Codekata Report:

Name: BUVANESH

Email: buvaneshr2018@gmail.com

1. Write a code to get the input in the given format and print the output in the given format

Sample Input:

2

Sample Output:

2

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Take an integer input from the user
number = int(input())

# Print the integer value
print(number)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4

Compilation Status: Passed

Execution Time:

0.01s

2. Write a code to get 2 integers A and N. Print the integer A, N times in separate line.

Sample Input:

2 3

Sample Output:

2
2
2

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```
no,no2= (int(no) for no in input().split() )
for i in range(0,no2):
print(no)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

5
5
5
5

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

10
10
10
10
10

Compilation Status: Passed

Execution Time:

0.01s

3. Given 3 numbers N , L and R. Print 'yes' if N is between L and R else print 'no'.Sample Testcase :INPUT32 6OUTPUTyes

Completion Status: Completed

Concepts Included:

mathematics

basics

Language Used: PYTHON 3

Source Code:

```
n=int(input())
l,r=(int(no) for no in input().split())
if n>l and n<r:
print('yes')
else:
print('no')
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.01s

4. Write a code to get an integer N and print values from 1 till N in a separate line.

Sample Input:

5

Sample Output:

1
2
3
4
5

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```
no=int(input())
for i in range(1,no+1):
print(i)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1
2
3
4
5
6
7
8
9
10

Compilation Status: Passed

Execution Time:

0.011s

5. Given a number N, print 'yes' if it is composite else print 'no'.Sample Testcase :INPUT123OUTPUTyes

Completion Status: Completed

Concepts Included:

mathematics

basics

Language Used: PYTHON 3

Source Code:

```
N=int(input())
count=0
for i in range(2,N):
if N%i==0:
```

```
count=count+1
if count>=1:
print('yes')
else:
print('no')
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.009s

6. Given numbers A,B find A^B.Input Size : 1 <= A <= 5 <= B <= 50Sample Testcase :INPUT3 4OUTPUT81

Completion Status: Completed

Concepts Included:

array

mathematics

basics

Language Used: PYTHON 3

Source Code:

```
import math
A,B=(int(no) for no in input().split())
result=math.pow(A,B)
print(round(result))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

243

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4

Compilation Status: Passed

Execution Time:

0.01s

7. Write a code to get the input and print it 5 times.

Sample Input:

4

Sample Output:

4
4
4
4
4

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```
no=int(input())
for i in range(0,5):
print(no)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

5

5
5
5
5

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

10
10
10
10
10

Compilation Status: Passed

Execution Time:

0.009s

8. Given 2 numbers N and M add both the numbers and check whether the sum is odd or even.Sample Testcase :INPUT9 2OUTPUTodd

Completion Status: Completed

Concepts Included:

basics

mathematics

Language Used: PYTHON 3

Source Code:

```
N,M=(int(no) for no in input().split())
```

```
result=(N+M)%2
if result==0:
print('even')
else:
print('odd')
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

odd

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

even

Compilation Status: Passed

Execution Time:

0.01s

9. Write a code to get an integer N and print the even values from 1 till N in a separate line.

Sample Input:

6

Sample Output:

2
4
6

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```
number=int(input())
for i in range(1,number+1):
if i%2==0:
print(i)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2
4
6
8
10
12
14
16
18
20
22
24
26
28

30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50

Compilation Status: Passed

Execution Time:

0.01s

10. Find the minimum among 10 numbers.Sample Testcase :INPUT5 4 3 2 1 7 6 10 8 9OUTPUT1

Completion Status: Completed

Concepts Included:

basics

mathematics

Language Used: PYTHON 3

Source Code:

```
numbers = input().split()
numbers = [int(num) for num in numbers]

minimum = min(numbers)
```

print(minimum)

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.012s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

11. Given 2 numbers N and K followed by elements of N .Print 'yes' if K exists else print 'no'.Sample Testcase :INPUT4 21 2 3 3OUTPUTyes

Completion Status: Completed

Concepts Included:

basics

array

Language Used: PYTHON 3

Source Code:

```
def check_element_existence(numbers, k):
"""Checks if element k exists in the list of numbers.

Args:
numbers: A list of numbers.
k: The element to check.

Returns:
True if k exists, False otherwise.
"""

return k in numbers

# Get input from the user
n, k = map(int, input().split())
numbers = list(map(int, input().split()))

# Check if k exists and print the result
if check_element_existence(numbers, k):
print("yes")
else:
print("no")
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.01s

12. You are given with a number ‘n’. You have to count the pair of two numbers a and b such that sum of two numbers are equal to n.

Note:Both numbers lie in range 1<=a,b<n

Sample Input:

5

Sample Output:

4

Completion Status: Completed

Concepts Included:

mathematics

integer

numbers

Language Used: PYTHON 3

Source Code:

```
def count_pairs(n):
"""Counts the number of pairs of integers a and b such that a + b = n.

Args:
```

n: The target sum.

Returns:
The number of pairs.
"""

```
count = 0
for a in range(1, n):
b = n - a
if 1 <= b < n:
count += 1
return count

# Get input from the user
n = int(input())

# Count and print the number of pairs
result = count_pairs(n)
print(result)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

22

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

11

Compilation Status: Passed

Execution Time:

0.009s

13. Write a code to get the input in the given format and print the output in the given format

Sample Input:

2 3 4 5 6 7 8

Sample Output:

2 3 4 5 6 7 8

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the input
input_line = input()

# Split the input string by spaces to get a list of integers
integer_list = input_line.split()

# Print the integers separated by space
print(' '.join(integer_list))
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 3 4 5 6 7 8

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

12 13 14 15 16 17 18

Compilation Status: Passed

Execution Time:

0.009s

14. Write a code to get the input in the given format and print the output in the given format.

Sample Input:

5 3
1 2 3 4 5

Sample Output:

5 3
1 2 3 4 5

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the first line of input
first_line = input()

# Read the second line of input
second_line = input()

# Print the first and second lines as the output
print(first_line)
print(second_line)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

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

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
4 2
1 4 3 2
```

Compilation Status: Passed

Execution Time:

0.01s

15. Write a code to get the input in the given format and print the output in the given format

Sample Input:

2 4
2 4
2 4

Sample Output:

2 4
2 4
2 4

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the first line of input
first_line = input()

# Read the second line of input
second_line = input()

# Read the third line of input
third_line = input()

# Print each line as the output
print(first_line)
print(second_line)
print(third_line)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 4
2 4
2 4

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 3
2 3
4 5

Compilation Status: Passed

Execution Time:

0.01s

16. Write a code to get the input in the given format and print the output in the given format

Sample Input:

2
4
5

Sample Output:

2 4 5

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the three integers from input
first_int = input()
second_int = input()
third_int = input()

# Print all three integers in a single line separated by space
print(first_int, second_int, third_int)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 4 5

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

9 9 9

Compilation Status: Passed

Execution Time:

0.01s

17. Write a code to get the input in the given format and print the output in the given format

Sample Input:

2 5
2 5 6
2 4 5

Sample Output:

2 5
2 5 6
2 4 5

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the first line of input
first_line = input()

# Read the second line of input
second_line = input()

# Read the third line of input
third_line = input()

# Print each line exactly as it was input
print(first_line)
print(second_line)
print(third_line)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 5
2 5 6
2 4 5

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 2
1 2 4
1 2 3

Compilation Status: Passed

Execution Time:

0.01s

18. Write a code to get the input in the given format and print the output in the given format

Sample Input:

guvi

Sample Output:

g u v i

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the input string
input_string = input()

# Print the characters of the string separated by space
print(' '.join(input_string))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

g u v i

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

c o d e k a t a

Compilation Status: Passed

Execution Time:

0.01s

19. Write a code to get the input in the given format and print the output in the given format.

Sample Input:

2.3 4.5 7.8

Sample Output:

2.3
4.5
7.8

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the input line containing three float values
input_floats = input()

# Split the input string into a list of float values
float_list = input_floats.split()

# Print each float value on a new line
for float_value in float_list:
print(float_value)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2.3
4.5
7.8

Compilation Status: Passed

Execution Time:

0.012s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1.2
3.4
5.6

Compilation Status: Passed

Execution Time:

0.011s

20. Write a code to get the input in the given format and print the output in the given format.

Sample Input:

guvigeek

Sample Output:

g
u
v
i
g
e
e
k

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the input string
input_string = input()

# Print each character of the string on a new line
for char in input_string:
print(char)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
g
u
v
i
g
e
e
k
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
c
o
d
e
```

Compilation Status: Passed

Execution Time:

0.01s

21. Write a code to get the input in the given format and print the output in the given format.

Sample Input:

guvi

Sample Output:

g,u,v,i

Completion Status: Completed

Concepts Included:

Input/Output

Language Used: PYTHON 3

Source Code:

```
# Read the input string
input_string = input()

# Print the characters of the string separated by a comma
print(','.join(input_string))
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

g,u,v,i

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

h,e,l,l,o

Compilation Status: Passed

Execution Time:

0.01s

22. A person saves his monthly saving according to given schema. He saves same amount of money which is equal to the money saved in immediate previous two months. Assume, initially he saved 1000 rupees and in first month he saved another 1000. Your task is to tell how much he had totally saved at the end of ‘n’ months

Sample Input:

1

Sample Output:

2000

Completion Status: Not Completed

Concepts Included:

mathematics

array

Language Used: PYTHON 3

Source Code:

```
# Read the number of months 'n'
```

```
n = int(input())

# Base cases: If n is 1, total savings is 2000 (1000 for initial and 1000 for the first
month)
if n == 1:
total_savings = 2000
else:
# Initial savings
first_month = 1000
second_month = 1000
total_savings = first_month + second_month

# Calculate the savings for the remaining months (if n > 1)
for i in range(2, n):
current_saving = first_month + second_month
total_savings += current_saving
first_month = second_month
second_month = current_saving

# Print the total savings
print(total_savings)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2000

Compilation Status: Failed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4000

Compilation Status: Failed

Execution Time:

0.011s

23. In a firm there is an intelligent employee. He said that he will not work on all those days which has factors more than 2. You are given with month and year calculate the no of working days of employee.

Sample Input:

May 2016

Sample Output:

11

Completion Status: Completed

Concepts Included:

mathematics

calendar

factorial

Language Used: PYTHON 3

Source Code:

```
import calendar

def is_prime(n):
"""Check if a number is prime."""
if n <= 1:
return False
for i in range(2, int(n**0.5) + 1):
if n % i == 0:
return False
return True

# Read the input
input_str = input()
month_str, year = input_str.split()
year = int(year)

# Convert month string to a month number
```

```
month = list(calendar.month_name).index(month_str)

# Get the number of days in the month
num_days = calendar.monthrange(year, month)[1]

# Count prime days
working_days = sum(1 for day in range(1, num_days + 1) if is_prime(day))

# Print the result
print(working_days)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

11

Compilation Status: Passed

Execution Time:

0.018s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

10

Compilation Status: Passed

Execution Time:

0.017s

24. You are given a number ‘n’. You have to tell whether a number is

great or not. A great number is a number whose sum of digits let (m) and product of digits let(j) when summed together gives the number back

m+j=n

Sample Input:

59

Sample Output:

Great

Completion Status: Completed

Concepts Included:

mathematics

numbers

Language Used: PYTHON 3

Source Code:

```
def is_great(n):
m = sum(int(digit) for digit in str(n))
j = 1
for digit in str(n):
j *= int(digit)
return m + j == n

n = int(input())
if is_great(n):
print("Great")
else:
print("no")
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Great

Compilation Status: Passed

Execution Time:

0.015s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.01s

25. Given a number N, print yes if the number is a multiple of 7 else print no.Sample Testcase :INPUT49OUTPUTyes

Completion Status: Completed

Concepts Included:

mathematics

Language Used: PYTHON 3

Source Code:

```
# Read the input number
N = int(input())

# Check if N is a multiple of 7
if N % 7 == 0:
print("yes")
```

```
else:
print("no")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.009s

26. You are given a large number made of only 0's and 1's.Your task is to find the max no of consecutive 1's. If there are no 1's print -1

Sample Input:

101011111

Sample Output:

5

Completion Status: Completed

Concepts Included:

mathematics

bit manipulation

binary

Language Used: PYTHON 3

Source Code:

```
# Read the input binary string
n = input()

# Split the string by '0' to get groups of consecutive '1's
consecutive_ones = n.split('0')

# Find the length of the longest group of '1's
max_consecutive_ones = max(len(group) for group in consecutive_ones)

# Check if there were any '1's in the original string
if max_consecutive_ones == 0:
print(-1)  # No '1's found
else:
print(max_consecutive_ones)  # Maximum count of consecutive '1's
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-1

Compilation Status: Passed

Execution Time:

0.01s

27. You are given a task to tell whether the number is pure or not. A pure number is a number whose sum of digits is multiple of 3.

O(1) time and O(1) space

Sample Input:

13

Sample Output:

not

Completion Status: Completed

Concepts Included:

mathematics

Language Used: PYTHON 3

Source Code:

```
# Read the input number as a string to easily iterate over digits
n = input().strip()

# Calculate the sum of digits
digit_sum = sum(int(digit) for digit in n)

# Check if the sum is a multiple of 3
if digit_sum % 3 == 0:
```

```
print("yes")
else:
print("not")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

not

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

not

Compilation Status: Passed

Execution Time:

0.009s

28. You are an employee of 'Rox Travel' channel.The channel has decided to give allowances to some customer who satisfy these conditions. The conditions are:

The customer should be born on or before july 22 1987

The month of D.O.B month should be of 31 days.

You are given with the D.O.B of all the employees.Your task is to print the employee index who are having chance to avail special offer.

Sample Input:

Input
4
23 MARCH 1996 23 MARCH 1986 22 JULY 1987 23 APRIL 1987

Sample Output:

2 3

Completion Status: Completed

Concepts Included:

mathematics

numbers

Language Used: PYTHON 3

Source Code:

```
from datetime import datetime

# Function to check if the date is on or before the specified date
def is_eligible(dob_str):
# Define the cutoff date
cutoff_date = datetime(1987, 7, 22)

# Convert the string to a datetime object
dob = datetime.strptime(dob_str, '%d %B %Y')

# Check if the birth date is on or before the cutoff date
if dob <= cutoff_date:
# Check if the month has 31 days
if dob.month in [1, 3, 5, 7, 8, 10, 12]:
return True
return False

# Read the number of employees
n = int(input())

# Read the D.O.B array as a single line, split by space
dob_array = input().strip().split(" ")

# Initialize a list to keep track of eligible employee indices
```

```
eligible_indices = []

# Loop through the D.O.B entries, 3 items at a time (day, month, year)
for i in range(0, len(dob_array), 3):
# Join the components into a single D.O.B string
dob_str = " ".join(dob_array[i:i+3])

# Check eligibility
if is_eligible(dob_str):
eligible_indices.append(i // 3 + 1)  # Use (i // 3 + 1) for 1-based index

# Print the eligible indices or -1 if none found
if eligible_indices:
print(" ".join(map(str, eligible_indices)))
else:
print(-1)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 3 4

Compilation Status: Passed

Execution Time:

0.018s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 3 4 5

Compilation Status: Passed

Execution Time:

0.018s

29. Assume your brother studies in class 2. He has to complete his homework on co-primes. As an elder sibling help him in finding whether the given two numbers is co-prime or not.

Sample Input:

3 5

Sample Output:

1

Completion Status: Completed

Concepts Included:

mathematics

Language Used: PYTHON 3

Source Code:

```
from math import gcd

# Read the two numbers
n, m = map(int, input().split())

# Check if the numbers are co-prime
if gcd(n, m) == 1:
print(1)  # They are co-prime
else:
print(0)  # They are not co-prime
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

30. You are provided with a number 'n'. Your task is to tell whether that number is saturated. A saturated number is a number which is made by exactly two digits.

Sample Input:

121

Sample Output:

Saturated

Completion Status: Completed

Concepts Included:

mathematics

numbers

Language Used: PYTHON 3

Source Code:

```
# Read the number as a string
n = input().strip()

# Use a set to find unique digits
unique_digits = set(n)

# Check if there are exactly two unique digits
if len(unique_digits) == 2:
print("Saturated")
else:
print("Unsaturated")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Saturated

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Saturated

Compilation Status: Passed

Execution Time:

0.01s

31. Write a program to print the sum of the first K natural numbers.Input Size : n <= 100000Sample Testcase :INPUT3OUTPUT6

Completion Status: Completed

Concepts Included:

basics

mathematics

Language Used: PYTHON 3

Source Code:

```
def sum_of_first_k_natural_numbers(k):
"""Calculates the sum of the first k natural numbers.

Args:
k: The number of natural numbers to sum.

Returns:
The sum of the first k natural numbers.
"""

sum = 0
for i in range(1, k + 1):
sum += i
return sum

# Get input from the user
k = int(input())

# Calculate and print the sum
result = sum_of_first_k_natural_numbers(k)
print(result)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

210

Compilation Status: Passed

Execution Time:

0.01s

32. Using the method of looping, write a program to print the table of 9 till N in the format as follows: (N is input by the user)

9 18 27...

Print NULL if 0 is input

Sample Input:

3

Sample Output:

9 18 27

Completion Status: Not Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```
# Input: Get the value of N from the user
N = int(input("Enter a positive integer: "))

# Check if N is 0 and print "NULL"
if N == 0:
print("NULL")
else:
# Loop to print the table of 9 up to 9 * N
for i in range(1, N + 1):
print(9 * i, end=" ")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Enter a positive integer: 9 18 27

Compilation Status: Failed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Enter a positive integer: 9

Compilation Status: Failed

Execution Time:

0.01s

33. You are given three numbers A, B & C. Print the largest among these three numbers.

Sample Input:

1
2
3

Sample Output:

3

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```
A = int(input())
B = int(input())
C = int(input())

largest = max(A, B, C)
print(largest)
```


Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.009s