequence of numbers and she is asking them to convert them into a valid time representation on Mars. For example, given the following sequence:

1	2	15	5	14	27
Year	season	days	hours	minutes	seconds

Her friends told her that this is a correct time representation as they know that:

- Years could be any value greater than or equal to zero.
- Season would be any of the four seasons, Spring 0, Summer 1, Autumn 2 and Winter 3.
- Days could be any value greater than zero.
- Hours should be: $0 \leq h < 24$
- Minutes should be: $0 \leq m < 60$
- Seconds should be: $0 \leq s < 60$

Suppose she gives them the following numbers: {0 0 195 30 45 93}, they would need to convert the time representation into a valid one as follows:

0	1	2	6	46	33
Year	season	days	hours	minutes	seconds

They did the conversion as follows:

- 93 seconds are equivalent to 60 seconds + 33 seconds. The 60 seconds would be added as one minute to the minutes value.
- 45 minutes receive one more minute from the seconds, therefore, it becomes 46 minutes.
- 30 hours are equivalent to 24 hours + 6 hours. The 24 hours would be added as one day to the days.
- 195 days is given; one additional day is added with 196 days total.

- The first season of the year is spring with 194 days. So 196 – 194 days is 2 days in the Summer season.

You are asked to write a program that will read the sequence of numbers and convert it into a valid time representation.

Further Reading

Input Format

The program reads a sequence of numbers (integer values) from the console.

Constraints

The input numbers can only be integers. The input sequence has up to six values. It is read as follows: $< years >< seasons >< days >< hours >< minutes >< seconds >$

Output Format

The output will be a line giving the correct time representation with six integer values seperated by spaces.

Sample Input 0

```
0 0 195 30 45 93
```

Sample Output 0

```
0 1 2 6 46 33
```

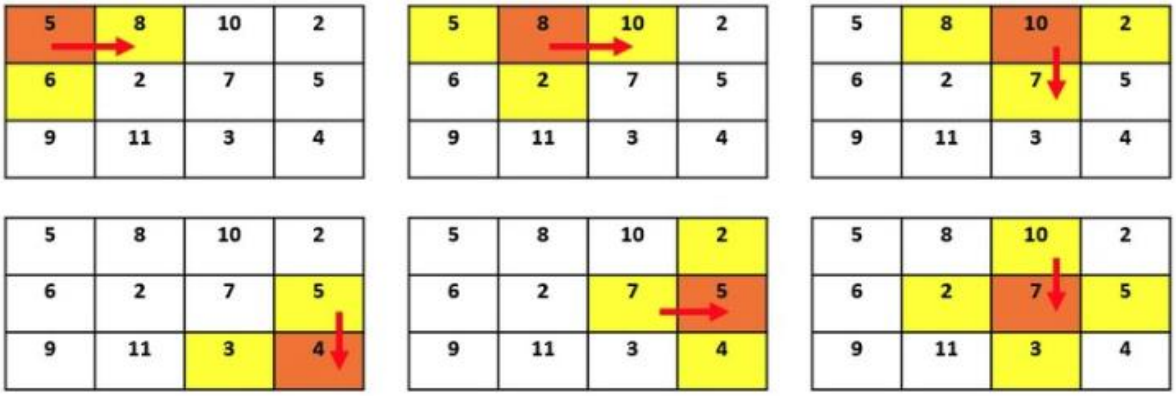
PacManFun

PacMan is the famous retro game that everyone loves. Help write a newer version of PacManFun that has slightly different rules.

The Game is played in a $M \times N$ grid with M rows and N columns. PacManFun is the users avatar that can go in only four directions, $East, West, North$ and $South$. The grid of size $M \times N$ has weights (chips) at each point, PacManFun determines the next move by looking at the highest number of chips in any of the 4 directions ($East, West, North$ and $South$). The avatar moves to the location with the largest number of chips.

The game always starts at position $(0, 0)$. PacManFun needs to reach position (M, N) to win the game. It looses the game otherwise.

The following shows a 3×4 grid.



PacManFun wins the game by accumulating: 5 + 8 + 10 + 7 + 5 + 4 = 39 chips.

Write a program that given a 3×4 grid determines if PacManFun wins the game afer consuming C number of chips.

Input Format

First Line consists of two integers M and N. There are M number of subsequent lines and N number of columns giving chip values as positive Integers.

Constraints

0 < M < 1000000

0 < N < 1000000

0 ≤ Chip value ≤ 1000000

A Chip location that has been visited, cannot be visited again.

Output Format

Only two integer values are printed on console W and C.

$W = \{0, 1\}$ 0 if the game is lost, 1 otherwise.

$C = \sum_{i=1}^{E} C_i$

where E is the length of the path. C_i is the integer chip value of the location i, C is the sum of all chip values in the path.

For any invalid input, the program prints -1 and terminates.

Sample Input 0

```
3 4
5 8 10 2
6 2 7 5
9 11 3 4
```

Sample Output 0

```
1 39
```

POWAGRAMs

The word "powagram" doesn't exist. We invented it for the sake of this programming jam. A powagram is any of the words that is obtained by replacing one letter (power 1) or more (power n) of a given word by another or more letters, then shuffling the word so that it means something according to the English Dictionary. In this problem, we will consider only power 1 powagrams. It is mandatory that the word obtained should have a meaning so that it is considered a powagram. In order to be counted, the word should be tested against an online dictionary.

Example: An English word "cab"; Rank with value 2; a letter to replace with value 'r'; will made two powagrams, "car" and "arc". Rank is the index of the letter to be replaced within the word.

Write a program that reads input and prints out the number of powagrams that can be produced.

You should refer to an online dictionary such as PyDictionary 2.0.1. PyDictionary. https://pypi.org/project/PyDictionary/

Input Format

The program reads three tokens from the console.

- The first token is the word.
- The second token is an integer value (rank)
- The third token is the character that will replace the letter at rank.

Constraints

The word size should be between 2 and 10 characters, both inclusive. The program should exclude other sizes' words. The generated words should be meaningful to be counted. Use an online dictionary like PyDictionary for python users.

Output Format

A single integer value is printed to the console.

Sample Input 0

```
cab 2 r
```

Sample Output 0

```
2
```

Smart Penalty System (Saher2)

We are all familiar with the Saher system for traffic violations management. Saher is an automated system that uses state-of-the-art hardware and AI technologies to control traffic all around the city. It manages the traffic via digital cameras network technology connected to the National Information Center (NIC). We see Saher almost everywhere in Saudi Arabia. Because of this system, traffic accidents and violations have decreased drastically. As a result, the safety of people and driving conditions have improved over time.

You are working in a research lab in your university to help improve the Saher system. The goal is to automate the violation recording by replacing the traffic constable with a robot. This will reduce the need for the physical presence of a human to record the traffic violations. At the moment your team has invested in using Cloud computing and Drones as a technology of choice. The drone will fly around a parking lot, scanning the cars number-plates and sending the information to the Cloud. This information would be processed in a back-end software.

Your team has to develop a back-end software system, that processes the violations posted by the drones. From the console, your program reads a string of inputs generated by the drone. It processes the information to compute the fine based on national id number of the user. There are two kinds of fines imposed, monetary and community service. The monetary fines are computed in Saudi Riyals. The community service fines are number of hours to be given for public service as a penalty. The details for the violation are given as violation codes, where each violation code has information including: violation code; violation description; violation degree; fine base amount;

hours; serious_offence

- Violation code: three letters code to identify the violation type.
- Violation description: a complete name or a description of the fine.
- Violation degree: there are three kinds of violations: Low, Medium, and High.
- Low: violations with no danger, e.g., parking in the wrong parking lot.
- Medium: violations that have no danger but affect the traffic conditions in parking lot, such as blocking roads or waiting in front of building entrances.
- High: violations that have dangers and affect the traffic conditions, such as speeding, not stopping for pedestrians, or driving in the opposite direction.
- Fine base amount: the base amount of the fine without Value Added Tax (VAT).
- Hours: local work on PSU premises if the violator can't or does not want to pay the traffic fine.
- serious_offence: this field will be yes or no. For high degree-fines, it will be yes otherwise, no.

The following are details for the violation codes:

- DPR;Double parking;Low;100;15;no;
- PWL;Parking in the wrong parking Lot;Low;100;15;no;
- PRS;Parking in reserved space;Low;100;20;no;
- PIE;Parking in entrances;Medium;500;30;yes;
- UMP;Using a mobile phone while driving;Medium;500;30;no;
- BMR;Blocking main roads and intersections;Medium;500;30;yes;
- DWD;Driving in the wrong direction;High;1000;50;yes;
- SPD;Speeding;High;1000;50;yes;
- NSP;Not stopping for pedestrians;High;1000;50;yes;

The fine is doubled if the violation occured 3 months prior to todays date.

Your program reads unprocessed data from the console, it processes the data and generates a report displaying the fines accumulated by users ordered by user ID numbers.

Input Format

The first line consists of an integer N followed by N number of lines each line consisting of ticket information issued per violation. Each line consists of

- Driver ID: the violator's legal ID (PSU or Saudi ID). The length of Driver ID is upto 10 digits.
- Date: the violation date. Date format is DD-MM-YYYY
- Plate number: the plate number of the vehicle with the violation. Format is 3 Characters followed by 4 digits.
- Violation code: the fine type.

Constraints

```
1 <= N <= 1000
```

For any invalid input, the program prints -1.

Output Format

The program displays N number of lines, each line consisting of Driver ID followed by the total fine accumulated (monetary AND/or community service hours). Today, we assume that for non-serious offence, the driver does not have to do any community service hours. The output is always ordered by the Driver ID.

Sample Input 0

```
3
1111111111;01-02-2023;AAA1111;PRS
2222222222;01-02-2023;BBB1234;NSP
2222222222;01-02-2023;BBB1234;SPD
```

Sample Output 0

```
1111111111;100
2222222222;2000;100
```

Explanation 0

Note, for driver with ID 1111111111, PRS is not a serious offence and does not require community service hours. Hence the output is 100 Saudi Riyals fine; no community service hours are assigned.

For driver with ID 2222222222, NSP and SPD both are serious offences, hence the monetary penalty is 2000 with 100 community service hours.

AI powered factory

Artificial Intelligence is massively used to solve optimization problems. *Genetic Algorithms (GA)* is one of the well-known bio-inspired algorithms that is successfully used in solving optimization problems. GA is an adaptive heuristic search and optimization algorithm, based on the evolutionary ideas of natural selection and genetics. GA can find the best solution near or equal to the Global Optima.

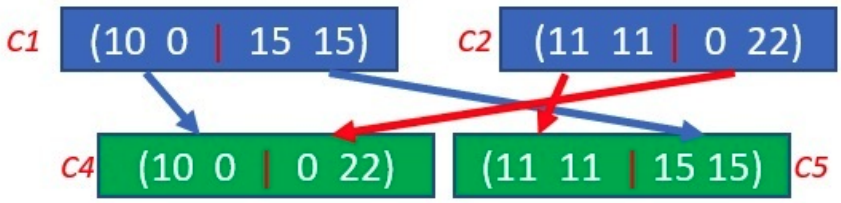
GA is based on the definition of a set of chromosomes that form a population. A chromosome is composed of multiple genes. The chromosomes in the population are then evaluated. The evaluation function is provided and gives the chromosomes a score based on how well they perform at the given task. The following defines the steps in a GA:

1. Initialization: A gene represents the quantity of the parts a manufacturing unit needs to produce a product. A chromosome: is a vector of integer numbers representing the genes. For example: a population of three chromosomes of 4 genes (3 manufacturing units each having 3 parts assemly line) can be given as $C1 = (10, 0, 15, 15)$, $C2 = (11, 11, 0, 22)$, and $C3 = (10, 5, 5, 5)$

2. Evaluation: The individuals in the population are then evaluated using a fitness function. The fitness function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. A simple fitness function to be used in the problem is the sum of genes (number of parts).
$F(C1) = 10 + 0 + 15 + 15 = 40$, $F(C2) = 11 + 11 + 0 + 22 = 44$, and
$F(C3) = 10 + 5 + 5 + 5 = 25$.

3. Selection: The chromosomes are then selected based on their fitness, the higher the fitness, the higher the chance of being selected. In the above, F(C1)=40; F(C2)=44; and F(C3)=25; therefore the two best chromosomes C1 and C2 are fit for regeneration.

4. Crossover: The best chromosomes then "reproduce" to create one or more offspring. The single point crossover is performed by setting the cross area to the half number of genes, and then switching the gene parts. Keep the offsprings having the best fitness value. Half of the population can be reproduced.



The half number of genes = 2. C1= (10,0,15,15) and C2 = (11,11,0,22); the new two offsprings will be: C4 = (10,0,0,22) and C5 = (11,11,15,15). The fitness function for C4 and C5 are F(C4)=32, F(C5)=52; therefore only C5 will be chosen for mutation. C4 will be discarded.

5. Mutation: The new offspring with the higest fitness function value is mutated by switching two genes (first and the last) i.e. after mutation, C5 = (**15**,11,15,**11**). The worst Chromosome i.e. C3 will be replaced by the new child. The new population is 3 with Chromosomes C1, C2, and C5.

6. Iterations & Output: Repeat step 2 onwards until a suitable solution has been found or a certain number of generations/epochs has passed. For the above, after one iteration, the best solution is C2 = (11,11,0,22) with profit 44.

Suppose the factory you work in has n number of manufacturing units (chromosomes) $C_1, C_2, C_3, \ldots, C_n$, each of which is capable of manufacturing a product with a limited quantity. Each manufacturing unit assembles a set of number (genes) of parts on the assembly line per day $x_1, x_2, x_3, \ldots, x_t$. The maximum limit (Fitness function) for assembling for each of M_i is set to 500 number of parts per product.

Write a program that reads input from the console, applies the GA on the input and generates an appropriate output providing the maximum value.

Note: For a chromosome with odd number of genes e.g. (10,2,7,3,5), during the crossover phase when new chromosomes are produced, the half number will be determined for chromosome as (10,2,7 | 3,5), i.e. the left half has one more gene.

Input Format

The first line consists of 3 integers, N, t and e, where N is the number of manufacturing units (chromosomes), t is the number of parts (genes), and e is the epochs (iterations). For the next N number of lines (each is a chromosome), each line contains t number of integers representing the genes.

Constraints

$0 < N < 1000$

$0 \leq x_i \leq 50$

Output Format

A single integer is printed giving the max profit. For any invalid input, the progarm prints -1.

Sample Input 0

```
3 4 1
10 0 15 15
11 11 0 22
10 5 5 5
```

Sample Output 0

```
44
```

Explanation 0

Chromosomes: C1= (10,0,15,15), C2 = (11,11,0,22), and C3 = (10,5,5,5)

Fitness: F(C1) = 40, F(C2) = 44, and F(C3) = 25

Selection: The best two chromosomes C1 and C2

Crossover: One-point and one child: C1= (10,0,|15,15) and C2 = (11,11, |0,22) the new two offsprings will be: C4 = (10,0,0,22) and C5 = (11,11, 15,15)

Mutation: switch two genes: C5 = (11,15,11,15) (C5 has the best fitness value)

New population: C1, C2, C5

Fitness: F(C1) = 40, F(C2) = 44, F(C5) = 52

F(C5) has the highest profit

Selling my 3D Game

Your team is developing a 3D Game. Your company plans to market the game to earn revenue before the game is launched. You seek bids from potential buyers of your game using an online medium. Up to 10 buyers will bid for your game at a time.

After receving the bids, you will decide to set a sale price. Each of these buyer will only purchase your 3D Game if the final price you set is either less than or equal to their bid.

Your team needs to finalize a sale price which guarantees maximum revenue.

Your program reads a number of potential buyers bids, from the console. The output will be the maximum possible revenue.

Input Format

The first line reads an integer N, the total number of potential buyers. Each of the subsequent N lines consists of an integer identifying the bid value for the buyer.

Constraints

```
N >= 2
Any bid value is a non-negative integer <=1000
```

Output Format

A single integer identifying the maximum possible revenue. For any invalid input, the program prints 0.

Sample Input 0

```
4
25
35
43
18
```

Sample Output 0

```
75
```

Explanation 0

Selecting 18 as the sale price will earn 72. Selecting 25 as the sale price will earn 75. Selecting 35 as the sale price will earn 70. Selecting 43 as the sale price will earn 43.

TROJENA1

His Royal Highness Mohammed bin Salman, Crown Prince and Chairman of the NEOM Company Board of Directors, announced the establishment of Trojena, the new global destination for mountain tourism, part of NEOM's plan and strategy to contribute to supporting and developing the tourism sector in the region.

His Royal Highness said: "Trojena will redefine mountain tourism for the world by creating a place based on the principles of ecotourism, highlighting our efforts to preserve nature and enhance the community's quality of life, which is aligned with the goals of the Kingdom's Vision 2030. It also confirms our commitment to be part of the global effort to protect the environment. Trojena will be an important addition to tourism in the region, a unique example of how Saudi Arabia is creating destinations based on its geographical and environmental diversity. This forward-looking vision will ensure that mountain tourism will be another revenue stream to support the Kingdom's economic diversification while still preserving its natural resources for future generations."

Trojena features a unique and innovative architecture, unlike any other in the world, where the captivating landscapes of NEOM mountains coexist in harmony with the tourist sites developed within them, offering a new and unprecedented tourism experience that reflects the future of living, working and entertainment in NEOM. Outdoor skiing is a unique feature of Trojena that will provide a unique experience never before witnessed in the region, especially in Gulf countries known for their desert climates. Amateurs and professionals alike will be able to enjoy the many ski runs of various difficulties with an array of contrasting and breathtaking views.

Customers/Visitors entering the Ski range will be given a pair of skis. There are a limited number of skis available in the stock, so naturally this creates a race condition. Skis come in various sizes, which for the sake of ease to understand this problem, are similar to a standard shoe size.

If a customer is able to find a ski that fits his/her shoe, he is satisfied giving a score of 0. Any difference in the size of the ski earn a dissatisfaction score (difference in size). You are to write a software to assign Skis to Customers minimizing the dissatisfaction as much as possible.

Input Format

The input will contain three lines. The first will give the number of customers N, the second is the sizes of the pairs of skis S_i, and the third is the list of people's shoe size P_i.

Constraints

$0 < N < 1000$

$0 \leq S < 1000$

$0 \leq P < 1000$

Output Format

Display an integer, giving the minimum disappointment. For any invalid input, the program prints -1.

Sample Input 0

```
2
2 5
4 1
```

Sample Output 0

```
2
```

Explanation 0

The Min disappointment is 2

Extrinsic Integers

Two positive integers are said to be extrinsic to each other if there is no common digit in their decimal representations. For example, 1357 and 2460 are extrinsic to each other, while 1234 and 5917 are not. Note that no positive integer is extrinsic to 1234506789.

You are asked to write a program to identify the closest positive integer number(s) that is(are) extrinsic to a given positive integer number X.

Input Format

Each test case consists of one line containing a positive integer number X.

Constraints

$1 \leq X \leq 10^{10}$

Output Format

For each test case, output, on one line the positive extrinsic integer number Y closest to the input number X. If there are two positive extrinsic integer numbers equally close to the input number X, output them both on one line, in ascending order, separated by a single space. If there is no positive integer number extrinsic to the input number X, output the word None.

Sample Input 0

```
1289
```

Sample Output 0

```
777
```

Sample Input 1

```
394
```

Sample Output 1

```
288 500
```

Sample Input 2

```
1903
```

Sample Output 2

```
2222
```

Sample Input 3

```
1025368497
```

Sample Output 3

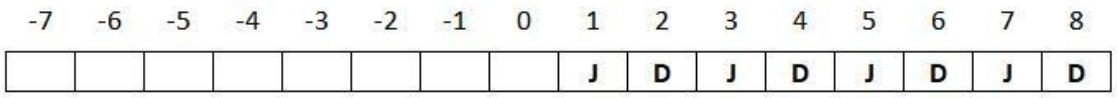
```
None
```

Saudi Arabia Railways Company

Saudi Arabia Railways Co. (SAR) plans to modernize its line operating from Riyadh to Dammam and add a new fast line from Riyadh to Jeddah. Two trains will be leaving daily Riyadh at approximately the same time, one to Jeddah and one to Dammam. The Riyadh railway station has X counters for the passengers to check in their luggage. At every counter, there are 2 identical containers, one for the luggage going to Jeddah and one for the luggage going to Dammam. Right before departure time, each pair of containers is moved by a powered carriage to a sorting zone. The carriage always moves 2 containers at a time, one for Jeddah and one for Dammam. After all the containers are moved, they line up in the sorting zone as follows:

J D J D J D J D J D ..... J D J D

with $2X$ luggage containers in a row, starting with a container for Jeddah then a container for Dammam, and so on. Before the containers can be loaded on the appropriate train, they must be arranged so that all the luggage containers for Dammam precede those for Jeddah. This is done by moving pairs of adjacent luggage containers (not necessarily Jeddah then Dammam), again by the powered carriage. For stability, the powered carriage must always carry two containers at a time, never only one. When the sorting begins, the containers locations are numbered from 1 (initially containing the leftmost Jeddah luggage container) to $2X$ (initially containing the rightmost Dammam luggage container). Initially, on the left of the first container, there are $2X$ empty spaces numbered from 0 to $2X - 1$ that can be used during the sorting phase, as shown below for the case $X = 4$.



Note that a pair of containers must always be moved to an empty space that is at least 2 containers wide.

Given the number of counters X in Riyadh railway station, and assuming that no counter ends up with an empty container, you are asked to write a program to find the shortest sequence of moves that will arrange the containers so that all Dammam containers are on the left of all Jeddah containers. At the end of the process, all the containers must be adjacent in a sequence of $2X$ locations, not necessarily starting a location 1.

Input Format

Each test case consists of one line containing the number of counters X.

Constraints

$3 \leq X \leq 30$

Output Format

For each test case, output the shortest sequence of moves needed to sort the containers, each move on one line in the form From A to B (separated by single spaces), where A and B are integers representing the move of the containers in locations A and $A+1$ to locations B and $B+1$.

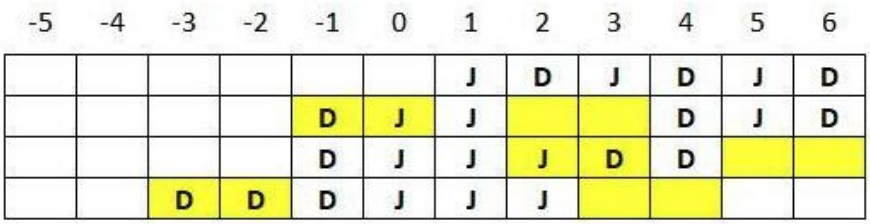
Sample Input 0

```
3
```

Sample Output 0

```
From 2 to -1
From 5 to 2
From 3 to -3
```

Explanation 0



Sample Input 1

```
7
```

Sample Output 1

```
From 8 to -1
From 5 to 8
From 12 to 5
From 3 to 12
From 9 to 3
From 0 to 9
From 13 to 0
```

Which candies can I get?

Ahmed is 10 and wants to buy some candies from a local market. He gets pocket-money from his mom which is usually a small sum. He visited the candies stand with many yummy candies placed on various racks. Each candy rack has a large label showing the price for each item.

Being greedy, Ahmed wants to get as many candies as possible. Ahmed is allowed to take only one candy from each rack. Help Ahmed determine which candies to buy.

Suppose that the items' prices are given in a sequence. For example:
$Prices = \{5, 2, 9, 1, 8, 3, 4\}$ represents the prices for items on each rack, i.e. item[0] on rack 0 has a price equal to 5, item[1] on rack 1 has a price equal to 2, etc...

Given these prices, suppose that Ahmad's budget is 10 Riyals, he can afford the items that have the following prices:

- Scenario 1: Items with prices = {1, 9}
- Scenario 2: Items with prices = {2, 3, 5}
- Scenario 3: Items with prices = {2, 8}
- .
- .
- Scenario n.

You are asked to write a program that would print all possible combinations of prices that he could afford. Your program will print the combination of prices that has a summation equal to the provided budget. The combination are ordered from lowest to highest price on each output line.

Input Format

The first line is a sequence of integer numbers separated by spaces which represents the prices of items in the store. It ends with a new line.

The second line is the budget that Ahmad can afford. It is an integer value.

Constraints

The input numbers can only be integers. If the program could not find a combination within the constraints, it would print 0.

Output Format

The output will be a line for each combination of prices. The total summation of the prices (per line) in the combination should equal to the budget provided in the input.

The order of printing the prices in each line should follow lowest to highest value.

Sample Input 0

```
2 4 5
6
```

Sample Output 0

```
2 4
```

Explanation 0

Only one combination totals the given budget 6.

My autonomous car

A self-driving car, also known as an autonomous car, driver-less car, or robotic car (robo-car), is a car that is capable of traveling without human input. Self-driving cars use sensors to perceive their surroundings, such as optical and thermographic cameras, radar, lidar, ultrasound/sonar, GPS, odometry and inertial measurement units. Control systems interpret sensory information to create a three-dimensional model of the surroundings. Based on the model, the car identifies appropriate navigation paths, and strategies for managing traffic controls (stop signs, etc.) and obstacles. (Source: Wikipedia)

(Image: Tesla dash board)

You will write a program to help navigate my autonomous car through roundabouts. A roundabout is a type of circular intersection or junction in which road traffic is permitted to flow in one direction around a central island, and priority is typically given to traffic already in the junction. Cars usually enter the roundabout from an exit points and moves aways using another exit point. There can be various exit-points in a round about.

Your program reads input from console, processes the information and prints the coordinates for the exit. For the sake of simplicity, we assume that my autonomous car will always take the first exit from the point of entry. Assume standard traffic direction for Saudi Arabia.

Input Format

The first line has the (x, y) coordinates of the roundabout and an integer N representing the number of exits in the roundabout.

The second line as (x, y) coordinates for the point of entry in the roundabout. In the subsequent $N - 1$ lines, the coordinates (x_i, y_i) of each exit i in the roundabout, where $i \in N$.

Constraints

$N > 0$

$-1000000 \leq x_i \leq 1000000$

$-1000000 \leq y_i \leq 1000000$

Output Format

The coordinates of the exit destination corresponding to the first exit of the roundabout. For any invalid input, the program prints -1.

Sample Input 0

```
1 1 4
1 0
1 2
0 1
2 1
```

Sample Output 0

```
2 1
```

Explanation 0