

Name: Shumaim Qamar
Roll no:BSDSF24A017

IDS ASSIGNMENT PYTHON PRACTICE IN HACKERRANK

Problem#1: If-Else

We have to input an integer n then check whether it is weird or not by using if-else statements.

- If n is odd, it is Weird
- If n is even and in the inclusive range of 2 and 5, it is not weird.
- If n is even and in the inclusive range of 6 and 20, it is weird.
- If n is even and above 20, it is not weird.

```
1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9
10
11  ✓ if __name__ == '__main__':
12  |     n = int(input().strip())
13  ✓ if n%2!=0:
14  |     print("Weird")
15  ✓ elif n%2 == 0 and 2 <= n <= 6:
16  |     print("Not Weird")
17  ✓ elif n%2 == 0 and 6<=n<=20:
18  |     print("Weird")
19  ✓ elif n%2==0 and n>=20:
20  |     print("Not Weird")
```

Line: 20 Col: 23

Problem#2: Loops

We have to input an integer from the user and then print square of integers till n using a for loop. Here we used the for loop syntax in which i will iterate from 0 to n-1 and prints the i*i mean it's square.

```
1  √ if __name__ == '__main__':  
2      n = int(input())  
3  
4  √   for i in range(0,n):  
5      print(i*i)
```

Problem#03: Swap Case

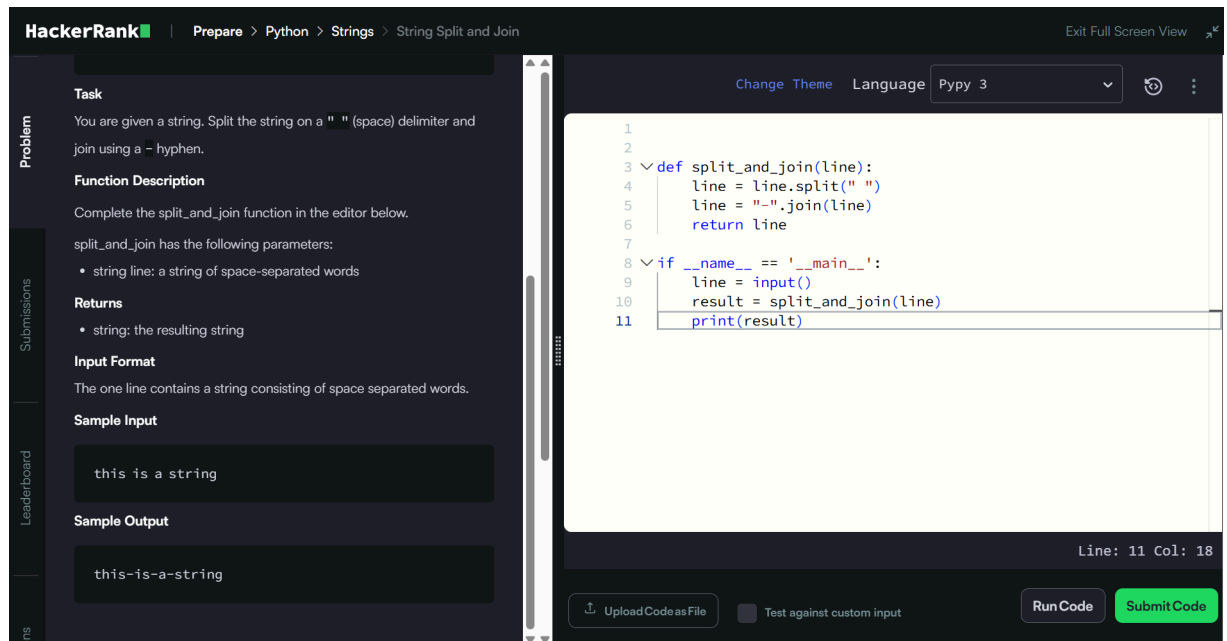
In this program a string is passed to a function swap_case and then it swap the upper case letters to lower case and lower case to upper case.

By using for loop and if-else conditions we can write code for this problem. upper() and lower() are the built-in functions which change the cases.

```
1  def swap_case(s):  
2      result = ""  
3      for ch in s:  
4          if 'A' <= ch <= 'Z':  
5              result += ch.lower()  
6          elif 'a' <= ch <= 'z':  
7              result += ch.upper()  
8          else:  
9              result += ch  
10     return result  
11  
12  √ if __name__ == '__main__':  
13      s = input()  
14      result = swap_case(s)  
15      print(result)
```

Problem#04: String Split and Join

In this problem, a string is passed to the function 'split_and_join'. In this function we have to replace space with '-'. But as String is immutable so we cannot make changes in the string directly. So with the help of a list we can make it possible. Split() and join() are the built-in functions in python specially made for this purpose. Firstly we have to split, which means changing the string to the list. And then we will use "-".join(line) to change back to string but with '-'.



HackerRank | Prepare > Python > Strings > String Split and Join | Exit Full Screen View

Problem

Task
You are given a string. Split the string on a " " (space) delimiter and join using a - hyphen.

Function Description
Complete the split_and_join function in the editor below.

split_and_join has the following parameters:

- string line: a string of space-separated words

Returns
• string: the resulting string

Input Format
The one line contains a string consisting of space separated words.

Sample Input
this is a string

Sample Output
this-is-a-string

Submissions

Leaderboard

Code Editor

```
1
2
3 def split_and_join(line):
4     line = line.split(" ")
5     line = "-".join(line)
6     return line
7
8 if __name__ == '__main__':
9     line = input()
10    result = split_and_join(line)
11    print(result)
```

Line: 11 Col: 18

Upload Code as File | Test against custom input | Run Code | Submit Code

Problem#05: Mutations

In this problem, we have to change the character from the given string and then print the modified string. As string is immutable so we will use list to solve this problem. First we will convert the string to list then put the character in the required position. And use the function join to again change into the string. Return the modified string.

HackerRank | Prepare > Python > Strings > Mutations

Exit Full Screen View

Problem

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

Input Format
The first line contains a string, `string`.
The next line contains an integer `position`, the index location and a string `character`, separated by a space.

Sample Input

STDIN	Function
-----	-----
abracadabra	<code>s = 'abracadabra'</code>
5 k	<code>position = 5, character = 'k'</code>

```

1 def mutate_string(string, position, character):
2     l = list(string)
3     l[position] = character
4     string = ''.join(l)
5     return string
6
7 if __name__ == '__main__':
8     s = input()
9     i, c = input().split()
10    s_new = mutate_string(s, int(i), c)
11    print(s_new)

```

Line: 11 Col: 17

Upload Code as File Test against custom input Run Code Submit Code

Problem#06: Symmetric Difference

In this program we have to find the non common elements in the two sets and display it in ascending order.

Here we will use the `append` function that adds the element at the end of the list.

And `Sort()` function that will sort the list of elements in ascending order.

HackerRank | Prepare > Python > Sets > Symmetric Difference

Exit Full Screen View

Problem

Task
Given 2 sets of integers, M and N , print their symmetric difference in ascending order. The term symmetric difference indicates those values that exist in either M or N but do not exist in both.

Input Format
The first line of input contains an integer, M .
The second line contains M space-separated integers.
The third line contains an integer, N .
The fourth line contains N space-separated integers.

Output Format
Output the symmetric difference integers in ascending order, one per line.

Sample Input

STDIN	Function
-----	-----
4	<code>set a size M = 4</code>
2 4 5 9	<code>a = {2, 4, 5, 9}</code>
4	<code>set b size N = 4</code>
2 4 11 12	<code>b = {2, 4, 11, 12}</code>

```

5 N = int(input())
6 b = list(map(int, input().split()))
7
8 M1 = []
9 for x in a:
10     if x not in M1:
11         M1.append(x)
12
13 N1 = []
14 for x in b:
15     if x not in N1:
16         N1.append(x)
17
18 only_M = []
19 for x in M1:
20     if x not in N1:
21         only_M.append(x)
22
23 only_N = []
24 for x in N1:

```

Line: 15 Col: 19

Upload Code as File Test against custom input Run Code Submit Code

```
13 N1 = []
14 for x in b:
15     if x not in N1:
16         N1.append(x)
17
18 only_M = []
19 for x in M1:
20     if x not in N1:
21         only_M.append(x)
22
23 only_N = []
24 for x in N1:
25     if x not in M1:
26         only_N.append(x)
27
28 sym_diff = only_M + only_N
29 sym_diff.sort()
30
31 for x in sym_diff:
32     print(x)
33
34 |
```

Line: 34 Col: 1

Problem#07: Arithmetic Operators

+, -, * operators are used to compute a and b. Then print the output.

HackerRank

Prepare > Python > Introduction > Arithmetic Operators

Exit Full Screen View

Problem

Submissions

Leaderboard

Check [Tutorial](#) tab to know how to solve.

Task

The 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.

Example

a = 3
b = 5

Print the following:

```
8
-2
15
```

Input Format

The first line contains the first integer, *a*.

Change Theme

Language Pypy 3

⌂ ⋮

```
1 if __name__ == '__main__':
2     a = int(input())
3     b = int(input())
4
5     print(a+b)
6     print(a-b)
7     print(a*b)
```

Line: 7 Col: 15

Problem#08: Division

// is used to output int integers and / gives the output in float no matter what their data types are.

HackerRank | Prepare | Python | Introduction | Python: Division | Exit Full Screen View

Check the [Tutorial](#) tab to know learn about division operators.

Task
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.

Example
 $a = 3$
 $b = 5$

- The result of the integer division $3 // 5 = 0$.
- The result of the float division is $3 / 5 = 0.6$.

Print:

```
0
0.6
```

Change Theme | Language | Pypy 3

```
1 if __name__ == '__main__':
2     a = int(input())
3     b = int(input())
4     print(a//b)
5     print(a/b)
```

Problem#09: String Validators

We used these built in functions to check if the string is valid or not and in this base we print True and False.

isalnum() - to check whether string has any alphanumeric characters

isalpha() - to check whether string has any alphabetic characters

isdigit() - to check whether the string has any digit character

isupper() - to check whether the string has any uppercase characters

islower() - to check whether the string has any lowercase characters

HackerRank | Prepare > Python > Strings > String Validators

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*.

Constraints
 $0 < \text{len}(S) < 1000$

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

```

1 if __name__ == '__main__':
2     s = input()
3
4     a,b,c,d,e = False, False, False, False, False
5
6     for ch in s:
7         if(ch.isalnum()):
8             a=True
9         if(ch.isalpha()):
10            b=True
11        if(ch.isdigit()):
12            c=True
13        if(ch.islower()):
14            d=True
15        if(ch.isupper()):
16            e=True
17    print(a)
18    print(b)
19    print(c)
20    print(d)
21    print(e)

```

Line: 17 Col: 5

Upload Code as File Test against custom input Run Code Submit Code

Problem#10: Find a string

In this program, we want to find the count of the sub-string in the given string. For this, we will use a loop to iterate through the string and for each iteration we will use the if condition to check whether the sub-string exists in that string or not. If True then count will increase by 1. len(string name) function is used to find the length of a string.

HackerRank | Prepare > Python > Strings > Find a string

Problem
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.

Constraints
 $1 \leq \text{len}(\text{string}) \leq 200$
Each character in the string is an ascii character.

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

Sample Input

```

ABCD CDC
CDC

```

Sample Output

```

2

```

Code:

```

1 def count_substring(string, sub_string):
2     count = 0;
3     for i in range(0, len(string)):
4         if string[i:i+len(sub_string)] == sub_string:
5             count+=1
6     return count
7
8 > if __name__ == '__main__':

```

Line: 8 Col: 27

Upload Code as File Test against custom input Run Code Submit Code

Problem#11: No Idea!

In this program, we are checking what is the count of the happiness

- We are happy if the element in the array is equal to the element in A set.
- We are not happy if the element in the array is equal to the element in B set.

map() converts each string into an integer.

Split() is used to split string by spaces into a list/set of strings.

The screenshot shows the HackerRank interface for the problem 'No Idea!'. The left sidebar contains the problem description, constraints, input/output formats, and sample input. The main area displays a Python solution. The problem description states: 'There is an array of n integers. There are also 2 disjoint sets, A and B , each containing m integers. You like all the integers in set A and dislike all the integers in set B . Your initial happiness is 0. For each i integer in the array, if $i \in A$, you add 1 to your happiness. If $i \in B$, you add -1 to your happiness. Otherwise, your happiness does not change. Output your final happiness at the end.' The constraints are: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$, and $1 \leq \text{Any integer in the input} \leq 10^9$. The input format specifies three lines: n and m , the array elements, and the elements of sets A and B . The output format requires a single integer for total happiness. The sample input is:
5 2
1 2 3 4 5
1 3 5
2 4

```
1 n, m = map(int, input().split())
2 arr = list(map(int, input().split()))
3 A = set(map(int, input().split()))
4 B = set(map(int, input().split()))
5
6
7 happiness = 0
8 for i in arr:
9     if i in A:
10        happiness+=1
11    elif i in B:
12        happiness-=1
13
14 print(happiness)
15
```

Problem#12: Print Function

Input a number then print a string having chr 1 to n numbers.

Here str() is used to typecast from int to string.

The screenshot shows the HackerRank interface for the problem 'Print Function'. The left sidebar contains the problem description, constraints, input/output formats, and sample input. The problem description states: 'The included code stub will read an integer, n , from STDIN. Without using any string methods, try to print the following: 123... n . Note that '...' represents the consecutive values in between.' The example shows $n=5$ resulting in the string '12345'. The input format specifies a single line with integer n . The constraints are $1 \leq n \leq 150$. The output format requires the list of integers from 1 through n as a string, without spaces. The sample input is 3. The sample output is 123.

```
1 if __name__ == '__main__':
2     n = int(input())
3     s=""
4     for i in range(1,n+1):
5         s+= str(i)
6
7     print(s)
```


Problem#13: Find the Runner-Up score

I have to find the runner-up(second largest number). First we find the max value then in each loop iteration use the if condition to check if $\text{score} \neq \text{max}$ and $\text{score} > \text{runner_up}$. In this way, we will find the runner-up at the end.

The screenshot shows the HackerRank interface for the problem "Find the Runner-Up Score!". The left sidebar contains navigation links: Problem, Submissions, Leaderboard, and Profile. The main content area on the left provides the problem details:

Problem
Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score. 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 $A[]$ of n integers each separated by a space.

Constraints

- $2 \leq n \leq 10$
- $-100 \leq A[i] \leq 100$

Output Format
Print the runner-up score.

Sample Input 0

```
5
2 3 6 6 5
```

Sample Output 0

```
5
```

The right side of the interface shows a code editor with a Python solution. The language is set to Pypy 3. The code is as follows:

```
1 if __name__ == '__main__':
2     n = int(input())
3     arr = list(map(int, input().split()))
4
5     max_score = max(arr)
6     runner_up = -999999
7
8     for score in arr:
9         if score != max_score and score > runner_up:
10             runner_up = score
11
12     print(runner_up)
```

At the bottom of the code editor, there are buttons for "Upload Code as File", "Test against custom input", "Run Code", and "Submit Code". The status bar at the bottom right indicates "Line: 12 Col: 16".