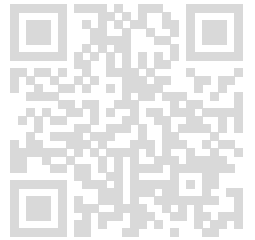# Codekata Report:

Name: Meena J

Email: meenait16bsc@gmail.com

1. You are given A = Length of a rectangle & B = breadth of a rectangle. Find its area "C".

(A and B are natural numbers)

Sample Input:

2
3

Sample Output:

6

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```
A = int(input())
B = int(input())
area = A * B
print(round(area,1))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

144

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

30

Compilation Status: Passed

Execution Time:

0.009s

2. You are provided with a number check whether its odd or even.

Print "Odd" or "Even" for the corresponding cases.

Note: In case of a decimal, Round off to nearest integer and then find the output. Incase the input is zero, print "Zero".
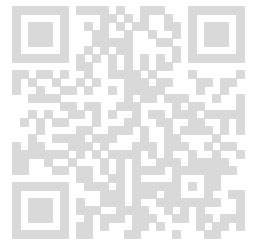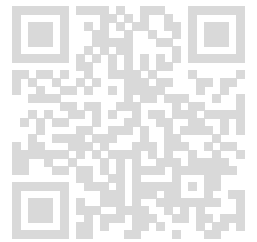
Sample Input:

2

Sample Output:

Even

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```python
N=int (input ())
if N==0:
print("Zero")
elif N%2==0:
print("Even")
else:
print("Odd")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Even

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Odd

Compilation Status: Passed

Execution Time:

0.01s

3. You are given three numbers A, B & C. Print the largest amongst these three numbers.
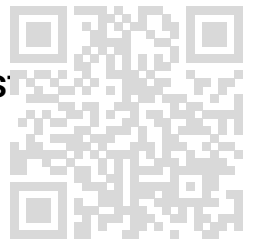
Sample Input:

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())
print(max(a,b,c))
```

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

4. 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: Completed

Concepts Included:

absolute beginner
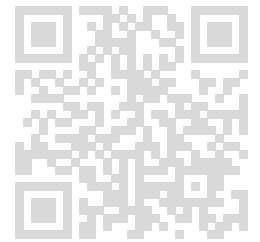
Language Used: PYTHON 3

Source Code:

```
n=int(input())

if (n>=1):

i=0
```

```
for i in range(1,n):

m=i*9

print(m,end=' ')

print((i+1)*9)

else:

print("NULL")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

9 18 27

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

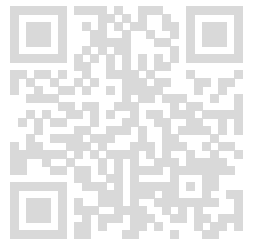**Output:**

9

Compilation Status: Passed

Execution Time:

0.01s
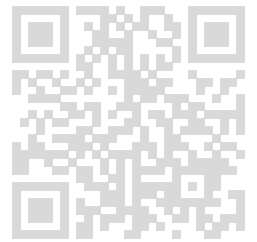
5. You are provided with two numbers. Find and print the smaller number.

Sample Input:

23 1

Sample Output:

1

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```
A,B = map(int,input().split())
print(min(A,B))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

32

Compilation Status: Passed

Execution Time:

0.009s

6. You will be provided with a number. Print the number of days in the month corresponding to that number.

Note: In case the input is February, print 28 days. If the Input is not in valid range print "Error".

Sample Input:

8

Sample Output:

31

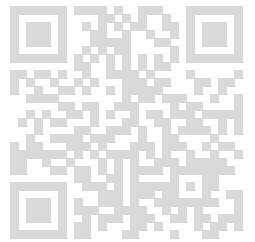**Completion Status:** Completed

Concepts Included:

absolute beginner
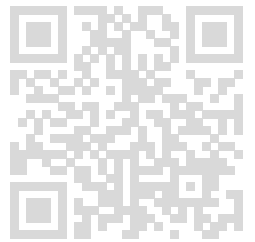
Language Used: PYTHON 3

Source Code:

```python
month = int(input())

if(month==2):
print(28)
elif(month==4|month==6|month==9|month==11):
print(30)
elif(month==1|month==3|month==5|month==7|month==8|month==10|month==12):
print(31)
else:
print('Error')
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Error

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Error

Compilation Status: Passed

Execution Time:

0.01s

7. Write a code to get an integer N and print the values from N to 1.
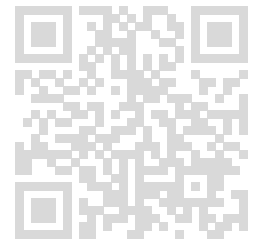
Sample Input:

10

Sample Output:

10
9
8
7

6
5
4
3
2
1

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```
N=int(input())

for i in range(N,0,-1):
print(i)
```
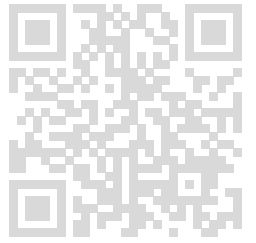
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

100
99
98
97
96
95
94
93
92
91
90
89
88

87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34

```
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
```

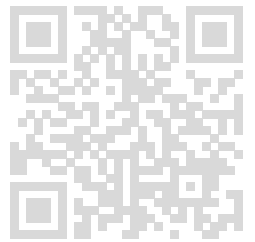**Compilation Status:** Passed

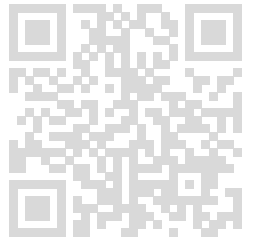**Execution Time:**

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
5
4
3
```

2
1

Compilation Status: Passed

Execution Time:

0.01s

8. Let "A" be a year, write a program to check whether this year is a leap year or not.

Print "Y" if its a leap year and "N" if its a common year.

Sample Input:

2020

Sample Output:

Y

Completion Status: Completed

Concepts Included:

absolute beginner

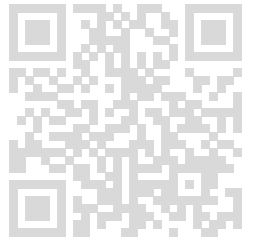**Language Used:** PYTHON 3

Source Code:

```
year = int(input())

if (year % 4) == 0:
if (year % 100) == 0:
if (year % 400) == 0:
print("Y")
else:
print("N")
else:
print("Y")
else:
print("N")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

N

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Y

Compilation Status: Passed

Execution Time:

0.01s

9. Write a code to get an integer N and print the sum of values from 1 to N.

Sample Input:

10

Sample Output:
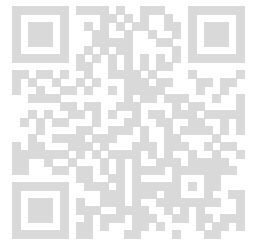
55

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```python
a = int(input())

total = a * (a + 1) // 2

print(total)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

5050

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1225

Compilation Status: Passed

Execution Time:

10. You are given with Principle amount($), Interest Rate(%) and Time (years) in that order. Find Simple Interest.

Print the output up to two decimal places (Round-off if necessary).

(S.I. = P*T*R/100)

Sample Input:

1000 2 5

Sample Output:

100.00

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```
p,t,r = map(float,input().split(" "))

si=(p*t*r)/100

simple = "{:.2f}".format(si)

print(simple)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

100.00

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

112.20

Compilation Status: Passed

Execution Time:

0.01s

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

Sample Input:

6

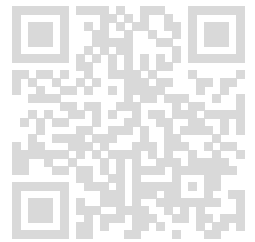**Sample Output:**

2
4
6

Completion Status: Completed
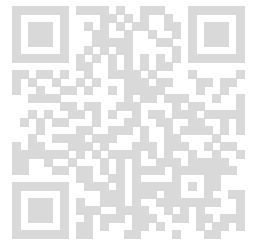
Concepts Included:

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:
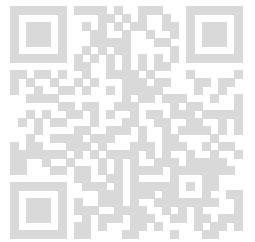
TestCase1:

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:

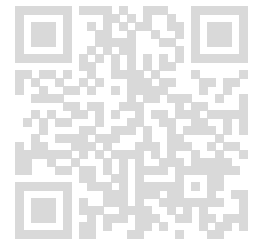**Input:**

< 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

12. You are given a number A in Kilometers. Convert this into B: Meters and C: Centi-Metres.

Sample Input:

2

Sample Output:

2000
200000

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

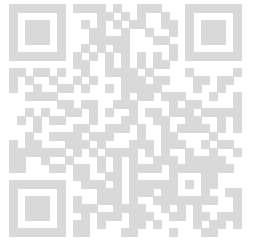**Source Code:**

```python
A =int(input())

meter=1000*A

print(meter)

centi_meter=100000*A

print(centi_meter)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2000
200000

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4000
400000

Compilation Status: Passed

Execution Time:

0.011s

13. Write a code to get 2 integers as input and find the HCF of the 2 integer without using recursion or Euclidean algorithm.
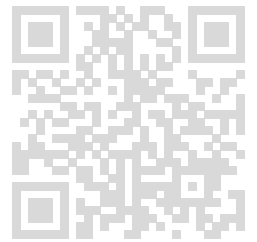
Sample Input:

2 3

Sample Output:

1

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```python
import math
x,y = map(int,input().split(" "))
print(math.gcd(x,y))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

19

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:
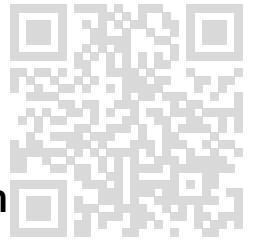
Input:

< hidden >

Expected Output:

< hidden >

Output:

15

Compilation Status: Passed

Execution Time:

14. Write a code get an integer number as input and print the sum of the digits.

Sample Input:

124

Sample Output:

7

Completion Status: Completed

Concepts Included:

absolute beginner

basics

Looping

Language Used: PYTHON 3

Source Code:

```
n = int(input())
r = 0
while n>0:
d = n%10
r = r+d
n = n//10
print(r)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

45

Compilation Status: Passed

Execution Time:
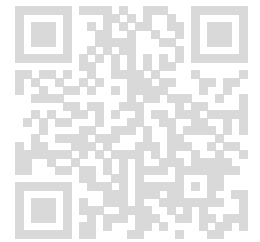
0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

49

Compilation Status: Passed

Execution Time:

0.01s

15. Given a string 'S' print the sum of weight of the String. A weight of character is defined as the ASCII value of corresponding character.

Sample Input:

abc

Sample Output:
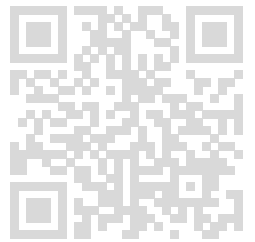
294

Completion Status: Completed

Concepts Included:

strings

Language Used: PYTHON 3

Source Code:

```python
s = input()
t = list(map(ord,s))
print(sum(t))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

294

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

201

Compilation Status: Passed

Execution Time:

0.01s

16. You are given two numbers. Your task is to multiply the two numbers and print the answer.

Sample Input:

99999 99998

Sample Output:

9999700002

Completion Status: Completed

Concepts Included:

strings

Language Used: PYTHON 3

Source Code:

```python
n=list(map(str,input().split()))
print(int(n[0])*int(n[1]))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

9999300006

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

9999700002

Compilation Status: Passed

Execution Time:
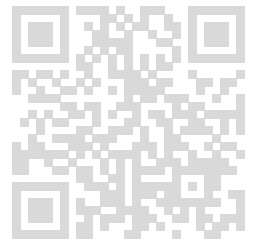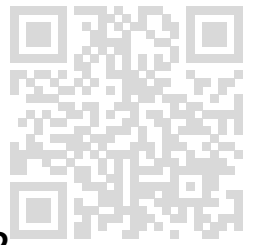
17. Print the position of first 1 from right to left, in binary representation of an Integer.Sample Testcase :INPUT18OUTPUT2

Completion Status: Completed

Concepts Included:

array

strings

mathematics

bitwise

Language Used: PYTHON 3

Source Code:

```
import math
n=int(input())
res=math.log2(n & -n)+1
print(round(res))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.011s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3

Compilation Status: Passed

Execution Time:

0.01s

18. You are given an array. Your task is to sort the array in given manner. Print the elements in increasing order of the frequency. If frequency is same print smaller one first.

Sample Input:

4
1 1 3 2

Sample Output:

2 3 1

Completion Status: Completed
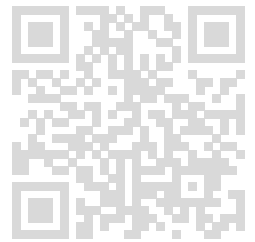
Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```
#taking number of elements as input
n = int(input())
# taking the list of number as input
arr= list(map(int,input().split()))

#importing the library for getting the counts of the elements
from collections import Counter

def frequencySort(nums):
cn = Counter(nums)
#alternatively this can be done using dict also
# dict1 = {}
```

```python
# for i in nums:
#     dict1[i]=nums.count(i)
# for k, v in sorted(dict1.items(), key = lambda kv: (kv[1],kv[0])):
#     out.append(v*[k])

#making a 2-dimensional array to store the values
out = []
for k, v in sorted(cn.items(), key = lambda kv: (kv[1],kv[0])):
out.append(v*[k])

# this gives the one-dimensional array
o = []
for i in out:
o+= i

#finally getting all the unique values from the array
final = []
for i in o:
if i not in final:
final.append(i)
return final

print(*frequencySort(arr))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:
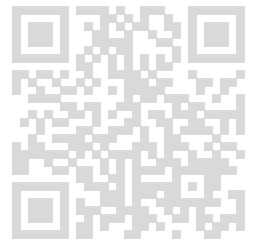
2 3 1

Compilation Status: Passed

Execution Time:
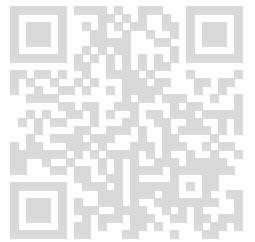
0.012s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3 13 1

Compilation Status: Passed

Execution Time:

0.011s

19. Given a string S, print it after changing the middle element to * (if the length of the string is even, change the 2 middle elements to *).Sample Testcase :INPUThelloOUTPUThe*lo

Completion Status: Completed

Concepts Included:

array

strings

Language Used: PYTHON 3

Source Code:

```
S = input().strip()
D = len(S)//2

if len(S)%2==0:
print(S[0:D-1]+"**"+S[D+1:])
else:
print(S[0:D]+"*"+S[D+1:])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

*

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

sa**ad

Compilation Status: Passed

Execution Time:

0.01s

20. Given a number N followed by N elements, find the second smallest element.If it cannot be found then print -1Input Size : N <= 100000 (ie do it in O(log n) time complexity)Sample Testcase :INPUT51 2 3 4 5OUTPUT2

Completion Status: Completed

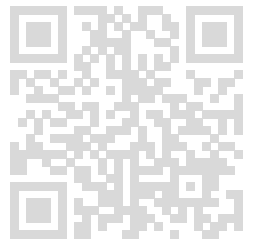**Concepts Included:**

array

Language Used: PYTHON 3

Source Code:

```
N=int(input())
no=list(map(int,input().split()))
no.sort()
if(no[0]==no[1]):
print(-1)

else:
print(no[1])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-5

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-1

Compilation Status: Passed
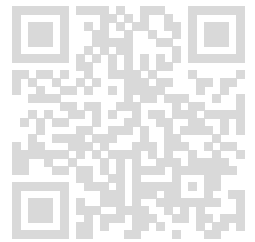
Execution Time:

0.009s

21. Prateek finds it difficult to judge the minimum element in the list of elements given to him. Your task is to develop the algorithm in order to find the minimum element.
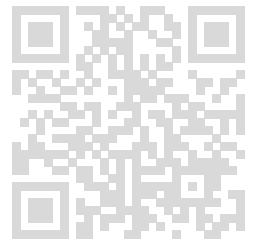
Sample Input:

5
3 4 9 1 6

Sample Output:

1

Completion Status: Completed

Concepts Included:

array

numbers

mathematics

Language Used: PYTHON 3

Source Code:

```
N=int(input())
l=list(map(int,input().split()))
l.sort()
print(l[0])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.009s
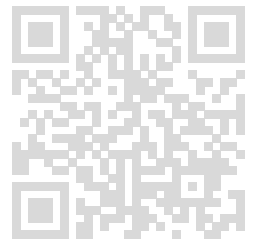
22. Ram is the CEO of an MNC. He wants to order the employee salaries in ascending order so that he can do a salary hike based on the salary values of employees. He selects you to do the task of sorting the salaries. Sort the salaries in ascending order and pass on the information to Ram.

Sample Input:

8
7000 8000 6500 1200 4000 2800 3000 5230

Sample Output:

1200 2800 3000 4000 5230 6500 7000 8000

Completion Status: Completed

Concepts Included:

sorting

array

Language Used: PYTHON 3

Source Code:

```
employee=input()
No=list(map(int,input().split()))
No.sort()
print(*No)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 2 3 4 5

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 8 9

Compilation Status: Passed

Execution Time:

0.01s

23. Given 2 numbers N and K followed by N elements,print the number of repetition of K otherwise print '-1' if the element not found.Sample Testcase :INPUT6 21 2 3 5 7 8OUTPUT0

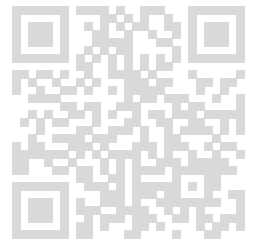**Completion Status:** Completed

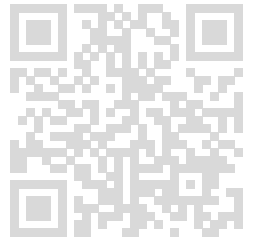**Concepts Included:**

basics

mathematics

array

Language Used: PYTHON 3

Source Code:

```
n,m=map(int,input().split())
b=map(int,input().split()[:n])
b=list(b)
```

```
count=-1
for i in b:
if(i==m):
count+=1
print(count)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:
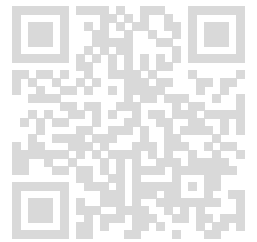
< hidden >

Expected Output:

< hidden >

Output:

-1

Compilation Status: Passed

Execution Time:

0.009s

24. Ria is always fascinated by the number 2. She always wants to know who came second in a race, the second person to set foot on the moon and so on. She is given a list of numbers and asked to find the maximum. As always, she reports the second highest number as the maximum because according to her, 2 is higher than

1. Find out which was the number that Ria would have reported, given a list of N numbers.

Sample Input:

10
1 9 8 7 6 5 2 3 4 10

Sample Output:

9

Completion Status: Completed

Concepts Included:

searching

array

Language Used: PYTHON 3

Source Code:

```
size=int(input())
elem=list(map(int,input().split()))
elem.sort()
print(elem[-2])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

76

Compilation Status: Passed

Execution Time:

0.01s

25. Given a sentence and string S, find how many times S occurs in the given sentence.If S is not found in the sentence print -1Input Size : |sentence| <= 1000000(complexity O(n)).Sample Testcase :INPUTI enjoy doing codekatacodekataOUTPUT1

Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

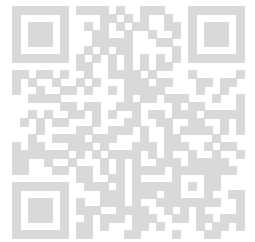**Source Code:**

```
S=input().split()
K=input()
res=S.count(K)
if (res>=1):
print(res)
else:
print("-1")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-1

Compilation Status: Passed

Execution Time:

0.01s

26. Given a number N print a right angled traingle structure with the starting level as single 1 and every immediate proceeding level with 2 more additional ones than the previous level .Repeat the pattern for N levels.Input Size : N <= 1000Sample Testcase :INPUT3OUTPUT11 1 11 1 1 1 1
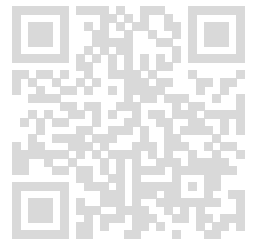
Completion Status: Completed
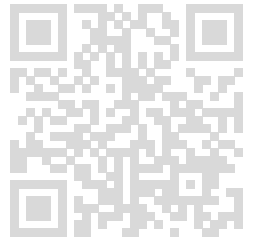
Concepts Included:

trees and graphs

array

companies

Language Used: PYTHON 3

Source Code:

```
N=int(input())
k='1'
for i in range(N):
print(k)
k=k+' 1 1'
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
1 1 1
1 1 1 1 1
1 1 1 1 1 1 1
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
1 1 1
1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
```

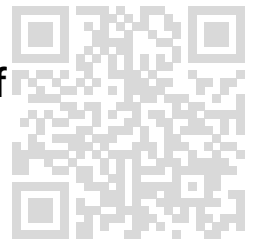**Compilation Status:** Passed

Execution Time:

0.01s

27. Pk finds it difficult to judge the minimum element in the list of elements given to him. Your task is to develop the algorithm in order to find the minimum element.

Note:Don't use sorting

Sample Input:

5
3 4 9 1 6

Sample Output:

1

Completion Status: Completed

Concepts Included:

mathematics

array

Language Used: PYTHON 3

Source Code:

```
N=int(input())
l=list(map(int,input().split()))
l.sort()
print(l[0])
```

Compilation Details:
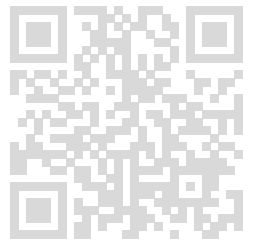
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.01s

28. Mr.Stark wants to order the employee ids, which are recorded in a 2D matrix, in ascending order. He wants to do it so as to allot a new id to a person who joins as a fresher. You are the CTO of the Stark industries and you are asked by Mr.Stark to sort the data.

Sample Input:

3 3
87 21 34
89 32 78
12 23 45
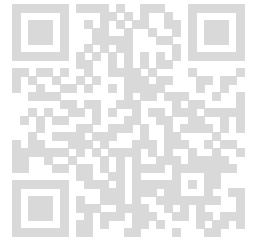
Sample Output:

12 21 23
32 34 45
78 87 89

Completion Status: Completed

Concepts Included:

sorting

array

Language Used: PYTHON 3

Source Code:

```python
n,m = map(int,input().split())
matrix=[]
for i in range(n):
row = list(map(int,input().split()))
matrix.append(row)

temp=[0]*(n*m)
k=0
for i in range(n):
for j in range(m):
temp[k]=matrix[i][j]
k+=1
temp.sort()
k=0
for i in range(n):
for j in range(m):
matrix[i][j]=temp[k]
k+=1

for row in matrix:
print(*row)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 2 3
7 8 9

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

23 34 54
56 64 89

Compilation Status: Passed

Execution Time:

0.01s

29. You are an intern at GUVI and the company wants to organise its data and delete unnecessary extra storage elements used. You are given k arrays of unequal dimensions. Sort the k arrays individually and concatenate them.

Sample Input:

3
2
98 12
6
1 2 3 8 5 9
1
11

Sample Output:

12 98 1 2 3 5 8 9 11

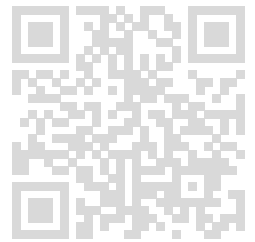**Completion Status:** Completed
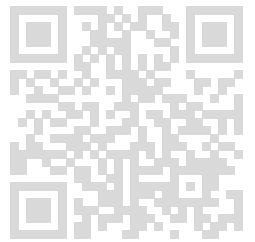
Concepts Included:

sorting

array

Language Used: PYTHON 3

Source Code:

```
a=int(input())
b=[]
for i in range(a):
```

```
x=int(input())
y=list(map(int,input().split()))
y.sort()
for j in range(len(y)):
b.append(y[j])
print(*b)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 2 3 10 12 43 66 76

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 45 67 9 12 56

Compilation Status: Passed
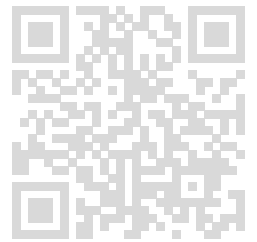
Execution Time:

0.01s

30. 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:

```
l = list(map(int,input().split()))
l.sort()
print(l[0])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

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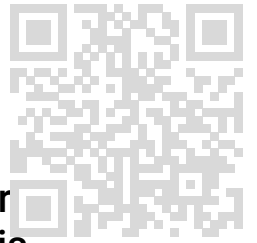
31. You are provided with a number 'n'. Your task is to tell whether that number is saturated. A saturated number is a number which is made by exactly two digits.

Sample Input:

121

Sample Output:

Saturated

Completion Status: Completed

Concepts Included:

mathematics

numbers

Language Used: PYTHON 3

Source Code:

```
inputt = input()
sett = list(set(inputt))

if(len(sett) == 2):
print("Saturated")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Saturated

Compilation Status: Passed

Execution Time:

0.01s
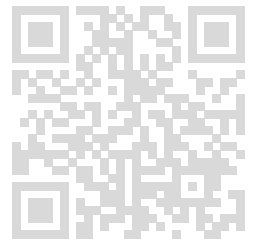
TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Saturated

Compilation Status: Passed

Execution Time:

0.009s

32. 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: Completed
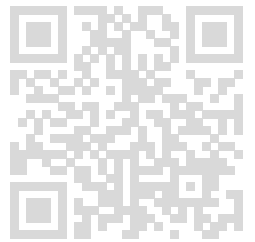
Concepts Included:

mathematics

array

Language Used: PYTHON 3

Source Code:

```
n=int(input())
def fib(n):
if n==1:
return 2000 # (0+1000)+1000
elif n==2:
return 4000 # (1000 + 2000)+1000
else:
return fib(n-1)+fib(n-2)+1000 # He saves same amount of money = the money saved
in immediate previous two months +1000
print(fib(n))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4000

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

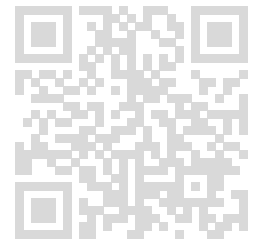< hidden >

Output:

7000

Compilation Status: Passed

Execution Time:

0.014s

33. Given 2 numbers N,M. Print 'yes' if their product is a perfect

square else print 'no'.Sample Testcase :INPUT5 5OUTPUTyes

Completion Status: Completed

Concepts Included:

mathematics

basics

Language Used: PYTHON 3

Source Code:

```python
N,M=map(int,input().split())
product=N*M
if product**0.5==N:
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 >
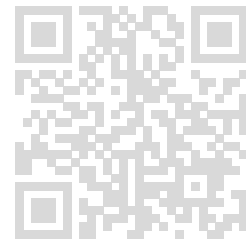
Output:

no

Compilation Status: Passed

Execution Time:

0.009s

34. 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:

```python
ui = input()
n = len(ui)

count = 0
result = 0
for i in range(0, n):
if ui[i] == '0':
count = 0
else:
count += 1
result = max(result, count)

print(result) if count > 0 else print(-1)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-1

Compilation Status: Passed

Execution Time:

0.01s

35. Write a code get an integer number as input and print the odd and even digits of the number separately.
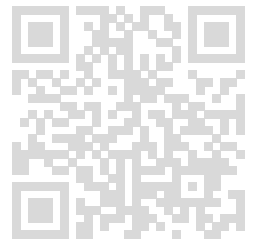
Sample Input:

1234

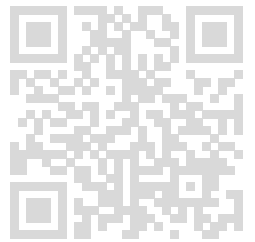**Sample Output:**

2 4
1 3

Completion Status: Completed

Concepts Included:

basics

absolute beginner

Looping

Language Used: PYTHON 3

Source Code:

```
a=int(input())
ev=[]
od=[]
b=list(map(int, str(a)))
for x in b:
if (x%2 ==0):
ev.append(x)
else:
od.append(x)
print(*sorted(ev))
print(*sorted(od))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 2 4
3 3 3

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 2 2 4
3 3 5 5 5

Compilation Status: Passed

Execution Time:

0.01s

36. Write a code to get 2 integers as input and add the integers without any carry.

Sample Input:

44 66

Sample Output:

0

Completion Status: Completed

Concepts Included:

basics

Looping

Language Used: PYTHON 3

Source Code:

```
a, b = list(map(int, input().split()))

c = a + b

d = str(c)
A = str(a)
B = str(b)

if c < 110 and ( len(d) > len(A) or len(d) > len(B)) :
print(d[-1])

elif c > 110 :
print((c - 100)/2)

else:
```
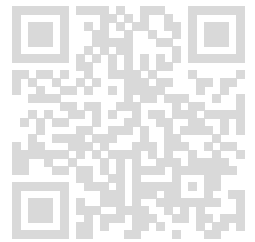
print(c)

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

9

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

10.0

Compilation Status: Passed

Execution Time:

0.009s

37. Given a string S consisting of 2 words reverse the order of two words .Input Size : |S| <= 10000000Sample Testcase :INPUThello world OUTPUTworld hello
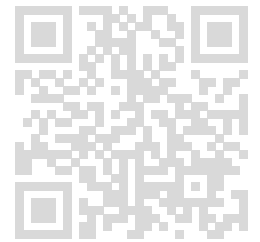
Completion Status: Completed

Concepts Included:

strings

basics

companies

Language Used: PYTHON 3

Source Code:

```
s=input().split()
a = s[::-1]
print(*a)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

world hello

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

a h

Compilation Status: Passed

Execution Time:

0.009s

38. Given a number N followed by N numbers. Find the smallest number and largest number and print both the indices(1 based indexing). Input Size : N <= 100000 Sample Testcase : INPUT 5 1 2 3 5 OUTPUT 1 5
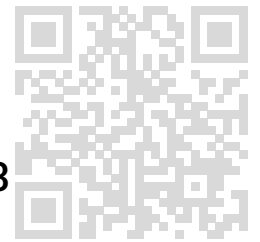
Completion Status: Completed

Concepts Included:

array

basics

Language Used: PYTHON 3

Source Code:

```
n=int(input())
N=input().split()
n=len(N)
A=max(N)
B=min(N)

print((N.index(B)+1),(N.index(A)+1))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 5

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4 1

Compilation Status: Passed

Execution Time:

0.009s

39. Let P represent Paper, R represent Rock and S represent Scissors. Given 2 out of the 3 determine which one wins. If its a draw print 'D'.Sample Testcase :INPUTR POUTPUTP

Completion Status: Completed

Concepts Included:

strings

basics

Language Used: PYTHON 3

Source Code:

```
N,M=input().split()
if((N=='P' and M =='R') or (N=='R' and M=='P')):
print("P")
elif((N=='S' and M=='P') or (N=='P' and M=='S')):
print("S")
elif((N=='R' and M=='S') or (N=='S' and M=='R')):
print("R")
else:
print("D")
```
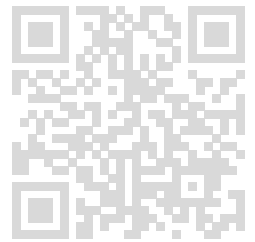
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

D

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

D

Compilation Status: Passed

Execution Time:

0.009s

40. Given 3 numbers A,B,C process and print 'yes' if they can form the sides of a triangle otherwise print 'no'.Input Size : A,B,C <= 100000Sample Testcase :INPUT3 4 5OUTPUTyes

Completion Status: Completed
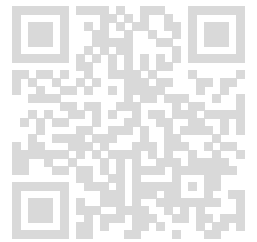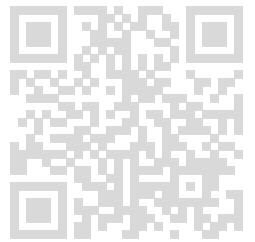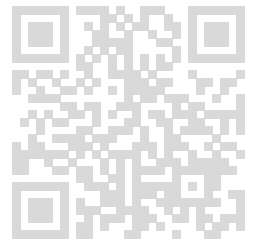
Concepts Included:

mathematics

basics

Language Used: PYTHON 3

Source Code:

```
a, b, c=input().split(" ")
a=int(a)
b=int(b)
c=int(c)
if c>a and c>b:
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

41. Iron Man wants to extract an infinity stone from a safe. The safe is protected by a password and Iron Man knows the clue to the password which is "sum one and two when sorted they are true". Decode the clue from the test case and help Iron Man open the safe.

Sample Input:

5
9 8 3 2 1

Sample Output:

3

Completion Status: Completed

Concepts Included:

sorting

array

Language Used: PYTHON 3

Source Code:

```
a = input()

b = input().split()


x = sorted([int(X) for X in b ])
# this used to convert array of conerted into list and sorted
print(x[0] + x[1])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

21

Compilation Status: Passed

Execution Time:

0.009s

42. You are given an array of non-negative integers representing height of walls at index i as Ai and the width of each block is 1. Compute how much air can be encapsulated between the walls of chamber.

Sample Input:

3
7 4 9

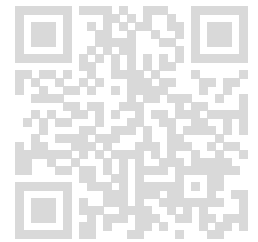**Sample Output:**

3

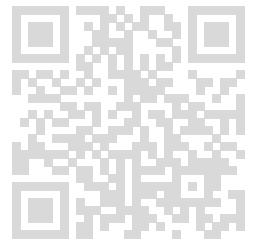**Completion Status:** Completed

Concepts Included:

array

mathematics

Language Used: PYTHON 3

Source Code:

```
n =int(input())
l = [int(x) for x in input().split()]
ans = 0
```

```
for i in range(1, len(l)-1):
if l[i] < l[i-1] and l[i] < l[i+1]:
ans += min(l[i+1], l[i-1]) - l[i]
print(ans)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:
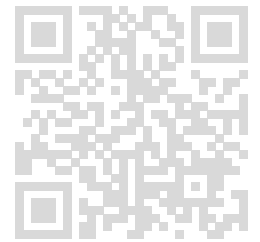
< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.009s

43. You are given with two arrays. Your task is to merge the array such that first array is in ascending order and second one in descending order.

Sample Input:

```
3 3
23 15 16
357 65 10
```

Sample Output:

15 16 23 357 65 10

Completion Status: Completed

Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```
n, m = list(map(int,input().split()))
N = list(map(int,input().split()))
M = list(map(int,input().split()))

x=sorted(N)
y=sorted(M,reverse=True)

L = x + y
print(*L)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

15 16 23 357 65 10

Compilation Status: Passed

Execution Time:

0.013s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 12 13 14 16 19 42 58 65 98 56 55 54 46 32 17 16

Compilation Status: Passed

Execution Time:

0.009s

44. You are given an array of numbers. Print the least occurring element. If there is more than 1 element print all of them in decreasing order of their value.

Sample Input:

```
9
1 6 4 56 56 56 6 4 2
```

Sample Output:
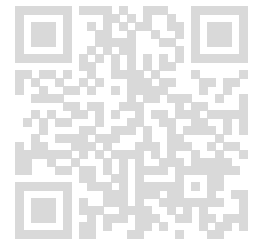
2 1

Completion Status: Completed

Concepts Included:

mathematics

array

Language Used: PYTHON 3

Source Code:

```
n = int(input())
arr = list(map(int, input().split()))

# count the occurrences of each number
freq = {}
for num in arr:
freq[num] = freq.get(num, 0) + 1
```
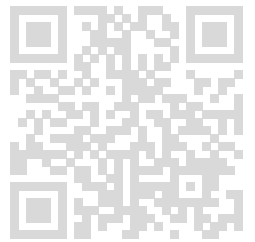
```python
# find the least occurring number
least_freq = min(freq.values())

# collect all numbers with least frequency
result = []
for num, count in freq.items():
if count == least_freq:
result.append(num)

# sort the result in decreasing order
result.sort(reverse=True)

# print the result
print(*result)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

76 25

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 0

Compilation Status: Passed

Execution Time:
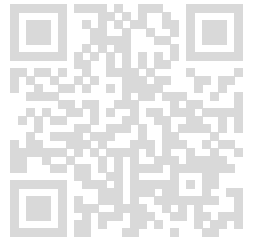
45. Write a code to get the input in the given format and print the output in the given format.

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:

```python
a = input()

b = input()

print(a)

print(b)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:
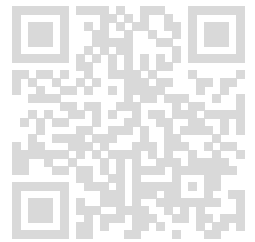
< hidden >

Output:

5 3
1 2 3 4 5

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4 2
1 4 3 2

Compilation Status: Passed

Execution Time:

0.01s

46. 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:

```
a = list(input().split())

b = list(input().split())

c = list(input().split())


print(*a)

print(*b)

print(*c,)
```

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
```
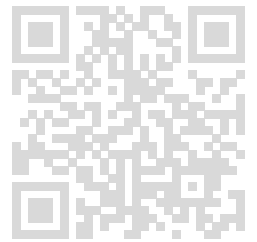
Compilation Status: Passed
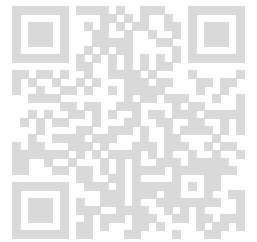
Execution Time:

0.009s

47. 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:

a = list(input().split())

b = list(input().split())

c = list(input().split())

print(*a)

print(*b)

print(*c,)

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:

0.009s

48. Loki wants to steal the tesseract but in order to do so, he has to rearrange the elements in an array in a specific manner which is mentioned in a clue. The clue says 'cursed are the odd and sorted are the even'. Loki manages to decode the clue which translates to "sort the even positioned elements of an array, starting from the element at index 0, in ascending order". Manipulate the array so as to help Loki steal the tesseract.

Sample Input:

5
3 9 1 44 6
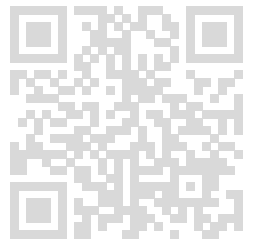
Sample Output:

1 9 3 44 6

Completion Status: Completed

Concepts Included:

sorting

array

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
arr=list(map(int,input().split()))
odd=[]
even=[]
res=[]
for i in range(n):
if i%2==0 or i==0:
even.append(arr[i])
#print(even)
else:
odd.append(arr[i])
#print(odd)
even.sort()
#print(even)

for i in range(len(even)):
res.append(even[i])
if i <len(odd):
res.append(odd[i])
print(*res)
```
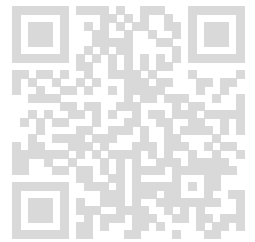
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 6 3 4 5 2 7

Compilation Status: Passed

Execution Time:
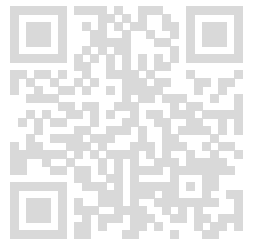
0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

23 9 39 5 45 47

Compilation Status: Passed

Execution Time:

0.01s

49. You are given given task is to print whether array is 'majestic' or not.A 'majsetic' array is an array whose sum of first three number is equal to last three number.

Sample Input:

7
1 2 3 4 6 0 0

Sample Output:

1

Completion Status: Completed

Concepts Included:

mathematics

array

Amazon

Facebook

United-Health-Group

guvi-learning-path

Language Used: PYTHON 3

Source Code:

```
n = int(input())
m = list(map(int,input().split()))
a = sum(m[0:3])
b = sum(m[-3:])
if a == b:
print(1)
else:
print(0)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:
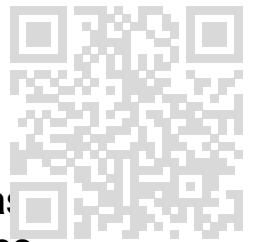
50. Your old mobile phone gets broken and so you want to purchas **a new smartphone and decide to go through all the online websites** **to find out which dealer has the best offer for a particular model.** **You document the prices of N dealers. Dealer ids start from 0 and** **go up to N. Find out which dealer has the best price for you.**

Constraints:

1 <= N <= 100

1 <= A[] <= 100000

Sample Input:

3
10000 11200 12030

Sample Output:

Dealer0

Completion Status: Completed

Concepts Included:

searching

array

Language Used: PYTHON 3

Source Code:

```
n=int(input())
price=list(map(int,input().split()))
best_p=float("inf")
id=0
for i in range(n):
if price[i]<best_p:
best_p=price[i]
id=i
```

```
print("Dealer"+str(id))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Dealer9

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

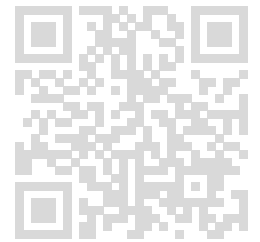**Expected Output:**

< hidden >

Output:

Dealer0

Compilation Status: Passed

Execution Time:

0.01s

51. You are provided with an array in which all elements are repeated thrice except one which is repeated twice.Your task is to print that number.

O(n) time and O(1) extra space

Sample Input:

5
13 12 13 12 13

Sample Output:

12

Completion Status: Completed

Concepts Included:

array

hashing

Language Used: PYTHON 3

Source Code:

```
n=int(input())
arr=list(map(int,input().split()))
c=[]
for i in arr:
if arr.count(i)==2:
if i not in c:
c.append(i)
if len(c)>=1:
print(*c)
```

Compilation Details:

TestCase1:

Input:

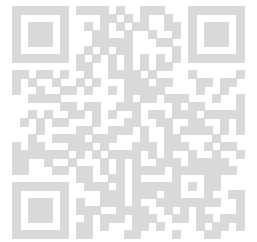< hidden >

Expected Output:

< hidden >

Output:

56

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2

Compilation Status: Passed

Execution Time:

0.01s

52. 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

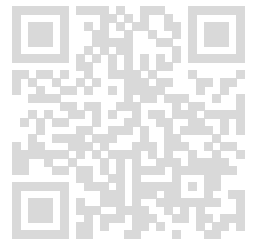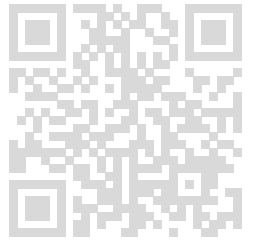**Completion Status:** Completed

Concepts Included:

mathematics

Language Used: PYTHON 3

Source Code:

```
from fractions import gcd
n,m=(int(no) for no in input().split())
if gcd(n,m)==1:
print("1")
else:
print("0")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.02s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

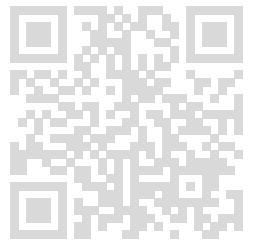**Compilation Status:** Passed

Execution Time:

0.019s

53. In XYZ country there is rule that car's engine no. depends upon car' number plate. Engine no is sum of all the integers present on car's Number plate.The issuing authority has hired you in order to provide engine no. to the cars.Your task is to develop an algorithm which takes input as in form of string(Number plate) and gives back

Engine number.

Sample Input:

HR05-AA-2669

Sample Output:

28

Completion Status: Completed

Concepts Included:

mathematics

strings

Language Used: PYTHON 3

Source Code:

```python
import re
x=input()
q=[]
y=re.split("\D+",x)
for i in range(len(y)):

z=list(map(int,("".join(y[i]))))
q.append(sum(z))
w=sum(q)
print(w)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

28

Compilation Status: Passed

Execution Time:

0.015s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

22

Compilation Status: Passed

Execution Time:

0.015s
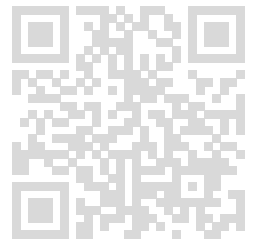
54. You are given with a string which comprises of some numbers. Your task is to find the largest integer by converting the string to the corresponding integer.

Sample Input:

I was born on 12 october 1998.

Sample Output:

1998

Completion Status: Completed

Concepts Included:

mathematics

strings

integer

Language Used: PYTHON 3

Source Code:

```
m = input()

res = ""

a = len(m)

for i in range(0,a):

if m[i]!='.':

res = res+m[i]
```
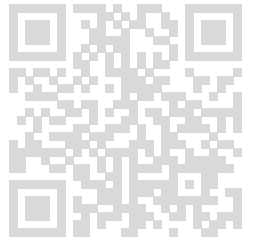
```
n = res.split(" ")

int_lst=[]

for i in n:

if i.isdigit()==True:

int_lst.append(i)

length = len(int_lst)

max_value = 0

for i in range(0,length):

if(int(max_value) < int(int_lst[i])):

max_value = int_lst[i]

print(max_value)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1947

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

28

Compilation Status: Passed

Execution Time:

0.01s
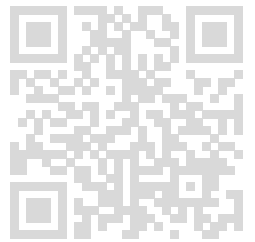
55. Indian PAN card issuing authority have found some fake PAN cards. They have hired you so that you can validate PAN card for them. Your task is to develop a suitable algorithm which could check if pan is valid or not

1)Pan must have uppercase letters only.

2)It must be of 10 character only

3)From index 1 to 5 all must be letters(A-Z),last index must be letter

4)Rest all must be integer Starting from 1

Sample Input:

HXTPS2142R

Sample Output:

pan

Completion Status: Completed

Concepts Included:

strings

Language Used: PYTHON 3

Source Code:

```
n=input()
s=False

if len(n)==10 and n.upper():
if n[9] not in range(0,9):
s=True
if n[0:5].isalnum():
```

```
s=False
if '0' in n:
s=False
if n[5:9].isnumeric():
s=True

if s:
print("pan")
else:
print("not pan")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

pan

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

not pan

Compilation Status: Passed

Execution Time:

0.009s
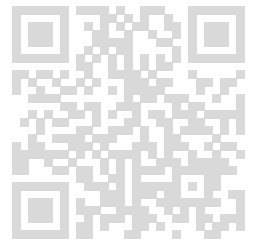
56. Guvi developed a new system to make sure no two usernames

are same. So, they hired you as a developer to develop this system. They have set some rules to do the same. If you see the same username that already exists, just add a number at the end of that username ,else print "Verified".
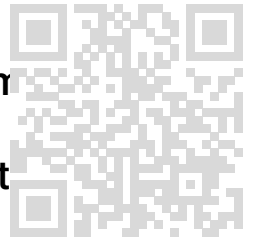
Sample Input:

4
abc aab abc aba

Sample Output:

Verified Verified abc1 aba

Completion Status: Completed

Concepts Included:

strings

Language Used: PYTHON 3

Source Code:

```python
a=int(input())
li=list(input().split(" "))
check=[]
output=[]
for i in li:
if i not in check:
check.append(i)
output.append("Verified")
else:
output.append(i+"1")
print(" ".join(output))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Verified Verified abc1 Verified

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Verified Verified abc1 Verified

Compilation Status: Passed

Execution Time:

0.009s

57. Write a program to get a string as input and reverse the string without using temporary variable.

Sample Input:

GUVI

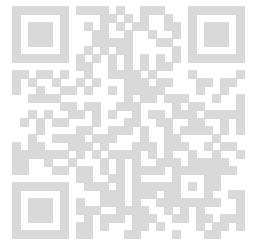**Sample Output:**

IVUG

Completion Status: Completed

Concepts Included:

absolute beginner

basics

bit manipulation

Looping

Language Used: PYTHON 3

Source Code:

```
str=input()
rev="".join(reversed(str))
print(rev)
```
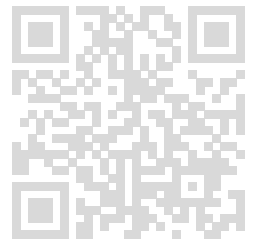
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

elgooG

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

koobecaf

Compilation Status: Passed

Execution Time:

0.009s

58. Let "A" be a string. Remove all the whitespaces and find it's length.

Sample Input:

Lorem Ipsum

Sample Output:

10

Completion Status: Completed

Concepts Included:

absolute beginner

Language Used: PYTHON 3

Source Code:

```python
A=str(input().replace(" ", ""))
print (len(A))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

10

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4
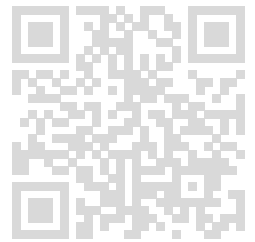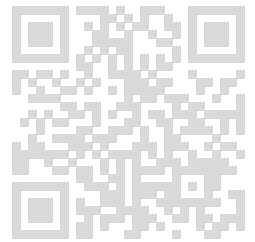
Compilation Status: Passed

Execution Time:

0.012s

59. you are given with array of numbers.you have to find whether array is beautiful or not. A beautiful array is an array whose sum of all numbers is divisible by 2, 3 and 5

**Sample Input:**

5
5 25 35 -5 30

Sample Output:

1

Completion Status: Completed

Concepts Included:

array

numbers

Language Used: PYTHON 3

Source Code:

```
n=int(input())
res=list(map(int,input().split()))
op=sum(res)
if op%2==0 and op%3==0 and op%5==0:
print("1")
else:
print("0")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.011s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0

Compilation Status: Passed

Execution Time:

0.009s

60. You are given with an array of numbers, Your task is to print the difference of indices of largest and smallest number.All number are unique.

Sample Input:

5
1 6 4 0 3

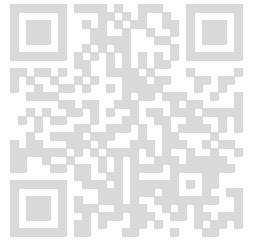**Sample Output:**

-2

Completion Status: Completed

Concepts Included:

array

numbers

Language Used: PYTHON 3
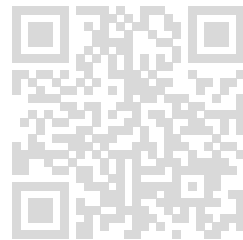
Source Code:

```
n=int(input())
nos=list(map(int,input().split()))
res=((nos.index(max(nos)))-(nos.index(min(nos))))
print(res)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-2

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

5

Compilation Status: Passed

Execution Time:

0.01s

61. Assume you are a student studying in school.You are given a task to find first negative integer for each and every window of size k.

Sample Input:

7
1 -2 -3 -4 5 6 -7
3

Sample Output:

-2 -2 -3 -4 -7

Completion Status: Completed

Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```python
n = int(input())
a = list(map(int,input().split(' ')))
k = int(input())
neg = []
for i in range(len(a)):
if(i+k>len(a)):
break
for j in range(i,i+k):
if a[j]<0:
neg.append(a[j])
break
else:
neg.append(0)
print(*neg)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-2 -2 -3 -4 -7

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0 0 0 0 0

Compilation Status: Passed

Execution Time:

0.01s

62. You are given with an array. For each element present in the array your task is to print the next smallest than that number. If it is not smallest print -1

Sample Input:

7
10 7 9 3 2 1 15

Sample Output:

7 3 3 2 1 -1 -1

Completion Status: Completed

Concepts Included:

array

Amazon

Flipkart

OYO-Rooms

Samsung

Snapdeal
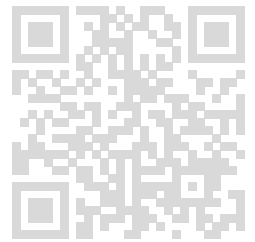
Zoho

guvi-learning-path

Language Used: PYTHON 3

Source Code:

```
n=int(input())
arr=list(map(int,input().split()))
c=[]
for i in range(n):
value=0
for j in range(i,n):
if arr[i]>arr[j]:
value=arr[j]
break
else:
value=-1

c.append(value)
print(*c)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

7 3 3 2 1 -1 -1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-1 -1 -1 -1 -1

Compilation Status: Passed

Execution Time:

0.009s
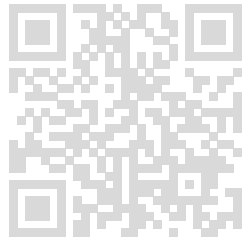
63. You are given with an circular array .Your task is calculate the difference between two consecutive number. And if difference is greater than 'k', print 1 else print 0

Sample Input:

5 15
50 65 85 98 35

Sample Output:

0 1 0 1 0

Completion Status: Completed

Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```python
n,k=(int(no) for no in input().split())
arr=list(map(int,input().split()))
a=[]
for i in range (n-1):
if abs(arr[i]-arr[i+1])>k:
a.append('1')
else:
a.append('0')

if abs(arr[-1]-arr[0])>k:
a.append('1')
else:
a.append('0')
print(*a)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 1 1 1 1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1 1 1 1 1 1 0 1 0 1

Compilation Status: Passed

Execution Time:

0.01s

64. Ramesh is a student and wants to find out if there is any other student in his class who has got the same marks as his, in maths. Help him to find out.

Sample Input:

2 10
1 2

Sample Output:

-1
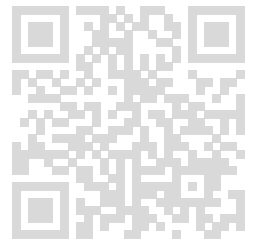
Completion Status: Completed

Concepts Included:

searching

array

Language Used: PYTHON 3

Source Code:

```
no,mark=(int(no) for no in input().split())
stu=list(map(int,input().split()))
res=[]
for i in range (no):
if stu[i]==mark:
res.append(i)
break
else:
res.append("-1")
print(*res)
```

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
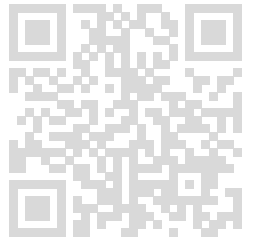
65. You are given an array of ids of prisoners. The jail authority found that there are some prisoners of same id. Your task is to help the authority in finding the common ids.

Sample Input:

7
1 1 11 121 131 141 98

Sample Output:

1

Completion Status: Completed

Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```
n=int(input())
arr=list(map(int,input().split()))
c=[]
for i in arr:
if arr.count(i)>=2:
if i not in c:
c.append(i)
if len(c)>=1:
print(*c)
else:
print("-1")
```

Compilation Details:
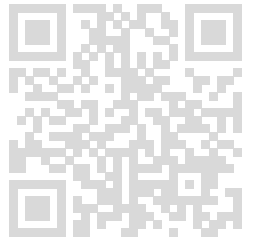
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

46

Compilation Status: Passed

Execution Time:

0.01s

66. Given a string S, print it without using semicolon in your program.Sample Testcase :INPUThello worldOUTPUThello world
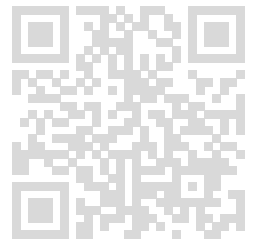
Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

Source Code:

```
S=str(input())
print(S)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

hello world

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

guvi geeks

Compilation Status: Passed

Execution Time:

0.01s

67. Ria is a 5 year old girl. Her mother wants to teach her how to sort words in the same order that they appear in a dictionary. She decides to write a program to sort a given set of strings based on their alphabetical order. Help Ria's mother to complete the program.
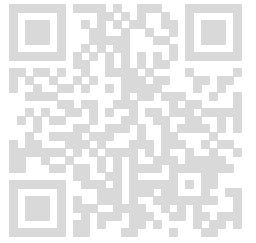
Sample Input:

3
InfinityWar EndGame Avengers

Sample Output:

Avengers EndGame InfinityWar

Completion Status: Completed

Concepts Included:

sorting

array

strings

Language Used: PYTHON 3

Source Code:

```python
no=int(input())
a=input().split()
l=[]
a.sort()
for i in a:
l.append(i)
print(*l)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

guvi online training

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

jc sboa

Compilation Status: Passed

Execution Time:

0.01s

68. You are given a number with duplicate digits your task is to remove the immediate duplicate digits and print the result

Sample Input:

1331

Sample Output:

11

Completion Status: Completed

Concepts Included:

strings

array

splay trees

Language Used: PYTHON 3

Source Code:

```python
from itertools import groupby
no=int(input())
new_no=[int(no) for no in str(no)]
res = [i for i, j in groupby(new_no) if sum(1 for x in j) < 2]
print(*res,sep="")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

11

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

156987

Compilation Status: Passed

Execution Time:

0.015s

69. Given a number N, print the odd digits in the number(space seperated) or print -1 if there is no odd digit in the given number.Input Size : N <= 100000Sample Testcase :INPUT2143OUTPUT1 3

Completion Status: Completed
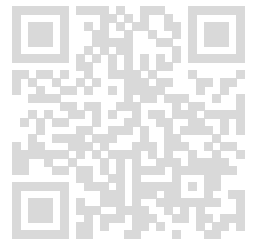
Concepts Included:

array

mathematics

Language Used: PYTHON 3

Source Code:

```
no=int(input())
ls=[]
for i in range(len(str(no))):
digit=no%10
if digit%2!=0:
ls.append(digit)
no=no//10
if not ls:
print("-1")
else:
ls.reverse()
print(*ls,sep=" ")
```

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 3

Compilation Status: Passed

Execution Time:

0.01s
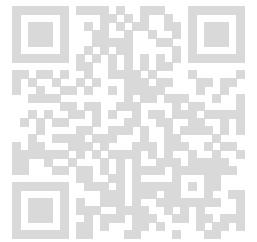
70. Given 2 numbers N and K followed by elements of N .Print 'yes' if K exists else print 'no'.Sample Testcase :INPUT4 21 2 3 3OUTPUTyes

Completion Status: Completed

Concepts Included:

basics

array

Language Used: PYTHON 3

Source Code:

```
N,K=(int(no) for no in input().split())
ele=list(map(int,input().split()))
count=0
i=1
if(i==K):
for i in range(1,N+1):
count=count+1
print("yes")
break
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

71. You are a passport issuer, but due to some problems in the system, there are redundant passport numbers. Your task is to delete all the duplicate passport numbers. You are given a list of passport numbers.

Sample Input:

5
A23 B56 B56 C79 D16

Sample Output:

A23 B56 C79 D16

Completion Status: Completed

Concepts Included:

array

set

Language Used: PYTHON 3

Source Code:

inp=int(input())
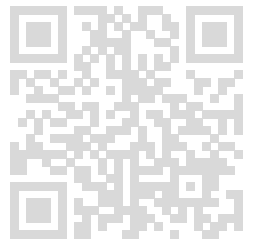
```
ele=list(map(str,input().split()))
res = []
[res.append(x) for x in ele if x not in res]
print(*res)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

11 12 13 A14 15 19 16 B18

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A23 B56 C79 D16

Compilation Status: Passed

Execution Time:

0.01s

72. Given a string 'S' swap the even and odd characters starting from index 1(Assume the index starts from 0).Input Size : |s| <= 10000000(complexity O(n))Sample Testcase :INPUTcodekataOUTPUTocedakat

Completion Status: Completed

Concepts Included:

basics

array

strings
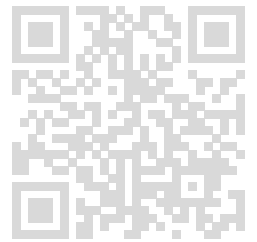
Language Used: PYTHON 3

Source Code:

```
S=input()
t=list(S)
t[::2],t[1::2]=t[1::2],t[::2]
res=''.join(t)
print(res)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

ugiv

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

ejardl

Compilation Status: Passed

Execution Time:

0.014s
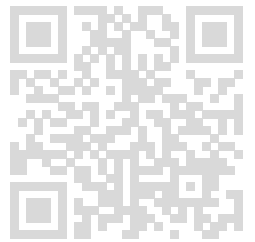
73. Given a string S, print the reverse of the string after removing the vowels.If the resulting string is empty print '-1'.Input Size : 1 <= N <= 100000Sample Testcase :INPUTcodekataOUTPUTtkdc

Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

Source Code:

```
S=input()
ls=""
for i in S:
if i not in "aeiouAEIOU":
ls=ls+i
if ls=="":
print("-1")
else:
print(ls[::-1])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

mhtyhr

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

-1

Compilation Status: Passed

Execution Time:

0.01s

74. Given a sentence S take out the extra spaces.If no extra space is present print the same as output.Input Size : |s| <= 100000(complexity O(n))Sample Testcase :INPUTcodekata challengeOUTPUTcodekata challenge

Completion Status: Completed

Concepts Included:

array

strings

Language Used: PYTHON 3

Source Code:

```
S=input()
res=" ".join(S.strip().split())
print(res)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

coding platform

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

guvi geeks

Compilation Status: Passed

Execution Time:

0.01s

75. Given a binary number convert it into octal format.Sample Testcase :INPUT1100100OUTPUT144

Completion Status: Completed

Concepts Included:

array

strings

Language Used: PYTHON 3

Source Code:

```
a=int(input(),2)
#print(a)
res=oct(a)
print(res[2::])
```
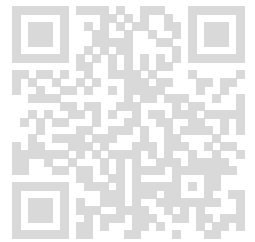
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

144

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

24

Compilation Status: Passed

Execution Time:

0.009s

76. Given a number N in decimal convert it into binary value.Input Size : N <= 100000 Sample Testcase :INPUT5OUTPUT101

Completion Status: Completed

Concepts Included:

mathematics

array
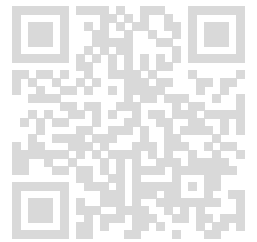
Language Used: PYTHON 3

Source Code:

```
n=int(input())
print(bin(n)[2::])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1000

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

110

Compilation Status: Passed

Execution Time:

0.01s

77. Given 2 strings S1 and s2, check whether they are case senitively equal without using any predefined function(case sensitive).If they are not same print 'no'Sample Testcase :INPUTguvi guviOUTPUTyes

Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

Source Code:

```
s1,s2=input().split()
if s1==s2:
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:

0.013s
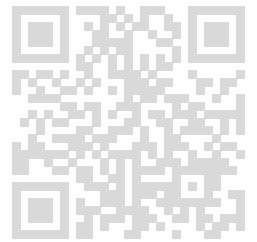
78. Given a binary number convert it to hexadecimal.Sample Testcase :INPUT1100100OUTPUT64

Completion Status: Completed

Concepts Included:

bitwise

array

strings

Language Used: PYTHON 3

Source Code:

```
a=int(input(),2)
res=hex(a)
print(res[2::])
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

64

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

14

Compilation Status: Passed

Execution Time:

0.01s

79. Given an array of N elements switch(swap) the element with the adjacent element and print the output.Sample Testcase :INPUT53 2 1 2 3OUTPUT2 3 2 1 3

Completion Status: Completed

Concepts Included:

mathematics

array

bitwise

basics

Language Used: PYTHON 3

Source Code:

```
n = int(input())
arr = list(map(int, input().split()))
for i in range(0,n-1,2):
temp = arr[i]
arr[i] = arr[i+1]
arr[i+1] = temp
print(*arr)
```

Compilation Details:
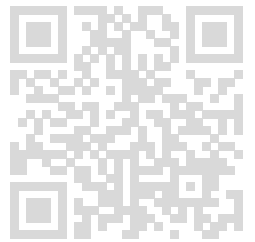
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

3 2 5 4 5 6

Compilation Status: Passed

Execution Time:

0.011s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 3 2 3 1

Compilation Status: Passed

Execution Time:

0.011s

80. Given a string S, print the encoded string by adding 3 to each character(a maps to d,b maps to e,c maps to f and so on).Input Size : 1 <= N <= 100000Sample Testcase :INPUTRADAROUTPUTUDGDU

Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

Source Code:
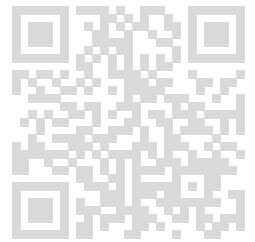
```
s=input()
ls=[]
for i in range(len(s)):
if (ord(s[i])== 88):
ls.append(chr(65))
elif (ord(s[i])==120):
ls.append(chr(97))
elif (ord(s[i])==89):
ls.append(chr(66))
elif (ord(s[i])==121):
ls.append(chr(98))
elif (ord(s[i])==90):
ls.append(chr(67))
elif (ord(s[i])==122):
ls.append(chr(99))
else:
res=ord(s[i])
sum=res+3
opt=chr(sum)
ls.append(opt)
print(*ls,sep="")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

ARUR

Compilation Status: Passed

Execution Time:

0.012s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

Compilation Status: Passed

Execution Time:

0.011s
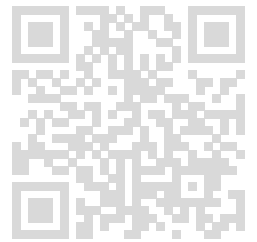
81. A number is given as input.Find the maximum number that can be formed using the digits.Input Size : N <= 10000000 Sample Testcase :INPUT4123OUTPUT4321

Completion Status: Completed

Concepts Included:

mathematics

array

strings

Language Used: PYTHON 3

Source Code:

```
N=[int(no) for no in input()]
ls=[]
N.sort(reverse=True)
ls.append(N)
print(*N,sep="")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

431

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:
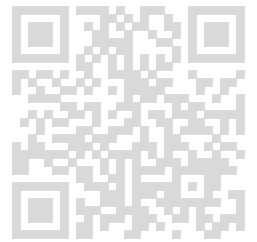
Input:

< hidden >

Expected Output:

< hidden >

Output:

00000

Compilation Status: Passed

Execution Time:

0.01s

82. Given 2 arrays print 'yes' if they are mirror images of each other,otherwise 'no'.Input Size : N <= 1000000Sample Testcase :INPUT41 2 3 44 3 2 1OUTPUTyes

Completion Status: Completed

Concepts Included:

array

companies

Language Used: PYTHON 3

Source Code:

```
no=int(input())
ele1=list(map(int,input().split()))
ele2=list(map(int,input().split()))

if ele1 == ele2[::-1]:
print("yes")
else:
print("no")
```
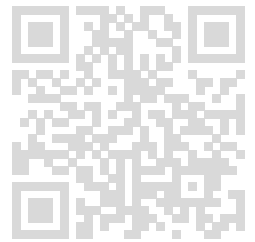
Compilation Details:

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:

0.015s

83. Given a string S, print 'yes' if the strings 'GUVI' and 'GEEK' is present case-sensitively in the string else print 'no'.Input Size : 1 <= 100Sample Testcase :INPUTVishal_Sundar prepared this questionOUTPUTno

Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

Source Code:

```
s=input().split()
sample=['GUVIGEEK']
l=[]
for i in range(len(s)):
if s[i] in sample:
l.append(s[i])

if len(l)>0:
print("yes")
else:
print("no")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.01s

84. You are given with string of words, we have to arrange them in reverse saturated order.
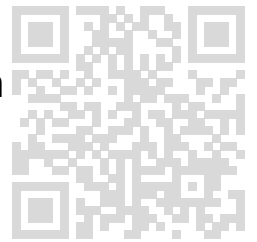
Sample Input:

I am kohli fan

Sample Output:

I ma ilhok naf

Completion Status: Completed

Concepts Included:

strings

array

Language Used: PYTHON 3

Source Code:

```python
l = str(input()).split()
a = []
for i in l:
a.append(i[::-1])
print(*a)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

I ma ilhok naf

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

ivug seigolonhcet

Compilation Status: Passed

Execution Time:

0.01s

85. Given an array print the number of subarrays that can be formed with it.Input Size : N <= 100000Sample Testcase :INPUT51 2 3 2 1OUTPUT15

Completion Status: Completed

Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```
n=int(input())
a=list(map(int,input().split()))
b=n*(n+1)//2
print(b)
```

Compilation Details:
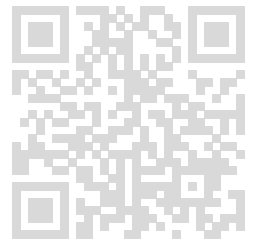
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

15

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

21

Compilation Status: Passed

Execution Time:

0.01s

86. Given a number N and an array of N elements, find the Bitwise XOR of the array elements.Input Size : N <= 100000Sample Testcase :INPUT22 4OUTPUT6

Completion Status: Completed

Concepts Included:
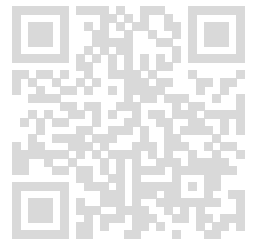
array

bitwise

bascis

Language Used: PYTHON 3

Source Code:

```
n = int(input())
arr = list(map(int, input().split()))
result = arr[0]
```

```
for i in range(1, n):
result = result ^ arr[i]

print(result)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4

Compilation Status: Passed

Execution Time:

0.015s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

7

Compilation Status: Passed

Execution Time:

0.01s

87. Write a code to get a integer n as input and calculate the smallest perfect power of 2 greater than n.

Sample Input:

48

Sample Output:

64

Completion Status: Completed

Concepts Included:

basics

bit manipulation

Looping

Language Used: PYTHON 3

Source Code:

```python
import math
n = int(input())
for i in range(int(math.sqrt(n))+2,1,-1):
if math.pow(2,i) <= n:
print(int(math.pow(2,i+1)))
break
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

64

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:
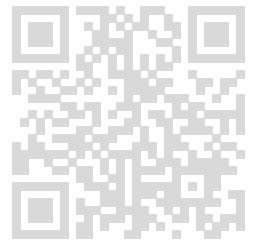
Input:

< hidden >

Expected Output:

< hidden >

Output:

256

Compilation Status: Passed

Execution Time:

0.01s

88.

Given a string as input, you have to reverse the string by keeping the punctuation and spaces intact. You have to modify the source string itself without creating another string.

Sample Input:

A man, in the boat says : I see 1-2-3 in the sky

Sample Output:

y kse, ht ni3 21ee sIsy : a sta o-b-e ht nin amA

Completion Status: Completed

Concepts Included:

basic io math - tf

Accolite

Amazon

D-E-Shaw

FactSet

MakeMyTrip

Microsoft

Nagarro

Samsung

guvi-learning-path

Language Used: PYTHON 3
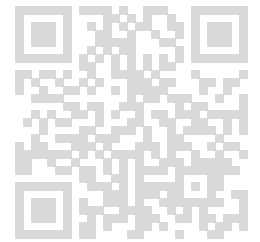
Source Code:

```
x=input()
y=list(x)
z=[]
l=[]
for i in y:
if i.isalnum():
l.append(i)
else:
z.append(i)
f=l[::-1]

kk=[]
for i in y:
if i.isalnum():
d=f.pop(0)
kk.append(d)
else:
kk.append(i)
else:
print("".join(kk))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

nlhi+!@@,iF

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

eXrl+!##,os:Oo7vnYrp'#hTQoXvBucRFhdZJ H;fZRnnlhii!F

Compilation Status: Passed

Execution Time:

0.009s
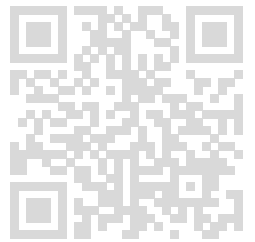
89. You are given a number n,ranging from 1 to n. Out of which one number is missing. Your task is to print that missing number.

Sample Input:

5
1 3 5 2

Sample Output:

4

Completion Status: Completed

Concepts Included:

array

Accolite

Adobe

Amazon

Cisco

D-E-Shaw

Intuit

Microsoft

Morgan

Stanley

Ola

Cabs

Payu

Qualcomm

Visa

guvi-learning-path

Language Used: PYTHON 3

Source Code:

```
n = int(input())
arr = list(map(int,input().split()))

total_sum = n * (n+1) // 2

given_sum = sum(arr)

missing_number = total_sum - given_sum

print(missing_number)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.009s
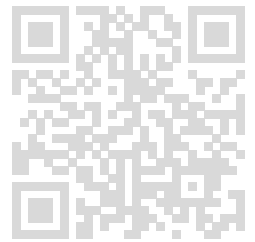
90. Given two strings S1 and S2,display 'yes' if given two strings are complementary otherwise display 'no'. If we join alphabets of both the strings we should get all 26 capital letters exactly once, then only we can call them as complementary.Sample Testcase :INPUTABDCFGIJKLMNOPQUVWXYZEHRSTOUTPUTyes

Completion Status: Completed

Concepts Included:

strings

companies

loop

Language Used: PYTHON 3

Source Code:

```
A =['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

A1=input()
A2=input()

cout=0
if len(A1)+len(A2)==len(A):
for i in A1:
if i in A:
A.remove(i)
for i in A2:
if i in A:
A.remove(i)
if len(A)==0:
print('yes')
else:
print('no')
else:
print('no')
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.014s

91. Write a code to generate an inverted half pyramid pattern using stars.

Sample Input:

5

Sample Output:

```
* * * * *
* * * *
* * *
```

* *
*

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n):
l=[]
for j in range(n-i):
l.append("*")
print(' '.join(l))


n=int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
* * * * *
* * * *
* * *
* *
*
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

Compilation Status: Passed

Execution Time:

0.013s

92. Write a code to generate a inverted half pyramid pattern using numbers.

Sample Input:

5

**Sample Output:**

```
12345
1234
123
12
1
```

Completion Status: Completed

Concepts Included:

patterns
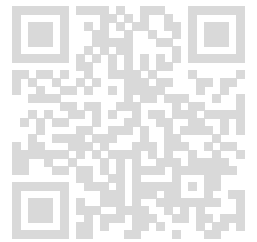
Language Used: PYTHON 3

Source Code:

```python
n = int(input())
for i in range(n):
p=1
```
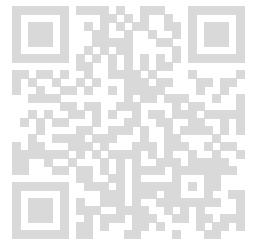
```
for j in range(i,n):
print(p,end="")
p+=1

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
12345
1234
123
12
1
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
123456
12345
1234
123
12
1
```

Compilation Status: Passed

Execution Time:
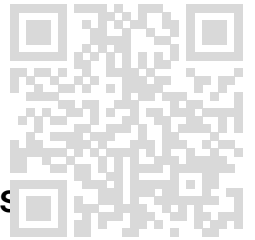
93. Write a code to generate a half pyramid pattern using numbers

Sample Input:

5

Sample Output:

```
1
22
333
4444
55555
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
R=int(input())
for i in range(R):
for j in range(i+1):
print(i+1,end="")
print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
22
333
4444
```

55555

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1
22
333
4444

Compilation Status: Passed

Execution Time:

0.01s

94. Generate a hollow inverted half pyramid pattern using numbers.

Sample Input:

5

Sample Output:

12345
1 4
1 3
12
1

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3
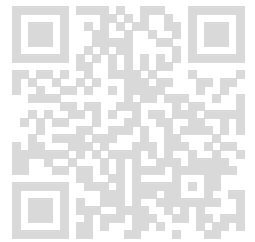
Source Code:

```
n = int(input())
for i in range(n):
p=1
for j in range(i,n):
if(i==0 or j==i or j==n-1):
print(p, end="")
else:
print("", end=" ")
p+=1

print()
```
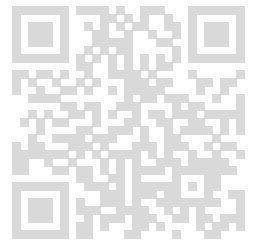
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
12345
1  4
1 3
12
1
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1234
1 3
12
```

1

Compilation Status: Passed

Execution Time:

0.009s
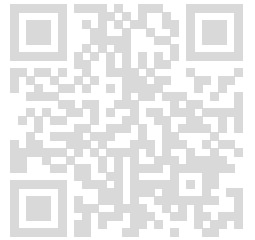
95. Write a code to generate a hollow rectangle using stars.

Sample Input:

3 5

Sample Output:

```
* * * * *
*       *
* * * * *
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(r, c):
for i in range(1, r+1):
l = []
if i == 1 or i == r:
for j in range(c):
l.append('*')
else:
l.append('*')
for j in range(1, c-1):
l.append(' ')
l.append('*')
print(*l)

r, c = map(int, input().split())
pattern(r, c)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
* * * * *
*       *
* * * * *
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
* * *
*   *
* * *
```

Compilation Status: Passed

Execution Time:

0.009s

96. Write a code to generate an inverted full pyramid pattern using stars.

Sample Input:

5

Sample Output:

```
* * * * *
* * * *
* * *
```

* *
*

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n):
for j in range(i):
print(' ', end = "")
l = []
for j in range(n-i):
l.append('*')
print(' '.join(l))

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

* * *
* *
*

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
* * * *
* * *
* *
*
```

Compilation Status: Passed

Execution Time:

0.009s

97. Write a code to generate a pyramid using stars.

Sample Input:

6

Sample Output:

```
*
**
***
****
*****
******
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(1,n+1):
for j in range(n-i):
print(" ",end="")
for j in range(i):
print("*",end="")
print()
```

```
n=int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
**
***
****
*****
******
```

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
**
***
****
*****
```

Compilation Status: Passed

Execution Time:

0.009s
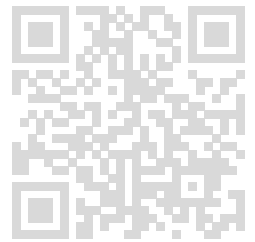
98. Write a code to generate a half pyramid number pattern.

Sample Input:

5

Sample Output:

```
12345
4321
123
21
1
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n):
if i % 2 == 0:
for j in range(n-i):
print(j+1, end = "")
else:
for j in range(n-i, 0, -1):
print(j, end = "")
print()
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:
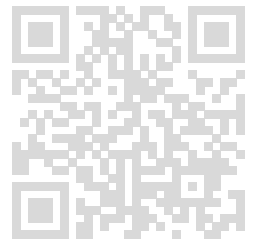
Input:

< hidden >

Expected Output:

< hidden >

Output:

12345

```
4321
123
21
1
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
12345678910
987654321
12345678
7654321
123456
54321
1234
321
12
1
```

Compilation Status: Passed

Execution Time:

0.01s

99. Generate a solid rectangle using stars.

Sample Input:

3 5

Sample Output:

```
* * * * *
* * * * *
* * * * *
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
R,C=(int(no) for no in input().split())

for i in range(1,R+1):
l=[]
for j in range(C):
l.append("*")
print(*l)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
* * * * *
* * * * *
* * * * *
```

Compilation Status: Passed

Execution Time:

0.012s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
```
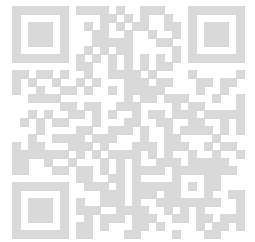
Compilation Status: Passed

Execution Time:

0.014s

100. Generate a full pyramid using numbers.

Sample Input:

5

Sample Output:

```
1
232
34543
4567654
567898765
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
def pattern(n):
for i in range(1, n+1):
for j in range(n-i):
print(' ', end = "")
l = []
for j in range(i, (2*i)):
l.append(str(j))
for j in range((2*i)-2, i-1, -1):
l.append(str(j))
print(''.join(l))

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
232
34543
4567654
567898765
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
232
34543
```

Compilation Status: Passed

Execution Time:

0.009s

101. Generate the following inverted character with star pattern.

bbbb*bbbb

bbb*bbb**

bb***bb**

b*****b**

****

Sample Input:

5

Sample Output:

```
bbbb*bbbb
bbb***bbb
bb*****bb
b*******b
*********
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
for i in range(n):

for j in range(i,n-1):
print("b",end="")

for j in range (i):
print("*",end="")

for j in range (i+1):
print("*",end="")

for j in range(i,n-1):
print("b",end="")


print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
bbbb*bbbb
bbb***bbb
bb*****bb
b*******b
*********
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
bbbbbb*bbbbbb
bbbbb***bbbbb
bbbb*****bbbb
bbb*******bbb
bb*********bb
b***********b
*************
```

Compilation Status: Passed

Execution Time:

0.009s

102. Generate the following pattern.

*****

b****

bb***

bbb**

bbbb*

Sample Input:

5
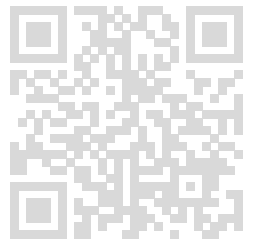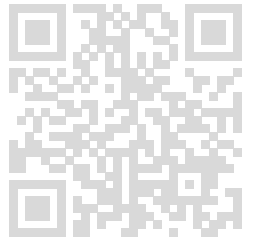
Sample Output:

```
*****
b****
bb***
bbb**
bbbb*
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
p=n
for i in range(n):
for j in range(i):
print("b",end="")

for j in range (i,n):
print("*", end="")
p=p-1

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*****
b****
bb***
bbb**
bbbb*
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
****
b***
bb**
bbb*
```
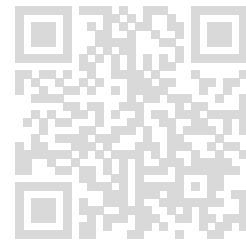
Compilation Status: Passed

Execution Time:

0.009s

103. In the IPL season's valedictory function the organizers have organized for a dance program. The dance has to be performed by men along with the points of the diagonals of the square of side 'n' and the females along with points of the borders. The remaining positions are filled by children. You have to determine their respective positions by writing a program.

Sample Input:

7

Sample Output:

```
M F F F F F M
F M C C C M F
```

```
F C M C M C F
F C C M C C F
F C M C M C F
F M C C C M F
M F F F F M
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n):
l = []
for j in range(n):
if i == j or i+j == n-1:
l.append('M')
elif i == 0 or j == 0 or i == n-1 or j == n-1:
l.append('F')
else:
l.append('C')
print(*l)

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
M F F M
F M M F
F M M F
M F F M
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
M F F F F F M
F M C C C M F
F C M C M C F
F C C M C C F
F C M C M C F
F M C C C M F
M F F F F F M
```

Compilation Status: Passed

Execution Time:

0.01s

104. Write a code to generate a pyramid pattern using stars from the given input size N.

Sample Input:

5

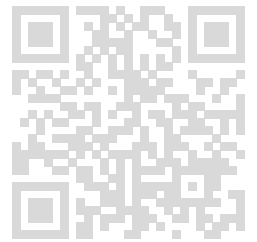### Sample Output:

```
*
* *
* * *
* * * *
* * * * *
```

Completion Status: Completed

Concepts Included:
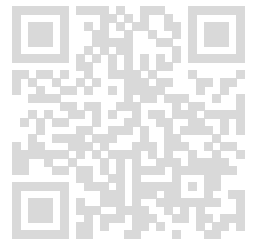
patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(1, n+1):
l = []
for j in range(i):
l.append('*')
print(*l)

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
* *
* * *
* * * *
* * * * *
```

Compilation Status: Passed

Execution Time:

0.015s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
* *
```

```
* * *
* * * *
* * * * *
* * * * * *
```

Compilation Status: Passed

Execution Time:

0.009s

105. Write a code to generate a full pyramid pattern using stars.

Sample Input:

5

Sample Output:

```
*
* *
* * *
* * * *
* * * * *
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):

for i in range(1, n+1):
for j in range(n-i):
print(' ', end = "")
l = []
for j in range(i):
l.append('*')
print(*l)
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

*
* *
* * *

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

*
* *

Compilation Status: Passed

Execution Time:

0.01s

106. Write a code to generate a aplhabet half pyramid pattern.

Sample Input:

5

Sample Output:

EDCBA
EDCB
EDC
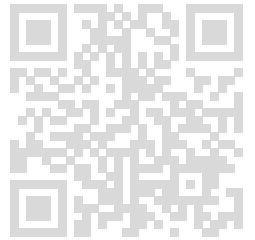
ED
E

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
let = 65
for i in range(n):
for j in range(n-1, i-1, -1):
print(chr(let+j), end = "")
print()

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

EDCBA
EDCB
EDC
ED
E

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
DCBA
DCB
DC
D
```

Compilation Status: Passed

Execution Time:

0.009s

107. Write a code to generate a half pyramid pattern using numbers.

Sample Input:

5

Sample Output:

```
12345
2345
345
45
5
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
def pattern(n):
for i in range(1,n+1):
for j in range(i,n+1):
print(j,end="")
print()
```
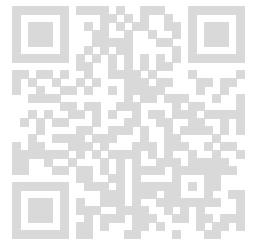
```
n=int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
12345678
2345678
345678
45678
5678
678
78
8
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1234567
234567
34567
4567
567
67
7
```

Compilation Status: Passed

Execution Time:

0.009s

108. Generate a half pyramid pattern using numbers.

Sample Input:

5

Sample Output:

1
12
123
1234
12345

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
n = int(input())
for i in range(n):
p=1
for j in range(i+1):
print(p,end="")
p+=1

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

```
12
123
1234
12345
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
12
123
```

Compilation Status: Passed

Execution Time:

0.01s

109. Generate the aplhabet pattern using nested loops.

Sample Input:

abcdef

Sample Output:

```
abcdef
b    e
c   d
d   c
e   b
fedcba
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(s):
n = len(s)
print(s)
for i in range(1, n-1):
res = s[i]
res += ' '*(n-2)
res += s[n-i-1]
print(res)
print(s[::-1])

s = input()
pattern(s)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
abcdef
b    e
c    d
d    c
e    b
fedcba
```

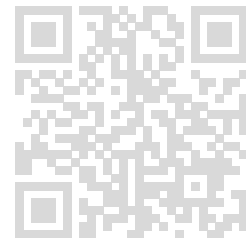**Compilation Status:** Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

abcd
b c
c b
dcba

Compilation Status: Passed

Execution Time:

0.013s

110. Write a code to generate a half pyramid pattern using numbers.

Sample Input:

5

Sample Output:

13579
3579
579
79
9

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(1, (2*n), 2):
l = []
for j in range(i, 2*n, 2):
l.append(str(j))
print(''.join(l))

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
13579
3579
579
79
9
```

Compilation Status: Passed

Execution Time:

0.015s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1357
357
57
7
```
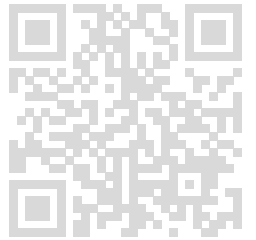
Compilation Status: Passed

Execution Time:

0.01s

111. Generate a half diamond pattern using stars and numbers in a palindromic pattern.

Sample Input:

3

Sample Output:

```
*
*1*
*121*
*12321*
*121*
*1*
*
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
def pattern(n):
print('*')
for i in range(1, n+1):
l = []
l.append('*')
for j in range(1, i+1):
l.append(str(j))
for j in range(i-1, 0, -1):
l.append(str(j))
l.append('*')
print(''.join(l))
for i in range(n-1, 0, -1):
l = []
l.append('*')
for j in range(1, i+1):
l.append(str(j))
for j in range(i-1, 0, -1):
l.append(str(j))
l.append('*')
print(''.join(l))
print('*')

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
*1*
*121*
*12321*
*121*
*1*
*
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
*1*
*121*
*12321*
*1234321*
*12321*
*121*
*1*
*
```
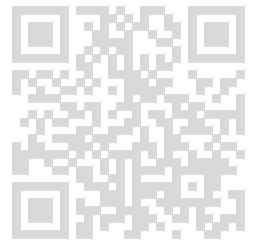
Compilation Status: Passed
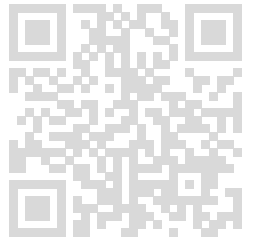
Execution Time:

0.012s

112. Write a code to generate a triangle character pattern.

Sample Input:

5

Sample Output:

A
A B
A B C
A B C D
A B C D E

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
c = 65
l = []
for i in range(n):
for j in range(n-i-1):
print(' ', end = "")
l.append(chr(c + i))
print(*l)

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A
A B
A B C
A B C D
A B C D E

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A
A B
A B C
A B C D

Compilation Status: Passed

Execution Time:

0.009s

113. Write a code to generate a half pyramid pattern using numbers.

Sample Input:

5

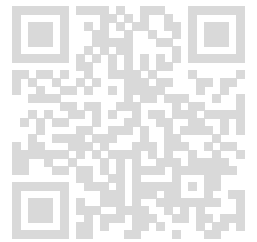**Sample Output:**

5
45
345
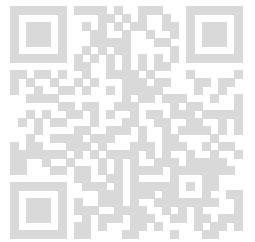2345
12345

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
n=int(input())
k=n
for i in range(n): #rows
p=k
l=[]
for j in range(i+1): #cols
#print(p,end=" ")
l.append(str(p))
p+=1
k-=1
print("".join(l))
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
7
67
567
4567
34567
234567
1234567
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
6
56
456
```

3456
23456
123456

Compilation Status: Passed

Execution Time:

0.01s

114. Write a code to generate an Rhombus Pattern using stars.

Sample Input:

4

Sample Output:

```
****
****
****
****
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
for i in range(n):
for j in range(i,n-1):
print(" ",end="")

for j in range(i+1): #cols
print("*",end="")

for j in range (i,n-1):
print("*", end="")
print()
```
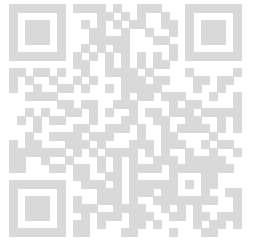
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
****
****
****
****
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*****
*****
*****
*****
*****
```

Compilation Status: Passed

Execution Time:

0.009s

115. Write a code to generate a square pattern using numbers.

Sample Input:

5

Sample Output:

5 5 5 5 5

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

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
k=n
for i in range(n): #rows
p=k
l=[]
for j in range(i+1): #cols
#print(p,end=" ")
l.append(p)
p+=1

for j in range (i,n-1):
#print(n, end=" ")
l.append(n)
k-=1
print(*l)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

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

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

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

Compilation Status: Passed

Execution Time:

0.01s

116. Write a code to generate a half pyramid pattern using numbers.

Sample Input:
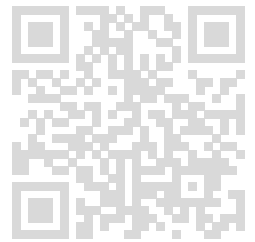
5

Sample Output:

```
1
24
135
2468
13579
```

Completion Status: Completed
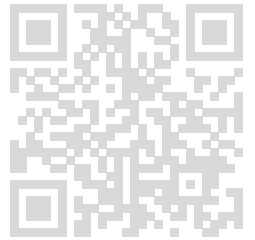
Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
def pattern(n):
o = ['1']
e = ['2']
for i in range(1, n+1):
if i % 2 == 1:
print(''.join(o))
else:
print(''.join(e))
o.append(str(int(o[-1]) + 2))
e.append(str(int(e[-1]) + 2))

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1
24
135
2468
13579

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1
24

117. Write a code to generate a alphabet pyramid pattern.

Sample Input:

5

Sample Output:

```
    A
   CCC
  EEEEE
 GGGGGGG
IIIIIIIII
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
c = 65
for i in range(1, n+1):
for j in range(n-i):
print(' ', end = "")
for j in range(2*i - 1):
print(chr(c), end = "")
print()
c += 2
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
A
CCC
EEEEE
GGGGGGG
IIIIIIIII
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
A
CCC
EEEEE
GGGGGGG
```

Compilation Status: Passed

Execution Time:

0.01s

118. Write a code to generate a right arrow using patterns.
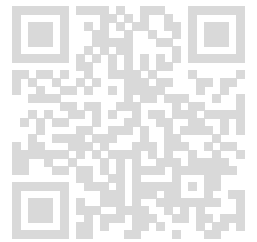
Sample Input:

5

Sample Output:
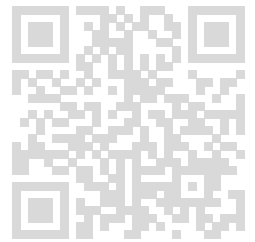
```
*
*
*
*
****
```

```
*
*
*
*
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n-1):
for j in range(n-i-1):
print(' ', end = "")
print('*')
print('*' * n)
for i in range(n-2, -1, -1):
for j in range(n-i-1):
print(' ', end = "")
print('*')
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

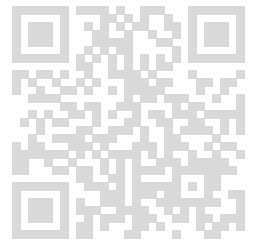**Input:**

< hidden >

Expected Output:

< hidden >

Output:

```
*
*
*
*
*****
*
*
*
*
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
*
*
****
*
*
*
```

Compilation Status: Passed

Execution Time:

0.009s

119. Write a code to generate an hollow rhombus Pattern using stars.
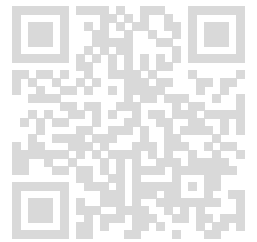
Sample Input:

4

Sample Output:

```
****
*  *
*  *
****
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(1, n+1):
for j in range(n-i):
print(' ', end = "")
if i == 1 or i == n: print('*' * n)
else:
s = '*'
s += ' ' * (n-2)
s += '*'
print(s)
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
****
*  *
*  *
****
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

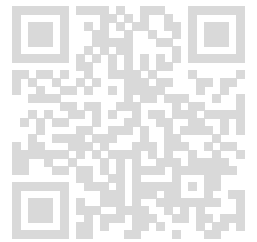**Expected Output:**

< hidden >

Output:

```
*****
*   *
*   *
*   *
*****
```

Compilation Status: Passed

Execution Time:

0.01s

120. Write a code to generate a solid half diamond pattern using stars.

Sample Input:

5

Sample Output:

```
*
**
***
****
*****
****
***
**
*
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
for i in range(n-1):
    for j in range(i+1):
        print("*",end="")
    print()

for i in range(n):
    for j in range(i,n):
        print("*",end="")
```

print()

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
**
***
****
*****
****
***
**
*
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

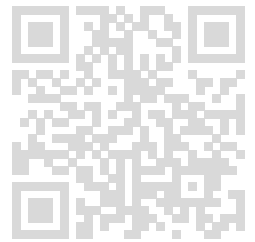### Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*
**
***
**
*
```

Compilation Status: Passed

Execution Time:

0.01s

121. Write a code to generate a aplhabet pyramid pattern.

Sample Input:

5

Sample Output:

```
A
BBB
CCCCC
DDDDDDD
EEEEEEEEE
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
c = 65
for i in range(n):
for j in range(n-i-1):
print(' ', end = "")
for j in range((2*i)+1):
print(chr(c+i), end = "")
print()
n = int(input())
pattern(n)
```

Compilation Details:

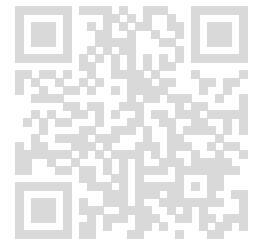**TestCase1:**

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
A
BBB
```

CCCCC
DDDDDDD
EEEEEEEEE

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A
BBB
CCCCC
DDDDDDD
EEEEEEEEE
FFFFFFFFFFF

Compilation Status: Passed

Execution Time:

0.009s

122. Write a code to generate a aplhabet half pyramid pattern.
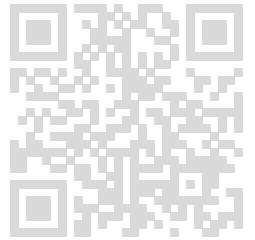
Sample Input:

5

Sample Output:

ABCDE
ABCD
ABC
AB
A

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n):
for j in range(i):
print(' ', end = "")
for j in range(n-i):
print(chr(65+j), end = "")
print()
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
ABCDE
ABCD
ABC
AB
A
```

Compilation Status: Passed

Execution Time:

0.014s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
ABCDEFGH
ABCDEFG
ABCDEF
ABCDE
ABCD
ABC
AB
A
```

Compilation Status: Passed

Execution Time:

0.009s

123. Write a code to generate a half pyramid number pattern.

Sample Input:

5

Sample Output:

```
1
2 4
1 3 5
2 4 6 8
1 3 5 7 9
```

Completion Status: Completed
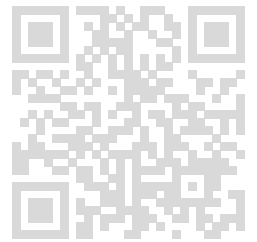
Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
o = ['1']
e = ['2']
for i in range(1, n+1):
if i % 2 == 1:
print(*o)
else:
print(*e)
o.append(str(int(o[-1]) + 2))
e.append(str(int(e[-1]) + 2))
```

```
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
2 4
1 3 5
2 4 6 8
1 3 5 7 9
2 4 6 8 10 12
1 3 5 7 9 11 13
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >
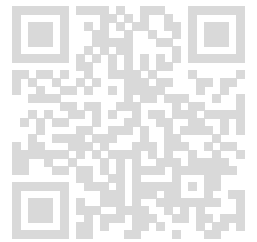
Expected Output:

< hidden >

Output:

```
1
2 4
1 3 5
2 4 6 8
```

Compilation Status: Passed

Execution Time:

0.01s

124. write a code to generate the following pattern.

```
**********

****bb****

***bbbb***

**bbbbbb**

*bbbbbbbb*

**bbbbbb**

***bbbb***

****bb****

**********
```

Sample Input:

10

Sample Output:

```
**********
****bb****
***bbbb***
**bbbbbb**
*bbbbbbbb*
**bbbbbb**
***bbbb***
****bb****
**********
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
n=int(input())
n=n//2

for i in range(n-1):

    for j in range(i,n):
        print("*",end="")

    for j in range (i):
        print("b",end="")

    for j in range (i):
        print("b",end="")

    for j in range(i,n):
        print("*",end="")

    print()

for i in range(n):

    for j in range(i+1):
        print("*",end="")

    for j in range (i,n-1):
        print("b", end= "")

    for j in range(i+1,n):
        print("b",end="")

    for j in range(i+1):
        print("*",end="")

    print()
```
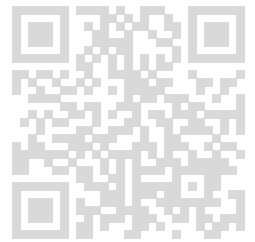
Compilation Details:
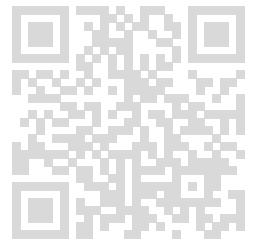
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*********
****bb****
***bbbb***
```

```
**bbbbbb**
*bbbbbbbb*
**bbbbbb**
***bbbb***
****bb****
*********
```

Compilation Status: Passed

Execution Time:

0.015s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
********
***bb***
**bbbb**
*bbbbbb*
**bbbb**
***bb***
********
```

Compilation Status: Passed

Execution Time:

0.01s

125. Write a code to generate a aplhabet half pyramid pattern.

Sample Input:

5

Sample Output:

```
EEEEE
DDDD
CCC
BB
A
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
n=int(input())
p=65+n-1
for i in range(n):
for j in range(i):
print(" ",end="")

for j in range (i,n):
print(chr(p), end="")
p=p-1

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
EEEEE
DDDD
CCC
BB
A
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
GGGGGGG
FFFFFF
EEEEE
DDDD
CCC
BB
A
```

Compilation Status: Passed

Execution Time:

0.01s

126. Write a code to generate a half pyramid pattern of mulitples of the given number.

Sample Input:

10

Sample Output:

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100
```
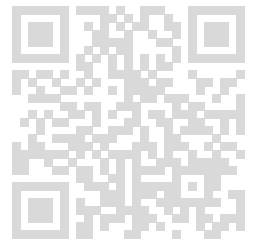
Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

def pattern(n):

```
for i in range(1, n+1):
l = []
for j in range(1, i+1):
l.append(i*j)
print(*l)

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100
```

Compilation Status: Passed
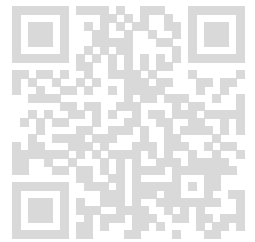
Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:
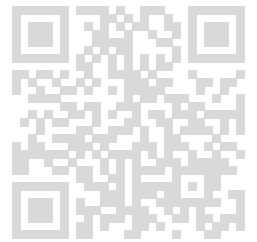
< hidden >

Output:

```
1
2 4
3 6 9
```

4 8 12 16

Compilation Status: Passed

Execution Time:

0.014s

127. Write a code to generate a pyramid pattern on numbers.

Sample Input:

5

Sample Output:

```
0
101
21012
3210123
432101234
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n):
for j in range(n-i-1):
print(' ', end = "")

for j in range(i, -1, -1):
print(j, end = "")
for j in range(1, i+1):
print(j, end = "")
print()
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0
101
21012
3210123
432101234

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

0
101
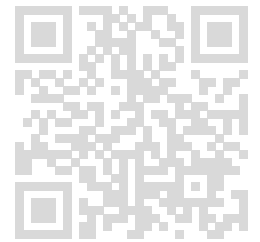21012
3210123

Compilation Status: Passed

Execution Time:

0.014s

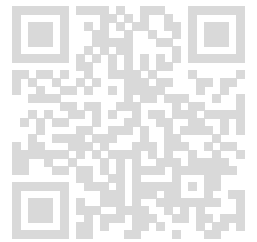## 128. Write a code to generate a alphabet pyramid pattern.

Sample Input:

5

Sample Output:

A

ABC
ABCDE
ABCDEFG
ABCDEFGHI

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())

for i in range(n):
p=65
for j in range(i+1,n):
print(" ",end="")

for j in range (i):
print(chr(p),end="")
p=p+1
for j in range (i+1):
print(chr(p),end="")
p=p+1

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A
ABC
ABCDE
ABCDEFG
ABCDEFGHI

Compilation Status: Passed

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
A
ABC
ABCDE
ABCDEFG
```

Compilation Status: Passed

129. Write a code to generate the X form of a number pattern.

Sample Input:
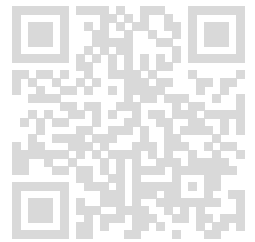
5

Sample Output:

```
1     1
2    2
3   3
4 4
5
4 4
3   3
2    2
1     1
```
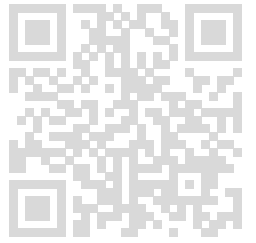
Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
    for i in range(1, n+1):
        for j in range(i-1):
            print(' ', end = "")
        print(i, end = "")
        for j in range(2*(n-i)-1):
            print(' ', end = "")
        if i != n: print(i)
        else: print()
    for i in range(n-1, 0, -1):
        for j in range(i-1):
            print(' ', end = "")
        print(i, end = "")
        for j in range(2*(n-i)-1):
            print(' ', end = "")
        if i != n: print(i)
        else: print()

n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

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

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

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

Compilation Status: Passed

Execution Time:

0.015s

130. Write a code to generate a pyramid pattern on numbers.

Sample Input:

5

Sample Output:

```
999999999
7777777
55555
333
1
```
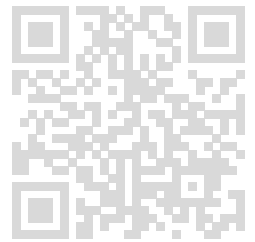
Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())
p=(n*2)-1
for i in range(n):
```

```python
for j in range(i):
    print(" ",end="")

for j in range (i,n-1):
    print(p, end= "")

for j in range(i,n):
    print(p,end="")
    p-=2

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

999999999
7777777
55555
333
1

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

7777777
55555
333
1

Execution Time:

0.009s
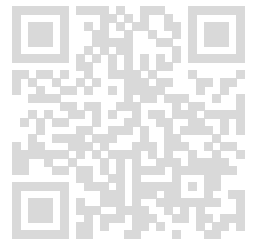
131. Write a code to generate a aplhabet pyramid pattern.

Sample Input:

5

Sample Output:

```
A
BAB
CBABC
DCBABCD
EDCBABCDE
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```
def pattern(n):
c = 65
for i in range(n):
for j in range(n-i-1):
print(' ', end = "")

for j in range(i, -1, -1):
print(chr(c+j), end = "")
for j in range(1, i+1):
print(chr(c+j), end = "")
print()
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A
BAB
CBABC
DCBABCD
EDCBABCDE

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A
BAB
CBABC
DCBABCD

Compilation Status: Passed

Execution Time:

0.01s

132. Write a code to generate a butterfly pattern printing using stars.
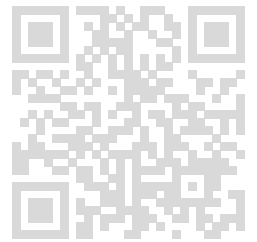
Sample Input:

5

Sample Output:

*   *
**   **

```
***   ***
**** ****
*********
*********
**** ****
***   ***
**    **
*      *
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
n=int(input())

for i in range(n):
for j in range(i+1):
print("*",end="")

for j in range(i,n-1):
print(" ", end="")

for j in range(i+1,n):
print(" ",end="")

for j in range(i+1):
print("*",end="")

print()

for i in range(n):
for j in range(i,n):
print("*",end="")

for j in range(i):
print("  ",end="")

for j in range(i-1):
print("", end="")

for j in range(i,n):
print("*",end="")

print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*       *
**     **
***   ***
**** ****
*********
*********
**** ****
***   ***
**     **
*       *
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:
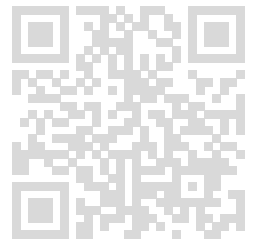
Input:

< hidden >

Expected Output:

< hidden >
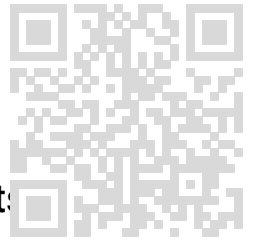
Output:

```
*     *
**   **
*** ***
*******
*******
*** ***
**   **
*     *
```

Compilation Status: Passed

Execution Time:

133. Write a code to generate a pyramid of numbers and aplhabets

Sample Input:

8

Sample Output:

```
A1
AB12
ABC123
ABCD1234
ABCDE12345
ABCDEF123456
ABCDEFG1234567
ABCDEFGH12345678
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3
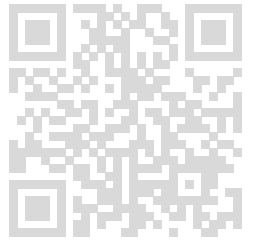
Source Code:

```python
n=int(input())

for i in range(n):
p=65
for j in range(i+1,n):
print(" ",end="")

for j in range(i+1):
print (chr(p),end="")
p+=1

p=1
for j in range(i+1): #cols
print(p,end="")
p=p+1
print()
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A1
AB12
ABC123
ABCD1234
ABCDE12345
ABCDEF123456
ABCDEFG1234567

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

A1
AB12
ABC123
ABCD1234
ABCDE12345
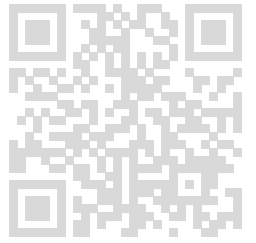ABCDEF123456

Compilation Status: Passed

Execution Time:

0.009s

134. Write a code to generate a square pattern using numbers.

Sample Input:

5

Sample Output:

```
1
1 4
1 4 9
1 4 9 16
1 4 9 16 25
1 4 9 16
1 4 9
1 4
1
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(1, n+1):
l = []
for j in range(1, i+1):
l.append(j**2)
print(*l)
for i in range(n-1, 0, -1):
l = []
for j in range(1, i+1):
l.append(j**2)
print(*l)
n = int(input())
pattern(n)
```

Compilation Details:
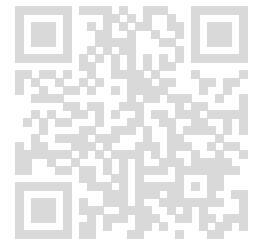
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
1 4
```

```
1 4 9
1 4 9 16
1 4 9 16 25
1 4 9 16
1 4 9
1 4
1
```

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
1
1 4
1 4 9
1 4 9 16
1 4 9
1 4
1
```

Compilation Status: Passed

Execution Time:

0.009s

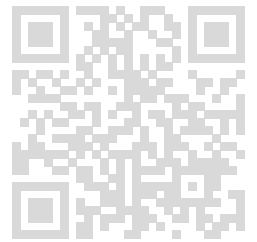## 135. Write a code to generate the hollow diamond inscribed in a rectangle using stars.

Sample Input:

5

Sample Output:

```
*********
****  ****
***   ***
**    **
```

```
*       *
*       *
**     **
***   ***
**** ****
*********
```

Completion Status: Completed

Concepts Included:

patterns

Language Used: PYTHON 3

Source Code:

```python
def pattern(n):
for i in range(n, 0, -1):
for j in range(i):
print('*', end = "")
for j in range(2*(n-i)):
print(' ', end = "")
for j in range(i):
print('*', end = "")

print()
for i in range(1, n+1):
for j in range(i):
print('*', end = "")
for j in range(2*(n-i)):
print(' ', end = "")
for j in range(i):
print('*', end = "")

print()
n = int(input())
pattern(n)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

```
*********
****  ****
***    ***
**      **
*        *
*        *
**      **
***    ***
****  ****
*********
```

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:
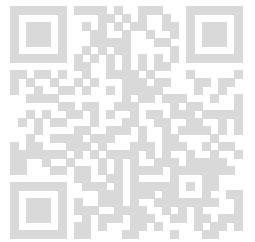
< hidden >

Expected Output:

< hidden >

Output:

```
********
***  ***
**    **
*      *
*      *
**    **
***  ***
********
```

Compilation Status: Passed

Execution Time:

0.01s

136. You are given an array of numbers. Print the least occurring element. If there is more than 1 element print all of them in decreasing order of their value.

Sample Input:
```
9
1 6 4 56 56 56 6 4 2
```

Sample Output:

2 1

Completion Status: Completed

Concepts Included:

array

Language Used: PYTHON 3

Source Code:

```python
n = int(input())
arr = list(map(int, input().split()))

# count the occurrences of each number
freq = {}
for num in arr:
freq[num] = freq.get(num, 0) + 1

# find the least occurring number
least_freq = min(freq.values())

# collect all numbers with least frequency
result = []
for num, count in freq.items():
if count == least_freq:
result.append(num)

# sort the result in decreasing order
result.sort(reverse=True)

# print the result
print(*result)
```
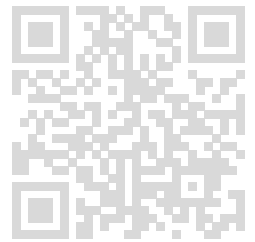
Compilation Details:
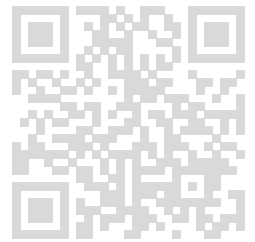
TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

2 1

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

99

Compilation Status: Passed

Execution Time:

0.009s

137. Given a string S consisting of only '(' and ')', print 'yes' if it is balanced otherwise print 'no'.Sample Testcase :INPUT(())OUTPUTyes

Completion Status: Completed
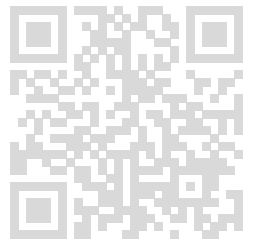
Concepts Included:

strings

array

data structures

Language Used: PYTHON 3

Source Code:

```
s=input()
openingbraces = ['(', '[', '{']
closingbraces = [')', ']', '}']
st = []
for x in s:
if x in openingbraces:
st.append(x)
if x in closingbraces:
```

```
if len(st) > 0:
ob = st.pop()
#print(ob)
else:
print("no")
if (x==')' and ob!='(') or (x==']' and ob!='[') or (x=='{' and ob!='}'):
print ("no")
if len(st)==0:
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

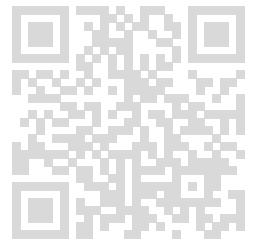**Compilation Status:** Passed

Execution Time:

0.009s

138. Given 2 strings and a number K, check whether they differ exactly by K characters.Input Size : |s| <= 100000(complexity O(nlogn) or O(n))Sample Testcase :INPUTcodekata codeguvi 4OUTPUTyes

Completion Status: Completed

Concepts Included:

array

strings

Language Used: PYTHON 3

Source Code:

```
n,m,k=input().split()
diff=0
for i in range(len(n)):
if n[i]!=m[i]:
diff+=1
if diff == int(k):
print('yes')
else:
print('no')
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

no

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

yes

Compilation Status: Passed

Execution Time:

0.009s

139. Given a string S convert each characters of the string into ASCII values and print the sum of the numbers.Input Size : |s| <= 100000 Sample Testcase :INPUTguviOUTPUT443

Completion Status: Completed

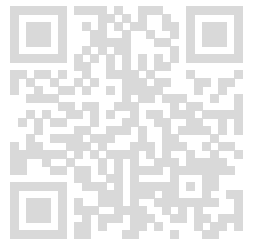**Concepts Included:**

array

strings

Language Used: PYTHON 3

Source Code:

```
s=input()
sum=0
for i in range(len(s)):
res=ord(s[i])
sum=sum+res
print(sum)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

119

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

291

Compilation Status: Passed

Execution Time:

0.015s

140. Given a string S of length N, write a program that would reverse every word in the string.Input Size : 1 <= N <= 100000Sample Testcases :INPUTHello WorldOUTPUTolleH dlroW

Completion Status: Completed
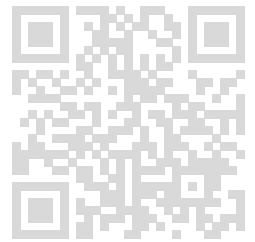
Concepts Included:
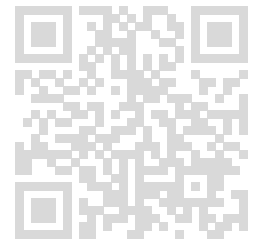
array

strings

Language Used: PYTHON 3

Source Code:

```
S=input()
words=S.split()
#for i in words:
new_word=[i[::-1] for i in words]
```

```
new_sentence=" ".join(new_word)
print(new_sentence)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

gnimmargorP htiw ++C

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

gninraeL si nuF

Compilation Status: Passed

Execution Time:

0.009s

141. Given a string S of length N, find whether the given string is a palindrome using stack or linked list and print 'yes' otherwise print 'no'.Input Size : 1 <= N <= 100000Sample Testcases :INPUTGuviGeekOUTPUTno
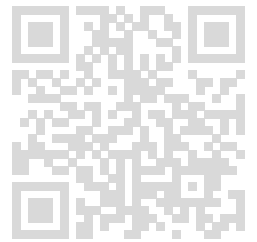
Completion Status: Completed

Concepts Included:

array

strings

data structures

companies

Language Used: PYTHON 3

Source Code:

```
s=input()
ls=[]


for i in s:
adding_value=ls.append(i)


lst_str=''.join(map(str,ls))


str_pop=''
i=0

while (i<len(s)):
str_pop=str_pop+ls.pop()
i=i+1

if (lst_str==str_pop):
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:

yes

Compilation Status: Passed

Execution Time:

0.01s

142. You are provided with an array. Your task is to print the count of subarray whose maximum element is strictly greater than 'k'.

Sample Input:

5 4
1 6 7 8 9

Sample Output:

14

Completion Status: Completed

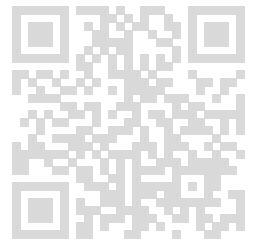**Concepts Included:**

array

24*7-Innovation-Labs

Accolite

Amazon

Citrix

D-E-Shaw

FactSet
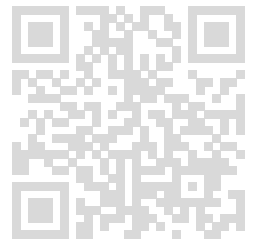
Language Used: PYTHON 3

Source Code:

```python
n, k = map(int, input().split())
arr = list(map(int, input().split()))

count = 0

for i in range(n):
for j in range(i, n):
sub_array = arr[i:j+1]
if max(sub_array) > k:
count += 1

print(count)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

14

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

65

Compilation Status: Passed

Execution Time:

0.01s

143. Check whether the given 4 points form a square or not.Example:INPUT10 1010 2020 2020 10OUTPUTyes
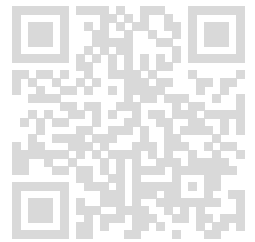
Completion Status: Completed
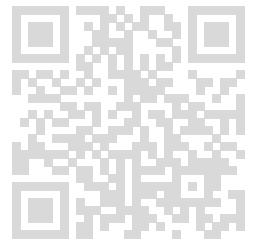
Concepts Included:

array

mathematics

companies

basics

Language Used: PYTHON 3

Source Code:

```python
import math

def dist(x1, y1, x2, y2):
return math.sqrt((x2 - x1)**2 + (y2 - y1)**2)

def angle(x1, y1, x2, y2, x3, y3):
a = dist(x1, y1, x2, y2)
b = dist(x2, y2, x3, y3)
c = dist(x3, y3, x1, y1)
angle = math.degrees(math.acos((a**2 + b**2 - c**2)/(2*a*b)))
return angle

x1, y1 = map(int, input().split())
x2, y2 = map(int, input().split())
x3, y3 = map(int, input().split())
x4, y4 = map(int, input().split())

if dist(x1, y1, x2, y2) == dist(x3, y3, x4, y4) and dist(x1, y1, x3, y3) == dist(x2, y2, x4, y4)
and angle(x1, y1, x2, y2, x4, y4) == 90 and angle(x2, y2, x3, y3, x4, y4) == 90:
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:

no

Compilation Status: Passed

Execution Time:

0.015s

144. Given 2 strings S1 and S2,work on the strings such that both string has the same number of characters.To adjust the length reduce number of exceeding characters from longer string.Sample Testcase :INPUTguvigeeksOUTPUTguvigeek

Completion Status: Completed

Concepts Included:

strings

Language Used: PYTHON 3

Source Code:

```
a,b = (list(map(str,(input().split(' ')))))[:2]
c = len(a)
d = len(b)
if c>d :
print(a[:d]+b)
elif d>c:
print(a+b[:c])
else: print(a+b)
```
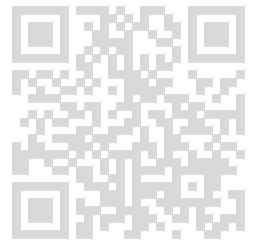
Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

learnpract

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

codiclub

Compilation Status: Passed

Execution Time:

0.01s

145. Given a number N, print the sum of squares of all its digits.Input Size : 1 <= N <= 100000Sample Testcase :INPUT12OUTPUT5

Completion Status: Completed
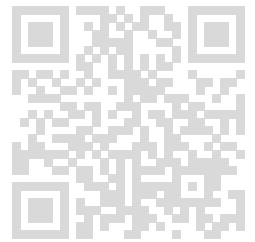
Concepts Included:

array

mathematics

Language Used: PYTHON 3

Source Code:

```
N=int(input())
sum=0
while N>0:
a=N%10
#print(a)
sum=sum+(pow(a,2))
#print("   ",sum)
N=N//10
```

```
print(sum)
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

4

Compilation Status: Passed

Execution Time:

0.009s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

Output:

144

Compilation Status: Passed

Execution Time:
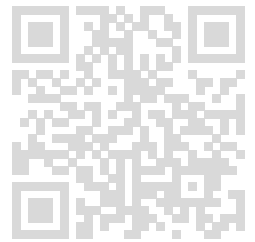
0.01s

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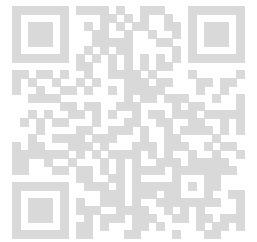
Completion Status: Completed

Concepts Included:

array

mathematics

basics

Language Used: PYTHON 3

Source Code:

```python
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

147. You are given a number 'n' you have to tell whether number is divisible by 8.

Print 1 if it is divisible and 0 if it is not.

Sample Input:

16

Sample Output:

1

Completion Status: Completed

Concepts Included:

bit manipulation

Language Used: PYTHON 3

Source Code:

```
n = int(input())
if n%8==0:
print("1")
else:
print("0")
```

Compilation Details:

TestCase1:

Input:

< hidden >

Expected Output:

< hidden >

Output:

1

Compilation Status: Passed

Execution Time:

0.01s

TestCase2:

Input:

< hidden >

Expected Output:

< hidden >

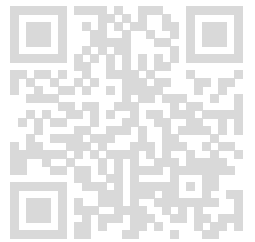**Output:**

0

Compilation Status: Passed

Execution Time:

0.009s