

WEEK-01-MCQ-Variables,Datatypes

1.Which one of the following is the correct way to format the string?

name = "Vijay"

age = 31

print("My name is {} and I am {} years old.".format(name, age))

a) My name is (name) and I am (age) years old.

b) My name is d and I am d years old.

c) My name is Vijay and I am 31 years old.

d) What would be the output if Vijay and 31 years old.

2.Which function converts a value to an integer in Python?

a) str() b) int() c) float() d)

bool()

3.What will be the return type of the input() function?

a) Integer b) String c) Boolean d)Float

4.Which function is used to print a message in Python?

a) print() b) show() c) display() d) echo()

5.The code to get input from the keyboard. (No need to assign to a variable)

Answer:

6.Which of the following is correct syntax to output ‘Hello Python’ in Python?

a) print("Hello Python") b) echo "Hello Python" c) which("Hello Python") d) echo("Hello Python")

7.What will be the output of the following python Code:

mystring='India is my country'

print(mystring[::-2])

a) str b) 'str' c) <class 'str'> d) class str

8.What will be the output of str('Hello')?

a) 'Hello' b) <class 'bool'> c) <class 'int'> d) <class 'str'>

9.Which function would you use to get a Boolean value?

a) str() b) int() c) bool() d) float()

10.Which of the following functions is a built-in function in python language?

a)val() b) print() c) printf() d)scanf()

11. one of the following is the correct extension of the Python file?

a) .pyth b) .python c) .cpp d) .p

12.Which of the following is a correct syntax to output “Hello Python” in Python?

- a)echo “Hello Python” b) print(“Hello Python”)
- c) print: “Hello Python” d)echo(“Hello Python”)

13.What do we use to define a block of code in the Python language?

- a) Indentation b)curly Braces c)Parenthesis d)Key

14.What is the correct syntax to create a variable in Python?

- a) var name : “value” b) name := “value”
- c)var name = “value” d) name = “value”

15.Which Python method can be used to format strings?

- a) string_format() b) format() c) format_string() d) str_format()

WEEK-01-CODING-Variables, Datatypes

1.You went on a tour to Ooty with your friends. As a part of the tour, you went boating with them. For the boat to remain stable, the number of people on one boat is restricted based on the weight of the people. You find that the boatman who is sailing your boat is so much greedy of money. For earning more, he takes too many people to travel in the boat at a time. So you want to check how many people can travel in the boat at a time so that the boat will not drown. Calculate the weight by considering the number of adults and number of children. Assume that an adult weighs 75 kg and children weigh 30 kg each. If the weight is normal, display Boat is stable, else display Boat will drown.

INPUT & OUTPUT FORMAT:

Input consists of 3 integers.

First input corresponds to the weight that the boat can handle.

Second input corresponds to the number of adults.

Third input corresponds to the number of children.

Program:

```
b=int(input())
a=int(input())
c=int(input())
wa=75
wc=30
tw=a*wa
tc=c*wc
tot=tw+tc
if tot <=b:
    print("Boat is stable")
else:
    print("Boat will drown")
```

Output:

	Input	Expected	Got	
✓	340 2 3	Boat is stable	Boat is stable	✓
✓	600 7 4	Boat will drown	Boat will drown	✓

Passed all tests! ✓

2. Alfred buys an old scooter for Rs. X and spends Rs. Y on its repairs. If he sells the scooter for Rs. Z ($Z > X + Y$). Write a program to help Alfred to find his gain percent. Get all the above-mentioned values through the keyboard and find the gain percent.

Input Format:

The first line contains the Rs X

The second line contains Rs Y

The third line contains Rs Z

Sample Input:

10000

250

15000

Sample Output:

46.34 is the gain percent.

For example:

Input	Result
45500 500 60000	30.43 is the gain percent.

Program:

```
x=float(input())
y=float(input())
z=float(input())
tot=x+y
gain=z-tot
g=(gain/tot)*100
print(f'{g:.2f} is the gain percent.')
```

Output:

	Input	Expected	Got	
✓	10000 250 15000	46.34 is the gain percent.	46.34 is the gain percent.	✓
✓	45500 500 60000	30.43 is the gain percent.	30.43 is the gain percent.	✓
✓	5000 0 7000	40.00 is the gain percent.	40.00 is the gain percent.	✓
✓	12500 5000 18000	2.86 is the gain percent.	2.86 is the gain percent.	✓

Passed all tests! ✓

3. In a Logistic the Parcels to be delivered in 4 locations (1st location 20%, 2nd location 40%, 3rd location 30% and 4th location 10%). write a python code to find the total no. of parcels after the delivery in 2 locations . use a format() to print the no of parcels delivered in each location

Input:

250

output:

Total Parcels is 250

1st Location 50 parcels

2nd Location 100 parcels

3rd Location 75 parcels

4th Location 25 parcels

Program:

```
tot=int(input())
```

```

p=[20,40,30,10]
p1=tot*p[0]/100
p2=tot*p[1]/100
p3=tot*p[2]/100
p4=tot*p[3]/100
print("Total Parcels is {}".format(tot))
print("1st Location {} parcels".format(int(p1)))
print("2nd Location {} parcels".format(int(p2)))
print("3rd Location {} parcels".format(int(p3)))
print("4th Location {} parcels".format(int(p4)))

```

Output:

	Input	Expected	Got	
✓	250	Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels	Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels	✓

Passed all tests! ✓

4. Justin is a carpenter who works on an hourly basis. He works in a company where he is paid Rs 50 for an hour on weekdays and Rs 80 for an hour on weekends. He works 10 hrs more on weekdays than weekends. If the salary paid for him is given, write a program to find the number of hours he has worked on weekdays and weekends.

Hint:

If the final result(hrs) are in -ve convert that to +ve using `abs()` function

The `abs()` function returns the absolute value of the given number.

```

number = -20
absolute_number = abs(number)
print(absolute_number)
# Output: 20

```

Sample Input:

450

Sample Output:

weekdays 10.38

weekend 0.38

For example:

Input	Result
450	weekdays 10.38 weekend 0.38

Program:

```
ts=int(input())
a=50
b=80
x=(ts-500)/130
if x<0:
    x=abs(x)
c=x+10
d=x
print(f"weekdays {c:.2f}")
print(f"weekend {d:.2f}")
```

Output:

	Input	Expected	Got	
✓	450	weekdays 10.38 weekend 0.38	weekdays 10.38 weekend 0.38	✓
✓	500	weekdays 10.00 weekend 0.00	weekdays 10.00 weekend 0.00	✓
✓	10000	weekdays 83.08 weekend 73.08	weekdays 83.08 weekend 73.08	✓
✓	6789	weekdays 58.38 weekend 48.38	weekdays 58.38 weekend 48.38	✓

Passed all tests! ✓

5. In a Lab 36% are Dell and 34% Lennovo and 28% are Acer and 2% are Samsung. write a python code to print total systems and brand wise count in the specific format using sep operator.

input: 150

output: Total System:150

Dell:54

Lennovo:51

Acer:42

Samsung:3

Program:

```
t=int(input())
d=(36/100)*t
l=(34/100)*t
a=(28/100)*t
s=(2/100)*t
print(f"Total
System:{t}",f"Dell:{int(d)}",f"Lennovo:{int(l)}",f"Acer:{int(a)}",f"Samsung:{int(s)}",sep='\n')
```

Output:

	Input	Expected	Got	
✓	150	Total System:150 Dell:54 Lennovo:51 Acer:42 Samsung:3	Total System:150 Dell:54 Lennovo:51 Acer:42 Samsung:3	✓

Passed all tests! ✓

6. In many jurisdictions, a small deposit is added to drink containers to encourage people to recycle them. In one particular jurisdiction, drink containers holding one liter or less have a \$0.10 deposit and drink containers holding more than one liter have a \$0.25 deposit. Write a program that reads the number of containers of each size(less and more) from the user. Your program should continue by computing and displaying the refund that will be received for returning those containers. Format the output so that it includes a dollar sign and always displays exactly two decimal places.

Sample Input

10

20

Sample Output

Your total refund will be \$6.00.

For example:

Input	Result
20 20	Your total refund will be \$7.00.

Program:

```
a=int(input())  
b=int(input())
```

x=0.10

y=0.25

```
tot=(a*x+b*y)  
print(f"Your total refund will be ${tot:.2f}.")
```

Output:

	Input	Expected	Got	
✓	20 20	Your total refund will be \$7.00.	Your total refund will be \$7.00.	✓
✓	11 22	Your total refund will be \$6.60.	Your total refund will be \$6.60.	✓
✓	123 200	Your total refund will be \$62.30.	Your total refund will be \$62.30.	✓
✓	76 38	Your total refund will be \$17.10.	Your total refund will be \$17.10.	✓

Passed all tests! ✓

	Input	Expected	Got	
✓	20 20	Your total refund will be \$7.00.	Your total refund will be \$7.00.	✓
✓	11 22	Your total refund will be \$6.60.	Your total refund will be \$6.60.	✓
✓	123 200	Your total refund will be \$62.30.	Your total refund will be \$62.30.	✓
✓	76 38	Your total refund will be \$17.10.	Your total refund will be \$17.10.	✓

Passed all tests! ✓

7. Write a program to convert strings to an integer and float and display its type.

Sample Input:

10

10.9

Sample Output:

10,<class 'int'>

10.9,<class 'float'>

For example:

Input	Result
10	10,<class 'int'>
10.9	10.9,<class 'float'>

Program:

```
a=int(input())
b=float(input())
print(f'{a},{type(a)}')
print(f'{round(b,1)},{type(b)}')
```

Output:

	Input	Expected	Got	
✓	10 10.9	10,<class 'int'> 10.9,<class 'float'>	10,<class 'int'> 10.9,<class 'float'>	✓
✓	12 12.5	12,<class 'int'> 12.5,<class 'float'>	12,<class 'int'> 12.5,<class 'float'>	✓
✓	89 7.56	89,<class 'int'> 7.6,<class 'float'>	89,<class 'int'> 7.6,<class 'float'>	✓
✓	55000 56.2	55000,<class 'int'> 56.2,<class 'float'>	55000,<class 'int'> 56.2,<class 'float'>	✓
✓	2541 2541.679	2541,<class 'int'> 2541.7,<class 'float'>	2541,<class 'int'> 2541.7,<class 'float'>	✓

Passed all tests! ✓

8. Write a program that returns the second last digit of the given number. Second last digit is being referred to the digit in the tens place in the given number.

For example, if the given number is 197, the second last digit is 9.

Note1 - The second last digit should be returned as a positive number. i.e. if the given number is -197, the second last digit is 9.

Note2 - If the given number is a single digit number, then the second last digit does not exist. In such cases, the program should return -1. i.e. if the given number is 5, the second last digit should be returned as -1

For example:

Input	Result
197	9
-197	9
5	-1

Program:

```
inp=int(input())
n=str(abs(inp))
if len(n)<2:
    print("-1")
else:
    r=int(n[-2])
    print(r)
```

Output:

	Input	Expected	Got	
✓	197	9	9	✓
✓	-197	9	9	✓
✓	5	-1	-1	✓

Passed all tests! ✓

9. Ramesh's basic salary is input through the keyboard. His dearness allowance is 40% of his basic salary, and his house rent allowance is 20% of his basic salary. Write a program to calculate his gross salary.

Sample Input:

10000

Sample Output:

16000

For example:

Input	Result
10000	16000

Program:

```
tot = int(input())
```

```
a = tot*40/100
```

```
b = tot*20/100
```

```
c=tot+a+b
```

```
print(c)
```

Output:

	Input	Expected	Got	
✓	10000	16000	16000.0	✓
✓	20000	32000	32000.0	✓
✓	28000	44800	44800.0	✓
✓	5000	8000	8000.0	✓

Passed all tests! ✓

10. In department 54% are boys and 46% are girls and 8% are hostel (boys/girls). write a python code to print total no of boys, girls and hostel students in the specific format using modulo operator.

input: 1500

output: Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120

Program:

```
tot=int(input())
```

```
b=tot*54/100
```

```
g=tot*46/100
```

```
h=tot*8/100
```

```
print(f"Total Students : {tot}, Boys : {int(b)}, Girls : {int(g)}, Hostel : {int(h)}")
```

Output:

	Input	Expected
✓	1500	Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120

Passed all tests! ✓

Got	
Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120	✓

WEEK-02-MCQ-Operators and Formating Output

1.What will be the output of the following statement?

print(15 + 20 / 5 + 3 * 2 - 1)

a. 19

b. 24.0

c.19.0

d.12

2. What is the output of the following expression?

z=2

z**=3

print(z)

a.0

b.3

c. Error

d.8

3. Which is the following is an Arithmetic operator in Python?

1)// (floor division) operator

2)& (binary and) operator

3)- (navigation) operator

4)>> (right shift) operator

a.2

b.1

c.3

d.4

4. What is the output of the following code

x = ["apple", "banana"]

y = ["apple", "banana"]

z = x

print(x is z)

print(x is y)

print(x == y)

a. True

False

True

b. False

False

True

c. True

False

True

d.True

True

5.What will be the value of x in the following Python expression, if the result of that expression is 2?

x>>2

a.1

b.2

c.4

d.8

6. What is the value of the expression $1+2**3*4+12*((100+4)*10-200//10)$?

- a. -23679 b. 12493 c. -24568 d.12273

7. What is the output of the following code

```
x = 5
```

```
y = 3
```

```
print(x == y)
```

- a . 5==3 b. False c. Error d. True

8. What is the value of the expression ?

```
print(100 / 25)
```

```
print(100//25)
```

- a.4.0 b. 4.0 c. 4.0 d.4
4.00 4.0 4 4

9. What is the output of the following code

```
x,y = 8, 2
```

```
print(x ** y)
```

```
print(x // y)
```

- a.64 b.64 c.0 d.64
0 8 64 4

10. What is the output of the following code

```
print(bool(0), bool(3.14159), bool(-3), bool(1.0+1j))
```

- a. False True False True b. True True False True
c. True True False False d. False True True True

11. What is the output of the following code: print 11//2?

- a. 5.0 b. 5.5 c. Error d.5

12. Which of the following is not a valid variable name in Python?

- a. var11 b. 5var c. var_name d. _var

13. What is the two's complement of -44?

- a. 11101011 b. 1011011 c.11010100 d.10110011

14. In the Python statement $x = a + 6 - c - d$:

a and b are _____

a + 6 - c-d is _____

a. operands, an equation

b.operands, an expression

c.terms, a group

d. operators, a statement

15.Which of the following statements assigns the value 35 to the variable x in Python:

a. $x \leftarrow 35$

b. $\text{int } x = 35$

c. $x := 35$

d. $x = 35$

WEEK-02-CODING-Operators and Formatting Output

1. Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7

Program:

```
num = int(input())
l = abs(num)
print(l%10)
```

Output:

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

2. Mr.Ram has been given a problem kindly help him to solve it. The input of the program is either 0 or 1. IF 0 is the input he should display "C" if 1 is the input it should display "D".There is a constraint that Mr. Ram should use either logical operators or arithmetic operators to solve the problem, not anything else.

Hint:Use ASCII values of C and D.

Input Format:

An integer x, $0 \leq x \leq 1$.

Output Format:

output a single character "C" or "D"depending on the value of x.

Input 1:

0

Output 1:

C

Input 2:

1

Output 1:

D

For example:

Input	Result
0	C

Program:

```
x = int(input())
result = chr(67+x)
print(result)
```

Output:

	Input	Expected	Got	
✓	0	C	C	✓
✓	1	D	D	✓

Passed all tests! ✓

3. Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: Tile sign of the input numbers should be ignored.

i.e. if the input numbers are 267 and 154, the sum of last two digits should be 11

if the input numbers are 267 and -154, the sum of last two digits should be 11

if the input numbers are -267 and 154, the sum of last two digits should be 11

if the input numbers are -267 and -154, the sum of last two digits should be 11

For example:

Input	Result
267 154	11
267 -154	11

Program:

```
num1 = int(input())
num2 = int(input())
l1=abs(num1) % 10
l2=abs(num2) % 10
result=l1+l2
print(result)
```

Output:

	Input	Expected	Got	
✓	267 154	11	11	✓
✓	267 -154	11	11	✓

Passed all tests! ✓

4. Complete the program to convert days into years, month and days. (Ignoring leap year and considering 1 month is 30 days)

Sample Test Cases

Test Case 1

Input

375

Output

YEARS: 1 MONTH: 0 DAYS: 10

Test Case 2

Input

200

Output

YEARS: 0 MONTH: 6 DAYS: 20

Program:

```
days = int(input())
year = days // 365
r_days = days % 365
month = r_days // 30
day = r_days % 30

print(f"YEARS: {year} MONTH: {month} DAYS: {day}")
```

Output:

	Input	Expected	Got	
✓	375	YEARS: 1 MONTH: 0 DAYS: 10	YEARS: 1 MONTH: 0 DAYS: 10	✓

Passed all tests! ✓

5. Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number. The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7

Program:

```
num = int(input())
l = abs(num)
print(l%10)
```

Output:

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

6. Write a program to find whether the given input number is Even.

If the given number is even, the function should return 2 else it should return 1.

Note: The number passed to the program can either be negative, positive or zero. Zero should be treated as Even.

For example:

Input	Result
100	2
1001	1

Program:

```
num = int(input())
if num % 2==0:
    result=2
else:
    result=1
print(result)
```

Output:

	Input	Expected	Got	
✓	100	2	2	✓
✓	1001	1	1	✓
✓	0	2	2	✓

Passed all tests! ✓

7. Note: Don't use if-else. Operators alone must be used .

A team from the Rotract club had planned to conduct a rally to create awareness among the Coimbatore people to donate blood. They conducted the rally successfully. Many of the Coimbatore people realized it and came forward to donate their blood to nearby blood banks. The eligibility criteria for donating blood are people should be above or equal to 18 and his/ her weight should be above 40. There was a huge crowd and staff in the blood bank found it difficult to manage the crowd. So they decided to keep a system and ask the people to enter their age and weight in the system. If a person is eligible he/she will be allowed inside.

Write a program and feed it to the system to find whether a person is eligible or not.

Input Format:

Input consists of two integers that correspond to the age and weight of a person respectively.

Output Format:

Display True(IF ELIGIBLE)

Display False (if not eligible)

Sample Input

19

45

Sample Output

True

For example:

Input	Result
18	False
40	

Program:

```
age = int(input())
weight = int(input())
result = (age>=18) and (weight > 40)
print(result)
```

Output:

	Input	Expected	Got	
✓	19 45	True	True	✓
✓	18 40	False	False	✓
✓	18 42	True	True	✓
✓	16 45	False	False	✓

Passed all tests! ✓

8. Write a python program that takes a integer between 0 and 15 as input and displays the number of '1' s in its binary form.(Hint:use python bitwise operator.

Sample Input

3

Sample Output:

2

Explanation:

The binary representation of 3 is 011, hence there are 2 ones in it. so the output is 2.

For example:

Input	Result
3	2

Program:

```
num = int(input())
if 0 <= num <=15:
    count = bin(num).count('1')
    print(count)
else:
    print("Give input between 0 and 15")
```

Output:

	Input	Expected	Got	
✓	3	2	2	✓
✓	5	2	2	✓
✓	15	4	4	✓

Passed all tests! ✓

9. In the 1800s, the battle of Troy was led by Hercules. He was a superstitious person. He believed that his crew can win the battle only if the total count of the weapons in hand is in multiple of 3 and the soldiers are in an even number of count. Given the total number of weapons and the soldier's count, Find whether the battle can be won or not according to Hercules's belief. If the battle can be won print True otherwise print False.

Input format:

Line 1 has the total number of weapons

Line 2 has the total number of Soldiers.

Output Format:

If the battle can be won print True otherwise print False.

Sample Input:

32

43

Sample Output:

False

For example:

Input	Result
32	False
43	

Program:

```
a = int(input())
b = int(input())
w = (a % 3 == 0)
s = (b % 2 == 0)
result = w and s
print(result)
```

Output:

	Input	Expected	Got	
✓	32 43	False	False	✓
✓	273 7890	True	True	✓
✓	800 4590	False	False	✓
✓	6789 32996	True	True	✓

Passed all tests! ✓

10. An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each gizmo weighs 112 grams. Write a program that reads the number of widgets and the number of gizmos from the user. Then your program should compute and display the total weight of the parts.

Sample Input:

10

20

Sample Output:

The total weight of all these widgets and gizmos is 2990 grams.

Program:

```
a = int(input())
b = int(input())
total = (a*75)+(b*112)
print(f"The total weight of all these widgets and gizmos is {total} grams.")
```

Output:

	Input	Expected
✓	10 20	The total weight of all these widgets and gizmos is 2990 grams.

Passed all tests! ✓

Got	
The total weight of all these widgets and gizmos is 2990 grams.	✓

WEEK-03-MCQ-Selection control

1.What is the output of the following snippet if 5 is given as the input?

```
c = input("Enter a number: ")  
print(c * 3)
```

a.2

b.8

c.15

d.555

2. What is the output of the given below program?

```
if 1 + 3 == 7:  
    print("Hello")  
else:  
    print("Know Program")
```

a. Know Program

b. Hello

c. Compiled Successfully, No Output

d. Error

3. What will be the output for the following code?

```
if False:  
    print("1001")  
else:  
    print("2002")
```

a. 1001

b. 2002

c. syntax error

4. What is the output of the following code snippet?

```
a = "Hi"  
b = "Arjuna"  
c = "Bhimaa"  
print("Hi", a, b, c)
```

a. Hi Arjuna Hi Bhimaa

b. Hi Arjuna Bhimaa Hi

c. Hi Arjuna Bhimaa

d. Hi Hi Arjuna Bhimaa

5. What is the output of the given below program?

```
a = 25  
if a > 15:  
    print("Hi")  
    if a <= 30:  
        print("Hello")  
Else:
```

```
print("Know Program")
```

- a. Hi Know Program b. Hello Know Program c. Hello d. Hi Hello

6. Python supports _____ types of control structures.

- a.3 b.1 c.4 d.2

7. x,y=1,2

```
if(x or y):  
    print("1")  
else:  
    print("0")
```

- a. Runtime error b. 1 c. Compile time error d. 0

8. a="REC"

```
If a in ("rec"):  
    print(a)  
print(a)
```

- a. false REC b. No output REC c. REC d. REC REC

9. What will be the output for the following code?

```
if 1-1:  
    print("python")  
else:  
    print("0 is false")
```

- a. Error b. python c. 0 is false

10. What will be output for the following code?

```
if 3 == 3:  
    print("Python is easy!")
```

- a. Python is easy! b. Error c. NO OUTPUT

11. Correct syntax of writing 'simple if' statement is _____

- a. if (condition) statements b. if condition -- statements
c. if condition statements d. if condition : statements

12. Number of elif in a program is dependent on the _____

- a. number of variables in a program
- b. All the Above
- c. number of conditions to be checked
- d. number of loops in a program

13. What keyword would you use to add an alternative condition to an if statement?

- a. elif
- b. else if
- c. elseif

14. Can we write if/else into one line in python?

- a. No
- b. Yes

15. Ahaana wants to make a fun program , if user enters any number a “Good” or “funny” message will appear . She is confused that which is the most suitable control to be used to make such program. Help her to choose correct option.

- a. if elif
- b. If
- c. if else
- d. Nested if

CODING:WEEK-3-CODING Selection control

1. Write a Python program that accepts three parameters. The first parameter is an integer. The second is one of the following mathematical operators: +, -, /, or *. The third parameter will also be an integer.

The function should perform a calculation and return the results. For example, if the function is passed 6 and 4, it should return 24.

Sample Input Format:

11

+

14

Sample Output Format:

25

Coding:

```
n1=int(input())
op=input().strip()
n2=int(input())
if op == '+':
    res=n1+n2
elif op == '-':
    res=n1-n2
elif op == '*':
    res=n1*n2
elif op == '/':
    if n2 !=0:
        res=n1/n2
    else:
        res="Division by zero is not valid"
else:
```

```

res="Invalid operator"

print(res)

```

Output:

	Input	Expected	Got	
✓	11 + 14	25	25	✓
✓	45 - 50	-5	-5	✓
✓	12 * 100	1200	1200	✓
✓	18 / 2	9.0	9.0	✓

Passed all tests! ✓

2. An automorphic number is a number whose square ends with the number itself.
 For example, 5 is an automorphic number because $5*5 = 25$. The last digit is 5 which same as the given number.

If it is an automorphic number display “Automorphic” else display “Not Automorphic”.

Input Format:

Take a Integer from Keyboard

Output Format:

Print Automorphic if given number is Automorphic number, otherwise Not Automorphic

Example input:

5

Output:

Automorphic

Example input:

25

Output:

Automorphic

Example input:

7

Output:

Not Automorphic

Coding:

```
num=int(input())
sq=num**2
a=str(num)
b=str(sq)
if b.endswith(a):
    print("Automorphic")
else:
    print("Not Automorphic")
```

Output:

	Input	Expected	Got	
✓	5	Automorphic	Automorphic	✓
✓	625	Automorphic	Automorphic	✓
✓	7	Not Automorphic	Not Automorphic	✓

Passed all tests! ✓

3. Write a program to calculate and print the Electricity bill where the unit consumed by the user is given from test case. It prints the total amount the customer has to pay. The charge are as follows:

Unit	Charge / Unit
Upto 199	@1.20
200 and above but less than 400	@1.50
400 and above but less than 600	@1.80
600 and above	@2.00

If bill exceeds Rs.400 then a surcharge of 15% will be charged and the minimum bill should be of Rs.100/-

Sample Test Cases

Test Case 1

Input

50

Output

100.00

Test Case 2

Input

300

Output

517.50

For example:

input	Result
100.00	120.00
500	1035.00

Coding:

```
a=float(input())
```

```
if a<= 199:
```

```
    b=a*1.20
```

```
elif 200 <= a <400:
```

```
    b=a*1.50
```

```
elif 400 <= a < 600:
```

```
    b=a*1.80
```

```
else:
```

```
    b=a*2.00
```

```
if b > 400:
```

```
    b+=b*0.15
```

```
if b< 100:
```

```
    b=100
```

```
print(f"{b:.2f}")
```

Output:

	Input	Expected	Got	
✓	50	100.00	100.00	✓
✓	100.00	120.00	120.00	✓
✓	500	1035.00	1035.00	✓
✓	700	1610.00	1610.00	✓

Passed all tests! ✓

4. The Chinese zodiac assigns animals to years in a 12 year cycle. One 12 year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the dragon, and 1999 being another year of the hare.

Year Animal

2000 Dragon

2001 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2008 Rat

2009 Ox

2010 Tiger

2011 Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

2010

Sample Output 1

2010 is the year of the Tiger.

Sample Input 2

2020

Sample Output 2

2020 is the year of the Rat.

Coding:

```
animals=["Dragon","Snake","Horse","Sheep","Monkey","Rooster","Dog","Pig","Rat","Ox","Tiger","Hare"]  
year=int(input())  
i=(year-2000)%12  
animal=animals[i]  
print(f"{year} is the year of the {animal}.")
```

Output:

	Input	Expected	Got	
✓	2010	2010 is the year of the Tiger.	2010 is the year of the Tiger.	✓
✓	2020	2020 is the year of the Rat.	2020 is the year of the Rat.	✓

Passed all tests! ✓

5. Find whether the given number is a Harshad number or not. Note that Harshad number is an integer that is divisible by the sum of its digits.

INPUT & OUTPUT FORMAT:

Input consists of 1 integer. If the given number is a Harshad Number, display "Harshad Number" or display "Not Harshad Number".

SAMPLE INPUT:

1729

SAMPLE OUTPUT:

Harshad Number

Explanation: 1729 is divisible by 19(1+7+2+9), hence 1729 is a Harshad Number

Coding:

```
num=int(input())  
s=str(num)  
x=sum(int(digit) for digit in s)  
if x==0:  
    print("Not Harshad Number")  
else:  
    if num % x==0:  
        print("Harshad Number")
```

```

else:
    print("Not Harshad Number")

```

Output:

	Input	Expected	Got	
✓	1729	Harshad Number	Harshad Number	✓
✓	64	Not Harshad Number	Not Harshad Number	✓

Passed all tests! ✓

6. Write a program that accepts 5 inputs and returns the count of how many of those 5 are odd.

For example,

If the five inputs are 12, 17, 19, 14, and 115, there are three odd numbers 17, 19 and 115. So, the program must return 3.

Similarly,

If the five inputs are 15, 0, -12, 19, and 28, there are two odd numbers 15 and 19. So, the program must return 2.

Observe that zero is considered an even number.

For example:

Input	Result
12	3
17	
19	
14	
115	
15	2
0	
-12	
19	

Input	Result
28	

	Input	Expected	Got	
✓	12 17 19 14 115	3	3	✓
✓	15 0 -12 19 28	2	2	✓

Passed all tests! ✓

7. A certain type of steel is used to test and give grade according to the following conditions.

1. Hardness of the steel must be greater than 50
2. Carbon content of the steel must be less than 0.7
3. Tensile strength must be greater than 5600

The grades awarded are as follows:

- Grade is 10 if all three conditions are met
- Grade is 9 if conditions (1) and (2) are met
- Grade is 8 if conditions (2) and (3) are met
- Grade is 7 if conditions (1) and (3) are met
- Grade is 6 if only one condition is met
- Grade is 5 if none of the three conditions are met

Write a program to display the grade of the steel, based on the values of hardness, carbon content and tensile strength of the steel, given by the user.

Input

53

0.6

5602

Output:

10

Coding:

```
a=float(input())
b=float(input())
c=float(input())
g=5
c1=a > 50
c2=b < 0.7
c3=c > 5600
if c1 and c2 and c3:
    g=10
elif c1 and c2:
    g=9
elif c2 and c3:
    g=8
elif c1 and c3:
    g=7
elif c1 or c2 or c3:
    g=6
print(g)
```

Output:

	Input	Expected	Got	
✓	53 0.6 5602	10	10	✓
✓	45 0 4500	6	6	✓

Passed all tests! ✓

8. Write a program to find the sum of the series $1 + 11 + 111 + 1111 + \dots + n$ terms (n will be given as input from the user and sum will be the output)

Sample Test Cases

Test Case 1

Input

4

Output

1234

Test Case 2

Input

6

Output

123456

Coding:

```
n=int(input())
a=0
b=1
for i in range(n):
    a+=b
    b=b*10+1
print(a)
```

Output:

	Input	Expected	Got	
✓	4	1234	1234	✓
✓	6	123456	123456	✓

Passed all tests! ✓

9. Write a program to determine the type of berth when the seat / berth number in the train is given.



Input Format:

Input consists of a single integer. Assume that the range of input is between 1 and 72.

Output Format:

Output consists of a single string. [Upper or Middle or Lower or Side Lower or Side Upper]

Sample Input 1:

9

Sample Output 1:

Lower Berth

Coding:

```
num=int(input())
if 1 <= num <=72:
    rem=num%8
    if rem==0:
        rem=8
    if rem in (1,2):
        res="Lower Berth"
    elif rem in (3,4):
        res="Middle Berth"
    elif rem in (5,6):
        res="Upper Berth"
    elif rem ==7:
        res="Side Lower Berth"
    elif rem==8:
        res= "Side Upper Berth"
    else:
        res="Invalid num"
print(res)
```

Output:

	Input	Expected	Got	
✓	9	Lower Berth	Lower Berth	✓
✓	72	Side Upper Berth	Side Upper Berth	✓

Passed all tests! ✓

10. Given an integer N, check whether N the given number can be made a perfect square after adding 1 to it.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

For example:

Input	Result
24	Yes

Coding:

```
import math
n=int(input())
m=n+1
r=int(math.isqrt(m))
if r*r==m:
    print("Yes")
else:
    print("No")
```

Output:

	Input	Expected	Got	
✓	24	Yes	Yes	✓
✓	26	No	No	✓

Passed all tests! ✓

WEEK-04-MCQ-Iteration control

1. True= False

```
while(True):  
    print(True)  
    break
```

What is the output of the following?

- a. No output b. Syntax Error c. True d. False

2. Which is a counter-controlled in python?

- a.for b.do-while c.while d.switch

3. A while loop in python is used for what type of iteration?

- a.discriminant b.indefinite c.definite d.indiscriminant

4. Predict the output of the program?

```
for x in range(4):  
    if x == 3: break  
    print(x)  
else:  
    print("Finally finished!")
```

What is the output of the following?

- a.00110 b.01110 c.00120 d.0120 e.ERROR

5. Predict the output of the program?

```
for x in range(4):  
    if x == 3: break  
    print(x)  
else:  
    print("Finally finished!")
```

Answer: ERROR

```
6.i=1  
while True:  
    if i%0o7==0:  
        break  
    print(i)  
    i+=1
```

Predict the output of the following?

- a. 1 2 3 4 5 6 7 b. 7 c. 1 2 3 4 5 6 d. 0 1 2 3 4 5 6 7

7. Which of the following is a loop in python?

- a. For b.Break c.Do-While d.If-Else

8. count = 0

```
while(True):  
    if count % 3 == 0:  
        print(count, end = " ")  
    if(count > 18):  
        break;  
    count += 1
```

Predict the output of the program?

- a. 0 3 6 9 12 15 b. Compilation error c. 0 3 9 12 18 d. 0 3 6 9 12 15 18

9. For loop in python is

- a. Exit Control Loop b. Entry Control Loop
c. Multi Control Loop d. Simple Loop

10. Which is a counter-controlled in python?

- a.while b.switch c.do-while d.for

11. Which of the following is a loop in python?

- a. Break b.Do-While c.For d.If-Else

12. numbers = (8, 9, 11, 20)

a = 1

for num in numbers:

 a = a * num print(a)

Predict the output of the program?

Answer:

13. For loop follows which principle?

- a. Don't Repeat Yourself (DRY)
- b. Single responsibility
- c. You Aren't Going to Need It(YAGNI)
- d. Open/closed

14. While loop can execute a set of statements till

- a. The condition starts executing
- b. The condition is True
- c. The condition stops executing
- d. The condition is False

15. How many times the loop run?

for i in range(-3):

 print(i)

Answer: 0

WEEK-04-CODING-Iteration control

1.A strobogrammatic number is a number that looks the same when rotated 180 degrees (looked at upside down).

Write a program to determine if a number is strobogrammatic. The number is represented as a string.

Example 1:

Input:

69

Output:

true

Example 2:

Input:

88

Output:

true

Example 3:

Input:

962

Output:

false

Example 4:

Input:

1

Output:

true

Coding:

```
x={'0' : '0', '1': '1', '6' : '9', '8': '8' , '9' : '6'}
num = input()
l,r=0,len(num) - 1
res=True
while l <= r:
    if num[l] not in x or x[num[l]] != num[r]:
        res=False
        break
    l +=1
    r -=1
if res:
    print("true")
else:
    print("false")
```

Output:

	Input	Expected	Got	
✓	69	true	true	✓
✓	88	true	true	✓

Passed all tests! ✓

2. Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: $2 \leq$ starting number of the range \leq ending number of the range ≤ 7919

Example1: If the starting and ending number of the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20, namely (2, 3, 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

Coding:

```
s = int(input())
e = int(input())
if 2 <= s <= e <= 7919:
    count = 0
    for num in range(s,e + 1):
        if num > 1:
            p=True
            if num <=3:
                p= True
            elif num % 2 == 0 or num % 3 ==0:
                p=False
            else:
                i = 5
                while i * i <= num:
                    if num % i == 0 or num % (i + 2) ==0:
                        p= False
                        break
                    i +=6
            if p:
                count += 1
    print(count)
```

Output:

	Input	Expected	Got	
✓	2 20	8	8	✓
✓	700 725	3	3	✓

Passed all tests! ✓

3. Write python program to print the following pattern based on input size.

Input:

3

Output:

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

For example:

Input	Result
4	1 2 3 4 5 6 7 8 9 10

Coding:

```
def show(s):  
    a=1  
    for i in range(1,s+1):  
        for j in range(s-i):  
            print(end=" ")  
  
        for j in range(i):  
            print(a,end=' ')  
            a +=1  
        print()  
  
s=int(input())  
show(s)
```

Output:

	Input	Expected	Got	
✓	3	1 2 3 4 5 6	1 2 3 4 5 6	✓
✓	4	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	✓

Passed all tests! ✓

4. An e-commerce company plans to give their customers a special discount for Christmas. They are planning to offer a flat discount. The discount value is calculated as the sum of all the prime digits in the total bill amount.

Write an algorithm to find the discount value for the given total bill amount.

Input

The input consists of an integer order value, representing the total bill amount.

Output

Print an integer representing the discount value for the given total bill amount.

Example Input

578

Output

12

Explanation:

Since 5 and 7 are the prime digits, then sum of $5+7=12$

Coding:

```
num={2,3,5,7}
i=input()
res=0
for char in i:
    if char.isdigit():
        d=int(char)
        if d in num:
            res +=d
print(res)
```

Output:

	Input	Expected	Got	
✓	578	12	12	✓
✓	456	5	5	✓
✓	7032	12	12	✓

Passed all tests! ✓

5. You are choreographing a circus show with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive direction.
- The first kangaroo starts at position x_1 and moves at a speed v_1 meters per jump.
 - The second kangaroo starts at position x_2 and moves at a speed of v_2 meters per jump and $x_2 > x_1$
 - You have to figure out to get both kangaroos at the same position at the same time as part of the show before k jumps. If it is possible, return YES, otherwise return NO.

Input Format:

x_1 -position of kangaroo1
 v_1 -Speed of kangaroo1
 x_2 -position of kangaroo2
 v_2 -Speed of kangaroo2
 k -jumps

Output Format:

Both kangaroos are at the same position within k jumps, YES, otherwise NO.

For example:

Input	Result
0	YES
3	
4	
2	
6	

Coding:

```
a1=int(input())
b1=int(input())
a2=int(input())
b2=int(input())
c=int(input())

if b1 == b2:
    if a1 == a2:
        print("YES")
    else:
        print("NO")
else:
    if (a2-a1) % (b1-b2) == 0:
        j = (a2-a1) // (b1-b2)
        if j >= 0 and j <= c:
            print("YES")
        else:
            print("NO")
    else:
        print("NO")
```

Output:

	Input	Expected	Got	
✓	0 3 4 2 6	YES	YES	✓
✓	0 3 2 4 8	NO	NO	✓

Passed all tests! ✓

6. Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number ≥ 1 and ≤ 25000 .

Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-repeated digit '9' in this number

If the given number is 1015, the program should return 2 because there are 2 non-repeated digits in this number, '0', and '5'.

If the given number is 108, the program should return 3 because there are 3 non-repeated digits in this number, '1', '0', and '8'.

If the given number is 22, the function should return 0 because there are NO non-repeated digits in this number.

For example:

Input	Result
292	1
1015	2
108	3
22	0

Coding:

```
def show(n):
    s=str(n)
    count={}
    for d in s:
        if d in count:
            count[d] +=1
        else:
            count[d] =1

    a=sum(1 for c in count.values() if c==1)
    return a

i=int(input())
print(show(i))
```

Output:

	Input	Expected	Got	
✓	292	1	1	✓
✓	1015	2	2	✓
✓	108	3	3	✓
✓	22	0	0	✓

Passed all tests! ✓

7. Let's print a chessboard!

Write a program that takes input:

Integer N(represents the rows and columns of a chessboard) and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

2

W

Output:

WB

BW

Coding:

```
n=int(input())
c=input()
if c == 'W':
    w='W'
    b='B'
else:
    w='B'
    b='W'
for i in range(n):
    row=[]
    for j in range(n):
        if (i+j)%2 == 0:
            row.append(w)
        else:
            row.append(b)
    print("".join(row))
```

Output:

	Input	Expected	Got	
✓	2 W	WB BW	WB BW	✓
✓	3 B	BWB WBW BWB	BWB WBW BWB	✓

Passed all tests! ✓

8. Given an integer N, check whether N the given number can be made a perfect square after adding to it.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

For example:

Input	Result
24	Yes

Coding:

```
n=int(input())
n= n+1
m=int(n**0.5)
p = m*m
if p==n:
    print("Yes")
else:
    print("No")
```

Output:

	Input	Expected	Got	
✓	24	Yes	Yes	✓
✓	26	No	No	✓

Passed all tests! ✓

9. Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: $2 \leq$ starting number of the range \leq ending number of the range ≤ 7919

Example1: If the starting and ending number of the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20. namely (2, 3, 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

For example:

Input	Result
2 20	8
700 725	3

Coding:

```
s= int(input())
e= int(input())
if 2 <= s <= e <= 7919:
    c=0
    for num in range(s,e+1):
        if num > 1:
            p=True
            if num <= 3:
                p=True
            elif num % 2 ==0 or num % 3 ==0:
                p = False
            else:
                i=5
                while i * i <= num:
                    if num % i ==0 or num % (i + 2)==0:
                        p= False
                        break
                    i +=6
            if p:
                c +=1
print(c)
```

Output:

	Input	Expected	Got	
✓	2 20	8	8	✓
✓	700 725	3	3	✓

Passed all tests! ✓

10. Write a program that given an integer 'n', prints the number of integers that are less than or equal to 'n' and co-prime to 'n'

Two integers a and b are said to be relatively prime or co-prime if the only positive integer that evenly divides both of them is 1. That is, the only common positive factor of the two numbers is 1. This is equivalent to their greatest common divisor being 1.

Input Format:

One line containing the value of 'n', where $1 \leq n \leq 10,000$

Output Format:

One line containing the number of integers that are co-prime to n and less than or equal to 'n'

Sample Test Cases

Test Case 1

Input
10

Output
4

Test Case 2

Input
23

Output
22

Test Case 3

Input
11

Output
10

Coding:

```
def show(n):
    res = n
    p=2
    while p*p <= n:
        if n % p == 0:
            while n%p ==0:
                n //=p
                res -= res //p
        p += 1
    if n > 1:
        res -= res //n
    return res
n=int(input())
print(show(n))
```

Output:

	Input	Expected	Got	
✓	10	4	4	✓
✓	23	22	22	✓
✓	11	10	10	✓

Passed all tests! ✓

WEEK-05-MCQ-Functions-BuiltIN, User Defined, Recursion

1. Python function always returns a value

Select one: a)True b)False

2. The return statement in function is used to _____.

- a. returns the control to the calling function
- b. Both return value and returns the control to the calling function
- c. None of the mentioned
- d. return value

3. Which of the following items are present in the function header?

- a. B. parameters list
- b. A. function name
- c. Both A and B
- d. return value

4. The process of dividing a computer program into separate independent blocks of code with specific functionalities is known as _____.

- a. Programming
- b. More Programming
- c. Modular Programming
- d. Step Programming

5. Functions which do not return any value is called _____.

- a. default function
- b. null function
- c. zero function
- d. void function

6. What is the output of the following function call?

```
def fun1(num):  
    return num + 25  
fun1(5)  
print(num)
```

- a.25
- b.5
- c.NameError
- d.30

7. What will be the output of the following Python code?

```
def maximum(x, y):  
    if x > y:  
        return x  
    elif x == y:  
        return 'The numbers are equal'
```

```
    else:  
        return y  
print(maximum(2, 3))
```

- a. None of the mentioned b. 3 c. 2 d. The numbers are equal

8. What will be the output of the following Python code?

```
def printMax(a, b):  
    if a > b:  
        print(a, 'is maximum')  
    elif a == b:  
        print(a, 'is equal to', b)  
    else:  
        print(b, 'is maximum')  
printMax(3, 4)
```

- a. 3 b. 4 c. 4 is maximum d. None of the mentioned

9. Function defined to achieve some task as per the programmer's requirement is called a

- a. All of the mentioned b. library function
c. user defined function d. built in functions

10. The function can be called in the program by writing function name followed by _____.

- a. { } b. () c. None of the mentioned d. []

11. Which of the following is not the scope of variable?

- a. Local b. Global c. Outside d. None of the mentioned

12. What is the output of the following function call?

```
def outer_fun(a, b):  
    def inner_fun(c, d):  
        return c + d  
    return inner_fun(a, b)  
  
    return a  
result = outer_fun(5, 10)      print(result)
```

- a. 5 b. Syntax Error c. (15,5) d. 15

13. Choose the correct statement

- a. We can create function with argument(s) and no return value.
- b. We can create function with no argument and with return value(s)
- c. We can create function with no argument and no return value.
- d. All of the mentioned**

14. Which of the following statement is a function call?

- a. sum()**
- b. function sum()
- c. call sum()
- d. def sum()

15. What is the output of the following function call?

```
def fun1(name, age=20):  
    print(name, age)
```

```
fun1('Emma', 25)
```

- a. age
- b. name
- c. Emma 25**
- d. Emma
20

WEEK-05-CODING-Functions-User Defined

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

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

Task:

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

return ugly if it is ugly, else return not ugly

Hint:

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

For example:

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

Coding:

```
def checkUgly(n):
    if n<=0:
        return "not ugly"

    for p in [2,3,5]:
        while n % p ==0:
            n //= p
    return "ugly" if n==1 else "not ugly"
```

Output:

	Test	Expected	Got	
✓	print(checkUgly(6))	ugly	ugly	✓
✓	print(checkUgly(21))	not ugly	not ugly	✓

Passed all tests! ✓

2. Write a function that returns the value of $a+aa+aaa+aaaa$ with a given digit as the value of a .

Suppose the following input is supplied to the program:

9

Then, the output should be:

$9+99+999+9999=11106$

Sample Input Format:

9

Sample Output format:

11106

For example:

Test	Result
<code>print(Summation(8))</code>	9872

Coding:

```
def Summation(n):
    a=str(n)
    t1=int(a)
    t2=int(a*2)
    t3=int(a*3)
    t4=int(a*4)

    total = t1+t2+t3+t4

    return total
```

Output:

	Test	Expected	Got	
✓	<code>print(Summation(8))</code>	9872	9872	✓
✓	<code>print(Summation(10))</code>	10203040	10203040	✓

Passed all tests! ✓

3. A strobogrammatic number is a number that looks the same when rotated 180 degrees (looked at upside down).

Write a program to determine if a number is strobogrammatic. The number is represented as a string.

Example 1:

Input:

69

Output:

true

Example 2:

Input:

88

Output:

true

Example 3:

Input:

962

Output:

false

Example 4:

Input:

1

Output:

true

For example:

Test	Result
print(Strobogrammatic(69))	true
print(Strobogrammatic(962))	false

Coding:

```
def Strobogrammatic(n: int) -> bool:  
    s={'0':'0', '1':'1', '6':'9', '8':'8', '9':'6'}
```

```

n=str(n)
l,r = 0 , len(n) - 1
while l <= r:
    if n[l] not in s or n[r] not in s:
        return 'false'
    if s[n[l]] != n[r]:
        return 'false'
    l +=1
    r -=1
return 'true'

```

Output:

	Test	Expected	Got	
✓	print(Strobogrammatic(69))	true	true	✓
✓	print(Strobogrammatic(88))	true	true	✓
✓	print(Strobogrammatic(962))	false	false	✓

Passed all tests! ✓

4. An e-commerce company plans to give their customers a special discount for Christmas. They are planning to offer a flat discount. The discount value is calculated as the sum of all the prime digits in the total bill amount.

Write an algorithm to find the discount value for the given total bill amount.

Constraints

$1 \leq \text{orderValue} < 10e100000$

Input

The input consists of an integer `orderValue`, representing the total bill amount.

Output

Print an integer representing the discount value for the given total bill amount.

Example Input

578

Output

12

For example:

Test	Result
print(christmasDiscount(578))	12

Coding:

```
def christmasDiscount(n):
    p={'2','3','5','7'}
    v=0
    for d in str(n):
        if d in p:
            v +=int(d)
    return v

#print(christmasDiscount(n))
```

Output:

	Test	Expected	Got	
✓	print(christmasDiscount(578))	12	12	✓

Passed all tests! ✓

5. complete function to implement coin change making problem i.e. finding the minimum number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

return the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value

Coding:

```
def coinChange(n):
    if n==0:
        return 0
    dp = [float('inf')]*(n+1)
    dp[0]=0
    c=[1,2,3,4]

    for a in range(1, n+1):
        for c1 in c:
            dp[a] = min(dp[a],dp[a-c1] + 1)
    return dp[n] if dp[n] != float('inf') else -1
```

Output:

	Test	Expected	Got	
✓	print(coinChange(16))	4	4	✓

Passed all tests! ✓

6. Complete a Recursive Function to find if a given number N can be expressed as a sum of two prime numbers.

Note: YOU MUST OPTIMIZE the logic to find whether a number is prime or not, as very large prime numbers are provided as input. If the logic is not optimized your program will NOT get executed within the given time limit.

Input Format:

First line contains number N.

Output Format:

Return either yes or no.

Boundary Conditions / Constraints:

$3 \leq N \leq 10^9$

Example Input/Output 1:

Input:

20

Output:

yes

Input:

23

Output:

no

Explanation:

20 can be expressed as 17+3

23 cannot be expressed as sum of two primes

For example:

Test	Result
print(checkPrimeSum(20))	yes
print(checkPrimeSum(23))	no

Coding:

```
import math
def ss(l):
    s=[True]*(l+1)
    s[0]=s[1]=False
    for st in range(2, int(math.sqrt(l))+1):
        if s[st]:
            for m in range(st*st,l+1,st):
                s[m]=False
    return {num for num,p in enumerate(s) if p}

def checkPrimeSum(n):
    if n<4:
        return "no"
    a=ss(n)
    for b in a:
        if(n-b) in a:
            return "yes"
```

Output:

	Test	Expected	Got	
✓	print(checkPrimeSum(20))	yes	yes	✓

Passed all tests! ✓

7. Complete the recursive function to return Binary Equivalent of an Integer using Recursion.

Sample Test Cases

Test Case 1

Input

10

Output

1010

Test Case 2

Input

257

Output

100000001

For example:

Test	Result
print(binayNumber(10))	1010
print(binayNumber(257))	100000001

Coding:

```
def binayNumber(n):
    if n==0:
        return "0"
    elif n==1:
        return "1"
    return binayNumber(n//2)+str(n%2)
```

Output:

	Test	Expected	Got	
✓	<code>print(binayNumber(10))</code>	1010	1010	✓
✓	<code>print(binayNumber(257))</code>	100000001	100000001	✓

Passed all tests! ✓

8. The notion of a palindrome was introduced previously. In this exercise you will write a recursive function that determines whether or not a string is a palindrome. The empty string is a palindrome, as is any string containing only one character. Any longer string is a palindrome if its first and last characters match, and if the string formed by removing the first and last characters is also a palindrome.

Write a program that reads a string from the user and uses your recursive function to determine whether or not it is a palindrome. Then your program should display an appropriate message for the user.

Sample Input

malayalam

Sample Output

That was a palindrome!

Sample Input

madan

Sample Output

That is not a palindrome.

Coding:

```
def isPalindrome(s):
    # Base case: The empty string is a palindrome. So is a string containing only 1 character.
    if len(s) <= 1:
        return True
    if s[0] == s[-1]:
        return isPalindrome(s[1:-1])
    else:
        return False

    # Recursive case: The string is a palindrome only if the first and last characters match,
    and
    # the rest of the string is a palindrome

# Check whether or not a string entered by the user is a palindrome
# Read the string from the user
```

```

line = input().strip()
# Check its status and display the result
if isPalindrome(line):
    print("That was a palindrome!")

else:
    print("That is not a palindrome.")

```

Output:

	Input	Expected	Got	
9.	Euclid was a Greek mathematician who lived approximately 2,300 years ago. His algorithm for computing the greatest common divisor of two positive integers, a and b, is both efficient and recursive. It is outlined below.	Passed all tests! ✓	Passed all tests! ✓	

If b is 0 then ✓

 return a

Else

 Set c equal to the remainder when a is divided by b

 Return the greatest common divisor of b and c

Write a Recursive function that implements Euclid's algorithm and uses it to determine the greatest common divisor of two integers entered by the user. Test your program with some very large integers. The result will be computed quickly, even for huge numbers consisting of hundreds of digits, because Euclid's algorithm is extremely efficient.

Coding:

```

def gcd(a,b):
    if b == 0:
        return a
    else:
        return gcd(b,a%b)

```

Output:

	Test	Expected	Got	
✓	print(gcd(8, 12))	4	4	✓
✓	print(gcd(720, 1000))	40	40	✓

Passed all tests! ✓

10. Given an integer number and you have to count the digits using recursion using Python program. In this program, you will be reading an integer number and counting the total digits, using a function `countDigits()` which will take a number as an argument and return the count after recursion process.

Input Format: The first and only line of the input contains a single integer `n`

Output Format: Output a single line denoting the number of digits in `n`.

For example:

Test	Result
<code>print(countDigits(800))</code>	3

Coding:

```
def countDigits(n):
    if n<0:
        n = -n
    if n<10:
        return 1
    else:
        return 1+ countDigits(n//10)
```

Output:

	Test	Expected	Got	
✓	<code>print(countDigits(12345))</code>	5	5	✓
✓	<code>print(countDigits(800))</code>	3	3	✓

Passed all tests! ✓

WEEK-06-MCQ-Strings and its operations.

1.Which of the following are valid string manipulation functions in Python?

- a. All of the mentioned b. count() c. strip() d. upper()

2. What is the output of the following?

```
i = 0  
while i < 3:  
    print(i)  
    i += 1  
else:  
    print(0)
```

- a. 0 1 2 b. 0 1 2 0 c. Error d. 0 1 2 3 0

3. What is the output of the following code?

```
print('ab cd ef'.title())
```

- a. None of the mentioned b. Ab cd ef c. Ab Cd Ef d. Ab cd eF

4. What is the output of the following code?

```
str1="vijay"  
for i in range(len(str1)):  
    print(i, end="")
```

- a. 01234 b. None of the above c. vijay d. No output

5.What is the output of the following?

```
my_string = 'arvijayakumar'  
for i in range(len(my_string)):  
    my_string[i].upper()  
print (my_string)
```

- a.Error b.arvijayakumar c.ARVIJAYAKUMAR d.None

6. What is the output of the following code?

```
print("rec. VIJAY".capitalize())
```

- a. Rec. vijay b. Rec. Vijay c. REC. VIJAY d. rec. vijay

7. What is the output of the following code ?

```
print('raining'. find('z'))
```

- a. " b. -1 c. Not Found d. Type error

8.What is the output of the following code?

```
my_string = "arvjayakumar"
```

```
i = "i"
```

```
while i in my_string:
```

```
    print(i, end = " ")
```

- a. arvjayakumar b. i i i i... c. a r v j a y a k u m a r d. None

9.What will be the output of below Python code?

```
str1="Application"  
str2=str1.replace('a","A')  
print(str2)
```

Answer:Appicaton

10. What is the output of the following Code?

```
str1="arvijayakumar"  
print(str1[::-1])
```

Answer:r

11. What is the output of the following Code?

```
print(chr(70))
```

Answer:e

12. What is the output of the following Code?

```
str1="6/4"  
print("str1")
```

Answer:5/4

13. What is the output of the following code?

```
print('Ab!2'.swapcase())
```

- a.AB!@ b.aB1@ c.aB!2 d.ab12

14. What is the output of the following code ?

```
str = "Welcome"
```

`str[2] = 'a' print(str)`

a.Welcomea
d.aWelcome

b.Error

c.Weacome

15. Which of the following will result in an error?

`str1="python"`

a. `print(str1[2])`
c. `str1[1]="x"`

b. `print(str1[0:9])`
d. None of the mentioned

WEEK-06-CODING-Strings

1. Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

Input Format:

The first line contains S.

Output Format:

The first line contains EXTENSION.
The second line contains DOMAIN.
The third line contains USERNAME.

Boundary Condition:

1 <= Length of S <= 100

Example Input/Output 1:

Input:

abcd@gmail.com

Output:

com
gmail
abcd

For example:

Input	Result
arvijayakumar@rajalakshmi.edu.in	edu.in rajalakshmi arvijayakumar

Coding:

```
s = input().strip()  
name, dom = s.split('@')  
i= dom.find(".")
```

```
d = dom[i + 1:]  
d1 = dom[:i]
```

```
print(d)  
print(d1)  
print(name)
```

Output:

	Input	Expected	Got	
✓	abcd@gmail.com	com gmail abcd	com gmail abcd	✓
✓	arvijayakumar@rajalakshmi.edu.in	edu.in rajalakshmi arvijayakumar	edu.in rajalakshmi arvijayakumar	✓

Passed all tests! ✓

2. Given a **non-empty** string s and an abbreviation abbr, return whether the string matches with the given abbreviation.

A string such as "word" contains only the following valid abbreviations:

```
["word", "1ord", "w1rd", "wo1d", "wor1", "2rd", "w2d", "wo2", "1o1d", "1or1", "w1r1", "1o2", "2r1", "3d",  
"w3", "4"]
```

Notice that only the above abbreviations are valid abbreviations of the string "word". Any other string is not a valid abbreviation of "word".

Note:

Assume s contains only lowercase letters and abbr contains only lowercase letters and digits.

Example 1:

Input

```
internationalization  
i12iz4n
```

Output

```
true
```

Explanation

Given **s** = "internationalization", **abbr** = "i12iz4n":

Return true.

Example 2:

Input

apple

a2e

Output

false

Explanation

Given **s** = "apple", **abbr** = "a2e":

Return false.

Coding:

```
def abb(w: str,a: str) -> str:  
    i=0  
    j=0  
    while i < len(w) and j < len(a):  
        if a[j].isdigit():  
            if a[j] == '0':  
                return "false"  
  
            num=0  
            while j < len(a) and a[j].isdigit():  
                num=num*10+int(a[j])  
                j += 1  
            i += num  
        else:  
            if i >= len(w) or w[i] != a[j]:  
                return "false"  
  
            i +=1  
            j +=1  
  
    return "true" if i == len(w) and j == len(a) else "false"
```

```
w=input()  
a=input()  
res=abb(w,a)  
print(res)
```

Output:

	Input	Expected	Got	
✓	internationalization i12iz4n	true	true	✓
✓	apple a2e	false	false	✓

Passed all tests! ✓

3. Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

Note: For the purpose of this problem, we define empty string as valid palindrome.

Example 1:

Input:

A man, a plan, a canal: Panama

Output:

1

Example 2:

Input:

race a car

Output:

0

Constraints:

s consists only of printable ASCII characters.

Coding:

```
s = input()  
f = ''.join(char.lower() for char in s if char.isalnum())  
p = f == f[::-1]  
print(1 if p else 0)
```

Output:

	Input	Expected	Got	
✓	A man, a plan, a canal: Panama	1	1	✓
✓	race a car	0	0	✓

Passed all tests! ✓

4. Consider the below words as key words and check the given input is key word or not.
 keywords: {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Input format:

Take string as an input from stdin.

Output format:

Print the word is key word or not.

Example Input:

break

Output:

break is a keyword

Example Input:

IF

Output:

IF is not a keyword

For example:

Input	Result
break	break is a keyword
IF	IF is not a keyword

Coding:

```
keywords =
{"break","case","continue","default","defer","else","for","func","goto","if","map","range","return",
"struct","type","var"}
i = input().strip()

if i in keywords:
    print(f"{i} is a keyword")
else:
    print(f"{i} is not a keyword")
```

Output:

	Input	Expected	Got	
✓	break	break is a keyword	break is a keyword	✓
✓	IF	IF is not a keyword	IF is not a keyword	✓

Passed all tests! ✓

5. A pangram is a sentence where every letter of the English alphabet appears at least once.

Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise.

Example 1:

Input:

thequickbrownfoxjumpsoverthelazydog

Output:

true

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

Input:

arvijayakumar

Output: false

Constraints:

1 <= sentence.length <= 1000

sentence consists of lowercase English letters.

For example:

Test	Result
print(checkPangram('thequickbrownfoxjumpsoverthelazydog'))	true
print(checkPangram('arvijayakumar'))	false

Coding:

```
def checkPangram(s):
    a=set('abcdefghijklmnopqrstuvwxyz')
    b=set(s)
    return "true" if a.issubset(b) else "false"
```

Output:

Test	Expected	Got	
✓ print(checkPangram('thequickbrownfoxjumpsoverthelazydog'))	true	true	✓
✓ print(checkPangram('arvijayakumar'))	false	false	✓

Passed all tests! ✓

6. Write a Python program to get one string and reverses a string. The input string is given as an array of characters char[].

You may assume all the characters consist of [printable ascii characters](#).

Example 1:

Input:

hello

Output:

olleh

Example 2:

Input:

Hannah

Output:

hannaH

Coding:

```
def r(s):
    return s[::-1]
n =input()
rs=r(n)
print(rs)
```

Output:

	Input	Expected	Got	
✓	hello	olleh	olleh	✓
✓	Hannah	hannaH	hannaH	✓

7 Assume that the given string has enough memory.

Passed all tests! ✓

Don't use any extra space(IN-PLACE)

Sample Input 1

a2b4c6

Sample Output 1

aabbccccc

Coding:

```
def r(s):
    s=list(s)
    n=len(s)
    i=0
    j=0
    res=[]

    while i < n:
        char = s[i]
        i+=1
        count=0

        while i<n and s[i].isdigit():
            count=count*10+int(s[i])
            i+=1

        res.extend([char]*count)

    return ''.join(res)

n=input()
a=r(n)
print(a)
```

Output:

	Input	Expected	Got	
✓	a2b4c6	aabbccccc	aabbccccc	✓
✓	a12b3d4	aaaaaaaaaaabbbddd	aaaaaaaaaaabbbddd	✓

Passed all tests! ✓

8. Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

Sample Input 1

thistest123string

123

Sample Output 1

8

Coding:

```
def res(s1,s2):
```

```
    i = s1.find(s2)
```

```
    return i
```

```
s1=input()
```

```
s2=input()
```

```
print(res(s1,s2))
```

Output:

	Input	Expected	Got	
✓	thistest123string 123	8	8	✓

Passed all tests! ✓

9. The program must accept **N** series of keystrokes as string values as the input. The character `^` represents undo action to clear the last entered keystroke. The program must print the string typed after applying the undo operations as the output. If there are no characters in the string then print **-1** as the output.

Boundary Condition(s):

1 <= N <= 100
1 <= Length of each string <= 100

Input Format:

The first line contains the integer N.
The next N lines contain a string on each line.

Output Format:

The first N lines contain the string after applying the undo operations.

Example Input/Output 1:

Input:

3
Hey ^ goooo^^glee^
lucke^y ^charr^ms
ora^^nge^^^

Output:

Hey google
luckycharms
-1

Coding:

def r(s):

 a=[]

 for char in s:

 if char == '^':

 if a:

 a.pop()

 else:

 a.append(char)

 return ''.join(a) if a else '-1'

def main():

```

import sys
input=sys.stdin.read
d=input().strip().split('\n')

n=int(d[0])
res=[]

for i in range(1,n+1):
    res.append(r(d[i]))

for res1 in res:
    print(res1)

if __name__ == "__main__":
    main()

```

Output:

	Input	Expected	Got	
✓	3 Hey ^ goooo^^glee^ lucke^y ^charr^ms ora^^nge^^^^	Hey google luckycharms -1	Hey google luckycharms -1	✓

Passed all tests! ✓

10. Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Constraints:

$1 \leq s.length \leq 10^4$
s consists of parentheses only '()'[]{}'.

For example:

Test	Result
print(ValidParenthesis("()"))	true
print(ValidParenthesis("()[]{}"))	true
print(ValidParenthesis("[]"))	false

Coding:

```
def ValidParenthesis(s):
    a={')':'(', '}':'{', ']':'['}
    st=[]

    for char in s:
        if char in a:
            e = st.pop() if st else '#'
            if a[char] != e:
                return "false"
        else:
            st.append(char)
    return "true" if not st else "false"
```

Output:

	Test	Expected	Got	
✓	print(ValidParenthesis("()"))	true	true	✓
✓	print(ValidParenthesis("()[]{}"))	true	true	✓
✓	print(ValidParenthesis("[]"))	false	false	✓

Passed all tests! ✓

WEEK-07-MCQ-Lists

1.What will be the output after the following statements?

```
m = [5, 10, 35]
```

```
del m[:]
```

```
print(m)
```

- a. [5, 10, 35] b. 5, 10, 35 c. [] d. [5, 35]

2. L=[1,5,9]

```
print(sum(L),max(L),min(L))
```

what will be the expected output for this following?

Answer:15 9 1

3. What will be the output after the following statements?

```
m = ['July', 'September', 'December']
```

```
n = m[0] + m[2]
```

```
print( n )
```

- a.SeptemberDecember b.JulyDecember c.JulySeptember d.July

4. L=['Amit","Anita","Zee","Longest Word"]

```
print(max(L))
```

what will be the expected output for this following?

Answer: Longest Word

5. Fill in the blanks with same word in both places

```
>>> import _____
```

```
>>> list1 = [1,2,3,4,5]
```

```
>>> list2 = _____copy(list1)
```

```
>>> list2
```

a.Pickle

b.Math

c.Copy

6. Write the output of the following :

```
def listchange(L):
    L.append(45)
    return
L1 = [1, 2, 3, 4]
listchange(L1)
print(L1)
```

what will be the expected output for this following?

Answer: [1,2,3,4,45]

7. What will be the output after the following statements?

```
m = 'A'
n = 'B'
o = 'C'
p = [m, n, o]
print(p)
```

- a. 'C' 'A' 'B' b. ['C' 'B' 'A'] c. ['A' 'B' 'C'] d. ['C', 'A', 'B']

8. What will be the output after the following statements?

```
m = [4, 8]
```

```
print(m * 3)
```

- a. [4, 8, 4, 8, 4, 8] b. [4, 8, 4, 8] c. [4, 8] d. [4, 8] * 3

9. What will be the output after the following statements?

```
m = [75, 23, 64]
```

```
n = m[0] + m[1]
```

```
print(n)
```

- a.75 b.23 c.98 d.64

10. What is the data type of m after the following statement?

```
m = ['July', 'September', 'December']
```

- a.String b.Tuple c.Dictionary d.List

WEEK-07-CODING-Lists

1. Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

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

Example Input:

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

Output:

```
1 2 3 4
```

Example Input:

```
6  
1  
1  
2  
2  
3  
3
```

Output:

```
1 2 3
```

For example:

Input	Result
5 1 2 2 3	1 2 3 4

Input	Result
4	
6 1 1 2 2 3 3	1 2 3

Coding:

```
def ele():
    n=int(input().strip())
    a=[]
    for _ in range(n):
        while True:
            e=int(input().strip())
            a.append(e)
            break
    d=set(a)
    d=sorted(d)
    print(' '.join(map(str,d)))
ele()
```

Output:

	Input	Expected	Got	
✓	5 1 2 2 3 4	1 2 3 4	1 2 3 4	✓
✓	6 1 1 2 2 3 3	1 2 3	1 2 3	✓

Passed all tests! ✓

2. Given an integer n, return a list of length n + 1 such that for each i (0 <= i <= n), ans[i] is the number of 1's in the binary representation of i.

Example:

Input: n = 2

Output: [0,1,1]

Explanation:

0 --> 0

1 --> 1

2 --> 10

Example2:

Input: n = 5

Output: [0,1,1,2,1,2]

Explanation:

0 --> 0

1 --> 1

2 --> 10

3 --> 11

4 --> 100

5 --> 101

Note: Complete the given function alone

For example:

Test	Result
print(CountingBits(5))	[0, 1, 1, 2, 1, 2]

Coding:

```
def CountingBits(n):
    res=[]
    for i in range(n+1):
        res.append(bin(i).count('1'))
    return res
```

Output:

	Test	Expected	Got	
✓	print(CountingBits(2))	[0, 1, 1]	[0, 1, 1]	✓
✓	print(CountingBits(5))	[0, 1, 1, 2, 1, 2]	[0, 1, 1, 2, 1, 2]	✓

Passed all tests! ✓

3. Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[i] - A[j] = k$, $i \neq j$.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

Example

Input

1
3
1
3
5
4

Output:

1

Input

1
3
1
3
5
99

Output

0

For example:

Input	Result
1 3 1 3 5 4	1
1 3	0

Input	Result
1 3 5 99	

Coding:

```
def pair(arr,n,k):
    i , j = 0, 1

    while i < n and j < n:
        if i != j and arr[j]-arr[i]==k:
            return 1
        elif arr[j] - arr[i] < k:
            j +=1
        else:
            i +=1
        if i == j:
            j +=1

    return 0

def main():
    t = int(input().strip())
    for _ in range(t):
        n = int(input().strip())
        arr = []
        for _ in range(n):
            arr.append(int(input().strip()))
        k = int(input().strip())
        print(pair(arr,n,k))

if __name__ == "__main__":
    main()
```

Output:

	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1 3 1 3 5 99	0	0	✓

Passed all tests! ✓

4. Complete the program to count frequency of each element of an array. Frequency of a particular element will be printed once.

Sample Test Cases

Test Case 1

Input

```
7  
23  
45  
23  
56  
45  
23  
40
```

Output

```
23 occurs 3 times  
45 occurs 2 times  
56 occurs 1 times  
40 occurs 1 times
```

Coding:

```
def fry(ar):  
    f={}  
    for e in ar:  
  
        if e in f:  
            f[e] +=1  
        else:  
            f[e] = 1  
  
    for e, value in f.items():  
        print(f"{e} occurs {value} times")  
  
n = int(input())  
ar=[]  
for i in range(n):  
    e=int(input())  
    ar.append(e)  
  
fry(ar)
```

Output:

	Input	Expected	Got	
✓	7	23 occurs 3 times	23 occurs 3 times	✓
	23	45 occurs 2 times	45 occurs 2 times	
	45	56 occurs 1 times	56 occurs 1 times	
	23	40 occurs 1 times	40 occurs 1 times	
	56			
	45			
	23			
	40			

Passed all tests! ✓

5. An array is monotonic if it is either **monotone increasing** or **monotone decreasing**. An array A is monotone increasing if for all $i \leq j$, $A[i] \leq A[j]$. An array A is monotone decreasing if for all $i \leq j$, $A[i] \geq A[j]$.

Write a program if n array is monotonic or not. Print "True" if is monotonic or "False" if it is not. Array can be monotone increasing or decreasing.

Input Format:

First line n-get number of elements

Next n Lines is the array of elements

Output Format:

True ,if array is monotone increasing or decreasing.

otherwise False is printed

Sample Input1

```
4
5
6
7
8
```

Sample Output1

True

Sample Input2

```
4
6
5
4
3
```

Sample Output2

True

Sample Input 3

4
6
7
8
7

Sample Output3

False

For example:

Input	Result
4	True
6	
5	
4	
3	

Coding:

n=int(input())

arr =[int(input()) for _ in range(n)]

inc = True
dec = True

```
for i in range(1,n):
    if arr[i] > arr[i-1]:
        dec = False
    if arr[i] < arr[i-1]:
        inc = False

if inc or dec:
    print("True")
else:
    print("False")
```

Output:

	Input	Expected	Got	
✓	4 6 5 4 3	True	True	✓
✓	4 3 5 7 4	False	False	✓
✓	4 1 6 9 2	False	False	✓
✓	4 9 6 4 2	True	True	✓
✓	3 2 1 4	False	False	✓

Passed all tests! ✓

6. Given two arrays of positive integers, for each element in the second array, find the total number of elements in the first array which are *less than or equal* to that element. Store the values determined in an array.

For example, if the first array is [1, 2, 3] and the second array is [2, 4], then there are 2 elements in the first array *less than or equal* to 2. There are 3 elements in the first array which are *less than or equal* to 4. We can store these answers in an array, *answer* = [2, 3].

Program Description

The program must return an array of m positive integers, one for each $\text{maxes}[i]$ representing the total number of elements $\text{nums}[j]$ satisfying $\text{nums}[j] \leq \text{maxes}[i]$ where $0 \leq j < n$ and $0 \leq i < m$, in the given order.

The program has the following:

- $\text{nums}[\text{nums}[0], \dots, \text{nums}[n-1]]$: first array of positive integers
- $\text{maxes}[\text{maxes}[0], \dots, \text{maxes}[n-1]]$: second array of positive integers

Constraints

- $2 \leq n, m \leq 10^5$
- $1 \leq \text{nums}[j] \leq 10^9$, where $0 \leq j < n$.
- $1 \leq \text{maxes}[i] \leq 10^9$, where $0 \leq i < m$.

Input Format For Custom Testing

Input from `stdin` will be processed as follows and passed to the program.

The first line contains an integer n , the number of elements in nums .

The next n lines each contain an integer describing $\text{nums}[j]$ where $0 \leq j < n$.

The next line contains an integer m , the number of elements in maxes .

The next m lines each contain an integer describing $\text{maxes}[i]$ where $0 \leq i < m$.

Sample Case 0

Sample Input 0

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

Sample Output 0

```
2  
4
```

Explanation 0

We are given $n = 4$, $\text{nums} = [1, 4, 2, 4]$, $m = 2$, and $\text{maxes} = [3, 5]$.

1. For $\text{maxes}[0] = 3$, we have 2 elements in nums ($\text{nums}[0] = 1$ and $\text{nums}[2] = 2$) that are $\leq \text{maxes}[0]$.
2. For $\text{maxes}[1] = 5$, we have 4 elements in nums ($\text{nums}[0] = 1$, $\text{nums}[1] = 4$, $\text{nums}[2] = 2$, and $\text{nums}[3] = 4$) that are $\leq \text{maxes}[1]$.

Thus, the program returns the array $[2, 4]$ as the answer.

Sample Case 1

Sample Input 1

```
5  
2  
10  
5  
4  
8  
4  
3  
1  
7  
8
```

Sample Output 1

```
1  
0
```

3
4

Explanation 1

We are given, $n = 5$, $\text{nums} = [2, 10, 5, 4, 8]$, $m = 4$, and $\text{maxes} = [3, 1, 7, 8]$.

1. For $\text{maxes}[0] = 3$, we have 1 element in nums ($\text{nums}[0] = 2$) that is $\leq \text{maxes}[0]$.
2. For $\text{maxes}[1] = 1$, there are 0 elements in nums that are $\leq \text{maxes}[1]$.
3. For $\text{maxes}[2] = 7$, we have 3 elements in nums ($\text{nums}[0] = 2$, $\text{nums}[2] = 5$, and $\text{nums}[3] = 4$) that are $\leq \text{maxes}[2]$.
4. For $\text{maxes}[3] = 8$, we have 4 elements in nums ($\text{nums}[0] = 2$, $\text{nums}[2] = 5$, $\text{nums}[3] = 4$, and $\text{nums}[4] = 8$) that are $\leq \text{maxes}[3]$.

Thus, the program returns the array $[1, 0, 3, 4]$ as the answer.

Coding:

```
import bisect
n= int(input())
nums =[int(input()) for _ in range(n)]
m= int(input())
maxes = [int(input()) for _ in range(m)]

nums.sort()

result = []
for maxval in maxes:
    count = bisect.bisect_right(nums, maxval)
    result.append(count)

for res in result:
    print(res)
```

Output:

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

7.The program must accept **N** integers and an integer **K** as the input. The program must print every K integers in descending order as the output.

-
Note: If **N % K != 0**, then sort the final **N%K** integers in descending order.

Boundary Condition(s):

$1 \leq N \leq 10^4$
-99999 \leq Array Element Value ≤ 99999

Input Format:

The first line contains the values of N and K separated by a space.
The second line contains N integers separated by space(s).

Output Format:

The first line contains N integers.

Example Input/Output 1:

Input:

7 3
48 541 23 68 13 41 6

Output:

541 48 23 68 41 13 6

Explanation:

The first three integers are 48 541 23, after sorting in descending order the integers are **541 48 23**.

The second three integers are 68 13 41, after sorting in descending order the integers are **68 41 13**.

The last integer is **6**.

The integers are **541 48 23 68 41 13 6**
Hence the output is **541 48 23 68 41 13 6**.

Coding:

```
N, K = map(int, input().split())
```

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

```
if len(a) < N:  
    a.extend([0] * (N - len(array)))
```

```
result = []
```

```
for i in range(0, N, K):  
    chunk = a[i:i + K]  
    result.extend(sorted(chunk, reverse=True))
```

```
print(" ".join(map(str, result)))
```

Output:

	Input	Expected	Got	
✓	7 3 48 541 23 68 13 41 6	541 48 23 68 41 13 6	541 48 23 68 41 13 6	✓

Passed all tests! ✓

8. Assume you have an array of length n initialized with all 0's and are given k update operations.

Each operation is represented as a triplet: **[startIndex, endIndex, inc]** which increments each element of subarray $A[\text{startIndex} \dots \text{endIndex}]$ (startIndex and endIndex inclusive) with **inc**.

Return the modified array after all k operations were executed.

Example:

Input:

5

3

1 3 2

2 4 3

0 2 -2

Output:

-2 0 3 5 3

Explanation:

Initial state:

length = 5, updates = [[1,3,2],[2,4,3],[0,2,-2]]

[0,0,0,0,0]

After applying operation [1,3,2]:

[0,2,2,2,0]

After applying operation [2,4,3]:

[0,2,5,5,3]

After applying operation [0,2,-2]:

[-2,0,3,5,3]

Coding:

```
n= int(input())
k= int(input())
```

```
a = [0]* n
for _ in range(k):
    SI, EI, inc= map(int, input().split())
    for i in range(SI, EI +1):
        a[i] += inc
print(" ".join(map(str,a)))
```

Output:

	Input	Expected	Got	
✓	5 3 1 3 2 2 4 3 0 2 -2	-2 0 3 5 3	-2 0 3 5 3	✓

Passed all tests! ✓

9. Given a matrix mat where every row is sorted in **strictly increasing** order, return the **smallest common element** in all rows.

If there is no common element, return -1.

Example 1:

Input:

```
4 5
1 2 3 4 5
2 4 5 8 10
3 5 7 9 11
1 3 5 7 9
```

Output:

```
5
```

Constraints:

- $1 \leq \text{mat.length}, \text{mat}[i].length \leq 500$
- $1 \leq \text{mat}[i][j] \leq 10^4$
- $\text{mat}[i]$ is sorted in strictly increasing order.

Coding:

```
import sys
from collections import Counter
f = input().strip()
n= int(f.strip()[0])
```

```
m=[]
```

```
for _ in range(n):
    r= list(map(int, input().split()))
    m.append(r)
```

```
counter=Counter()
```

```
for r in m:
    u = set(r)
    for e in u:
        counter[e] +=1
```

```
small = float('inf')
```

```

for e, count in counter.items():
    if count == n and e < small:
        small = e
if small == float('inf'):
    print(-1)
else:
    print(small)

```

Output:

	Input	Expected	Got	
✓	4 5 1 2 3 4 5 2 4 5 8 10 3 5 7 9 11 1 3 5 7 9	5	5	✓

Passed all tests! ✓

10. Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p^{th} element of the list, sorted ascending. If there is no p^{th} element, return 0.

Example

$n = 20$

$p = 3$

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if $p = 3$, then 4 is returned. If $p > 6$, 0 would be returned.

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^9$

The first line contains an integer n , the number to factor.

The second line contains an integer p , the 1-based index of the factor to return.

Sample Case 0

Sample Input 0

10

3

Sample Output 0

5

Explanation 0

Factoring $n = 10$ results in {1, 2, 5, 10}. Return the $p = 3^{\text{rd}}$ factor, 5, as the answer.

Sample Case 1

Sample Input 1

10

5

Sample Output 1

0

Explanation 1

Factoring $n = 10$ results in $\{1, 2, 5, 10\}$. There are only 4 factors and $p = 5$, therefore 0 is returned as the answer.

Sample Case 2**Sample Input 2**

1

1

Sample Output 2

1

Explanation 2

Factoring $n = 1$ results in $\{1\}$. The $p = 1$ st factor of 1 is returned as the answer.

For example:

Input	Result
10 3	5
10 5	0
1 1	1

Coding:

```
def fa(n,p):  
    f=[i for i in range(1,n+1) if n%i ==0]  
    f.sort()
```

```
if p <= len(f):  
    return f[p-1]  
else:  
    return 0
```

```
n = int(input())  
p = int(input())  
print(fa(n,p))
```

Output:

	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

WEEK-08-MCQ-Tuple-Set

1.What is the output of the following code?

```
aTuple = (10, 20, 30, 40, 50, 60, 70, 80)
```

```
print(aTuple[2:5], aTuple[:4], aTuple[3:])
```

- a. (10, 20, 30, 40) (40, 50, 60, 70, 80)
- b. (30, 40, 50)(40, 50, 60, 70, 80)
- c. (30, 40, 50) (10, 20, 30, 40)
- d. (30, 40, 50) (10, 20, 30, 40) (40, 50, 60, 70, 80)

2. Select which is true for Python tuple?

- a. We can change the tuple once created
- b. A tuple maintains the order of items
- c. A tuple is unordered
- d. None of these

3. Choose the correct option.

- a. In Python, a tuple can contain either string or integer but not both at a time.
- b. In Python, a tuple can contain both integers and strings as its elements.
- c. In Python, a tuple can contain only strings as its elements.
- d. In Python, a tuple can contain only integers as its elements.

4. What will be printed when the following code executes?

```
a = ("Python Programming")
```

```
print type(a)
```

- a. <class 'tuple'>
- b. <class 'str'>
- c. str
- d. <class 'int'>

5. What is the output of the given below program?

```
t1 = (1,2,3)
t2 = (4,5,6)
x = t1+t2
print(x)
```

- a.(1,2,3)(4,5,6)
- b.Error
- c.(1,2,3,3,2,1)
- d.(1,2,3,4,5,6)

6. Write the Output of the following Code?

```
t = (15,83,83,52,60,45,52,85,100)  
print(min(t)+max(t)+t.count(52))
```

- a.100 b.2 c.117 d.Error

7. Which of the following is a Python tuple?

- a.{1,3,8,9,41} b.(1,4,5,6,7) c.[1,2,3,4] d.("Wonder")

8. Which of the following Python code will create a set?

- (i) set1=set((0,9,0))
(ii) set1=set([0,2,9])
(iii) set1={}

a. iii b. ii c. All of the above d. i,ii

9. What will be the output of following Python code?

```
list1=[1,3,4,2]  
x=list1.pop(2)  
print(set([x]))
```

- a.{1,3,4} b.{4} c.{1,3,2} d.{2}

10. What will the below Python code do?

```
set1={2,3}  
set2={3,2}  
set3={2,1}  
if(set1==set2):  
    print("yes")  
else:  
    print("no")  
if(set1==set3):  
    print("yes")  
else:  
    print("no")
```

- a. No, No b. Yes, No
c. "==" is not supported for set in Python d. Yes, Yes

11. What will be the output of the below Python code?

```
t1=(55,12,78,64,25)
```

```
t1.pop(12)
```

```
print(tuple1)
```

a.(12)

b.Error

c.12

d.(55,78,64,25)

12. What is the output of the following

```
set1 = {1, 2, 3, 4, 5}
```

```
set2 = {6, 7, 1, 3, 4, 8, 2, 5}
```

```
print(set1.issubset(set2))
```

```
print(set2.issuperset(set1))
```

a.FalseTrue

b.TrueTrue

c.FalseFalse

d.TrueFalse

13. What will be the output of below Python code?

```
tupl=("python","", "programming", "Computer")
```

```
print(tupl[-3:0])
```

a.Error

b.()

c.Computer

d.(Computer)

14. What is the output of the given below program?

```
my_t1 = (1, 2, 3, 4)
```

```
my_t1.append( (5, 6, 7) )
```

```
print(len(my_t1))
```

a.Error

b.2

c.1

d.5

15. Find the output of the given Python program?

```
t1 = (1,2,3,(4,5))
```

```
t2 = (3,2,1,(4,5))
```

```
print(t1>t2)
```

a.Error

b.Error

c.True

d.False

WEEK-08-CODING-Tuple-Set

1. There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly.

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

Example 1:

Input: text = "hello world", brokenLetters = "ad"

Output:

1

Explanation: We cannot type "world" because the 'd' key is broken.

For example:

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

Coding:

```
def main(t,l):
    b = set(l.lower())
    ws = t.split()
    c= 0
    for w in ws:
        if all(letter.lower() not in b for letter in w):
            c +=1
    return c
```

```
t = input().strip()
b = input().strip()
res = main(t,b)
print(res)
```

Output:

	Input	Expected	Got	
✓	hello world ad	1	1	✓
✓	Welcome to REC e	1	1	✓
✓	Faculty Upskilling in Python Programming ak	2	2	✓

Passed all tests! ✓

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

Examples:

Input: str = "01010101010"

Output: Yes

Input: str = "REC101"

Output: No

For example:

Input	Result
01010101010	Yes
010101 10101	No

Coding:

```
s=input()
c=set(s)
if c <= {'0', '1'}:
    print("Yes")
else:
    print("No")
```

Output:

	Input	Expected	Got	
✓	01010101010	Yes	Yes	✓
✓	REC123	No	No	✓
✓	010101 10101	No	No	✓

Passed all tests! ✓

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

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

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

Example 1:

Input: s = "AAAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"

Output: ["AAAAAACCCC", "CCCCCAAAAA"]

Example 2:

Input: s = "AAAAAAAAAAAAAA"

Output: ["AAAAAAAAAA"]

For example:

Input	Result
AAAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAAACCCC CCCCCAAAAA

Coding:

```
def show(s):
    se = set()
    r = set()
    for i in range(len(s)-9):
        sq = s[i:i+10]
        if sq in se:
            r.add(sq)
        else:
            se.add(sq)
    return list(r)
```

```
n = input().strip()
res = show(n)
for resut in res:
    print(resut)
```

Output:

	Input	Expected	Got	
✓	AAAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAAACCCC CCCCCAAAAA	AAAAAACCCC CCCCCAAAAA	✓
✓	AAAAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✓

Passed all tests! ✓

4. Given an array of integers `nums` containing $n + 1$ integers where each integer is in the range $[1, n]$ inclusive. There is only **one repeated number** in `nums`, return *this repeated number*. Solve the problem using set.

Example 1:

Input: `nums = [1,3,4,2,2]`

Output: 2

Example 2:

Input: `nums = [3,1,3,4,2]`

Output: 3

For example:

Input	Result
1 3 4 4 2	4

Coding:

```
def show(num):
    s=set()
    for n in num:
        if n in s:
            return n
        s.add(n)
    return -1

i = input()
num = list(map(int,i.split()))
res = show(num)
print(res)
```

Output:

	Input	Expected	Got	
✓	1 3 4 4 2	4	4	✓
✓	1 2 2 3 4 5 6 7	2	2	✓

Passed all tests! ✓

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

1. If the first element of the tuple has the smallest value in the entire tuple, remove it.
2. Otherwise, move the first element to the end of the tuple.

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

Constraints

- The input tuple `nums` contains distinct integers.
- The operations must be performed using tuples and sets to maintain immutability and efficiency.
- Your function should accept the tuple `nums` as input and return the total number of operations as an integer.

Example:

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

Output: 5

Explanation:

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

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

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

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

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

Total operations: 5

For example:

Test	Result
<code>print(count_operations((3, 4, -1)))</code>	5

Coding:

```
def count_operations(nums: tuple) -> int:  
    # Your implementation here  
    from collections import deque  
  
    nums = deque(nums)  
    c = 0  
    while nums:  
        m = min(nums)  
        if nums[0] == m:  
            nums.popleft()  
        else:  
            nums.append(nums.popleft())  
  
        c += 1  
  
    return c
```

Output:

	Test	Expected	Got	
✓	print(count_operations((3, 4, -1)))	5	5	✓
✓	print(count_operations((1, 2, 3, 4, 5)))	5	5	✓
✓	print(count_operations((5, 4, 3, 2, 1)))	15	15	✓
✓	print(count_operations((42,)))	1	1	✓
✓	print(count_operations((-2, 3, -5, 4, 1)))	11	11	✓

Passed all tests! ✓

6. Write a program to eliminate the common elements in the given 2 arrays and print only the non-repeating

elements and the total number of such non-repeating elements.

Input Format:

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

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

Sample Input:

5 4

1 2 8 6 5

2 6 8 10

Sample Output:

1 5 10

3

Sample Input:

5 5

1 2 3 4 5

1 2 3 4 5

Sample Output:

NO SUCH ELEMENTS

For example:

Input	Result
5 4 1 2 8 6 5 2 6 8 10	1 5 10 3
5 5 1 2 3 4 5 1 2 3 4 5	NO SUCH ELEMENTS

Coding:

```
def main(ar1,ar2):
    s1 = set(ar1)
    s2 = set(ar2)
    c = s1.intersection(s2)

    u1 = s1 - c
    u2 = s2 - c

    u = u1.union(u2)
    return list(u)

z = input().strip().split()
z1,z2 = int(z[0]),int(z[1])

a1 = list(map(int,input().strip().split()))
a2 = list(map(int,input().strip().split()))

res = main(a1,a2)

if res:
    print(" ".join(map(str,sorted(res))))
    print(len(res))
else:
    print("NO SUCH ELEMENTS")
```

Output:

	Input	Expected	Got	
✓	5 4 1 2 8 6 5 2 6 8 10	1 5 10 3	1 5 10 3	✓
✓	3 3 10 10 10 10 11 12	11 12 2	11 12 2	✓
✓	5 5 1 2 3 4 5 1 2 3 4 5	NO SUCH ELEMENTS	NO SUCH ELEMENTS	✓

Passed all tests! ✓

7. Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

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

Example Input:

5

1 2 2 3 4

Output:

1 2 3 4

Example Input:

6

1 1 2 2 3 3

Output:

1 2 3

For example:

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

Coding:

```
def show(arr):
    d = set(arr)
    print(" ".join(map(str,sorted(d))))
n = int(input())
arr =[int(input().strip()) for _ in range(n)]
show(arr)
```

Output:

Input	Expected	Got	
5 1 2 2 3 4	1 2 3 4	1 2 3 4	✓
6 1 1 2 2 3 3	1 2 3	1 2 3	✓
5 11 22 11 22 11	11 22	11 22	✓
10 1 2 3 4 5 1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	✓

Passed all tests! ✓

8. Check if a set is a subset of another set.

Example:

Sample Input1:

mango apple

mango orange

mango

output1:

yes

set3 is subset of set1 and set2

input2:

mango orange

banana orange

grapes

output2:

no

For example:

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

Coding:

```
def main(s1,s2,ss):
    s1 = set(s1.split())
    s2 = set(s2.split())
    ss = set(ss.split())

    if ss.issubset(s1) and ss.issubset(s2):
        print("yes")
        print("set3 is subset of set1 and set2")
    else:
        print("No")

s1 = input().strip()
s2 = input().strip()
ss = input().strip()

res = main(s1,s2,ss)
```

Output:

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

Passed all tests! ✓

9. Given a tuple and a positive integer k, the task is to find the count of distinct pairs in the tuple whose sum is equal to K. **Examples:**

Input: t = (5, 6, 5, 7, 7, 8), K = 13

Output: 2

Explanation:

Pairs with sum K(= 13) are { (5, 8), (6, 7), (6, 7) }.

Therefore, distinct pairs with sum K(= 13) are { (5, 8), (6, 7) }. Therefore, the required output is 2.

For example:

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

Coding:

```
def show(t,K):
    s=set()
    p=set()

    for num in t:
        com = K- num
        if com in s:
            p.add((min(num, com), max(num, com)))
            s.add(num)
    return len(p)

tup = input()
t = tuple(map(int , tup.split(',')))
K = int(input())
res=show(t,K)
print(res)
```

Output:

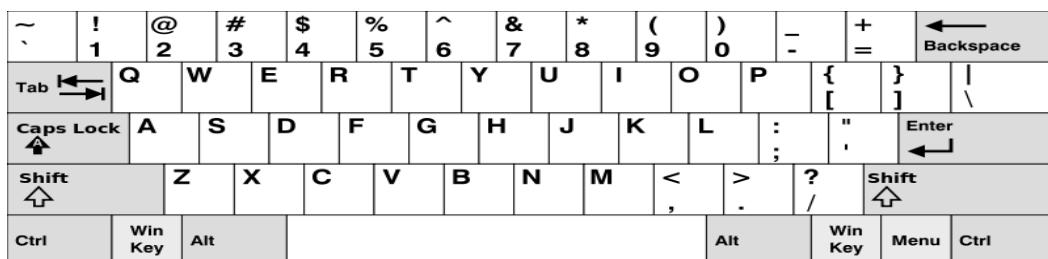
	Input	Expected	Got	
✓	5,6,5,7,7,8 13	2	2	✓
✓	1,2,1,2,5 3	1	1	✓
✓	1,2 0	0	0	✓

Passed all tests! ✓

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

In the **American keyboard**:

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



Example 1:

Input: words = ["Hello", "Alaska", "Dad", "Peace"]

Output: ["Alaska", "Dad"]

Example 2:

Input: words = ["omk"]

Output: []

Example 3:

Input: words = ["adsdf", "sfd"]

Output: ["adsdf", "sfd"]

For example:

Input	Result
4 Hello Alaska Dad Peace	Alaska Dad

Input	Result
2 adsfd afd	adsfd afd

Coding:

```
def word(w):
    r1=set("qwertyuiop")
    r2=set("asdfghjkl")
    r3=set("zxcvbnm")
    res=[]
    for wo in w:
        l = set(wo.lower())
        if l <= r1 or l <= r2 or l <= r3:
            res.append(wo)
    return res
i = int(input())
w=[]
for _ in range(i):
    wo = input()
    w.append(wo)
res =word(w)
if res:
    for wo in res:
        print(wo)
else:
    print("No words")
```

Output:

	Input	Expected	Got	
✓	4 Hello Alaska Dad Peace	Alaska Dad	Alaska Dad	✓
✓	1 omk	No words	No words	✓
✓	2 adsfd afd	adsfd afd	adsfd afd	✓

Passed all tests! ✓

WEEK-9-MCQ-Dictionary

1. Which of the following is feature of Dictionary?

- a. Dictionary is mutable.
 - b. Keys are unique within a dictionary.
 - c. Keys must be of an immutable data type.
 - d. All of the mentioned

2. Which of the following is not method of dictionary?

- a. update() b. del() c. pop() d. len()

3. Which of the following are true of Python dictionaries:

- a) All the keys in a dictionary must be of the same type.
 - b) Items are accessed by their position in a dictionary.
 - c) A dictionary can contain any object type except another dictionary.
 - d) Dictionaries can be nested to any depth.
 - e) Dictionaries are mutable.
 - f) Dictionaries are accessed by key.

a)d,e,f b)c,d,e c)b,c

4. `pop()` function delete and _____ the element of dictionary.

 - a. return
 - b. display
 - c. not return
 - d. add

5. Only values (without keys) can be printed in dictionary?

- a True b False

6. In dictionary Keys and values are separated by

- a. Colon(:) b. Comma(,) c. Semicolon(:) d. dot()

7. Choose the correct statement, in reference to the following code: D1.update(D2) #D1 and D2 are dictionaries

- a. None of the mentioned
 - b. It will create a new dictionary.
 - c. It will merge all the elements of dictionary 'D1' in dictionary 'D2'.
 - d. It will merge all the elements of dictionary 'D2' in dictionary 'D1'.

8. Both keys and values are unique in dictionary.

- a.True b.False

9. Dictionaries in python are _____.

- a. Mutable data type b. Both Non-Mutable data type and Mapping data type
c. Non-Mutable data type d. Mapping data type

10. _____ function returns the value corresponding to the key passed as the argument.

- a. values() b. get() c. del() d. update()

11. Which of the following are immutable data type? A. String B. Tuple C. List D. Dictionary

- a)b and d b) c and d c) a and c d) a and b

12. All elements in dictionary are separated by _____.

- a. Comma(,) b. Colon (:) c. Semicolon(;) d. dot(.)

13. Which one of the following is correct?

- a. A dictionary can have two same keys with different values.
b. A python, a dictionary can neither have two same keys nor two same values.
c. A dictionary can have two same keys or same values but cannot have two same key-value pair
d. A dictionary can have two same values with different keys.

14. Which of the following is an example of dictionary?

- a. C = () b. L = [] c. D = {} d. None of the mentioned

15. In Python, Dictionaries are immutable

Select one: True False

WEEK-09-CODING-Dictionary

- Given a number, convert it into corresponding alphabet.

Input	Output
1	A
26	Z
27	AA
676	YZ

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

1 <= num <= 4294967295

Sample Input 1

26

Sample Output 1

Z

For example:

Test	Result
print(excelNumber(26))	Z

Coding:

```
def excelNumber(n):
    res = []
    while n > 0:
        n -= 1
        rem = n % 26
        res.append(chr(rem + ord('A')))
        n //= 26

    res.reverse()
    return ''.join(res)
```

Output:

	Test	Expected	Got	
✓	print(excelNumber(26))	Z	Z	✓
✓	print(excelNumber(27))	AA	AA	✓

Passed all tests! ✓

2. Objective:

Develop a Python program that takes an input string from the user and counts the number of occurrences of each vowel (a, e, i, o, u) in the string. The program should be case-insensitive, meaning it should treat uppercase and lowercase vowels as the same.

Description:

Vowels play a significant role in the English language and other alphabet-based languages. Counting vowels in a given string is a fundamental task that can be applied in various text processing applications, including speech recognition, linguistic research, and text analysis. The objective of this problem is to create a Python script that accurately counts and displays the number of times each vowel appears in a user-provided string.

Program Requirements:

Input:

First line reading String as input. The string can contain any characters, including letters, numbers, and special characters.

Output:

Display the number of occurrences of each vowel in the string.

The output should list each vowel followed by its count.

Example:

Consider the following example for better understanding:

- **Input:** "Python Programming"
- **Output**

```
a = 1  
e = 0  
i = 1  
o = 2  
u = 0
```

For example:

Input	Result
Hello World	a = 0 e = 1 i = 0 o = 2 u = 0
Python	a = 0 e = 0 i = 0 o = 1 u = 0

Coding:

```
def main(s):  
    s = s.lower()  
    v = {'a':0,'e':0,'i':0,'o':0,'u':0}  
  
    for char in s:  
        if char in v:  
            v[char] += 1  
  
    for vowel in ['a','e','i','o','u']:  
        print(f'{vowel} = {v[vowel]}')  
  
s = input()  
main(s)
```

Output:

	Input	Expected	Got	
✓	Hello World	a = 0 e = 1 i = 0 o = 2 u = 0	a = 0 e = 1 i = 0 o = 2 u = 0	✓
✓	AEIOU aeio u	a = 2 e = 2 i = 2 o = 2 u = 2	a = 2 e = 2 i = 2 o = 2 u = 2	✓
✓	Python	a = 0 e = 0 i = 0 o = 1 u = 0	a = 0 e = 0 i = 0 o = 1 u = 0	✓
✓	abcdefghijklmnopqrstuvwxyz	a = 1 e = 1 i = 1 o = 1 u = 1	a = 1 e = 1 i = 1 o = 1 u = 1	✓
✓	12345!@#\$%AEIOU	a = 1 e = 1 i = 1 o = 1 u = 1	a = 1 e = 1 i = 1 o = 1 u = 1	✓

Passed all tests! ✓

3. In the game of Scrabble™, each letter has points associated with it. The total score of a word is the sum of the scores of its letters. More common letters are worth fewer points while less common letters are worth more points. The points associated with each letter are shown below:

Points Letters

1 A, E, I, L, N, O, R, S, T and U

2 D and G

3 B, C, M and P

4 F, H, V, W and Y

5 K

8 J and X

10 Q and Z

Write a program that computes and displays the Scrabble™ score for a word. Create a dictionary that maps from letters to point values. Then use the dictionary to compute the score.

A Scrabble™ board includes some squares that multiply the value of a letter or the value of an entire word. We will ignore these squares in this exercise.

[Sample](#) Input

REC

[Sample](#) Output

REC is worth 5 points.

For example:

Input	Result
REC	REC is worth 5 points.

Coding:

```
def main(w):
    letter = {
        'A':1,'E':1,'I':1,'L':1,'N':1,'O':1,'R':1,'S':1,'T':1,'U':1,
        'D':2,'G':2,
        'B':3,'C':3,'M':3,'P':3,
        'F':4,'H':4,'V':4,'W':4,'Y':4,
        'K':5,
        'J':8,'X':8,
        'Q':10,'Z':10
    }
    t = 0
    w = w.upper()

    for l in w:
        if l in letter:
            t +=letter[l]
    return t

n = input()
s = main(n)
print(f"{n} is worth {s} points.)
```

Output:

	Input	Expected	Got	
✓	GOD	GOD is worth 5 points.	GOD is worth 5 points.	✓
✓	REC	REC is worth 5 points.	REC is worth 5 points.	✓

Passed all tests! ✓

4. Create a student dictionary for n students with the student name as key and their test mark assignment mark and lab mark as values. Do the following computations and display the result.

- 1.Identify the student with the highest average score
- 2.Identify the student who has the highest Assignment marks
- 3.Identify the student with the Lowest lab marks
- 4.Identify the student with the lowest average score

Note:

If more than one student has the same score display all the student names

Sample input:

4

James 67 89 56

Lalith 89 45 45

Ram 89 89 89

Sita 70 70 70

Sample Output:

Ram

James Ram

Lalith

Lalith

For example:

Input	Result
4 James 67 89 56 Lalith 89 45 45 Ram 89 89 89 Sita 70 70 70	Ram James Ram Lalith Lalith

Coding:

```
def main(m):  
    return sum(m) / len(m)  
n = int(input())  
s = {}  
for _ in range(n):  
    d = input().split()  
    n = d[0]  
    m = list(map(float,d[1:]))  
    s[n] = m  
  
ma = max(main(m) for m in s.values())
```

```

ha = [n for n,m in s.items() if main(m) == ma]

mas = max(m[1] for m in s.values())
has = [n for n,m in s.items() if m[1] == mas]

ml = min(m[2] for m in s.values())
ll = [n for n, m in s.items() if m[2] == ml]

mi = min(main(m) for m in s.values())
la = [n for n, m in s.items() if main(m) == mi]

print(" ".join(ha))
print(" ".join(has))
print(" ".join(sorted(ll)))
print(" ".join(la))

```

Output:

	Input	Expected	Got	
✓	4 James 67 89 56 Lalith 89 45 45 Ram 89 89 89 Sita 70 70 70	Ram James Ram Lalith Lalith	Ram James Ram Lalith Lalith	✓
✓	3 Raja 95 67 90 Aarav 89 90 90 Shadhana 95 95 91	Shadhana Shadhana Aarav Raja	Shadhana Shadhana Aarav Raja	✓

Passed all tests! ✓

5. A sentence is a string of single-space separated words where each word consists only of lowercase letters. A word is uncommon if it appears exactly once in one of the sentences, and does not appear in the other sentence.

Given two sentences s1 and s2, return a list of all the uncommon words. You may return the answer in any order.

Example 1:

Input: s1 = "this apple is sweet", s2 = "this apple is sour"

Output: ["sweet","sour"]

Example 2:

Input: s1 = "apple apple", s2 = "banana"

Output: ["banana"]

Constraints:

$1 \leq s1.length, s2.length \leq 200$

$s1$ and $s2$ consist of lowercase English letters and spaces.

$s1$ and $s2$ do not have leading or trailing spaces.

All the words in $s1$ and $s2$ are separated by a single space.

Note:

Use dictionary to solve the problem

For example:

Input	Result
this apple is sweet	sweet sour
this apple is sour	

Coding:

```
def main(s1,s2):  
    from collections import Counter
```

```
    c1 = Counter(s1.split())  
    c2 = Counter(s2.split())
```

```
    u1 = {word for word in c1 if c1[word] == 1 and word not in c2}  
    u2 = {word for word in c2 if c2[word] == 1 and word not in c1}
```

```
    u = list(u1.union(u2))  
    return ''.join(u)
```

```
s1 = input()  
s2 = input()  
print(main(s1,s2))
```

Output:

	Input	Expected	Got	
✓	this apple is sweet this apple is sour	sweet sour	sweet sour	✓
✓	apple apple banana	banana	banana	✓

Passed all tests! ✓

6. You are given a string `word`. A letter is called **special** if it appears both in lowercase and uppercase in `word`.

Your task is to return the number of **special** letters in `word`.

Constraints

- The input string `word` will contain only alphabetic characters (both lowercase and uppercase).
- The solution must utilize a dictionary to determine the number of special letters.
- The function should handle various edge cases, such as strings without any special letters, strings with only lowercase or uppercase letters, and mixed strings.

Examples

Example 1:

Input: `word = "aaAbcBC"`

Output: 3

Explanation:

The special characters in `word` are 'a', 'b', and 'c'.

Example 2:

Input: `word = "abc"`

Output: 0

Explanation:

No character in `word` appears in uppercase.

For example:

Test	Result
<code>print(count_special_letters("AaBbCcDdEe"))</code>	5

Coding:

```
def count_special_letters(word: str) -> int:  
    # Your implementation here  
    l = {}  
    u = {}  
  
    for char in word:  
        if char.islower():  
            l[char] = True  
        elif char.isupper():  
            u[char.lower()] = True  
    c = 0  
    for char in l:  
        if char in u:  
            c += 1  
    return c
```

Output:

	Test	Expected	Got	
✓	print(count_special_letters("AaBbCcDdEe"))	5	5	✓
✓	print(count_special_letters("ABCDE"))	0	0	✓
✓	print(count_special_letters("abcde"))	0	0	✓

Passed all tests! ✓

7. A company wants to send its quotation secretly to its client. The company decided to encrypt the amount they are sending to their client with some special symbols so that the equation amount will not be revealed to any external person. They used the special symbols !,@,#,\$,%,&,*,>,< for 0,1,2,3,4,5,6,7,8,9 respectively. Write a python code to help the company to convert the amount to special symbols.

(Value rounded off to 2 decimal points)

Input

n: Float data type which reads amount to send

Output

s: : String data type which displays symbols

Sample Testcase 1

Input

10000

Output

@!!!!.!!

Sample Testcase2

1234.56

Output

@#\$%.^&

For example:

Input	Result
1345.23	@\$%^.#\$
15000.5 9	@^!!!.^<
156789	@^&*><.!!

Coding:

```
def main(a):
    s={'0':'!', '1':'@', '2':'#', '3':'$', '4':'%', '5':'^', '6':'&', '7':'*', '8':">>, '9':'<'}
    am = f'{a:.2f}'
    res = ''.join(s[char] for char in am if char in s)
    return res

n = float(input())
print(main(n))
```

Output:

	Input	Expected	Got	
✓	1345.23	@\$%^.#\$	@\$%^.#\$	✓
✓	15000.59	@^!!!.^<	@^!!!.^<	✓
✓	1234	@#\$%.!!	@#\$%.!!	✓
✓	156789	@^&*><.!!	@^&*><.!!	✓

Passed all tests! ✓

8. Given an array of names of candidates in an election. A candidate name in the array represents a vote cast to the candidate. Print the name of candidates received Max vote. If there is tie, print a lexicographically smaller name.

Examples:

Input : votes[] = {"john", "johnny", "jackie", "johnny", "john", "jackie", "jamie", "jamie", "john", "johnny", "jamie", "johnny", "john"};

Output : John

We have four Candidates with name as 'John', 'Johnny', 'jamie', 'jackie'. The candidates John and Johny get maximum votes. Since John is alphabetically smaller, we print it. Use dictionary to solve the above problem

Sample Input:

```
10
John
John
Johny
Jamie
Jamie
Johny
Jack
Johny
Johny
```

Jackie

Sample Output:

Johny

Coding:

```
from collections import defaultdict
```

```
def main(vs):
    vc = defaultdict(int)

    for v in vs:
        vc[v.lower()] += 1

    mv = max(vc.values())
    mc = [c for c in vc if vc[c] == mv]

    return min(mc).capitalize()

n = int(input())
vs = []
for _ in range(n):
    v = input()
    vs.append(v)

print(main(vs))
```

Output:

	Input	Expected	Got	
✓	10 John John Johny Jamie Jamie Johny Jack Johny Johny Jackie	Johny	Johny	✓
✓	6 Ida Ida Ida Kiruba Kiruba Kiruba	Ida	Ida	✓

Passed all tests! ✓

9. Give a dictionary with value lists, sort the keys by summation of values in value list.

Input : test_dict = {'Gfg' : [6, 7, 4], 'best' : [7, 6, 5]}

Output : {'Gfg': 17, 'best': 18}

Explanation : Sorted by sum, and replaced.

Input : test_dict = {'Gfg' : [8,8], 'best' : [5,5]}

Output : {'best': 10, 'Gfg': 16}

Explanation : Sorted by sum, and replaced.

Sample Input:

2

Gfg 6 7 4

Best 7 6 5

Sample Output

Gfg 17

Best 18

For example:

Input	Result
2 Gfg 6 7 4 Best 7 6 5	Gfg 17 Best 18

Coding:

```
def main():
    import sys
    input = sys.stdin.read

    d = input().strip().split('\n')
    e = int(d[0])
    dd = {}

    for i in range(1, e+1):
        p = d[i].split()
        key = p[0]
        v = list(map(int, p[1:]))
        s = sum(v)
        dd[key] = s

    sd = dict(sorted(dd.items(), key=lambda item:item[1]))

    for key, res in sd.items():
        print(f'{key} {res}')
main()
```

Output:

	Input	Expected	Got	
✓	2 Gfg 6 7 4 Best 7 6 5	Gfg 17 Best 18	Gfg 17 Best 18	✓
✓	2 Gfg 6 6 Best 5 5	Best 10 Gfg 12	Best 10 Gfg 12	✓

Passed all tests! ✓

10. A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence.

Example 1:

Input:

is2 sentence4 This1 a3

Output:

This is a sentence

Explanation: Sort the words in s to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2:

Input:

Myself2 Me1 I4 and3

Output:

Me Myself and I

Explanation: Sort the words in s to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

$2 \leq s.length \leq 200$

s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9.

The number of words in s is between 1 and 9.

The words in s are separated by a single space.

s contains no leading or trailing spaces.

Coding:

```
def show(s):
    ws = s.split()
    p = []
    for w in ws:
        if w[-1].isdigit():
            pos = int(w[-1])
            o = w[:-1]
            p.append((pos,o))
            p.sort()
    os = ''.join(w for _,w in p)
    return os

n = input()
print(show(n))
```

Output:

	Input	Expected	Got	
✓	is2 sentence4 This1 a3	This is a sentence	This is a sentence	✓

Passed all tests! ✓

WEEK-10-MCQ-Linear and Binary

1. In the context of searching, what is a successful search?

- a. When the search algorithm finishes
- b. When the list contains duplicate elements
- c. When the list is sorted
- d. When the element is found in the list

2. What is the key characteristic of binary search?

- a. It can be applied only if the list is sorted
- b. It works on unsorted lists
- c. It always starts from the beginning of the list
- d. It compares elements sequentially

3. In binary search, how is the middle element determined?

- a. By starting from the first element
- b. By comparing each element sequentially
- c. By using a hash function
- d. By dividing the list length by two

4. Given an array arr = {45, 77, 89, 90, 94, 99, 100} and key = 100; What are the mid values (corresponding array elements) generated in the first and second iterations?

- a. 90 and 99
- b. 90 and 100
- c. 89 and 94
- d. 94 and 99

5. _____ search takes a sorted/ordered list and divides it in the middle.

- a. Binary
- b. Both (1) & (3)
- c. Linear
- d. Hash

6. Which of the following is not a limitation of binary search algorithm?

- a. Binary search algorithm is not efficient when the data elements more than 1500
- b. Requirement of sorted array is expensive when a lot of insertion and deletions are needed
- c. Must use a sorted array
- d. There must be a mechanism to access middle element directly

7. In linear search, if the target element is not found in the list, what is the result?

- a. The last element is returned
- b. An error is raised
- c. The first element is returned
- d. The search is considered unsuccessful

8. What is the best-case time complexity of linear search?

- a. $O(n \log n)$
- b. $O(\log n)$
- c. $O(n)$
- d. $O(1)$

9. Which of the following scenarios is best suited for applying binary search?

- a. When the list is very small
- b. When the list contains duplicate elements
- c. When the list is unsorted
- d. When the list is sorted

10. What happens when the element is found in linear search?

- a. The search stops immediately
- b. The search starts over from the beginning
- c. The search backtracks to find duplicate elements
- d. The search continues until the end of the list

11.What is the first step in binary search?

- a.Sort the list
- b.Divide the list into two equal parts
- c.Compare the target element with the middle element in the list
- d.Compare the target element with the first element in the list

12Which of the following is a conventional searching technique?

- a.Linear search
- b.Binary search
- c.Hashing
- d.Dynamic search

13.What is the time complexity of linear search in the worst case?

- a. $O(\log n)$
- b. $O(n)$
- c. $O(n \log n)$
- d. $O(1)$

14.In which situation is linear search more efficient than binary search?

- a.When the list is large and unsorted
- b.When the list is small and sorted
- c.When the list is small and unsorted
- d.When the list is large and sorted

15.Which of the following best describes the process of a linear search?

- a.Sorting the list before searching
- b.Dividing the list in half repeatedly
- c.Skipping every second element
- d.Checking each element sequentially

1. Balanced strings are those that have an equal quantity of 'L' and 'R' characters.

Given a balanced string s , split it in the maximum amount of balanced strings.

Return the maximum amount of split balanced strings.

Example 1:

Input:

RLRRLLRLRL

Output:

4

Explanation: s can be split into "RL", "RLL", "RL", "RL", each substring contains same number of 'L' and 'R'.

Example 2:

Input:

RLLLLRRRRLR

Output:

3

Explanation: s can be split into "RL", "LLLRRR", "LR", each substring contains same number of 'L' and 'R'.

Example 3:

Input:

LLLLRRRR

Output:

1

Explanation: s can be split into "LLLLRRRR".

Constraints:

$1 \leq s.length \leq 1000$

$s[i]$ is either 'L' or 'R'.

s is a balanced string.

Program:

```
def BalancedStrings(s,l=0,r=0,count=0):
```

```

for i in s :
    if i=='L' :
        l+=1
    elif i=='R' :
        r+=1
    if l==r :
        count+=1
return count

```

	Test	Expected	Got	
✓	print(BalancedStrings('RLRRLRLRL'))	4	4	✓
✓	print(BalancedStrings('RLLLLRRRLR'))	3	3	✓

Passed all tests! ✓

2. You are given an $m \times n$ integer matrix `matrix` with the following two properties:

- Each row is sorted in non-decreasing order.
- The first integer of each row is greater than the last integer of the previous row.

Given an integer `target`, return `True` if `target` is in `matrix` or `False` otherwise.

You must write a solution in $O(\log(m * n))$ time complexity.

Example 1:

1	3	5	7
10	11	16	20
23	30	34	60

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3
Output: True

Example 2:

1	3	5	7
10	11	16	20
23	30	34	60

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 13
Output: False

Program:

```
def searchMatrix(matrix:list[list[int]], target: int) -> bool:  
  
    for i in range(len(matrix)):  
  
        for j in range(len(matrix[0])):  
  
            if matrix[i][j]==target:  
  
                return True  
  
    return False
```

	Test	Expected	Got	
✓	print(searchMatrix([[1,3,5,7],[10,11,16,20],[23,30,34,60]], 13))	False	False ✓	
✓	print(searchMatrix([[1,3,5,7],[10,11,16,20],[23,30,34,60]], 3))	True	True ✓	

Passed all tests! ✓

3. A list contains N numbers and you want to determine whether two of the numbers sum to a given number K. For example, if the input is 8, 4, 1, 6 and K is 10, the answer is yes (4 and 6). A number may be used twice.

Input Format

The first line contains a single integer n , the length of list

The second line contains n space-separated integers, list[i].

The third line contains integer k.

Output Format

Print Yes or No.

Sample Input

7

0 1 2 4 6 5 3

1

Sample Output

Yes

Program:

```
a=int(input())
p=input()
b=list(map(int,p.split()))
count=0
c=int(input())
for i in range(len(b)):
    for j in range(i+1,len(b)):
        if (b[i]+b[j]==c):
            print("Yes")
            count=1
```

```
break
```

```
if count==1:
```

```
    break
```

```
if count==0:
```

```
    print("No")
```

	Input	Expected	Got	
✓	5 8 9 12 15 3 11	Yes	Yes	✓
✓	6 2 9 21 32 43 43 1 4	No	No	✓
✓	6 13 42 31 4 8 9 17	Yes	Yes	✓

Passed all tests! ✓

4. Given an array `nums` containing n distinct numbers in the range $[0, n]$, return *the only number in the range that is missing from the array*.

Example 1:

Input: `nums = [3,0,1]`

Output: 2

Explanation: $n = 3$ since there are 3 numbers, so all numbers are in the range $[0,3]$. 2 is the missing number in the range since it does not appear in `nums`.

Example 2:

Input: `nums = [0,1]`

Output: 2

Explanation: $n = 2$ since there are 2 numbers, so all numbers are in the range $[0,2]$. 2 is the missing number in the range since it does not appear in `nums`.

Example 3:

Input: `nums = [9,6,4,2,3,5,7,0,1]`

Output: 8

Explanation: $n = 9$ since there are 9 numbers, so all numbers are in the range $[0,9]$. 8 is the missing number in the range since it does not appear in `nums`.

program:

```
def missingNumber(n):
```

```

count=0

flag=0

p=len(n)-1

for i in range(p):

    count+=1

    if count not in n:

        flag=1

    if flag==1:

        break

    if flag==1:

        return count

else:

    return n[p]+1

```

	Test	Expected	Got	
✓	print(search([-1,0,3,5,9,12],9))	4	4	✓
✓	print(search([-1,0,3,5,9,12],2))	-1	-1	✓

Passed all tests! ✓

5. Given an array of integers `nums` which is sorted in ascending order, and an integer `target`, write a function to search `target` in `nums`. If `target` exists, then return its index. Otherwise, return `-1`.

You must write an algorithm with $O(\log n)$ runtime complexity.

Example 1:

`Input: nums = [-1,0,3,5,9,12], target = 9`

`Output: 4`

`Explanation: 9 exists in nums and its index is 4`

Example 2:

Input: nums = [-1,0,3,5,9,12], target = 2
Output: -1
Explanation: 2 does not exist in nums so return -1

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
- $-10^4 < \text{nums}[i], \text{target} < 10^4$
- All the integers in `nums` are **unique**.
- `nums` is sorted in ascending order.

Programdef search(num: list[int], target: int) -> int:

 count=0

 flag=0

 for i in range(len(num)):

 if num[i]==target:

 count=i

 flag=1

 break

 if flag==1:

 return count

 else:

 return -1

	Test	Expected	Got	
✓	print(missingNumber([3,0,1]))	2	2	✓
✓	print(missingNumber([0,1]))	2	2	✓
✓	print(missingNumber([9,6,4,2,3,5,7,0,1]))	8	8	✓

Passed all tests! ✓

6. Write a Python program for binary search.

For example:

Input	Result
1,2,3,5,8 6	False
3,5,9,45,42 42	True

Program:

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

```
b=int(input())
```

```
c=0
```

```
flag=0
```

```
d=len(a)
```

```
a.sort()
```

```
while c<d:
```

```
    p=(c+d)//2
```

```
    if a[p]==b:
```

```
        print("True")
```

```
        flag=1
```

```
        break
```

```
    elif b<a[p]:
```

```
        d=p
```

```
    else:
```

```
        c=p+1
```

```

if flag==0:
    print("False")

```

	Input	Expected	Got	
✓	1,2,3,5,8 6	False	False	✓
✓	3,5,9,45,42 42	True	True	✓
✓	52,45,89,43,11 11	True	True	✓

Passed all tests! ✓

7. Given a list, find peak element in it. A peak element is an element that is greater than its neighbors.

An element $a[i]$ is a peak element if

$A[i-1] \leq A[i] \geq A[i+1]$ for middle elements. $[0 < i < n-1]$

$A[i-1] \leq A[i]$ for last element $[i=n-1]$

$A[i] \geq A[i+1]$ for first element $[i=0]$

Input Format

The first line contains a single integer n , the length of A .

The second line contains n space-separated integers, $A[i]$.

Output Format

Print peak numbers separated by space.

Sample Input

5

8 9 10 2 6

Sample Output

10 6

Program:

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

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

```
c=[]
```

```
d=len(b)-1
```

```
if a>1:
```

```
    if b[0]>b[1]:
```

```
        c.append(b[0])
```

```
    if b[d]>b[d-1]:
```

```
        c.append(b[d])
```

```
for i in range(1,d-1):
```

```
    m=i-1
```

```
    n=i+1
```

```
    if b[i]>b[m] and b[i]>b[n]:
```

```
        c.append(b[i])
```

```
c.sort(reverse=True)
```

```
print(*c)
```

	Input	Expected	Got	
✓	7 15 7 10 8 9 4 6	15 10 9 6	15 10 9 6	✓
✓	4 12 3 6 8	12 8	12 8	✓

Passed all tests! ✓

8. Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

Input Format:

The first line contains S1.

The second line contains S2.

The third line contains N.

Output Format:

The first line contains the N characters present in S1 which are also present in S2.

Boundary Conditions:

$2 \leq N \leq 10$

$2 \leq \text{Length of } S1, S2 \leq 1000$

Example Input/Output 1:

Input:

```
abcbde
cdefghbb
3
```

Output:

```
bcd
```

Note:

b occurs twice in common but must be printed only once.

Program:

```
a=input()
```

```
b=input()
```

```
c=int(input())
```

```
d=""
```

```
count=0
```

```
for i in a:
```

```
    if count>=c:
```

```
        break
```

```
    if i in b and i not in d:
```

```
        d+=i
```

```
    count+=1
```

```
print(d)
```

	Input	Expected	Got	
✓	abcbde cdefghbb 3	bcd	bcd	✓

Passed all tests! ✓

9. String should contain only the words are not palindrome.

Sample Input 1

```
Malayalam is my mother tongue
```

Sample Output 1

```
is my mother tongue
```

program:

```
w=input().split(' ')
```

```
u=""
```

```
for i in w:
```

```
    i=i.lower()
```

```
    if i!=i[::-1]:
```

```
        u+=i+" "
```

```
print(u)
```

	Input	Expected	Got	
✓	Malayalam is my mother tongue	is my mother tongue	is my mother tongue	✓

Passed all tests! ✓

10. Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints

1<= string length <= 200

Sample Input 1

```
experience  
enc
```

Sample Output 1

```
xpri
```

program:

```
a=input()
```

```
b=input()
```

```
c=""
```

```
for i in a:
```

```
    if i not in b:
```

```
        c+=i
```

```
print(c)
```

	Input	Expected	Got	
✓	experience enc	xpri	xpri	✓

Passed all tests! ✓

WEEK-11-MCQ-Bubble and Merge Sort

1.What is the primary advantage of the divide-and-conquer approach in sorting algorithms?

- a.It only works on small datasets
- b.It simplifies the sorting process by using only one pass
- c.**It allows for efficient parallel processing and sorting of data**
- d.It avoids the need for recursion

2Which of the following best describes the term "sorting" in computer science?

- a.Merging two datasets
- b.Removing duplicates from a list
- c.**Arranging data in a specific order**
- d.Finding a specific element in a list

3.Which of the following is not an in-place sorting algorithm?

- a.**Merge sort**
- b.Heap sort
- c.Selection sort
- d.Quick sort

4.What is sorting in the context of computer science?

- a.Inserting data into a list
- b.Deleting data from a list
- c.**Arranging data in a particular format**
- d.Searching for data in a list

5.Very slow way of sorting is_____

- a.Quick sort
- b.**Bubble sort**
- c.Heap sort
- d.Insertion sort

6.What is one of the first steps in a divide-and-conquer algorithm like Merge Sort?

- a.Comparing each element with the others
- b.Sorting the entire list sequentially
- c.Combining sorted sublists
- d.**Dividing the input into smaller subproblems**

7.What is mean by stable sorting algorithm?

- a.A sorting algorithm is stable if it doesn't preserve the order of duplicate keys
- b.A sorting algorithm is stable if it preserves the order of non-duplicate keys
- c.A sorting algorithm is stable if it preserves the order of duplicate keys
- d.A sorting algorithm is stable if it preserves the order of all keys

8.Which of the following is a key reason for the importance of sorting algorithms?

- a.Sorting makes it harder to search for items
- b.Sorting is rarely used in programming
- c.Sorting helps in finding duplicates quickly
- d.Sorting decreases the efficiency of selection operations

9.What type of problems can sorting help solve efficiently?

- a.Selection
- b.Searching
- c.Duplicates
- d.All of the above

10.In Merge Sort, what happens after the two halves of the list are sorted?

- a.They are compared element by element
- b.They are split again into smaller sublists
- c.They are discarded
- d.They are combined to form a single sorted list

11.The process of placing or rearranging a collection of elements into a particular order is known as

- a.Searching
- b.Sorting
- c.Rearranging
- d.Merging

12.What is one advantage of sorting a list before performing a search operation?

- a.It allows for faster searching
- b.It increases the number of comparisons needed
- c.It makes the search operation slower
- d.It has no effect on the search operation

13.Which algorithm typically follows a divide-and-conquer structure?

- a.Bubble Sort
- b.Merge Sort
- c.Binary Search
- d.Linear Search

14.In the context of sorting, what does the divide-and-conquer approach involve?

- a.Dividing the input into parts, solving each part, and combining the solutions
- b.Sorting data in a single pass
- c.Sorting data sequentially
- d.Rearranging data without sorting

15.What is one of the key advantages of using the built-in sorted() function in Python?

- a.It sorts data out of the box efficiently
- b.It is less efficient than custom sorting algorithms
- c.It only works with integer arrays
- d.It requires external libraries

1.Given a list of integers, sort the array in ascending order using the *Bubble Sort* algorithm above. Once sorted, print the following three lines:

1. List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.
2. First Element: firstElement, the *first* element in the sorted list.
3. Last Element: lastElement, the *last* element in the sorted list.

For example, given a worst-case but small array to sort: a=[6,4,1]. It took 3 swaps to sort the array. Output would be

```
Array is sorted in 3 swaps.  
First Element: 1  
Last Element: 6
```

Input Format

The first line contains an integer,n , the size of the list a .

The second line contains n, space-separated integers a[i].

Constraints

- $2 \leq n \leq 600$
- $1 \leq a[i] \leq 2 \times 10^6$.

Output Format

You must print the following three lines of output:

1. List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.
2. First Element: firstElement, the *first* element in the sorted list.
3. Last Element: lastElement, the *last* element in the sorted list.

Sample Input 0

```
3  
1 2 3
```

Sample Output 0

List is sorted in 0 swaps.

First Element: 1

Last Element: 3

Program:

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

```
count=0
```

```

b=list(map(int,input().split()))
for i in range(len(b)):
    for j in range(i+1,len(b)):
        if b[i]>b[j]:
            b[i],b[j]=b[j],b[i]
            count+=1
print("List is sorted in %d swaps."%count)
print("First Element: %d"%b[0])
print("Last Element: %d"%b[-1])

```

	Input	Expected	Got	
✓	3 3 2 1	List is sorted in 3 swaps. First Element: 1 Last Element: 3	List is sorted in 3 swaps. First Element: 1 Last Element: 3	✓
✓	5 1 9 2 8 4	List is sorted in 4 swaps. First Element: 1 Last Element: 9	List is sorted in 4 swaps. First Element: 1 Last Element: 9	✓

Passed all tests! ✓

2. To find the frequency of numbers in a list and display in sorted order.

Constraints:

$1 \leq n, arr[i] \leq 100$

Input:

1 68 79 4 90 68 1 4 5

output:

1 2

4 2

5 1

68 2

79 1

90 1

For example:

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

Answer:(penalty regime: 0 %)

```
a=list(map(int,input().split()))
for i in range(len(a)):
    for j in range(len(a)):
        if a[i]>a[j]:
            a[i],a[j]=a[j],a[i]
b=set(a)
for i in b:
    print(i,a.count(i))
```

	Input	Expected	Got	
✓	4 3 5 3 4 5	3 2 4 2 5 2	3 2 4 2 5 2	✓
✓	12 4 4 4 2 3 5	2 1 3 1 4 3 5 1 12 1	2 1 3 1 4 3 5 1 12 1	✓
✓	5 4 5 4 6 5 7 3	3 1 4 2 5 3 6 1 7 1	3 1 4 2 5 3 6 1 7 1	✓

Passed all tests! ✓

3. Write a Python program to sort a list of elements using the merge sort algorithm.

For example:

Program:

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

4. Given an array of integers `arr`, replace each element with its rank.

The rank represents how large the element is. The rank has the following rules:

- Rank is an integer starting from 1.
- The larger the element, the larger the rank. If two elements are equal, their rank must be the same.
- Rank should be as small as possible.

Example 1:

Input: arr = [40,10,20,30]

Output: [4,1,2,3]

Explanation: 40 is the largest element. 10 is the smallest. 20 is the second smallest. 30 is the third smallest.

Example 2:

Input: arr = [100,100,100]

Output: [1,1,1]

Explanation: Same elements share the same rank.

Example 3:

Input: arr = [37,12,28,9,100,56,80,5,12]

Output: [5,3,4,2,8,6,7,1,3]

Constraints:

- $0 \leq \text{arr.length} \leq 10^5$
- $-10^9 \leq \text{arr}[i] \leq 10^9$

Program:

```
def arrayRankTransform( arr: list[int] ) -> list[int]:
```

```
    a=list(set(arr))
```

```
    a.sort()
```

```
    b={} 
```

```
    count=0
```

```
    for i in a:
```

```
        count+=1
```

```
b.update({i:count})
```

```
p=[]
```

```
for i in arr:
```

```
    p.append(b[i])
```

```
return p
```

	Test	Expected	Got	
✓	print(arrayRankTransform([40,10,20,30]))	[4, 1, 2, 3]	[4, 1, 2, 3]	✓
✓	print(arrayRankTransform([100,100,100]))	[1, 1, 1]	[1, 1, 1]	✓
✓	print(arrayRankTransform([37,12,28,9,100,56,80,5,12]))	[5, 3, 4, 2, 8, 6, 7, 1, 3]	[5, 3, 4, 2, 8, 6, 7, 1, 3]	✓

Passed all tests! ✓

5. Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. You read an list of numbers. You need to arrange the elements in ascending order and print the result. The sorting should be done using bubble sort.

Input Format: The first line reads the number of elements in the array. The second line reads the array elements one by one.

Output Format: The output should be a sorted list.

Program:

```
a=input()
```

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

```
for i in range(len(b)):
```

```
    for j in range(len(b)):
```

```
        if b[i]<b[j]:
```

```
            b[i],b[j]=b[j],b[i]
```

```
print(*b)
```

	Input	Expected	Got	
✓	6 3 4 8 7 1 2	1 2 3 4 7 8	1 2 3 4 7 8	✓
✓	6 9 18 1 3 4 6	1 3 4 6 9 18	1 3 4 6 9 18	✓
✓	5 4 5 2 3 1	1 2 3 4 5	1 2 3 4 5	✓

Passed all tests! ✓

6. Given an array of integers `nums`, sort the array in **increasing** order based on the frequency of the values. If multiple values have the same frequency, sort them in **decreasing** order.

Print the *sorted array*.

Example 1:

Input:

6
1 1 2 2 2 3

Output:

3 1 1 2 2 2

Explanation: '3' has a frequency of 1, '1' has a frequency of 2, and '2' has a frequency of 3.

Example 2:

Input:

5
2 3 1 3 2

Output:

1 3 3 2 2

Explanation: '2' and '3' both have a frequency of 2, so they are sorted in decreasing order.

Example 3:

Input:

9
-1 1 -6 4 5 -6 1 4 1

Output:

5 -1 4 4 -6 -6 1 1 1

Constraints:

- `1 <= nums.length <= 100`
- `-100 <= nums[i] <= 100`

Program:

```
a=input()  
  
b=list(map(int,input().split()))  
  
c=set(b)  
  
d={}  
  
for i in c:  
  
    co=b.count(i)  
  
    d.update({i:co})  
  
b.clear()  
  
b=dict(sorted(d.items(),key=lambda item:(item[1],-item[0])))  
  
p=[]  
  
for i in b:  
  
    count=b[i]  
  
    for j in range(b[i]):  
  
        p.append(i)  
  
print(*p)
```

	Input	Expected	Got	
✓	6 1 1 2 2 2 3	3 1 1 2 2 2	3 1 1 2 2 2	✓
✓	5 2 3 1 3 2	1 3 3 2 2	1 3 3 2 2	✓
✓	9 -1 1 -6 4 5 -6 1 4 1	5 -1 4 4 -6 -6 1 1 1	5 -1 4 4 -6 -6 1 1 1	✓

Passed all tests! ✓

7. Given an integer array `nums` sorted in **non-decreasing** order, return *an array of the squares of each number sorted in non-decreasing order*.

Example 1:

Input: `nums = [-4, -1, 0, 3, 10]`

Output: `[0, 1, 9, 16, 100]`

Explanation: After squaring, the array becomes `[16, 1, 0, 9, 100]`.

After sorting, it becomes `[0, 1, 9, 16, 100]`.

Example 2:

Input: `nums = [-7, -3, 2, 3, 11]`

Output: `[4, 9, 9, 49, 121]`

Program:

```
def sortedSquares(n: list[int]) -> list[int]:  
    a=[]  
    for i in n:  
        a.append(i**2)  
    for i in range(0,len(a)):  
        for j in range(i+1,len(a)):  
            if a[i]>a[j]:  
                a[i],a[j]=a[j],a[i]  
    return a
```

	Test	Expected	Got	
✓	<code>print(sortedSquares([-4, -1, 0, 3, 10]))</code>	<code>[0, 1, 9, 16, 100]</code>	<code>[0, 1, 9, 16, 100]</code>	✓
✓	<code>print(sortedSquares([-7, -3, 2, 3, 11]))</code>	<code>[4, 9, 9, 49, 121]</code>	<code>[4, 9, 9, 49, 121]</code>	✓

Passed all tests! ✓

8. The problem is that we want to reverse a array in $O(N)$ linear time complexity and we want

the algorithm to be in-place as well!

For example: input is `[1,2,3,4,5]` then the output is `[5,4,3,2,1]`

Input

5

1 2 3 4 5

Output

5 4 3 2 1

Program:

```
a=input()  
b=list(map(int,input().split()))  
print(*b[::-1])
```

	Input	Expected	Got	
✓	5 1 2 3 4 5	5 4 3 2 1	5 4 3 2 1	✓
✓	10 0 2 4 6 8 1 3 5 7 9	9 7 5 3 1 8 6 4 2 0	9 7 5 3 1 8 6 4 2 0	✓

Passed all tests! ✓

9. Given an integer array `nums`, return an integer array `counts` where `counts[i]` is the number of smaller elements to the right of `nums[i]`.

Example 1:

Input: `nums = [5,2,6,1]`

Output: `[2,1,1,0]`

Explanation:

To the right of 5 there are 2 smaller elements (2 and 1).

To the right of 2 there is only 1 smaller element (1).

To the right of 6 there is 1 smaller element (1).

To the right of 1 there is 0 smaller element.

Example 2:

Input: `nums = [-1]`

Output: `[0]`

Example 3:

Input: `nums = [-1,-1]`

Output: `[0,0]`

Program:

```
def countSmaller(num: list[int]) -> list[int]:
```

```
    a=[]
```

```
    for i in range(0,len(num)):
```

```
        count=0
```

```
        for j in range(i+1,len(num)):
```

```
            if num[i]>num[j]:
```

```

        count+=1
        a.append(count)
    return a

```

	Test	Expected	Got	
✓	print(countSmaller([5,2,6,1]))	[2, 1, 1, 0]	[2, 1, 1, 0]	✓
✓	print(countSmaller([50,20,60,10]))	[2, 1, 1, 0]	[2, 1, 1, 0]	✓
✓	print(countSmaller([-1]))	[0]	[0]	✓
✓	print(countSmaller([-1, -1]))	[0, 0]	[0, 0]	✓

Passed all tests! ✓

10. Objective:

Develop a Python program to find the k-th maximum value in a given list of integers. The program should handle various edge cases, including lists with duplicate values, empty lists, and invalid values of k. The k-th maximum value refers to the k-th largest distinct element in the list.

Background:

Finding the k-th maximum value in a list is a common problem in computer science, often encountered in fields like data analysis, competitive programming, and software development. This problem requires an understanding of sorting algorithms, data structures, and efficient problem-solving techniques. By solving this problem, one gains insights into how to handle large datasets and optimize performance in practical applications.

Problem Description:

Given a list of integers, the task is to determine the k-th maximum value in the list. The program should meet the following requirements:

1. Input:

- A list of integers, which can contain both positive and negative values.
- An integer k, representing the position of the maximum value to find.

2. Output:

- The k-th maximum value in the list.

- If k is greater than the number of distinct elements in the list or if the list is empty, the program should return an appropriate message indicating the error.

Constraints:

- The list may contain duplicate values.
- The value of k should be a positive integer.
- The list may contain up to 10^6 elements, and each element can be as large as 10^9 in magnitude.

Examples:

Consider the following examples for better understanding:

1. Example 1:

- **Input:** list = [3, 1, 5, 4, 2], k = 2
- **Output:** 4
- **Explanation:** The distinct elements in the list are [1, 2, 3, 4, 5]. The 2nd maximum value is 4.

2. Example 2:

- **Input:** list = [7, 7, 7, 7, 7], k = 1
- **Output:** 7
- **Explanation:** The distinct elements in the list are [7]. The 1st maximum value is 7.

3. Example 3:

- **Input:** list = [2, 1, 2, 1, 2], k = 3
- **Output:** -1
- **Explanation:** The distinct elements in the list are [1, 2]. There is no 3rd maximum value.

For example:

Input	Result
5 3 1 5 4 2 2	4
6 7 7 7 7 7 7 1	7
10	-1

Input	Result
2 1 2 1 2 1 2 1 2 1 3	

Program:

```
a=input()
b=list(set(map(int,input().split())))
c=int(input())
if c>len(b):
    print(-1)
else:
    for i in range(len(b)):
        for j in range(len(b)):
            if b[i]>b[j]:
                b[i],b[j]=b[j],b[i]
    for i in range(c):
        count=b[i]
    print(count)
```

	Input	Expected	Got	
✓	5 3 1 5 4 2 2	4	4	✓
✓	6 7 7 7 7 7 7 1	7	7	✓
✓	10 2 1 2 1 2 1 2 1 2 1 3	-1	-1	✓

Passed all tests! ✓

WEEK-12-MCQ-Files

1.Which of the following methods is used to read a single line from a file?

- a.readline() b.readline(size) c.read() d.readlines()

2.Which of the following is data file?

- a.none of the given choices b.Both text and binary file
c.Binary file d.Text file

3.How do you check if a file is closed in Python?

- a.file.closed b.file.closed() c.file.isclosed() d.file.is_closed()

4.To open a file a.txt for reading, we use _____

- a.infile = open(file = "a.txt", "r")
b. infile = open("a.txt", "r")
c.infile = open("a.txt", "r")
d.infile = open(file = "a.txt", "r")

5.what happens if you try to open a file in read mode ('r') that does not exist?

- a.An error is raised b.The operation is ignored
c.An empty file object is returned d.A new file is created

6.Which method is used to close a file in Python?

- a.terminate() b.end() c.close() d.finish()

7.What does the flush() method do in file handling?

- a.Moves the file cursor to the beginning
b Deletes the file content
c.Closes the file
d.Flushes the internal buffer to the file

8.Which of the following modes opens a file for writing in binary format?

- a.rb b.r+ c.wb d.w

9.What does the following code do?

fo = open("foo.txt", "w")

- aOpens an existing file in read mode b.Opens a file in binary mode
cCreates a new file for writing d.Appends to an existing file

10.What will the read() method return if used on an empty file?

- a.None b.An empty string c.EOFError d.0

11.What is the difference between r+ and w+ modes in file handling?

a.r+ is for binary read and write, w+ is for text read and write

b.r+ is for read and write, w+ is for append only

c.r+ is for read and append, w+ is for read and write

d.r+ is for read and write without truncating, w+ is for read and write with truncating

12.What will be the output of the following code if the file "example.txt" contains "Python Programming"?

```
f = open("example.txt", "r")
print(f.read(6))    f.close()
```

- a.An error is raised b.Python c.Python Programming d.Program

13. file_object.seek(offset [, reference_point])

What is the default, the value of reference_point?

- a.1 b.0 c.2 d.null e.garbage

14.What happens if you try to open a file in write mode ('w') and the file already exists?

a.It does nothing b.It raises an error

c.It appends data to the file d.It clears the old data and starts fresh

15.How can you read a file line by line efficiently in Python?

a.Using the readline() method b.Using the readlines() method

c.Using a for loop d.Using the read() method

Write a Python program to count the frequency of each word in a given text file.

Description:

1. Input:
 - String as input.
2. Output:
 - A list of words with their corresponding frequency count to be write in a file "output.txt"

Example:

- Input File Content:

```
apple orange apple banana apple orange
```

Output:

```
apple: 3
orange: 2
banana: 1
```

For example:

Test	Input	Result
<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	apple orange apple banana apple orange	apple: 3 banana: 1 orange: 2

Answer:(penalty regime: 0 %)

Feedback

```
n=input()
l=".join([c for c in n if c.isalnum() or c.isspace()])
l=l.split()
l=sorted(l,key=str.lower)
d={}
for i in l:
    d[i.lower()]=str(l.count(i))
s=""
for i in d:
    s+=(i+': '+d[i]+'\n')
f=open('output.txt','w')
f.write(s)
f.close()
```

	Test	Input
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	apple orange apple banana apple orange
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	Hello world! Hello everyone. Welcome to the world of pro
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	One fish two fish Red fish blue fish

Create a Python program to delete a specific line from a text file based on a given line number.

Description:

1. Input:

- A text file with multiple lines.
- A line number to delete.

2. Output:

- The updated file with the specified line removed in file "output.txt".

Example:

● Input File Content:

"Line one.

Line two.

Line three.

Line four."

2

Updated line two.

Output:

Line one.

Line three.

Line four.

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 2	Line one. Line three. Line four.

```
fi=input()  
n=int(input())  
o='output.txt'  
with open(fi,'r') as f:  
    l=f.readlines()  
l.remove(l[n-1])  
with open(o,'w')as f:  
    f.writelines(l)
```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 2	Line one. Line three. Line four.	Line one. Line three. Line four.	✓
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 3	Line A. Line B.	Line A. Line B.	✓

Passed all tests! ✓

Correct

Last for this submission - 1.00 / 1.00

Develop a Python program to copy the contents of one file to another file.

Description:

1. Input:
 - Source file and destination file names.
2. Output:
 - The content of the source file copied to the destination file.

For example:

Test	Input	Result

```

with open('output1.txt', 'r') as file:
    text = file.read()
    print(text)

```

input1.txt output1.txt	This is the source file. It contains multiple lines of text. Here is another line.
---------------------------	--

```

i=input()
o=input()
with open(i,'r')as f:
    with open(o,'a')as f1:
        f1.write(f.read())

```

	Test	Input	Expected	Got
✓	with open('output1.txt', 'r') as file: text = file.read() print(text)	input1.txt output1.txt	This is the source file. It contains multiple lines of text. Here is another line.	This It c Here
✓	with open('output2.txt', 'r') as file: text = file.read() print(text)	input2.txt output2.txt	Hello, world! Python programming is amazing. Let's copy this text to another file.	Hello Pyth Let'
✓	with open('output3.txt', 'r') as file: text = file.read() print(text)	input3.txt output3.txt	Single line.	Sing

Passed all tests! ✓

Write a Python program to reverse the contents of a specific line in a text file based on a given line number.

Description:

1. Input:
 - A text file with multiple lines.
 - A line number to reverse.
2. Output:
 - The updated file with the specified line's contents reversed in file "output.txt".

Example:

- Input File Content:

"Line one.
Line two.
Line three.

Line four."

3

Output:

Line one.
Line two.
eerht eniL.
Line four.

For example:

Test	Input	Result
<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	input1.txt 3	Line one. Line two. eerht eniL. Line four.

```
f1=input()  
n=int(input())  
with open(f1,'r')as f:  
    l=f.readlines()  
l[n-1]=l[n-1][:-1]  
l[n-1]=l[n-1][2:]+\'.\n'  
with open('output.txt','w') as f:  
    f.writelines(l)
```

	Test	Input	Expected	Got	
✓	<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	input1.txt 3	Line one. Line two. eerht eniL. Line four.	Line one. Line two. eerht eniL. Line four.	✓
✓	<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	input2.txt 2	Line A. B eniL. Line C.	Line A. B eniL. Line C.	✓

Passed all tests! ✓

Create a Python program to write to a specific line in a text file, replacing the existing content of that line.

Description:

1. Input:
 - A text file with multiple lines.
 - A line number to write to.
 - New content for the specified line.
2. Output:
 - The updated file with the specified line replaced by the new content in file "output.txt".

Example:

- Input File Content:


```
"Line one.  
Line two.  
Line three.  
Line four."  
2
```

Updated line two.

Output:

```
Line one.  
Updated line two.  
Line three.  
Line four.
```

For example:

Test	Input	Result
<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	<pre>input1.txt 2 Updated line two.</pre>	<pre>Line one. Updated line two. Line three. Line four.</pre>

```
i=input()
n=int(input())
```

```

s=input()
with open(i,'r')as f:
    l=f.readlines()
l[n-1]=s+'\n'
with open('output.txt','w')as f:
    f.writelines(l)

```

	Test	Input	Expected	Got
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 2 Updated line two.	Line one. Updated line two. Line three. Line four.	Line one. Updated line two. Line three. Line four.
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 2 Line B Updated.	Line A. Line B Updated. Line C.	Line A. Line B Updated. Line C.

Passed all tests! ✓

Develop a Python program to identify and print all palindrome words from a given text file.

Description:

1. Input:
 - A text file containing multiple words.
2. Output:
 - A list of palindrome words found in the file name as 'output.txt'.

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text)	input1.tx t	madam arora malayala m

```

i=input()
with open(i,'r') as f:

```

```

l=f.read()
l=l.split()
s=""
for i in l:
    if i==i[::-1]:
        s+=i+"\n"
with open('output.txt','w')as f:
    f.write(s)

```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt	madam arora malayalam	madam arora malayalam	✓

Passed all tests! ✓

Develop a Python program to read a text file and count the total number of words in the file.

Description:

1. Input:
 - A text file containing several lines of text.
 - File name you should get as input.
2. Output:
 - The total number of words in the file.

For example:

Input	Result
input2.txt	Total words: 14
input3.txt	Total words: 15

```

import re
i=input()

```

```

with open(i,'r')as f:
    l=f.readlines()
l1=[]
for i in l:
    x=i.split()
    l1.extend(x)
print('Total words:',len(l1))

```

	Input	Expected	Got	
✓	input1.txt	Total words: 6	Total words: 6	✓
✓	input2.txt	Total words: 14	Total words: 14	✓
✓	input3.txt	Total words: 15	Total words: 15	✓

Passed all tests! ✓

Develop a Python program to read a specific line from a text file based on a given line number.

Description:

1. Input:
 - A text file with multiple lines.
 - A line number to read.
2. Output:
 - The content of the specified line.

input1.txt:

Line one.
 Line two.
 Line three.
 Line four.

For example:

Input	Result

input1.tx t 3	Line three.
---------------------	----------------

```
i=input()
n=int(input())
with open(i,'r')as f:
    l=f.readlines()
print(l[n-1])
```

	Input	Expected	Got	
✓	input1.txt 3	Line three.	Line three.	✓
✓	input2.txt 3	Line C.	Line C.	✓

Passed all tests! ✓

Write a Python program to append a new line at a specific position in a text file, shifting existing lines down.

Description:

1. Input:
 - A text file with multiple lines.
 - A line number to insert the new line at.
 - New content for the new line.
2. Output:
 - The updated file with the new line inserted at the specified position, shifting the existing lines down in file "output.txt".

Example:

- Input File Content:
 "Line one.
 Line two.
 Line three.

Line four."

3

Inserted line..

Output:

Line one.

Line two.

Inserted line.

Line three.

Line four.

For example:

Test	Input	Result
<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	input1.txt 3 Inserted line.	Line one. Line two. Inserted line. Line three. Line four.

```
i=input()  
n=int(input())  
s=input()  
s+="\n'  
with open(i,'r')as f:  
    l=f.readlines()  
if n-1==len(l):  
    l[-1]+="\n'  
l.insert(n-1,s)  
with open('output.txt','w')as f:  
    f.writelines(l)
```

Test	Input	Expected	Got	
✓ with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 3 Inserted line.	Line one. Line two. Inserted line. Line three. Line four.	Line one. Line two. Inserted line. Line three. Line four.	✓
✓ with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 4 Inserted line D.	Line A. Line B. Line C. Inserted line D.	Line A. Line B. Line C. Inserted line D.	✓

Passed all tests! ✓

Create a Python program to find the longest word in a text file.

- Input:
 - A text file containing multiple lines of text.
- Output:
 - The longest word in the file.

For example:

Input	Result
input1.txt	Longest word: containing

```
i=input()
with open(i,'r') as f:
    l=f.read()
l=l.split()
s=max(l,key=len)
print('Longest word:',s)
```

	Input	Expected	Got
✓	input1.txt	Longest word: containing	Longest word: containing
✓	input2.txt	Longest word: thousand	Longest word: thousand
✓	input3.txt	Longest word: supercalifragilisticexpialidocious	Longest word: supercalifragilistic

Passed all tests! ✓