Task1: Hello Python

-Print hello python

-Print hello + your name

Task2:   
Given an integer, , perform the following conditional actions:

* If  n is odd, print Weird
* If  n is even and in the inclusive range of  to , print Not Weird
* If n is even and in the inclusive range of  to , print Weird
* If  n is even and greater than , print Not Weird

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Task3:

he provided code stub reads two integers from STDIN, a and b . Add code to print three lines where:

1. The first line contains the sum of the two numbers.
2. The second line contains the difference of the two numbers (first - second).
3. The third line contains the product of the two numbers.

Task4:

The provided code stub reads two integers, a and b, from STDIN.

Add logic to print two lines. The first line should contain the result of integer division,  a// b. The second line should contain the result of float division, a /b .

No rounding or formatting is necessary.

**Task5:**  
The provided code stub reads and integer, , from STDIN. For all non-negative integers i < n , print i^2 .

Task6:

An extra day is added to the calendar almost every four years as February 29, and the day is called a *leap day*. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used to identify leap years:

* The year can be evenly divided by 4, is a leap year, unless:
  + The year can be evenly divided by 100, it is NOT a leap year, unless:
    - The year is also evenly divisible by 400. Then it is a leap year.

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. [Source](http://www.timeanddate.com/date/leapyear.html)

**Task:**

Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean True, otherwise return False.

Note that the code stub provided reads from STDIN and passes arguments to the is\_leap function. It is only necessary to complete the is\_leap function

Task 7:

The included code stub will read an integer, n , from STDIN.

Without using any string methods, try to print the following:

1234…n. Example n=5. Print 12345

Task8:

Apply your knowledge of the .add() operation to help your friend Rupal.  
  
Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of N country stamps.  
  
Find the total number of distinct country stamps.

**Input Format**

The first line contains an integer N, the total number of country stamps.  
The next N lines contains the name of the country where the stamp is from.

Task9:

Let's learn about list comprehension! You are given three integers  and  representing the dimensions of a cuboid along with an integer . Print a list of all possible coordinates given by ( i, j, k) on a 3D grid where the sum of i+j+k  is not equal to n . Here,0<=i<=x, 0<=j<=y, 0<=z<=k . Please use list comprehensions rather than multiple loops, as a learning exercise.

Task 10:

Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score (2nd place ). You are given n  scores. Store them in a list and find the score of the runner-up.

**Input Format**

The first line contains n. The second line contains an array a[n]  of  integers each separated by a space.

Constrains: 2<=n<=10 and -100<=a[i]<=100

Task11:

Given the names and grades for each student in a class of N students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

**Note:** If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

**Output Format**

Print the name(s) of any student(s) having the second lowest grade in. If there are multiple students, order their names alphabetically and print each one on a new line.

Task 12:

The provided code stub will read in a dictionary containing key/value pairs of name:[marks] for a list of students. Print the average of the marks array for the student name provided, showing 2 places after the decimal.

Task 13:

Consider a list (list = []). You can perform the following commands:

1. insert i e: Insert integer  at position .
2. print: Print the list.
3. remove e: Delete the first occurrence of integer .
4. append e: Insert integer  at the end of the list.
5. sort: Sort the list.
6. pop: Pop the last element from the list.
7. reverse: Reverse the list.

Initialize your list and read in the value of  followed by  lines of commands where each command will be of the  types listed above. Iterate through each command in order and perform the corresponding operation on your list.

**Sample Input 0**

12

insert 0 5

insert 1 10

insert 0 6

print

remove 6

append 9

append 1

sort

print

pop

reverse

print

**Sample Output 0**

[6, 5, 10]

[1, 5, 9, 10]

[9, 5, 1]

Task 14: **Task**  
Given an integer,n , and  space-separated integers as input, create a tuple,t, of those n integers. Then compute and print the result of  hash(t).

**Note:** [hash()](https://docs.python.org/3/library/functions.html#hash) is one of the functions in the \_\_builtins\_\_ module, so it need not be imported.

**Output Format**

Print the result of  hash(t).

**Sample Input 0**

2

1 2

**Sample Output 0**

3713081631934410656

Task15:

You are given a string and your task is to *swap cases*. In other words, convert all lowercase letters to uppercase letters and vice versa.

**For Example:**

Www.HackerRank.com → wWW.hACKERrANK.COM

Pythonist 2 → pYTHONIST 2

**Function Description**

Complete the *swap\_case* function in the editor below.

*swap\_case* has the following parameters:

* *string s:* the string to modify

**Sample Input :**

HackerRank.com presents "Pythonist 2".

**Sample Output :**

hACKERrANK.COM PRESENTS "pYTHONIST 2".

Task16:

You are given a string “ hi every body”, print it as hi-every-body.

Hint: use split then join methods☺

Task17: You are given the firstname and lastname of a person on two different lines. Your task is to read them and print the following:

Hello firstname lastname! You just delved into python.

**Sample Input 0**

Ross

Taylor

**Sample Output 0**

Hello Ross Taylor! You just delved into python.

Task 18:

**Task**  
Read a given string, change the character at a given index and then print the modified string.  
**Function Description**

Complete the *mutate\_string* function in the editor below.

*mutate\_string* has the following parameters:

* *string string:* the string to change
* *int position:* the index to insert the character at
* *string character:* the character to insert

**Returns**

* *string:* the altered string
* **Sample Input**
* STDIN Function

----- --------

abracadabra s = 'abracadabra'

5 k position = 5, character = 'k'

* **Sample Output**
* abrackdabra

Task19:

In this challenge, the user enters a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

**NOTE:** String letters are case-sensitive.

**Input Format**

The first line of input contains the original string. The next line contains the substring.

**Output Format**

Output the integer number indicating the total number of occurrences of the substring in the original string.

**Sample Input**

ABCDCDC

CDC

**Sample Output**

2

Task20:

Python has built-in string validation methods for basic data. It can check if a string is composed of alphabetical characters, alphanumeric characters, digits, etc.

**Have a look here:**

[str.isalnum()](https://docs.python.org/2/library/stdtypes.html#str.isalnum) **,** [str.isalpha()](https://docs.python.org/2/library/stdtypes.html#str.isalpha)**, …,** [str.isupper()](https://docs.python.org/2/library/stdtypes.html#str.isupper)**.**

**Task**

You are given a string **S**.  
Your task is to find out if the string **S** contains: alphanumeric characters, alphabetical characters, digits, lowercase and uppercase characters.

**Input Format**

A single line containing a string **S.**

**Output Format**

In the first line, print True if **S** has any alphanumeric characters. Otherwise, print False.  
In the second line, print True if **S** has any alphabetical characters. Otherwise, print False.  
In the third line, print True if **S** has any digits. Otherwise, print False.  
In the fourth line, print True if **S** has any lowercase characters. Otherwise, print False.  
In the fifth line, print True if **S** has any uppercase characters. Otherwise, print False.

**Sample Input**

qA2

**Sample Output**

True

True

True

True

True

**Task21:**

In Python, a string of text can be aligned left, right and center.

**.ljust(width)**

This method returns a left aligned string of length width.

>>> width = 20

>>> print 'HackerRank'.ljust(width,'-')

HackerRank----------

**.center(width)**

**.rjust(width)**

**Task**

You are given a partial code that is used for generating the HackerRank Logo of variable thickness.  
Your task is to replace the blank (\_\_\_\_\_\_) with rjust, ljust or center.

**Input Format**

A single line containing the thickness value for the logo.

**Constraints**

The thickness must be an odd number.

**Output Format**

Output the desired logo.

**Sample Input**

5

**Sample Output**

H

HHH

HHHHH

HHHHHHH

HHHHHHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHHHHHHHHHHHHHHHHHHHHHH

HHHHHHHHHHHHHHHHHHHHHHHHH

HHHHHHHHHHHHHHHHHHHHHHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHH HHHHH

HHHHHHHHH

HHHHHHH

HHHHH

HHH

H

Task22:

You are given a string  and width .  
Your task is to wrap the string into a paragraph of width .

**Function Description**

Complete the *wrap* function in the editor below.

*wrap* has the following parameters:

* *string string:* a long string
* *int max\_width:* the width to wrap to

**Returns**

* *string:* a single string with newline characters ('\n') where the breaks should be
* **Sample Input 0**
* ABCDEFGHIJKLIMNOQRSTUVWXYZ
* 4
* **Sample Output 0**
* ABCD
* EFGH
* IJKL
* IMNO
* QRST
* UVWX
* YZ

### We wrote our own function to do that. But there are built in methods under the module [Textwrap](https://docs.python.org/2/library/textwrap.html#module-textwrap)

[textwrap.wrap()](https://docs.python.org/2/library/textwrap.html#textwrap.wrap)

[textwrap.fill()](https://docs.python.org/2/library/textwrap.html#textwrap.fill)

Task 23: Mr. Vincent works in a door mat manufacturing company. One day, he designed a new door mat with the following specifications:

* Mat size must be X. ( is an odd natural number, and  is  times .)
* The design should have 'WELCOME' written in the center.
* The design pattern should only use |, . and - characters.

**Sample Designs**

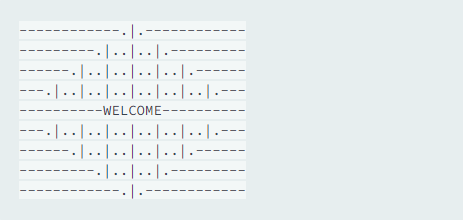
A screenshot of a computer screen

Description automatically generated

**Sample Input**

9 27

**Sample Output**



Task 24:

Given an integer, n, print the following values for each integer i from 1 to n:

1. Decimal
2. Octal
3. Hexadecimal (capitalized)
4. Binary

**Function Description**

Complete the *print\_formatted* function in the editor below.

*print\_formatted* has the following parameters:

* *int number:* the maximum value to print

**Prints**

The four values must be printed on a single line *in the order specified above* for each  from  to . Each value should be space-padded to match the width of the *binary* value of  and the values should be separated by a single space.

**Sample Input**

17

**Sample Output**

1 1 1 1

2 2 2 10

3 3 3 11

4 4 4 100

5 5 5 101

6 6 6 110

7 7 7 111

8 10 8 1000

9 11 9 1001

10 12 A 1010

11 13 B 1011

12 14 C 1100

13 15 D 1101

14 16 E 1110

15 17 F 1111

16 20 10 10000

17 21 11 10001

Task25:

You are given an array of integers, Your task is to create paires of them, such that every pair consists of equal numbers. Each array element may belong to one pair only. Is it possible to use all of the integers?

Write a function:

Def solution(A):

That, given an array A consisting of N integers, returns whether it is possible to split all integers into pairs.

Examples:

* 1. Given A=[1,2,2,1], your function should return Ture, as the pairs are (A[0], A[3]) (both have value 1) and others also have value 2.
  2. A=[7,7,7] return False
  3. A=[1,2,2,3] return False.

Task26:

Write a function solution that, given a string S Consisting of N characters, returns the alphabetically smallest string that can be obtained by removing exactly one letter from S.

Example:

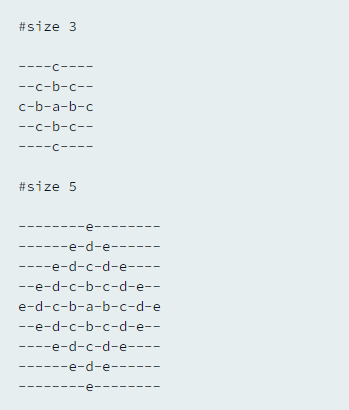
Given S= “acb”, by removing one letter, you can obtain ‘ac’, ‘ab’ or ‘cb’. Your function return “ab” (after removing ‘c’ ) since it is alphabetically smaller than “ac” and “bc”.

Another Example: Given “codility”. Your function should return “cdility”. Which can be obtained by removing the second letter.

Task27:

You are given an integer, N. Your task is to print an alphabet rangoli of N. (Rangoli is a form of Indian folk art based on creation of patterns.)

Different sizes of alphabet rangoli are shown below:



The center of the rangoli has the first alphabet letter *a*, and the boundary has the  alphabet letter (in alphabetical order).

**Function Description**

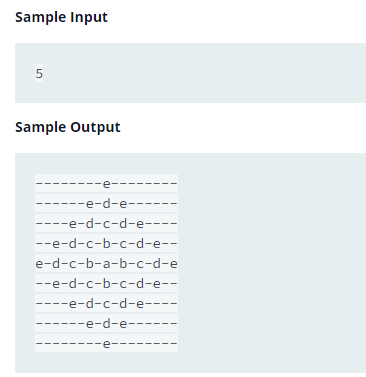
Complete the *rangoli* function in the editor below.

*rangoli* has the following parameters:

* *int size:* the size of the rangoli

**Returns**

* *string:* a single string made up of each of the lines of the rangoli separated by a newline character (\n)



Task28:

You are asked to ensure that the first and last names of people begin with a capital letter in their passports. For example, alison heck should be capitalised correctly as Alison Heck.

**Constraints**

* The string consists of alphanumeric characters and spaces.

**Input Format**

A single line of input containing the full name, S.

**Note:** in a word only the first character is capitalized. Example 12abc when capitalized remains 12abc.

**Output Format**

Print the capitalized string, .

**Sample Input**

chris alan

**Sample Output**

Chris Alan