**InfyTQ Training-2022**

**Practice Problems List**

**1. Concatenation**

**Input**: a string of comma separated numbers. The numbers 5 and 8 are present in the list. Assume that 8 always comes after 5.

Case 1: num1 = add all numbers which do not lie between 5 and 8 in the input.  
Case 2: num2= numbers formed by concatenating all numbers from 5 to 8.  
**Output**: sum of num1 and num2

Example: 1) 3,2,6,5,1,4,8,9  
Num1 : 3+2+6+9 =20  
Num2: 5148  
output: 5148+20 = 5168

**2. Palindrome**

Write a python function nearest\_palindrome() which can accepts a number and return the nearest greater palindrome number.  
**Input**: 12300 -> output: 12321  
**Input**: 12331 -> output: 12421

**3. Parenthesis matching**

A non-empty string instr containing only parenthesis (,),{.},[,] it return outstr based on following,  
– instr is properly nested and return 0  
– instr not properly nested, return position of element in instr  
– position start from 1  
**Input**: {([])} Output: 0  
**Input**: (])()] Output: 3  
**Input**: [[()]  Output: n+1 for last element i.e 5+1=6

**4. Sum of factor**

For a given list of numbers find the its factors and add the factors then if the sum of all factor is present in original list, sort it and print it  
Ex:  
**Input**: 0,1,6  
Factors: 0 = 0, sum =0  
1 = 1, sum =1  
6 =1,2,3 = sum =6  
**Output**: 0,1,6  
If the sum is not present in the list, then return -1.

**5. Special character out**

Write a python program that it should consist of special char numbers and chars. if there are even numbers of special chars Then the series should start with even followed by odd  
**Input**: 19@a42&516  
**Output**: 492561  
If there are odd numbers of special chars then the output will be starting with odd followed by even  
**Input**: 5u6@25g7#@  
**Output**: 56527  
If there are any number of additional digits append them at last

**6. Password generation**

Given input of array of string in format <emnp name> <emp number> separated by ‘,’ Emp  
should contain only alphabets and employee number. You have to generate password for  
Ex:  
**input**: Robert: 36787, Tina: 68721, Jo:56389  
**Output**: tiX  
**Conditions**: len of robert is 6 and 6 is present in emp number  
robert (36787), so return the alphabet at position 6 that is t.  
Now len of tina is 4 and 4 is not present in the 68721 so select the number which is max and less than the len of tina so select 2 return the alphabet that is at position 2 that is i.  
Now In of Jo is 2 it is not present in 56389 and there is not present any number which is less than 2 so return X.

**7. Even number**

A string which is a mixture of letter and integer and special char from which find the largest even number that can generated from the available digit after removing the duplicates.  
If an even number is not formed then return-1.  
Ex: infosys@337  
O/p:-1  
Hello#81@21349  
O/p:983412

**8. Matrix problem**

Read m’m>4  
N=m+1  
Take m\*n matrix  
If any num is consecutive for 3 times either in a row, column, diagonals print the num, if there multiple num print min of those num  
Ex: m=6 take 6\*7 matrix  
2 3 4 5 6 2 4 3  
2 3 4 7 6 7 6 2  
2 3 5 5 5 5 2 5  
2 3 1 1 2 1 3 6  
1 1 1 1 9 0 3 5  
2 3 1 1 5 1 2 7  
O/p=1

**9. String rotation**

**Input:** rhdt:246, ghftd:1246  
Expl: here every string is associated with the number separated by : if sum of squares of digits is even then left rotate the string by 1, if square of digits is odd then rotate the string right by 2 position.

E.g- 2\*2+4\*4+6\*6=56 which is even so rotate rhdt ->trhd.

E.g 1\*1+2\*2+4\*4+6\*6=57 which is odd then  
rotate string by 2 at left ‘ghftd”  
**output**: ftdgh

**10. Pronic number**

**Input**:- 93012630  
**Output**-2,6,12,30,930,  
We should divide the total number into substrings and we should verify each num is pronic num or not if pronic we should print that num

Pronic: means it is a multiple of two consecutive integers  
Ex: 6->2\*3 it’s a pronic  
12->3\*4 it’s a pronic  
Input: 12665042  
Output: 2,6,12,42,650

**11. Longest palindrome**

Find the longest palindrome from a string  
**Input**: moomso  
Possible cases  
Moom, mom, oso, ooo, omo  
Longest is moom so output: moom

### 12. Self Sufficient

**Problem Statement – Abhijeet** is one of those students who tries to get his own money by part time jobs in various places to fill up the expenses for buying books. He is not placed in one place, so what he does, he tries to allocate how much the book he needs will cost, and then work to earn that much money only. He works and then buys the book respectively. Sometimes he gets more money than he needs so the money is saved for the next book. Sometimes he doesn’t. In that time, if he has stored money from previous books, he can afford it, otherwise he needs money from his parents.

Now his parents go to work and he can’t contact them amid a day. You are his friend, and you have to find how much money minimum he can borrow from his parents so that he can buy all the books.

He can Buy the book in any order.

**Function Description:**

Complete the function with the following parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| N | Integer | How many Books he has to buy that day. |
| EarnArray[ ] | Integer array | Array of his earnings for the ith book |
| CostArray[ ] | Integer array | Array of the actual cost of the ith book. |

**Constraints:**

* 1 <= N <= 10^3
* 1 <= EarnArray[i] <= 10^3
* 1 <=  CostArray[i] <= 10^3

**Input Format:**

* First line contains N.
* Second N lines contain The ith earning for the ith book.
* After that N lines contain The cost of the ith book.

**Output Format:**The minimum money he needs to cover the total expense.

**Sample Input 1:**

3

[3 4 2]

[5 3 4]

**Sample Output 1:**

3

### 13. Amusement Park

**Problem Statement** – Aashay loves to go to WONDERLA, an amusement park. They are offering students who can code well with some discount. Our task is to reduce the cost of the ticket as low as possible.

They will give some k turns to remove the cost of one ticket where the cost of tickets are combined and given as string. Help Aashay in coding as he is not good in programming and get a 50%  discount for your ticket.

**Constraints:**

* 1 <= number of tickets <= 10^5
* 1 <= K <= number of tickets

**Input Format for Custom Testing:**

* The first line contains a string, Tickets, denoting the given cost of each ticket.
* The next line contains an integer, K, denoting the number of tickets that is to be removed.

**Sample Cases:**

* Sample Input 1203  
  3
* Sample Output 10

### 14. HR issues

**Problem statement -:** Shovon is an HR in a renowned company and he is assigning people to work. Now he is assigning people work in a fashion where if he assigns somework a work of cost 2, the next person will be strictly getting a job with cost equal or more than 2. Given that Shovon’s company has infinite work and a number of employees, how many distributions can be possible. The cost of jobs can go 0 to 9.

**Function Description:**

Complete the special\_numbers function in the editor below. It has the following parameter(s):

**Parameters:**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| N | Integer | The number of depts. |
| arr[ ] | Integer array | The number of employees in each dept.. |

Return: The function must return an INTEGER denoting the sum of answers for all distinct distributions.

**Constraints:**

* 1 <= n <= 100
* 1 <= arr[i] <= 200

**Sample Cases:**

* **Sample Input 1**2  
  4  
  1
* **Sample Output 1**725
* DescriptionThe ans if m = 1 is 1o, which is all numbers from 0 to 9  
  The ans for m = 2 is 55  
  The answer for m = 3 is 220  
  The answer for m = 4 is 715  
  So fun(4) + fun(1) = 725

### 15. Airport authority

**Problem Statement** -: In an airport, the Airport authority decides to charge some minimum amount to the passengers who are carrying luggage with them. They set a threshold weight value, say T, if the luggage exceeds the weight threshold you should pay double the base amount. If it is less than or equal to threshold then you have to pay $1.

**Function Description:**

Complete the weight machine function in the editor below. It has the following parameter(s):

**Parameters:**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| N | Integer | number of luggage |
| T | Integer | weight of each luggage |
| weights[ ] | Integer array | threshold weight |

**Returns**: The function must return an INTEGER denoting the required amount to be paid.

**Constraints:**

* 1 <= N <= 10^5
* 1 <= weights[i] <= 10^5
* 1 <= T <= 10^5

**Input Format for Custom Testing:**

* The first line contains an integer, N, denoting the number of luggage.
* Each line i of the N subsequent lines (where 0 <= i <n) contains an integer describing weight of ith luggage.
* The next line contains an integer, T, denoting the threshold weight of the boundary wall.

**Sample Cases:**

* **Sample Input 1**4  
  1  
  2  
  3  
  4  
  3
* **Sample Output 1**5

**16. Forming a Special String**

**Input type: String.**

**Output type: number.**

**Print out the maximum even number which can be generated by making combinations of numbers present in the given string. Each number can be used only once i.e. No Redundancy.**

**Example 1**

**Input String** = Infytq@218412

Intermediate step (List of numbers \*//redundancy removed\*) = [2,1,8,4]

**Output Number** = 8412

**Example 2**

Input : Hello#81@21349

Output: 983412

**Example 3**

Input String = someString&337

Intermediate step (List of numbers \*//redundancy removed\*) = [3,7]

Output Number = -1

**17. Uniformity**

You are given a string that is formed from only three characters ‘a’, ‘b’, ‘c’. You are allowed to change atmost ‘k’ characters in the given string while attempting to optimize the uniformity index.

Note: The uniformity index of a string is defined by the maximum length of the substring that contains same character in it.

**Input**

The first line of input contains two integers **n** (the size of string) and **k**. The next line contains a string of length **n**.

**Output**

A single integer denoting the maximum uniformity index that can be achieved.

**Constraints**

1 <= n <= 10^6

0 <= k <= n

String contains only ‘a’, ‘b’, ‘c’.

**Sample Input 0**

6 3

abaccc

**Sample Output 0**

6

**Explanation**

First 3 letters can be changed to ‘c’ and we can get the string ‘cccccc’

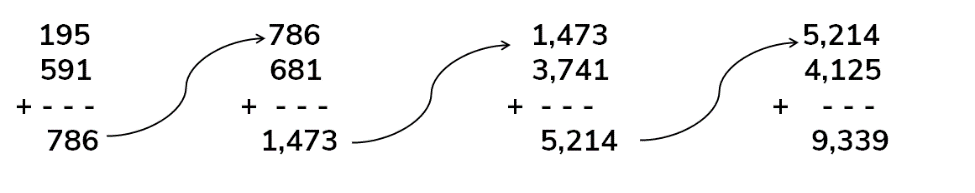
### 18. Identify Palindrome

For a given positive number num, identify the palindrome formed by performing the following operations-

* Add num and its reverse
* Check whether the sum is palindrome or not. If not, add the sum and its reverse and repeat the process until a palindrome is obtained

**For example:**

If original integer is 195, we get 9,339, as the resulting palindrome after the fourth addition:



**Input format:** Read num from the standard input stream.

**Output format**: Print the palindrome calculated to the standard output stream.

**Example 1**

Sample input:

124

Sample output:

545

Explanation:

The sum of 124 and its reverse 421 is 545 which is a palindrome.

**Example 2**

Sample input:

4

Sample output:

8

Explanation:

The sum of 4 and its reverse 4 is 8 which is a palindrome.

**19. Special Reverse**

**For a given string with letters and special characters, print the string with the letters reversed, but without changing the position of the special characters.**

**Example: Input: in@#f^ytq&!**

**Output: qt@#y^fni&!**

**Input Format: One line – the string to be reversed**

**Output format: The reversed string**

**20. Formatting a Matrix:**

**Consider a M x N matrix, M1 and P x Q matrix, M2 consisting of positive integers. Create and print outmatrix based on the below logic.**

**For every number num1 from M1 at row r1 and column c1:**

* **Multiply with each of the numbers, num2 of M2 such that num2 is at M2[r1][r1], M2[r1][c1], M2[c1][r1], M2[c1][c1], if exists.**
* **Identify and add the product that occurs maximum number of times to outmatrix[r1][c1].**
* **If there are multiple products with maximum number of times, then the product with the lowest value will be held by outmatrix[r1][c1].**
* **If both r1 and c1 don’t exist in M2, then outmatrix[r1][c1] = -1**

**Assumption: Zero is positive Integer**

**Input Format:**

**• First line contains an integer n, denoting number of rows in M1**

**• Second line contains an integer p, denoting number of rows in M2**

**• Next n lines contain m space separated integers, the contents on M1**

**• Next p lines contain q space separated integers, the contents on M2**

**Output Format: Output n lines of m space separated integers out outmatrix.**

**Example:**

**Input:**

2

2

1 2

3 4

3

4

Output: 3 6

9 -1

Explanation:

• **For (0, 0):** Here r1=0 and c1=0. Multiplying with the element at (0,0) of M2

results in 1\*3=3. The only product is 3 which is occurring once. Hence

outmatrix[0][0] = 3

• **For (0, 1):** Here r1=0 and c1=1. Multiplying with each of the elements at (0,0),

(0,1), (1,0), (1,1) of M2 results in 2\*3=6, 2\*4=8. (0,1) & (1,1) don't exist in M2

and hence ignored. Products are '6' occurring once and '8' occurring once.

Both the products are occurring once. Hence substituting the lowest value

product i.e. '6' to outmatrix at (0,1).

• **For(1, 0):** Here r1=1 and c1=0. Multiplying with each of the elements at (0,0)

(0,1), (1,0), (1,1) of M2 results in 3\*3=9, 3\*4=12, (0,1) & (1,1) don't exist in

M2 and hence ignored. Products are '9' occurring once and '12' occurring

once. Both the products are occurring once. Hence substituting the lowest

value product i. e. '9' to outmatrix at (1,0)

• **For (1, 1):** Here r1=1 and c1=1, (1,1) doesn't exist in M2. Hence

outmatrix[1][1] = -1

**21. String Rotation-II**

Given a string and a number separated by a colon(:), output the string as follows:

• If the sum of squares of the digits of the number following the string is even, rotate the string to right by 1 and print it.

• If the sum of squares of the digits of the number following the string is odd, rotate the string to the left by 2 and print it.

**Example 1:**

Input: dkns:348

Output: nsdk

Explanation: Sum of squares of digits of 348 are: 3\*3 + 4\*4 + 8\*8 = 9+16+64 = 89, which is odd, therefore left rotation by 2 places.

**Example 2:**

Input: jaipq:974

Output: qjaip

Explanation: 9\*9 + 7\*7 + 4\*4 = 81 + 49 + 16 = 146, which is even, hence right rotation by 1.

22. **Lane Problem**

Problem: A, B, C, D are lanes with capacity of 10 slots each.  
  
Input Format:  
i) Point 1: First take the inputs for the 4 lanes with booked slots 1-10 for each Lane (4 inputs in 4 lines and the values are separated by commas)  
ii) Point 2: If no slot is booked for a lane ‘-1’ will be given as an input for that lane.  
iii) Point 3: At 5th line take input for number of waiting cars.  
  
To perform:  
Point 1: For all waiting cars to be parked first find out the lane with maximum free slots if 2 lane have the same number of highest free slots, then prefer A -> D flow.  
  
Point 2: Fill each waiting car, in the selected lane and mark it as the sequence number for the lane with free slots A, A2,…….A10 till all waiting cars are parked. If A has already filled 5 slots then fill waiting slots from A6-A10  
  
Point 3: If the waiting cars are not completely filled, then fill the next highest free lane continue till all waiting cars get parked.  
  
Note :If no slot is free and cars can’t be parked then print capital ‘X‘ (without quotes).

**Sample Input:**

|  |
| --- |
| A1,A2,A3,A4  B1,B2,B3  C1,C2  D1,D2,D3,D4,D5  10 |

**Sample Output:**

|  |
| --- |
| C3 C4 C5 C6 C7 C8 C9 C10 B4 B5 |

**Explanation:** Here the Slot C has the highest number of available spaces (10 (total space) – 2 (Occupied spaces) = 8 (left out spaces) ). So first 8 cars out of 10 cars in the waiting list are parked in the slot, the remaining 2 cars are parked into Slot B, which is the second highest slot with available spaces (10-3 = 7).

# **23. Unique Pair whose Sum of Digits are Same**

#### Problem:Consider a non-empty input array (inarr) containing non-zero positive Integer. From input array, identify the unique pairs of integers such that each of the integers in the pair have the same sum of the digits. Print outnum, the number of unique pairs identified satisfying the criteria. If no such pair of integers can be identified print -1. Input format: Read the array inarr (input array) with the elements separated by ‘,’ (comma) Output format: Print outnum or -1 accordingly

#### Sample Input:

|  |
| --- |
| 34,89,6,321,53,45,2211,81 |

#### **Sample Output:**

|  |
| --- |
| 4 |

#### Explanation: All the possible combinations are: (6, 321), (6, 2211), (321, 2211), (45, 81) First combination: 6, 3+2+1 = 6 Second combination: 6, 2+2+1+1 = 6 Third combination: 3+2+1 = 6, 2+2+1+1 = 6 Fourth combination: 4+5 = 9, 8+1 = 6 Hence the output will be: 4 (Output)

# **24. Combination And Its Sum**

#### Problem:A set of number will be given and a sum will also be given. Print the number of combinations possible of length 4 which sums up to be the given sum. Input format: First line contains the given set of numbers. Second line contains a single integer denoting the sum.

#### Sample Input:

|  |
| --- |
| -1,1,0,2, -2  0 |

#### **Sample Output:**

|  |
| --- |
| 3 |

#### Explanation: All the possible combinations of length 4 which will add up to be 0 are (-1, 1, 2, -2), (0, 0, 1, -1), (0, 0, -2, 2) First combination: -1+1+2+(-2) = 0 Second combination: 0+0+1+(-1) = 0 Third combination: 0+0+(-2)+2 = 0 Hence, the output will be 3.

25. **:Minimum Distinct Element after Deletion**

Problem: Given a list (combination of repeated and distinct elements) and number of elements deletion X. You have to delete any X elements from the list so that list will have minimum distinct number and return the length of the minimum distinct list.  
  
Input Format: The first line contains an integer T. T is the number of independent testcases. For each testcase the first line contains an integer denoting the number of elements you can delete. Next line contains the list. (Inputs are separated by space here.)

Sample Input:

2

2

1 1 2 4

3

1 1 1 2 2 2 4 5 6

Sample Output:

1

2

Explanation: Here, 2 and 4 are removed so that it has the minimum distinct element 1 which is repeated (repetition are removed) so the length of the minimum distinct list is 1  
  
For the second case, 4, 5 and 6 are removed so that it has the minimum distinct element 1 and 2 which is repeated (repetition are removed) so the length of the minimum distinct list is 2

**26. Equal String**

Check if two strings can be made equal by swapping one character among each other

Given two strings **A** and **B** of length **N**, the task is to check whether the two strings can be made equal by swapping any character of **A** with any other character of **B** only once.

**Examples:**

***Input:****A = “SEEKSFORGEEKS”, B = “GEEKSFORGEEKG”****Output:****Yes  
“****S****EEKSFORGEEKS” and “GEEKSFORGEEK****G****”  
can be swapped to make both the strings equal.*

***Input:****A = “GEEKSFORGEEKS”, B = “THESUPERBSITE”****Output:****No*

Ref: <https://www.geeksforgeeks.org/check-if-two-strings-can-be-made-equal-by-swapping-one-character-among-each-other/>

**27. Longest subsequence having difference atmost K**

Given a string **S** of length **N** and an integer **K**, the task is to find the length of longest sub-sequence such that the difference between the ASCII values of adjacent characters in the subsequence is not more than K.

**Examples:**

**Input:** N = 7, K = 2, S = "afcbedg"

**Output:** 4

**Explantion:**

Longest special sequence present

in "afcbedg" is a, c, b, d.

It is special because |a - c| <= 2,

|c - b| <= 2 and | b-d| <= 2

**Input:** N = 13, K = 3, S = "geeksforgeeks"

**Output:** 7

Ref: <https://www.geeksforgeeks.org/longest-subsequence-having-difference-atmost-k/>

**28. Check whether the given string is Palindrome using Stack**

Given a string **str**, the task is to find whether the given string is a palindrome or not using stack.

**Examples:**

***Input:****str = “geeksforgeeks”****Output:****No*

***Input:****str = “madam”****Output:****Yes*

Ref: <https://www.geeksforgeeks.org/check-whether-the-given-string-is-palindrome-using-stack/>

**29. Minimum number of operations to convert a given sequence into a Geometric Progression**

Given a sequence of N elements, only three operations can be performed on any element at most one time. The operations are:

1. Add one to the element.
2. Subtract one from the element.
3. Leave the element unchanged.

Perform any one of the operations on all elements in the array. The task is to find the minimum number of operations(addition and subtraction) that can be performed on the sequence, in order to convert it into a [Geometric Progression](https://en.wikipedia.org/wiki/Geometric_progression). If it is not possible to generate a GP by performing the above operations, print -1.

**Examples**:  
***Input****: a[] = {1, 1, 4, 7, 15, 33}****Output****: The minimum number of operations are 4.  
Steps:*

1. *Keep a1 unchanged*
2. *Add one to a2.*
3. *Keep a3 unchanged*
4. *Subtract one from a4.*
5. *Subtract one from a5.*
6. *Add one to a6.*

*The resultant sequence is {1, 2, 4, 8, 16, 32}*

***Input****: a[] = {20, 15, 20, 15}****Output****: -1*

Ref: <https://www.geeksforgeeks.org/minimum-number-of-operations-to-convert-a-given-sequence-into-a-geometric-progression/>

**30. Charging Robot**

There is a robot which wants to go the charging point to charge itself.   
The robot moves in a 2-D plane from the original point (0,0).  The robot can move toward **UP, DOWN, LEFT** and **RIGHT** with given steps.  
The trace of robot movement is shown as the following:

**UP 5**

**DOWN 3**

**LEFT 3**

**RIGHT 2**

Then, the output of the program should be:

**2**

The numbers after the direction are steps.   
Write a program to compute the distance between the current position after a sequence of movement and original point. If the distance is a float, then just print the nearest integer (use **round()** function for that and then convert it into an integer).  
**Input Format:**  
The first line of the input contains a number n which implies the number of directions to be given.  
The next n lines contain the direction and the step separated by a space.  
**Output Format:**  
Print the distance from the original position to the current position.   
Example:  
Input:  
4

UP 5

DOWN 3

LEFT 3

RIGHT 2  
Output:  
2  
Explanation:   
After the movements, the robot is at the position (-1, 2). Distance from the (0, 0) to the point (-1, 2) is calculated as (−1)2+(2)2−−−−−−−−−−√  
The round value of which is 2.0, and int value is 2.  
NOTE: Import math library and use the **sqrt()** function of the **math** library to compute the distance.  
Guide to calculate the distance from the point (a,b) to (c,d) is [here](https://www.cut-the-knot.org/pythagoras/DistanceFormula.shtml)   
Guide to use math sqrt function is [here](https://www.geeksforgeeks.org/python-math-function-sqrt/).

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 4  UP 5  DOWN 3  LEFT 3  RIGHT 2 | 2 |
| Test Case 2 | 6  UP 5  RIGHT 6  DOWN 3  LEFT 2  RIGHT 8  DOWN 5 | 12 |
| Test Case 3 | 10  DOWN 10  LEFT 11  RIGHT 1  DOWN 5  LEFT 8  LEFT 7  DOWN 3  UP 6  DOWN 3  UP 4 | 27 |
| Test Case 4 | 5  DOWN 3  DOWN 2  DOWN 4  DOWN 4  UP 10 | 3 |
| Test Case 5 | 5  DOWN 3  UP 4  LEFT 6  RIGHT 3  LEFT 1 | 4 |
| Test Case 6 | 12  RIGHT 12  LEFT 3  UP 2  UP 2  DOWN 5  LEFT 7  RIGHT 4  UP 6  UP 6  DOWN 1  DOWN 1  LEFT 2 | 10 |
| Test Case 7 | 12  RIGHT 12  LEFT 3  UP 4  UP 2  DOWN 5  LEFT 4  RIGHT 4  UP 2  UP 6  DOWN 14  DOWN 1  LEFT 9 | 6 |
| Test Case 8 | 4  UP 0  LEFT 0  RIGHT 0  DOWN 0 | 0 |

***31. Minimum* operations required to convert X to Y by multiplying X with the given co-primes**

Given four integers **X**, **Y**, **P** and **Q** such that **X ≤ Y** and **gcd(P, Q) = 1**. The task is to find minimum operation required to convert **X** to **Y**. In a single operation, you can multiply **X** with either **P** or **Q**. If it is not possible to convert **X** to **Y** then print **-1**.

**Examples:**

***Input:****X = 12, Y = 2592, P = 2, Q = 3****Output:****6  
(12 \* 2) -> (24 \* 3) -> (72 \* 2) -> (144 \* 3) -> (432 \* 3) -> (1296 \* 2) ->2592*

***Input:****X = 7, Y = 9, P = 2, Q = 7****Output:****-1  
There is no way we can reach 9 from 7 by  
multiplying 7 with either 2 or 7*

Ref: <https://www.geeksforgeeks.org/minimum-operations-required-to-convert-x-to-y-by-multiplying-x-with-the-given-co-primes/>

# **32. Minimum number of operations required to reduce N to 1**

Given an integer element ‘N’, the task is to find the minimum number of operations that need to be performed to make ‘N’ equal to 1.  
The allowed operations to be performed are:

1. Decrement N by 1.
2. Increment N by 1.
3. If N is a multiple of 3, you can divide N by 3.

**Examples:**

***Input:****N = 4****Output:****2  
4 – 1 = 3  
3 / 3 = 1  
The minimum number of operations required is 2.*

***Input:****N = 8****Output:****3  
8 + 1 = 9  
9 / 3 = 3  
3 / 3 = 1  
The minimum number of operations required is 3.*

Ref: <https://www.geeksforgeeks.org/minimum-number-of-operations-required-to-reduce-n-to-1/>

**33. Island**

You are a poor person in an island. There is only one shop in this island, this shop is open on all days of the week except for Sunday. Consider following constraints:

* N – Maximum unit of food you can buy each day.
* S – Number of days you are required to survive.
* M – Unit of food required each day to survive.

Currently, it’s Monday, and you need to survive for the next S days.   
**Find the minimum number of days (ceil value) on which you need to buy food from the shop so that you can survive the next S days,**or determine that it isn’t possible to survive.  
**Input Format:**The first line of the input contains three numbers S, N and M separated by space.  
**Output Format:**If it is possible to survive the print the number of days, otherwise print 'NO' without quotes.  
Example-1  
Input:  
10 16 2  
Output:  
2  
Example-2  
Input:  
10 20 30  
Output:  
NO  
**Explanation 1:** One possible solution is to buy a box on the first day (Monday), it’s sufficient to eat from this box up to 8th day (Monday) inclusive. Now, on the 9th day (Tuesday), you buy another box to survive the 9th and 10th day.  
  
**Explanation 2***: You can’t survive even if you buy food because the maximum number of units you can buy in one day is less the required food for one day.*

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 10 14 3 | 3 |
| Test Case 2 | 12 9 8 | NO |
| Test Case 3 | 15 7 5 | 11 |
| Test Case 4 | 18 17 12 | 13 |
| Test Case 5 | 10 6 2 | 4 |
| Test Case 6 | 10 8 5 | 7 |
| Test Case 7 | 12 9 8 | NO |
| Test Case 8 | 12 10 2 | 3 |

**34. Power of 2**

Write a program to find whether a given number is a power of **2** or not.  
**Input format:**  
The first line of the input contains the number n for which you have to find whether it is a power of 2 or not.  
**Output Format:**Print 'YES' or 'NO' accordingly without quotes.  
**Example:**Input:  
32  
Output:  
YES  
Input:  
26  
Output:  
NO  
**Explanation:**In the first example, 32 is actually 25 so the answer is YES.  
The second number is not a power of 2 hence the answer is NO.

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 22 | NO |
| Test Case 2 | 16 | YES |
| Test Case 3 | 82 | NO |
| Test Case 4 | 512 | YES |
| Test Case 5 | 1024 | YES |
| Test Case 6 | 4096 | YES |
| Test Case 7 | 16384 | YES |
| Test Case 8 | 65536 | YES |

**35. Fill missing entries of a magic square**

Given a 3X3 matrix **mat** with it’s left diagonal elements missing (set to **0**), considering the sum of every row, column and diagonal of the original matrix was equal, the task is to find the missing diagonal elements and print the original matrix.

**Examples:**

***Input:****mat[][] = {{0, 7, 6}, {9, 0, 1}, {4, 3, 0}}****Output:*** *2 7 6  
9 5 1  
4 3 8  
Row sum = Column sum = Diagonal sum = 15*

***Input:****mat[][] = {{0, 1, 1}, {1, 0, 1}, {1, 1, 0}}****Output:*** *1 1 1  
1 1 1  
1 1 1*

Ref: <https://www.geeksforgeeks.org/fill-missing-entries-of-a-magic-square/>

36. **Game of Alphabets**

Find the winner of the Game to Win by erasing any two consecutive similar alphabets

Given a string consisting of lower case alphabets.

**Rules of the Game:**

* A player can choose a pair of similar consecutive characters and erase them.
* There are two players playing the game, the player who makes the last move wins.

The task is to find the winner if A goes first and both play optimally.

**Examples:**

**Input:** str = "kaak"

**Output:** B

**Explanation:**

Initial String: "kaak"

A's turn:

removes: "aa"

Remaining String: "kk"

B's turn:

removes: "kk"

Remaining String: ""

Since B was the last one to play

B is the winner.

**Input:** str = "kk"

**Output:** A

Ref: <https://www.geeksforgeeks.org/find-the-winner-of-the-game-to-win-by-erasing-any-two-consecutive-similar-alphabets/>

**37. Absolute Game**

# Problem: Predict the winner of the game on the basis of absolute difference of sum by selecting numbers

Given an array of **N** numbers. Two players **X** and **Y** play a game where at every step one player selects a number. One number can be selected only once. After all the numbers have been selected, player **X** wins if the **absolute difference** between the sum of numbers collected by **X** and **Y** is divisible by **4**, else **Y** wins.  
**Note:** Player X starts the game and numbers are selected optimally at every step.

**Examples:**

***Input:****a[] = {4, 8, 12, 16}****Output:****X  
X chooses 4  
Y chooses 12  
X chooses 8  
Y chooses 16  
|(4 + 8) – (12 + 16)| = |12 – 28| = 16 which is divisible by 4.  
Hence, X wins*

***Input:****a[] = {7, 9, 1}****Output:****Y*

Ref: <https://www.geeksforgeeks.org/predict-the-winner-of-the-game-on-the-basis-of-absolute-difference-of-sum-by-selecting-numbers/>

**38. Larger than Right Sub array**

Find all elements in an array that are greater than all elements present to their right.

Description: Given an unsorted array of integers, print all elements which are greater than all elements present to its right.

For Example:

Input: [10, 4, 6, 3, 5]

Output: [10, 6, 5]

39. **Colouring Array**

Minimum number of colours required to colour a Circular Array.

Description: Given a circular array arr[] containing N integers, the task is to find the minimum number of colours required to colour the array element such that two adjacent elements having different values must not be coloured the same.

For Example:

Input: arr[] = {1, 2, 1, 1, 2}

Output: 2

Explanation:

Minimum 2 type of colours are required.

We can distribute colour as {r, g, r, r, g} such that no adjacent element having different value are coloured same.

**40. Array VS XOR**

Minimum Bitwise XOR operations to make any two array elements equal

Given an array arr[] of integers of size **N** and an integer **K**. One can perform the Bitwise XOR operation between any array element and **K** any number of times. The task is to print the minimum number of such operations required to make any two elements of the array equal. If it is not possible to make any two elements of the array equal after performing the above-mentioned operation then print -1.

**Examples:**

***Input :****arr[] = {1, 9, 4, 3}, K = 3****Output :****-1****Explanation :****No possible to make any two elements equal*

***Input :****arr[] = {13, 13, 21, 15}, K = 13****Output :****0****Explanation :****Already exsits two same elements*

Ref: <https://www.geeksforgeeks.org/minimum-bitwise-xor-operations-to-make-any-two-array-elements-equal/>

# 