WEEK - 8

Tuples & Sets

1) Coders here is a simple task for you, Given string str. Your task is to check whether it is a binary string or not by using python set.

Input	Result
01010101010	Yes
REC101	No

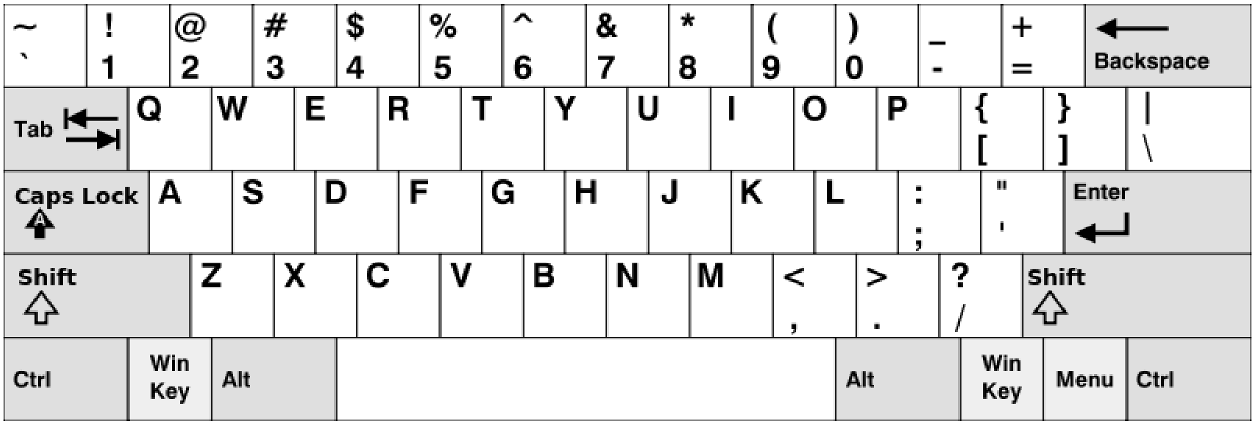
PROGRAM :

```
 n=input()
if n.isdigit():
        print(“Yes”)
else:
        print(“No”)
```

2) Given an array of string words, return the words that can be typed using letters of the alphabet on only one row of the American keyboard like the image below.

In the American keyboard:

- The first row consists of the characters "qwertyuiop",
- The second row consists of the characters "asdfghjkl", and
- The third row consists of the characters "zxcvbnm".


~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Backspace
Tab Q W E R T Y U I O P { [}] | \
Caps Lock A S D F G H J K L : ; " ' Enter
Shift Z X C V B N M < , > . ? / Shift
Ctrl Win Key Alt Alt Win Key Menu Ctrl

Input	Result

["Hello","Alaska","Dad","Peace"]	["Alaska","Dad"]
["omk"]	[]

```
n=int(input())
1=[]
a=[]
for i in range(n):
        1.append(input())
                for i in 1:
                        k1='qwertyuiop' k2='asdfghjkl' k3='zxcvbnm'
                        w=set(i.lower())
                        if w.issubset(k1) or w.issubset (k2) or w.issubset(k3):
                                a.append(i)
if len(a)==0:
        print('No words')
else:
        for i in a:
                print(i)
```

3) Write a program to eliminate the common elements in the given 2 arrays and print only the non-repeating elements and the total number of such non-repeating elements.

Input	Result

5 4
1 2 8 6 5
2 6 8 10
	1 5 10
	3

5 5
1 2 3 4 5
1 2 3 4 5
	NO SUCH ELEMENTS

PROGRAM :

```
n,m=map(int,input().split())
11-list (map(int,input().split()))
12=list(map(int,input().split()))
|13=[]
for i in 11:
        if i not in 12:
                13.append(i)
for i in 12:
        if i not in 11:
                13.append(i)
if len(13)>0:
        print(*13)
        print(len(13))
else:
        print("NO SUCH ELEMENTS")
```

4) The DNA sequence is composed of a series of nucleotides abbreviated as 'A', 'C', 'G', and 'T'.For example, "ACGAATTCCG" is a DNA sequence.

When studying DNA, it is useful to identify repeated sequences within the DNA.

Given a string s that represents a DNA sequence, return all the 10-letter-long sequences (substrings) that occur more than once in a DNA molecule.

Input	Result
AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

PROGRAM :

```
Str = input()
map = dict()
res = []
```

```
for i in range(len(Str) - 9):
     s = Str[i:i + 10]
    if s in map:
        map[s] += 1
    else:
        if map[s] == 2:
            res.append(s)
map[s] = 1
for i in res:
    print(i)
```

5) Check if a set is a subset of another set.

Test	Input	Result
1	mango apple mango orange mango	yes set3 is subset of set1 and set2
2	mango orange banana orange grapes	No

PROGRAM :

```
s1=set(input())

s2=set(input())

s3=set(input())

if (s3.issubset(s1) and s3.issubset(s2)):

    print("yes")

    print("set3 is subset of set1 and set2")

else:

    print("No")
```

6) There is a malfunctioning keyboard where some letter keys do not work.

All other keys on the keyboard work properly.

Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard.

Input	Result
hello world ad	1
Faculty Upskilling in Python Programming Ak	2

PROGRAM :

```
s=input().split()
b=input()
C = 0
for i in bl:
        for j in range(len(s)):
                if (i in s[j] or i.upper() in s[j]): c += 1
                        print(len(s) - c)
```

7) Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive.

There is only one repeated number in nums, return this repeated number.

Solve the problem using set.

Input	Result
1 3 4 4 2	4

PROGRAM :

```
s=input()
```

```
c=[]
for i in s:
    if i not in c:
        c.append(i)
d=[]
for i in s:
    if i in c:
        d.append(i)
print(d[4])
```

8) You are given an integer tuple nums containing distinct numbers. Your task is to perform a sequence of operations on this tuple until it becomes empty. The operations are defined as follows:

If the first element of the tuple has the smallest value in the entire tuple, remove it.

Otherwise, move the first element to the end of the tuple.

You need to return an integer denoting the number of operations required to make the tuple empty.

The input tuple nums contains distinct integers.

The operations must be performed using tuples and sets to maintain immutability and efficiency.

Your function should accept the tuple nums as input and return the total number of operations as an integer.

Input: nums = (3, 4, -1)

Output: 5

Explanation:

Operation 1: [3, 4, -1] -> First element is not the smallest, move to the end -> [4, -1, 3]

Operation 2: [4, -1, 3] -> First element is not the smallest, move to the end -> [-1, 3, 4]

Operation 3: [-1, 3, 4] -> First element is the smallest, remove it -> [3, 4]

Operation 4: [3, 4] -> First element is the smallest, remove it -> [4]

Operation 5: [4] -> First element is the smallest, remove it -> []

Total operations: 5

Test	Result
print(count_operations((3, 4, -1)))	5

PROGRAM :

```
def count_operations (nums: tuple) -> int:
# Your implementation here
    arr= list(nums)
    operations =0
    i = 0
    while len(arr) != 0:
        m = min(arr)
        if (arr[i] == m):
            arr.remove(arr[i])
            operations += 1
        else:
            arr.append(arr[i])
            arr.remove(arr[i])
            operations += 1
    return operations
```

9) Program to print all the distinct elements in an array.

Input Format:First line take an Integer input from stdin which is array length n.Second line take n Integers which is inputs of array.

Output Format:Print the Distinct Elements in Array in single line which is space Separated

Input	Result
5 1 2 2 3 4	1 2 3 4

```
n=int(input())
a=[]
for i in range(n):
	a.append(int(input())
s=set(a)
print(*s)
```

10) Given a tuple and a positive integer k, the task is to find the count of distinct pairs in the tuple whose sum is equal to K.

Input: t = (5, 6, 5, 7, 7, 8), K = 13; Output: 2
Explanation:
Pairs with sum K(= 13) are {(5, 8), (6, 7), (6, 7)}.
Therefore, distinct pairs with sum K(= 13) are { (5, 8), (6, 7) }.
Therefore, the required output is 2.

Input	Result
1,2,1,2,5 3	1
1,2 0	0

PROGRAM :

```
s=input()
s1=[]
k=int(input())
```

```
 for i in s:
        if i.isdigit():
                s1.append(int(i))
s3=[]
s2=set(s1)
for i in s2:
        for j in s2:
                if i+j==k:
                        s3.append(i)
1=set(s3)
print(len(1)//2)
```