

For example:

Test	Result
print(checkUgly(6))	ugly
print(checkUgly(21))	not ugly

Ex. No. : 7.7

Date:

Register No.: 230701358

Name: TARUN C

Ugly number

A number is considered to be ugly if its only prime factors are 2, 3 or 5.

[1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...] is the sequence of ugly numbers.

Task:

complete the function which takes a number n as input and checks if it's an ugly number.

return ugly if it is ugly, else return not ugly

Hint:

An ugly number U can be expressed as: $U = 2^a * 3^b * 5^c$, where a, b and c are nonnegative integers.

Program:

```
def checkUgly(n):
```

```
    if n<=0:
```

```
        return "not ugly"
```

```
    for p in [2,3,5]:
```

```
        while n%p==0:
```

```
            n//=p
```

```
    if n==1:
```

```
        return "ugly"
```

```
    else:
```

```
        return "not ugly"
```

```
n=input()
```


08 – Tuple/Set

Examples:

Input: str = "01010101010"

Output: Yes

Input: str = "REC101"

Output: No

For example:

Input	Result
01010101010	Yes
010101 10101	No

Ex. No. : 8.1

Date:

Register No.: 230701358

Name: TARUN C

Binary String

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.

Program:

```
n=input()
a=set(n)
if a=={'0','1'}:
    print("Yes")
else:
    print("No")
```

Examples:

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.

For example:

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

Ex. No. : 8.2

Date:

Register No.: 230701358

Name: TARUN C

Check Pair

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**.

Program:

```
t=str(input())
tu=t.split(',')
k=int(input())
numbers=[]
for i in tu:
    if i.isdigit():
        numbers.append(int(i))
d=0
s=set()
pairs=set()
for num in numbers:
    c=k-num
    if c in s:
        pairs.add((min(num,c),max(num,c)))
    s.add(num)
d=len(pairs)
print(d)
```


Example 1:

Input: s = "AAAAACCCCCAAAAACCCCCAAAAAGGGTTT"

Output: ["AAAAACCCCC","CCCCAAAAA"]

Example 2:

Input: s = "AAAAAAAAAAAAA"

Output: ["AAAAAAAAA"]

For example:

Input	Result
AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCAAAAA

Ex. No. : 8.3

Date:

Register No.: 230701358

Name: TARUN C

DNA Sequence

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. You may return the answer in **any order**.

Program:

```
s = input()
sequences = set()
repeated_sequences = set()

for i in range(len(s) - 9):
    sequence = s[i:i+10]
    if sequence in sequences:
        repeated_sequences.add(sequence)
    else:
        sequences.add(sequence)

for sequence in repeated_sequences:
    print(sequence)
```

Example 1:**Input:** nums = [1,3,4,2,2]**Output:** 2**Example 2:****Input:** nums = [3,1,3,4,2]**Output:** 3**For example:**

Input	Result
1 3 4 4 2	4

Ex. No. : 8.4

Date:

Register No.: 230701358

Name: TARUN C

Print repeated no

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](#).

Program:

```
nums = [int(x) for x in input().split()]
```

```
seen = set()
```

```
for num in nums:
```

```
    if num in seen:
```

```
        print(num)
```

```
        break
```

```
    seen.add(num)
```

Sample Input:

5 4
1 2 8 6 5
2 6 8 10

Sample Output:

1 5 10
3

Sample Input:

5 5
1 2 3 4 5
1 2 3 4 5

Sample Output:

NO SUCH ELEMENTS

For example:

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

Ex. No. : 8.5

Date:

Register No.: 230701358

Name: TARUN C

Remove repeated

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 Format:

The first line contains space-separated values, denoting the size of the two arrays in integer format respectively.

The next two lines contain the space-separated integer arrays to be compared.

Program:

```
s = input()
a, b = s.split()
c = int(a)d = int(b)
al = input()
a2 = input()
bl = al.split()
b2 = a2.split()
c1 = list(map(int, bl))
c2 = list(map(int, b2))
set1 = set(c1)
set2 = set(c2)
n = set1.symmetric_difference(set2)
if n:
    print(*n)
```