

Lecture 1 : Flowchart Practice Questions

1-Tut : **Average of three numbers**

[Send Feedback](#)

You are given three numbers. You need to calculate and print their average value. Draw a flowchart for this process.

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

2-Tut: **Check Number**

[Send Feedback](#)

You are given a single number. You need to print one of the following outputs according to the number's nature.

Print 1, if the number is positive

Print -1, if it's negative

Print 0, if it's equal to 0

Draw a flowchart for this process.

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

3-Tut : **Valid Triangle**

[Send Feedback](#)

You are given 3 numbers. Each number represents the length of a line. You need to figure out whether these lines can form a valid triangle.

If a valid triangle can be formed, print "Yes", otherwise print "No".

Draw a flowchart for this process

A triangle is a **valid triangle**, If and only If, the sum of any two sides of a triangle is greater than the third side. For Example, let A, B and C are three sides of a triangle. Then, $A + B > C$, $B + C > A$ and $C + A > B$

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

4-Tut : **Find Product**

[Send Feedback](#)

You are given a single non-negative integer, N. You need to calculate and print N factorial (N!)

N factorial is defined as the product of all integers from 1 to N (both inclusive)

Draw a flowchart for this process

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

5-Tut : **Print Even Numbers**

[Send Feedback](#)

You are given a single positive integer, N. You need to print all even integers that occur between 1 and N (both inclusive).

Draw a flowchart for this process

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

6-Assignment : **Check triangle**

[Send Feedback](#)

You are given the lengths of 3 sides of a valid triangle. You need to print any one of the following outputs depending on the triangle's nature.

Print 1, if the triangle is equilateral

Print 0, if it's isosceles

Print -1, if it's scalene

Draw a flowchart for this process.

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

7-As: **Sum of evens**

[Send Feedback](#)

You are given a single positive integer, N. You need to calculate and print the sum of all even numbers till N(inclusive)

Draw a flowchart for this process

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

8-As : **Find GCD**

[Send Feedback](#)

You are given two numbers. You need to calculate and print their greatest common divisor (GCD).

Draw a flowchart for this process.

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

9-As : **All primes**

[Send Feedback](#)

You are given a single positive integer, N. You need to print all prime numbers that occur in the range 1 to N (both inclusive).

Draw a flowchart for this process

Note : You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

10-As : **All fibonacci numbers**

[Send Feedback](#)

You are given a single non-negative integer, N. You need to print all numbers that:

- (i) occur in the range 0 to N (both inclusive)
- (ii) are a part of the fibonacci sequence

Draw a flowchart for this process

Note 1: The first two terms of the fibonacci sequence are 0 and 1.

Note 2: You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

11-As : **Member of Fibonacci**

[Send Feedback](#)

You are given a single non-negative integer, N. You need to find out whether N is a part of the fibonacci sequence.

Print "Yes" if it is and "No" if it's not.

Draw a flowchart for this process

Note 1: The first two terms of the fibonacci sequence are 0 and 1.

Note 2: You don't need to submit the problem. Just attempt in your notebook and ask doubts if you need help.

Lecture 2 : Introduction to Python Practice Questions

1-Tut : **Output Question**

[Send Feedback](#)

What will the following code segment print?

```
print("Career")
```

```
print("Labs")
```

Options

CareerLabs

Career Labs

Career Labs(in next line)

"Career""Labs"

Correct Answer : C

2-Tut : **Output Question**

[Send Feedback](#)

What will be the output of the given code segment?

```
a = 10
```

```
b = 20
```

```
multiple = a*b
```

```
print("multiple")
```

Options

20

200

multiple

None of the above

Correct Answer C

3-Tut : **Output Question**

[Send Feedback](#)

What will be the output of the given code segment?

```
a = 10
```

```
b = 20
```

```
multiple = a*b
```

```
print(multiple)
```

Options

20
200
multiple
None of the above

Correct Answer: B

4-Tut : Python Variable Name

[Send Feedback](#)

Select correct variable name(s) -

One or more options may be correct

Options

var1
var_1
1var
_var1

Correct Answer: A,B,D

5-Tut : Python Variables

[Send Feedback](#)

What will be the result of the following code in Python ?

```
x = 10  
x = "abcd"  
print(x)
```

Options

10
abcd
Error

Correct Answer B

Note : variables are not same as in c++/ Java etc.. , in python x will store the address of the variable where 10 is stored , also in cpp/java we can;t update the variable to another data type but in python we can , now x will store the address of variable where “abcd” is stored.

6-Tut : Python Variable Types

[Send Feedback](#)

Consider the python code below -

```
x = "abcd"  
x = 10
```

What is the type of x after the code executes ?

Options

str

int

Correct Answer B

7-Tut : Check for Equality

[Send Feedback](#)

Will id1 and id2 have the same value?

```
a = 10
```

```
id1 = id(a)
```

```
b = a + 2-2
```

```
id2 = id(b)
```

Options

Yes

No

Can't say

Correct Answer : Yes

Note :For Numbers in range [-5,256] python does the auto optimisation : (no new space for same value,

Ex : a = 10, b = 10 : both a and b will contain the same address(& of variable where 10 is stored)

Also in case of updates like a = a+1; (in python id will be changed to new id where 11 is stored, but in case of c++/ java etc.. same variable with updated value)

Limit of integer, Arithmetic operators

Note : in python there is no limit bcz here declaring variable is not allocated space also here variable stores the address of the "content variable".

8-Tut : Output Question

[Send Feedback](#)

What will be the output of the following statement?

```
print(17//10)
```

Options

1.7

1

2

None of the above

Correct Answer : B

9-Tut : Output Question

[Send Feedback](#)

What will be the output of the following statement?

```
print(17/10)
```

Options

1.7

1

2

None of the above

Correct Answer : A

Taking Inputs :

10-Tut : Output Question

[Send Feedback](#)

What will be the output of the code if the input provided is 40 and 57 ?

```
a = input()
```

```
b = input()
```

```
C = a+b
```

```
print(C)
```

Note that the double quotes given in the options is to denote that it is a string. It wouldn't appear in the final output.

Options

97

"40+57"

"4057"

None of the above

Correct Answer : C

11-Tut : Output Question

[Send Feedback](#)

What will be the output of the code if input provided is 40 and 57 ?

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

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

```
C = a+b
```

```
print(C)
```

Options

97

"40+57"

"4057"

None of the above

Correct Answer : A

12-Tut : **Output Question**

[Send Feedback](#)

What will be the output of the code if input provided is “abc” and “def”?

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

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

```
C = a+b
```

```
print(C)
```

Options

abcdef

abc+def

Value Error

None of the above

Correct Answer : C

13-Tut : **Find average Marks**

[Send Feedback](#)

Write a program to input marks of three tests of a student (all integers). Then calculate and print the average of all test marks.

Input format :

3 Test marks (in different lines)

Output format :

Average

Sample Input 1 :

3

4

6

Sample Output 1 : 4.333333333333333

Sample Input 2 :

5

10

5

Sample Output 2 : 6.666666666666667

CODE :

1. `a = int(input())`
2. `b = int(input())`
3. `c = int(input())`
4. `print((a+b+c)/3)`

14-Ass : Find X raised to power N

[Send Feedback](#)

You are given two integers: X and N. You have to calculate X raised to power N and print it.

Input format:

The first line of input contains an integer X ($1 \leq X \leq 100$)

The second line of input contains an integer N ($1 \leq N \leq 10$)

Constraints:

Time Limit: 1 second

Output format:

The first and only line of output contains the result.

Sample Input:

10

4

Sample Output: 10000

1. [# Write your code here](#)
2. [X = int\(input\(\)\)](#)
3. [N = int\(input\(\)\)](#)
4. [print\(X**N\)](#)
- 5.

15-Ass: Arithmetic Progression

[Send Feedback](#)

You are given the first three entries of an arithmetic progression. You have to calculate the common difference and print it.

Input format:

The first line of input contains an integer a ($1 \leq a \leq 100$)

The second line of input contains an integer b ($1 \leq b \leq 100$)

The third line of input contains an integer c ($1 \leq c \leq 100$)

Constraints:

Time Limit: 1 second

Output format:

The first and only line of output contains the result.

Sample Input:

1

3

5

Sample Output: 2

1. [# Write your code here](#)
2. [a = int\(input\(\)\)](#)
3. [b = int\(input\(\)\)](#)
4. [c = int\(input\(\)\)](#)
5. [print\(c-b\)](#)

16-Ass: Rectangular Area

[Send Feedback](#)

You are given a rectangle in a plane. The coordinates of one of its diagonals are provided to you. You have to print the total area of the rectangle.

The coordinates of the rectangle are provided as four integral values: x_1 , y_1 , x_2 , y_2 . It is given that $x_1 < x_2$ and $y_1 < y_2$.

Input format:

The first line of input contains an integer x_1

The second line of input contains an integer y_1

The third line of input contains an integer x_2

The fourth line of input contains an integer y_2

Constraints:

$1 \leq x_1 \leq 10$

$1 \leq y_1 \leq 10$

$1 \leq x_2 \leq 10$

$1 \leq y_2 \leq 10$

Time Limit: 1 second

Output format:

The first and only line of output contains the result.

Sample Input:

```
1
1
3
3
```

Sample Output: 4

Explanation:

The given coordinates of the diagonal are $(x_1, y_1) = (1, 1)$ and $(x_2, y_2) = (3, 3)$.

The area of the rectangle can then easily be calculated as:

$$(3 - 1) * (3 - 1) = 2 * 2 = 4$$

1. **# Write your code here**
2. **$x_1 = \text{int}(\text{input}())$**
3. **$y_1 = \text{int}(\text{input}())$**
4. **$x_2 = \text{int}(\text{input}())$**
5. **$y_2 = \text{int}(\text{input}())$**
6. **$\text{print}((x_2 - x_1) * (y_2 - y_1))$**
- 7.

L3: Conditional and Loops Practice Question

1-Tut: Predict the Output

[Send Feedback](#)

Output of the following program will be :

```
n = 15
#Check If the number is between 1 to 10
if n>=1 and n<=10:
    print("too low")

#Check If the number is between 11 to 20
elif n>=10 and n<=20:
    print("medium")

#Check If the number is between 21 to 30
elif n>=20 and n<=30:
    print("large")
#Check if the number is greater than 30
else:
    print("too large")
```

Options

too low
medium
large
too large

Correct Answer : B

2-Tut : Predict the Output

[Send Feedback](#)

Output of the following program will be :

```
n = 10
#Check If the number is between 1 to 10
if n>=1 and n<=10:
    print("too low")

#Check If the number is between 10 to 20
elif n>=10 and n<=20:
    print("medium")

#Check If the number is between 20 to 30
elif n>=20 and n<=30:
    print("large")
```

```
#Check if the number is greater than 30
else:
```

```
    print("too large")
```

Options

too low

medium

large

too large

Correct Answer: A

3-Tut : **Figure out the output**

[Send Feedback](#)

What will the following code segment print?

```
x = 15
if x <= 15:
    print("Inside if")
else:
    print("Inside else")
```

Options

Inside If

Inside else

Inside If Inside else

Correct Answer: A

4-Tut : **Multiple Ifs**

[Send Feedback](#)

Consider the following piece of code -

```
x = 5
if x < 6:
    print("Hello")
if x == 5:
    print("Hi")
else:
    print("Hey")
```

Which of the above 3 print statement(s) will be executed?

Options print("Hello") , print("Hi") , print("Hey") , All 3 will execute

Correct Answer : Hello , Hi

Relational and Logical Operators, else if

5-Tut : **Check number**

[Send Feedback](#)

Given an integer n, find if n is positive, negative or 0.

If n is positive, print "Positive"

If n is negative, print "Negative"

And if n is equal to 0, print "Zero".

Input Format :

Integer n

Output Format :

"Positive" or "Negative" or "Zero" (without double quotes)

Constraints :

$-100 \leq n \leq 100$

Sample Input 1 : 10

Sample Output 1 : Positive

Sample Input 2 : -10

Sample Output 2 : Negative

1. # Read input as specified in the question
2. # Print output as specified in the question
3. `n = int(input())`
4. `if(n == 0):`
5. `print("Zero")`
6. `elif(n > 0):`
7. `print("Positive")`
8. `else:`
9. `print("Negative")`

Nested Conditional

6-Tut : **Conditional Question**

[Send Feedback](#)

What will the following code segment print?

```
if (10 < 0) and (0 < -10):
    print("A")
elif (10 > 0) or False:
    print("B")
else:
    print("C")
```

Options

A
B
C
B & C

Correct Answer : B

7-Tut : Conditional Question

[Send Feedback](#)

What will be the following code segment print?

if True or True:

 if False and True or False:

 print('A')

 elif False and False or True and True:

 print('B')

 else:

 print('C')

else:

 print('D')

Options

A
B
C
D
B & D

Correct Answer: B

While Loop

8-Tut : Sum of n numbers

[Send Feedback](#)

Given an integer n, find and print the sum of numbers from 1 to n.

Note : Use while loop only.

Input Format : Integer n

Output Format :Sum

Constraints :

1 <= n <= 100

Sample Input : 10

Sample Output : 55

```
1. # Read input as specified in the question
2. # Print output as specified in the question
3. n = int(input())
4. sum = 0
5. i = 1
6. while(i <= n):
7.     sum = sum + i
8.     i = i+1
9. print(sum)
```

9-Tut : Sum of Even Numbers

[Send Feedback](#)

Given a number N, print sum of all even numbers from 1 to N.

Input Format : Integer N

Output Format : Required Sum

Sample Input 1 : 6

Sample Output 1 : 12

```
1. ## Read input as specified in the question.
2. ## Print output as specified in the question.
3. N = int(input())
4. sum = 0
5. i = 2
6. while(i <= N):
7.     sum = sum + i
8.     i = i+ 2
9. print(sum)
```

Primality Checking

10-Tut: Predict the Output

[Send Feedback](#)

What will be the output of the following code segment?

```
i=0
while i<10:
    print(i)

    i=i+1
```

Options

Numbers from 0 to 9 will be printed

Only 0 will be printed

Indentation Error

None of the above

Correct Answer : C

11-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code segment?

```
i=0
```

```
while i<10:
```

```
    print(i)
```

```
    i = i+1
```

Options

Numbers from 0 to 9 will be printed

Infinite times 0 will be printed

Indentation Error

None of the above

Correct Answer : B

12-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code segment?

```
i=0
```

```
while i<10:
```

```
    print(i)
```

```
    i= i+1
```

Options

Numbers from 0 to 9 will be printed

Infinite times 0 will be printed

Indentation Error

None of the above

Correct Answer : A

Nested Loops

13-Tut : Fahrenheit to Celsius

[Send Feedback](#)

Given three values - Start Fahrenheit Value (S), End Fahrenheit value (E) and Step Size (W), you need to convert all Fahrenheit values from Start to End at the gap of W, into their corresponding Celsius values and print the table.

Input Format :

3 integers - S, E and W respectively

Output Format :

Fahrenheit to Celsius conversion table. One line for every Fahrenheit and corresponding Celsius value. On Fahrenheit value and its corresponding Celsius value should be separate by tab ("t")

Constraints :

$0 \leq S \leq 80$

$S \leq E \leq 900$

$0 \leq W \leq 40$

Sample Input 1:

0

100

20

Sample Output 1:

0 -17

20 -6

40 4

60 15

80 26

100 37

Sample Input 2:

20

119

13

Sample Output 2:

20 -6

33 0

46 7

59 15

72 22

85 29

98 36

111 43

Explanation For Input 2:

We need to start calculating the Celsius values for each of the Fahrenheit Value which starts from 20. So starting from 20 which is the given Fahrenheit start value, we need to compute its corresponding Celsius value which computes to -6. We print this information as <Fahrenheit Value> a tab space"\t" <Celsius Value> on each line for each step of 13 we take to get the next value of Fahrenheit and extend this idea till we reach the end that is till 119 in this case. You may or may not exactly land on the end value depending on the steps you are taking.

Read input as specified in the question

```
1. # Print output as specified in the question
2. S = int(input())
3. E = int(input())
4. W = int(input())
5.
6. while(S <= E):
7.     p = (S - 32)*5/9
8.     p = int(p)
9.     print(S,"t", p)
10.    S = S + W
```

14-Ass : Calculator

Send Feedback

Write a program that performs the tasks of a simple calculator. The program should first take an integer as input and then based on that integer perform the task as given below.

1. If the input is 1, then 2 integers are taken from the user and their sum is printed.
2. If the input is 2, then 2 integers are taken from the user and their difference(1st number - 2nd number) is printed.
3. If the input is 3, then 2 integers are taken from the user and their product is printed.
4. If the input is 4, then 2 integers are taken from the user and the quotient obtained (on dividing 1st number by 2nd number) is printed.
5. If the input is 5, then 2 integers are taken from the user and their remainder(1st number mod 2nd number) is printed.
6. If the input is 6, then the program exits.
7. For any other input, then print "Invalid Operation".

Note: Each answer in the next line.

Input format:

Take integers as input, in accordance with the description of the question.

Constraints:

Time Limit: 1 second

Output format:

The output lines must be as prescribed in the description of the question.

Sample Input:

3
1
2
4
4
2
1
3
2
7
6

Sample Output:

2
2
5

Invalid Operation

Explanation of the sample input

The first number given is 3, so that means two more numbers will be given and we'll have to multiply them and show the result. The two numbers are 1 and 2. Their product is 2, so 2 is displayed first in the output. Similarly, all the numbers are processed in groups of three. The first number tells the operation and the next two numbers tell on which numbers the operation is done. This applies to numbers from 1 to 5. If the input is 6 (like it is at the end), two more numbers will NOT be provided, you simply have to exit the program. Also, if the input is any number except 1 to 6 (like 7 which is at the second last), then you simply have to print "Invalid Operation"

```

1. # Write your code here
2. n=int(input())
3. while (n!=6):
4.     if n==1:
5.         a=int(input())
6.         b=int(input())
7.         print(a+b)
8.     if n==2:
9.         a=int(input())
10.        b=int(input())
11.        print(a-b)
12.    if n==3:
13.        a=int(input())
14.        b=int(input())
15.        print(a*b)
16.    if n==4:
17.        a=int(input())
18.        b=int(input())
19.        print(a//b)
20.    if n==5:
21.        a=int(input())
22.        b=int(input())
23.        print(a%b)
24.    if n < 1 or n > 6:
25.        print("Invalid Operation")
26.    n=int(input())

```

15-Ass : Reverse of a number

[Send Feedback](#)

Write a program to generate the reverse of a given number N. Print the corresponding reverse number.

Note : If a number has trailing zeros, then its reverse will not include them. For e.g
The reverse of 10400 will be 401 instead of 00401.

Input format :

Integer N

Output format :

Corresponding reverse number

Constraints:

$0 \leq N < 10^8$

Sample Input 1 :

1234

Sample Output 1 :

4321

Sample Input 2 :

1980

Sample Output 2 :

891

```
1. #Write Your Code Here
2. n=int(input())
3. rev=0
4. while(n>0):
5.     dig=n%10
6.     rev=rev*10+dig
7.     n=n//10
8. print(rev)
```

16-Ass : Palindrome number

[Send Feedback](#)

Write a program to determine if a given number is palindrome or not. Print true if it is palindrome, false otherwise.

Palindromes are the numbers for which the reverse is exactly the same as the original one. For eg. 121

Sample Input 1 : 121

Sample Output 1 : true

Sample Input 2 : 1032

Sample Output 2 : false

```
1. number = int(input())
2.
3. reverse = 0
4. temp = number
5.
6. while(temp > 0):
7.     Reminder = temp % 10
8.     reverse = (reverse * 10) + Reminder
9.     temp = temp //10
10. if(number == reverse):
11.     print("true")
12. else:
```

13. `print("false")`

17-Ass : Sum of even & odd

[Send Feedback](#)

Write a program to input an integer N and print the sum of all its even digits and sum of all its odd digits separately.

Digits mean numbers, not the places! That is, if the given integer is "13245", even digits are 2 & 4 and odd digits are 1, 3 & 5.

Input format :

Integer N

Output format :

Sum_of_Even_Digits Sum_of_Odd_Digits

(Print first even sum and then odd sum separated by space)

Constraints

$0 \leq N \leq 10^8$

Sample Input 1:

1234

Sample Output 1:

6 4

Sample Input 2:

552245

Sample Output 2:

8 15

Explanation for Input 2:

For the given input, the even digits are 2, 2 and 4 and if we take the sum of these digits it will come out to be $8(2 + 2 + 4)$ and similarly, if we look at the odd digits, they are, 5, 5 and 5 which makes a sum of $15(5 + 5 + 5)$. Hence the answer would be, $8(\text{evenSum})$ <single space> $15(\text{oddSum})$

1. `## Note : For printing multiple values in one line, put them inside print separated by space.`
2. `## You can follow this syntax for printing values of two variables val1 and val2 separated by space`
3. `## print(val1, " ", val2)`
4. `Number=int(input())`
5. `even=0`
6. `odd=0`
7. `while(Number>0):`
8. `Reminder = Number %10`
9. `if(Reminder % 2 == 0):`
10. `even=even + Reminder`
11. `else:`
12. `odd= odd + Reminder`
13. `Number = Number //10`
14. `print(even," ",odd)`

18-Ass : Nth Fibonacci Number

[Send Feedback](#)

Nth term of Fibonacci series $F(n)$, where $F(n)$ is a function, is calculated using the following formula -

$$F(n) = F(n-1) + F(n-2),$$

$$\text{Where, } F(1) = F(2) = 1$$

Provided N you have to find out the Nth Fibonacci Number.

Input Format :

The first line of each test case contains a real number 'N'.

Output Format :

For each test case, return its equivalent Fibonacci number.

Constraints:

$$1 \leq N \leq 10000$$

Where 'N' represents the number for which we have to find its equivalent Fibonacci number.

Time Limit: 1 second

Sample Input 1: 6

Sample Output 1: 8

Explanation of Sample Input 1:

Now the number is '6' so we have to find the "6th" Fibonacci number

So by using the property of the Fibonacci series i.e

[1, 1, 2, 3, 5, 8]

So the "6th" element is "8" hence we get the output.

1. `## Read input as specified in the question.`
2. `## Print output as specified in the question.`
- 3.
4. `n = int(input())`
5. `s = 1`
6. `e = 1`
7. `if(n < 3):`
8. `print(e)`
9. `else:`
10. `i = 3`
11. `while(i <= n):`
12. `temp = e`
13. `e = s + e`
14. `s = temp`
15. `i = i + 1`
16. `print(e)`

L4 : Patterns1 Practice Questions

1-Tut : Code : Square Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
4444
4444
4444
4444
```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints

0 <= N <= 50

Sample Input 1:

```
7
```

Sample Output 1:

```
7777777
7777777
7777777
7777777
7777777
7777777
7777777
```

Sample Input 1:

```
6
```

Sample Output 1:

```
666666
666666
666666
666666
666666
666666
```

1. `##` Read input as specified in the question
2. `##` Print the required output in given format
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `while(j <= N):`
8. `print(N,end="")`
9. `j=j+1`
10. `print()`
11. `i = i+ 1`

Square Pattern, Triangular Pattern

2-Tut : Code : Triangular Star Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

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

Note : There are no spaces between the stars (*).

Input format : Integer N (Total no. of rows)

Output format :

Pattern in N lines

Constraints

$0 \leq N \leq 50$

Sample Input 1: 5

Sample Output 1:

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

Sample Input 2: 6

Sample Output 2:

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

1. `##` Read input as specified in the question
2. `##` Print the required output in given format
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `while(j <= i):`
8. `print("*",end="")`
9. `j = j + 1`
10. `print()`
11. `i = i + 1`

3-Tut : Code : Triangle Number Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
1
22
333
4444
```

Input format : Integer N (Total no. of rows)

Output format :

Pattern in N lines

Constraints

0 <= N <= 50

Sample Input 1: 5

Sample Output 1:

```
1
22
333
4444
55555
```

Sample Input 2: 6

Sample Output 2:

```
1
22
333
4444
55555
666666
```

1. `## Read input as specified in the question`
2. `## Print the required output in given format`
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `while(j <= i):`
8. `print(i, end="")`
9. `j = j+1`
10. `print()`
11. `i = i + 1`

4-Tut : Code : Reverse Number Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
1
21
321
```

4321

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints

$0 \leq N \leq 50$

Sample Input 1:

5

Sample Output 1:

1

21

321

4321

54321

Sample Input 2:

6

Sample Output 2:

1

21

321

4321

54321

654321

1. `## Read input as specified in the question`
2. `## Print the required output in given format`
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `p = i`
8. `while(j <= i):`
9. `print(p, end="")`
10. `p = p - 1`
11. `j = j+1`
12. `print()`
13. `i = i + 1`

Character Pattern

5-Tut : Code : Character Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

A

BC

CDE

DEFG

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints

0 <= N <= 13

Sample Input 1:

5

Sample Output 1:

A
BC
CDE
DEFG
EFGHI

Sample Input 2: 6

Sample Output 2:

A
BC
CDE
DEFG
EFGHI
FGHIJK

1. `##` Read input as specified in the question
2. `##` Print the required output in given format
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `startchar = chr(ord('A') + i - 1)`
7. `j = 1`
8. `while(j <= i):`
9. `charp = chr(ord(startchar) + j -1)`
10. `print(charp, end="")`
11. `j = j + 1`
12. `print()`
13. `i = i + 1`

6-Tut : Code : Interesting Alphabets

[Send Feedback](#)

Print the following pattern for the given number of rows.

Pattern for N = 5

E
DE
CDE
BCDE
ABCDE

Input format : N (Total no. of rows)

Output format :

Pattern in N lines

Constraints 0 <= N <= 26

Sample Input 1: 8

Sample Output 1:

```
H
GH
FGH
EFGH
DEFGH
CDEFGH
BCDEFGH
ABCDEFGH
```

Sample Input 2: 7

Sample Output 2:

```
G
FG
EFG
DEFG
CDEFG
BCDEFG
ABCDEFG
```

1. `##` Read input as specified in the question.
2. `##` Print output as specified in the question.
3. `##` Read input as specified in the question
4. `##` Print the required output in given format
5. `N = int(input())`
6. `i = 1`
7. `while(i <= N):`
8. `startchar = chr(ord('A') + N - i)`
9. `j = 1`
10. `while(j <= i):`
11. `charp = chr(ord(startchar) + j -1)`
12. `print(charp, end="")`
13. `j = j + 1`
14. `print()`
15. `i = i + 1`

7-Ass : Number Pattern 1

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
1
11
111
1111
```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 5

Sample Output :

```
1
11
111
1111
11111
```

1. **##** Read input as specified in the question.
2. **##** Print output as specified in the question.
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `while(j <= i):`
8. `print(1, end="")`
9. `j = j + 1`
10. `print()`
11. `i = i + 1`

8-Ass : Number Pattern 2

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
1
11
202
3003
```

Input format : Integer N (Total no. of rows)

Constraints:

`1 <= n <= 50`

Output format : Pattern in N lines

Sample Input : 5

Sample Output :

```
1
11
202
3003
40004
```

1. **##** Read input as specified in the question.
2. **##** Print output as specified in the question.
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `while(j <= i):`
8. `if(i == 1):`
9. `print(1,end="")`
10. `elif(j == 1 or j == i):`

```

11.         print(i-1, end=")
12.     else:
13.         print(0,end = ")
14.     j = j + 1
15.     print()
16.     i = i + 1

```

9-Ass: Number Pattern 3

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```

1
11
121
1221

```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 5

Sample Output :

```

1
11
121
1221
12221

```

```

1.  ## Read input as specified in the question.
2.  ## Print output as specified in the question.
3.  N = int(input())
4.  i = 1
5.  while( i <= N):
6.      j = 1
7.      while(j <= i):
8.          if(i == 1):
9.              print(1,end=")
10.         elif(j == 1 or j == i):
11.             print(1, end=")
12.         else:
13.             print(2,end = ")
14.         j = j + 1
15.     print()
16.     i = i + 1

```

10-Ass : Number Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```

1234

```


123
12
1

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 5

Sample Output :

12345
1234
123
12
1

1. `##` Read input as specified in the question.
2. `##` Print output as specified in the question.
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `while(j <= N-i+1):`
8. `print(j,end="")`
9. `j = j + 1`
10. `print()`
11. `i = i + 1`

11-Ass: Alpha Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 3

A
BB
CCC

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints

`0 <= N <= 26`

Sample Input 1: 7

Sample Output 1:

A
BB
CCC
DDDD
EEEE
FFFFF
GGGGGG

Sample Input 2: 6

Sample Output 2:

A

BB

CCC

DDDD

EEEE

FFFFF

1. `## Read input as specified in the question.`
2. `## Print output as specified in the question.`
3. `N = int(input())`
4. `i = 1`
5. `while(i <= N):`
6. `j = 1`
7. `startchar = chr(ord('A')+i-1)`
8. `while(j <= i):`
9. `print(startchar,end="")`
10. `j = j + 1`
11. `print()`
12. `i = i + 1`

L5 : Patterns2 Practice Questions

1-Tut : Code : Inverted Number Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
4444
333
22
1
```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints :

0 <= N <= 50

Sample Input 1:

```
5
```

Sample Output 1:

```
55555
4444
333
22
1
```

Sample Input 2:6

Sample Output 2:

```
666666
55555
4444
333
22
1
```

1. **## Read input as specified in the question**
2. **## Print the required output in given format**
3. **N = int(input())**
4. **i = 1**
5. **while(i <= N):**
6. **j = N - i + 1**
7. **k = j**
8. **while(j>=1):**
9. **print(k,end="")**
10. **j=j-1**
11. **print()**
12. **i=i+1**

2-Tut : Code : Mirror Number Pattern

[Send Feedback](#)

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
• • • 1
• • 12
• 123
1234
```

The dots represent spaces.

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints

0 <= N <= 50

Sample Input 1: 3

Sample Output 1:

```
1
12
123
```

Sample Input 2: 4

Sample Output 2:

```
1
12
123
1234
```

```
1. N = int(input())
2. i = 1
3. while(i <= N):
4.     j = 1
5.     while(j <= N-i):
6.         print(' ',end="")
7.         j = j +1
8.     s = 1
9.     while(s <= i):
10.        print(s,end="")
11.        s = s + 1
12.    print()
13.    i=i+1
```

3-Tut: Code : Star Pattern

[Send Feedback](#)

Print the following pattern

Pattern for N = 4

```

. . . *
. . ***
. *****
*****

```

The dots represent spaces.

Input Format : N (Total no. of rows)

Output Format : Pattern in N lines

Constraints :

0 <= N <= 50

Sample Input 1 : 3

Sample Output 1 :

```

*
***
*****

```

Sample Input 2 : 4

Sample Output 2 :

```

*
***
*****
*****

```

```

1. N = int(input())
2. i = 1
3. while(i <= N):
4.     spaces = 1
5.     while(spaces <= N-i):
6.         print(' ',end="")
7.         spaces = spaces +1
8.     star = 1
9.     while(star <= 2*i - 1):
10.        print('*',end="")
11.        star = star + 1
12.    print()
13.    i=i+1

```

4-Tut: Code : Triangle of Numbers

[Send Feedback](#)

Print the following pattern for the given number of rows.

Pattern for N = 4

```

. . . 1
. . 232
. 34543
4567654

```

The dots represent spaces.

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints :

$0 \leq N \leq 50$

Sample Input 1:5

Sample Output 1:

```
1
232
34543
4567654
567898765
```

Sample Input 2:4

Sample Output 2:

```
1
232
34543
4567654
```

```
1. N = int(input())
2. i = 1
3. while(i <= N):
4.     spaces = 1
5.     while(spaces <= N-i):
6.         print(' ',end=")
7.         spaces = spaces +1
8.     num = i
9.     j = 1
10.    while(j <= i ):
11.        print(num,end=")
12.        num = num + 1
13.        j = j + 1
14.
15.    p = i - 1
16.    k = 2*i - 2
17.    while(p >= 1):
18.        print(k,end=")
19.        k = k -1
20.        p = p -1
21.    print()
22.    i=i+1
```

5-Tut : Code : Diamond of stars

[Send Feedback](#)

Print the following pattern for the given number of rows.

Note: N is always odd.

Pattern for N = 5

```
  • • ★
  • ★★★
★★★★★
  • ★★★
  • • ★
```

The dots represent spaces.

Input format : N (Total no. of rows and can only be odd)

Output format : Pattern in N lines

Constraints :

1 <= N <= 49

Sample Input 1: 5

Sample Output 1:

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

Sample Input 2: 3

Sample Output 2:

```
*
***
*
```

```
1. N = int(input())
2. firsthalf = (N+1)//2
3. secondhalf = N//2
4.
5. rows = 1
6. while(rows <= firsthalf):
7.     spaces = 1
8.     while(spaces <= firsthalf-rows):
9.         print(" ",end=")
10.        spaces = spaces + 1
11.        stars = 1
12.        while(stars <= (2*rows -1) ):
13.            print("*",end=")
14.            stars = stars + 1
15.        print()
16.        rows = rows + 1
17. rows = secondhalf
18. while(rows >= 1):
19.     spaces = 1
```

```

20. while(spaces <= firsthalf - rows):
21.     print(" ",end=")
22.     spaces = spaces + 1
23. stars = 1
24. while(stars <= (2*rows -1) ):
25.     print("*",end=")
26.     stars = stars + 1
27. print()
28. rows = rows - 1
29. .

```

6-Ass: Number Pattern

[Send Feedback](#)

Print the following pattern for n number of rows.

For eg. N = 5

```

1      1
12     21
123    321
1234   4321
1234554321

```

Sample Input 1 : 4

Sample Output 1 :

```

1      1
12     21
123    321
12344321

```

```

1. N = int(input())
2. row = 1
3. while(row <= N):
4.     fwd = 1
5.     while(fwd <= row):
6.         print(fwd,end=")
7.         fwd = fwd + 1
8.     space = 1
9.     while(space <= 2*(N-row) ):
10.        print(" ",end=")
11.        space = space+1
12.    bwd = row
13.    while(bwd >= 1):
14.        print(bwd,end=")
15.        bwd = bwd -1
16.    print()
17.    row = row + 1

```

7-Ass : Zeros and Stars Pattern

[Send Feedback](#)

Print the following pattern

Pattern for N = 4

```
*000*000*
0*00*00*0
00*0*0*00
000**000
```

Input Format : N (Total no. of rows)

Output Format : Pattern in N lines

Sample Input 1 : 3

Sample Output 1 :

```
*00*00*
0*0*0*0
00***00
```

Sample Input 2 : 5

Sample Output 2 :

```
*0000*0000*
0*000*000*0
00*00*00*00
000*0*0*000
0000***0000
```

1. `##` Read input as specified in the question.
2. `##` Print output as specified in the question.
3. `lines=int(input())`
4. `i=1`
5. `j=1`
6. `while i<=lines:`
7. `j=1`
8. `while j<=lines:`
9. `if i==j:`
10. `print("**", end=" ", flush=True)`
11. `else :`
12. `print("0", end=" ", flush=True)`
13. `j=j+1`
14. `j=j-1;`
15. `print("**", end=" ", flush=True)`
16. `while j>=1:`
17. `if i==j:`
18. `print("**", end=" ", flush=True)`
19. `else :`
20. `print("0", end=" ", flush=True)`
21. `j=j-1`
22. `print("");`
23. `i=i+1`

8-Ass: Pyramid Number Pattern

[Send Feedback](#)

Print the following pattern for the given number of rows.

Pattern for N = 4

```
1
212
32123
4321234
```

Input format : N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 5

Sample Output :

```
1
212
32123
4321234
543212345
```

1. `## Read input as specified in the question.`
2. `## Print output as specified in the question.`
3. `N = int(input())`
4. `row = 1`
5. `while(row <= N):`
6. `space = 1`
7. `while(space <= N-row):`
8. `print(" ",end="")`
9. `space = space + 1`
10. `num = row`
11. `while(num >= 1):`
12. `print(num,end="")`
13. `num = num - 1`
14. `num = 2`
15. `k = row-1`
16. `while(k >= 1):`
17. `print(num,end="")`
18. `k = k - 1`
19. `num = num + 1`
20. `print()`
21. `row = row + 1`

9-Ass: Arrow pattern

[Send Feedback](#)

Print the following pattern for the given number of rows. Assume N is always odd.

Note : There is space after every star.

Pattern for N = 7

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

```

* * * *
* * *
* *
*

```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 11

Sample Output :

```

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * *
* * *
* *
*

```

1. **##** Read input as specified in the question.
2. **##** Print output as specified in the question.
3. `N = int(input())`
4. `firsthalf = (N+1)//2`
5. `sh = N//2`
6. `row = 1`
7. `while(row <= firsthalf):`
8. `space = 1`
9. `while(space <= row - 1):`
10. `print(' ',end="")`
11. `space = space + 1`
12. `star = 1`
13. `while(star <= row):`
14. `print("* ",end="")`
15. `star = star + 1`
16. `print()`
17. `row = row + 1`
- 18.
19. `row = sh`
20. `while(row >= 1):`
21. `space = 1`
22. `while(space <= row - 1):`
23. `print(' ',end="")`
24. `space = space + 1`
25. `star = 1`
26. `while(star <= row):`
27. `print("* ",end="")`

```
28.     star = star + 1
29.     print()
30.     row = row - 1
```

L6: More on Loops Practice Questions

For Loops and Range Method :

1-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
for i in range(1,5,2):  
    print(i,end=' ')
```

Options

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

Correct Answer : B

2-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code ?

```
for i in range(5):  
    print(i,end= ' ')
```

Options

- 0 1 2 3 4 5
- 1 2 3 4 5
- 0 1 2 3 4
- 1 3 5

Correct Answer: C

Note : range(start, stop, stride) : default value : start = 0, stride = 1 : if only one entry for range() function then it is for 'stop' otherwise default order will follow for more than 1 entry. Also start is included and stop is excluded.

3-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
i=1  
while i<5:  
    if i==3:  
        break  
    print(i,end=" ")
```

```
i = i + 1
```

Options

1 2 3 4

1 2

1 2 3

Infinite Loop

Correct Answer : B

4-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
i=1
while i<3:
    j=1
    while j<5:
        if j==3:
            break
        print(j,end=" ")
        j = j + 1
    i = i + 1
```

Options

1 2 1 2

1 2

1 2 4 1 2 4

Infinite Loop

Correct Answer : A

Note: else keyword with loops in python.(other languages like c++ / java don't support it.)

5-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
i=1
while i<5:
    if i == 6:
        break
    print(i,end=" ")
    i = i + 1
else:
    print("Else is also printed")
```

Options

1 2 3 4 5 Else is also printed

1 2 3 4 Else is also printed

1 2 3 4 5

1 2 3 4

Correct Answer : B

6-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
i=1
while i<5:
    if i == 3:
        break
    print(i,end=" ")
    i = i + 1
else:
    print("Else is also printed")
```

Options

1 2 3 4 5 Else is also printed

1 2 Else is also printed

1 2

1 2 3 4

Correct Answer : C

Note : continue keyword : take care of stride when you use it with while loop otherwise it will be trapped in infinite loop

7-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
i=1
while i<5:
    if i==3:
        continue
    print(i,end=" ")
    i = i + 1
```

Options

1 2 3 4

1 2

1 2 Infinite Loop

1 2 4

Correct Answer : C

8-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
i=1
while i<3:
    j=0
    while j<5:
        j = j +1
        if j==3:
            continue
        print(j,end=" ")
    i = i +1
```

Options

1 2 1 2

1 2 3 4 1 2 3 4

1 2 4 5 1 2 4 5

1 2 4 1 2 4

Correct Answer : C

Note : Pass Keyword : just a dummy statement / way to write just like in c++/java blank if block, for loop.

Ex : for i in range(1,4)

Pass

print("end") o/p will be : end.

9-Ass : Binary Pattern

[Send Feedback](#)

Print the following pattern for the given number of rows.

Pattern for N = 4

1111

000

11

0

Input format : N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 5

Sample Output :

11111

0000

111

00

1

1. `## Read input as specified in the question.`
2. `## Print output as specified in the question.`
3. `N = int(input())`
4. `row = 1`
5. `while(row <= N):`
6. `col = N - row + 1`
7. `p = row % 2`
8. `while(col >= 1):`
9. `print(p,end="")`
10. `col = col - 1`
11. `print()`
12. `row = row + 1`

10-Ass : Print Number Pyramid

[Send Feedback](#)

Print the following pattern for a given n.

For eg. N = 6

123456

23456

3456

456

56

6

56

456

3456

23456

123456

Sample Input 1 : 4

Sample Output 1 :

1234

234

34

4

34

234

1234

```
1. ## Read input as specified in the question.
2. ## Print output as specified in the question.
3. N = int(input())
4. row = 1
5. while(row <= N):
6.     space = row - 1
7.     while(space >= 1):
8.         print(" ",end = "")
9.         space = space - 1
10.
11.    p = row
12.    while(p <= N):
13.        print(p,end="")
14.        p=p+1
15.    print()
16.    row = row + 1
17. row = N - 1
18. while(row >= 1):
19.     space = row - 1
20.     while(space >= 1):
21.         print(" ",end = "")
22.         space = space - 1
23.
24.    p = row
25.    while(p <= N):
26.        print(p,end="")
27.        p=p+1
28.    print()
29.    row = row - 1
```

11-Ass : Diamond of Stars

[Send Feedback](#)

Print the following pattern for the given number of rows.

Note: N is always odd.

Pattern for N = 5

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

The dots represent spaces.

Input format : N (Total no. of rows and can only be odd)

Output format : Pattern in N lines

Constraints : $1 \leq N \leq 49$

Sample Input 1: 5

Sample Output 1:

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

Sample Input 2: 3

Sample Output 2:

```
*  
***  
*
```

1. `##` Read input as specified in the question.
2. `##` Print output as specified in the question.
3. `N = int(input())`
4. `fh = (N+1)//2`
5. `sh = N//2`
6. `row = 1`
7. `while(row <= fh):`
8. `space = fh - row`
9. `while(space >= 1):`
10. `print(" ",end="")`
11. `space = space - 1`

```

12. star = 2*row - 1
13. while(star >= 1):
14.     print("*",end =")
15.     star=star-1
16.     print()
17.     row = row + 1
18. row = sh
19. while(row >= 1):
20.     space = fh - row
21.     while(space >= 1):
22.         print(" ",end=")
23.         space = space - 1
24.     star = 2*row - 1
25.     while(star >= 1):
26.         print("*",end =")
27.         star=star-1
28.         print()
29.         row = row - 1

```

12-Ass : Rectangular numbers

[Send Feedback](#)

Print the following pattern for the given number of rows.

Pattern for N = 4

44444444

4333334

4322234

4321234

4322234

4333334

44444444

Input format : N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 3

Sample Output :

33333

32223

32123

32223

33333

```

1. n = int(input())

```

```

2. for i in range(1,n+1):
3.     temp = n
4.     for j in range(1,i):
5.         print(temp,end="")
6.         temp = temp - 1
7.     for j in range(1,(2*n) - (2*i) + 2):
8.         print(n-i+1,end="")
9.     for j in range(1,i):
10.        temp = temp+1
11.        print(temp,end="")
12.    print()
13. for i in range(n-1,0,-1):
14.     temp = n
15.     for j in range(1,i):
16.         print(temp,end="")
17.         temp = temp - 1
18.     for j in range(1,(2*n) - (2*i) + 2):
19.         print(n-i+1,end="")
20.     for j in range(1,i):
21.         temp = temp+1
22.         print(temp,end="")
23.    print()

```

13-Ass: Print the pattern

[Send Feedback](#)

Print the following pattern for the given number of rows.

Pattern for N = 5

```

1  2  3  4  5
11 12 13 14 15
21 22 23 24 25
16 17 18 19 20
6  7  8  9  10

```

Input format : N (Total no. of rows)

Output format : Pattern in N lines

Sample Input : 4

Sample Output :

```

1 2 3 4
9 10 11 12
13 14 15 16
5 6 7 8

```

```
1. n = int(input())
2. startValue = 1
3. for i in range(1,n+1):
4.     for j in range(startValue,startValue + n):
5.         print(j,end=" ")
6.     print()
7.     if(i==(n+1)//2):
8.         if((n%2)!=0):
9.             startValue = n*(n-2)+1
10.        else:
11.            startValue = n*(n-1) + 1
12.    elif((i>(n+1)//2)):
13.        startValue = startValue - 2*n
14.    else:
15.        startValue = startValue + 2*n
```

L7 : Functions Practice Questions in python

1-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
def func(a):  
    a = a + 10  
    return a  
a = 5  
func(a)  
  
print(a)
```

Answer

[Type here : 5](#)

Correct Answer

2-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
def square(a):  
    ans = a*a  
    return ans  
  
a = 4  
a = square(a)  
print(a)
```

Answer

[Type here : 16](#)

Correct Answer

3-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
a = 14  
def f():  
    a=12  
f()  
  
print(a)
```

Options

12

14

a is not defined

None of the above

Correct Answer : B

4-Tut: Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
a=14
def f():
    global a
    a=12
f()
print(a)
```

Options

12

14

a is not defined

None of the above

Correct Answer : A

5-Tut : What will be the output of the following code?

```
a = 14
def f():
    a = 12
    return a
a = f()
print(a)
```

Options

12

14

a is not defined

None of them

Correct Answer : A

[Default Parameters in functions](#)

6-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
def function(a,b,c=1):
```



```
    return a+b-c
value = function(10,12)
print(value)
```

Options

21

22

23

None of the above

Correct Answer : A

7-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code ?

```
def function(a,b,c=1):
    return a+b-c
value = function(10,12,5)

print(value)
```

Options

21

22

23

17

Correct Answer : D

8-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
def function(a,b,c=1,d=5):
    return a+b+c+d
value = function(1,2,d=7)

print(value)
```

Options

9

11

3

10

Correct Answer : B

9-Ass : Fahrenheit to Celsius Function

[Send Feedback](#)

Given three values - Start Fahrenheit Value (S), End Fahrenheit value (E) and Step Size (W), you need to convert all Fahrenheit values from Start to End at the gap of W, into their corresponding Celsius values and print the table.

Input Format : 3 integers - S, E and W respectively

Output Format : Fahrenheit to Celsius conversion table. One line for every Fahrenheit and Celsius Fahrenheit value. Fahrenheit value and its corresponding Celsius value should be separate by tab ("\t")

Constraints :

$0 \leq S \leq 1000$

$0 \leq E \leq 1000$

$0 \leq W \leq 1000$

Sample Input 1:

0
100
20

Sample Output 1:

0 -17
20 -6
40 4
60 15
80 26
100 37

Sample Input 2:

120
200
40

Sample Output 2:

120 48
160 71
200 93

Explanation for Sample Output 2 :

Start value is 120, end value is 200 and step size is 40. Therefore, the values we need to convert are 120, $120 + 40 = 160$, and $160 + 40 = 200$.

The formula for converting Fahrenheit to Celsius is:

$\text{Celsius Value} = (5/9) * (\text{Fahrenheit Value} - 32)$

Plugging 120 into the formula, the celsius value will be $(5 / 9) * (120 - 32) \Rightarrow (5 / 9) * 88 \Rightarrow (5 * 88) / 9 \Rightarrow 440 / 9 \Rightarrow 48.88$

But we'll only print 48 because we are only interested in the integral part of the value.

```
1. def printTable(start,end,step):
2.     #Implement Your Code Here
3.     while start <= end:
4.         cel = ( ( start-32) * 5) / 9 )
5.         print(start, int(cel))
6.         start = start + step
```

```

7.
8. s = int(input())
9. e = int(input())
10. step = int(input())
11. printTable(s,e,step)

```

10-Ass : Fibonacci Member

Send Feedback

Given a number N, figure out if it is a member of the fibonacci series or not. Return true if the number is a member of the fibonacci series, else false.

Fibonacci Series is defined by the recurrence

$$F(n) = F(n-1) + F(n-2)$$

where $F(0) = 0$ and $F(1) = 1$

Input Format : Integer N

Output Format : true or false

Constraints : $0 \leq n \leq 10^4$

Sample Input 1 :5

Sample Output 1 :true

Sample Input 2 :14

Sample Output 2 : false

```

1. def checkMember(n):
2.     #Implement Your Code Here
3.     f0 = 0
4.     f1 = 1
5.     if n == 0:
6.         return True
7.     else:
8.         while(f1 <= n):
9.             if(f1 == n):
10.                 return True
11.                 temp = f1
12.                 f1 = f0 + f1
13.                 f0 = temp
14.             else:
15.                 return False
16.         #pass
17.
18. n=int(input())
19. if(checkMember(n)):
20.     print("true")
21. else:
22.     print("false")

```

Method 2 : Property of fibonacci series : $5*n*n + 4$ or $5*n*n - 4$ is a perfect square

```
1. import math
2. def isPerfectSquare(x):
3.     s = int(math.sqrt(x))
4.     return s*s == x
5. def checkMember(n):
6.     return isPerfectSquare(5*n*n + 4) or isPerfectSquare(5*n*n - 4)
7.
8. n=int(input())
9. if(checkMember(n)):
10.    print("true")
11. else:
12.    print("false")
```

Note : $!=$, is not : are two diff things “is not” check whether they are pointing to the same location or not whereas “ $!=$ ” checks values are equal or not

11-Ass : **Palindrome number**

[Send Feedback](#)

Write a program to determine if a given number is palindrome or not. Print true if it is palindrome, false otherwise.

Palindromes are the numbers for which the reverse is exactly the same as the original one. For eg. 121

Sample Input 1 : 121

Sample Output 1 : true

Sample Input 2 : 1032

Sample Output 2 : false

```
1. def checkPalindrome(num):
2.     if num == 0:
3.         return True
4.     num1 = num
5.     numstr=""
6.     while num1 != 0:
7.         r = num1 % 10
8.         numstr += str(r)
9.         num1 = num1//10
10.    num1 = int(numstr)
11.    return num1 == num
12. num = int(input())
13. isPalindrome = checkPalindrome(num)
14. if(isPalindrome):
15.     print('true')
16. else:
17.     print('false')
```

12-Ass: Check Armstrong

[Send Feedback](#)

Write a Program to determine if the given number is Armstrong number or not. Print true if number is armstrong, otherwise print false.

An Armstrong number is a number (with digits n) such that the sum of its digits raised to n th power is equal to the number itself.

For example,

371, as $3^3 + 7^3 + 1^3 = 371$

1634, as $1^4 + 6^4 + 3^4 + 4^4 = 1634$

Input Format : Integer n

Output Format : true or false

Sample Input 1 : 1

Sample Output 1 : true

Sample Input 2 : 103

Sample Output 2 : false

```
1. ## Read input as specified in the question.
2. ## Print output as specified in the question.
3. def armstrong(n):
4.     psod = 0
5.     n1 = n
6.     while n1 != 0:
7.         r = n1 % 10
8.         psod += r**length
9.         n1 = n1 // 10
10.    if psod == n:
11.        print('true')
12.    else:
13.        print('false')
14.
15. N = input()
16. length = len(N)
17. N = int(N)
18. armstrong(N)
19.
```

L8 : Test 1

1-Question : Number Star pattern 1

[Send Feedback](#)

Print the following pattern for a given number of rows.

Input format :

Integer N (Total number of rows)

Output Format : Pattern in N lines

Sample Input : 5

Sample Output :

```
5432*
543*1
54*21
5*321
*4321
```

```
1. n=int(input())
2. for i in range(1,n+1):
3.     for j in range(n,0,-1):
4.         if j==i:
5.             print("*",end="")
6.         else:
7.             print(j,end="")
8.     print()
```

2-Question : Zeros and Stars Pattern

[Send Feedback](#)

Print the following pattern

Pattern for N = 4

```
*000*000*
0*00*00*0
00*0*0*00
000***000
```

Input Format : N (Total no. of rows)

Output Format : Pattern in N lines

Sample Input 1 : 3

Sample Output 1 :

```
*00*00*
0*0*0*0
00***00
```

Sample Input 2 :5

Sample Output 2 :

```
*0000*0000*  
0*000*000*0  
00*00*00*00  
000*0*0*000  
0000***0000
```

1. **##** Read input as specified in the question.
2. **##** Print output as specified in the question.
3. `n=int(input())`
4. `l=[[0]*(2*n+1) for i in range(n)]`
5. `for i in range(n):`
6. `for j in range(n):`
7. `if i==j:`
8. `l[i][j]="*"`
9. `for i in range(n):`
10. `l[i][n]="*"`
11. `#print(l)`
12. `for i in range(n):`
13. `for j in range(2*n+1):`
14. `if i+j==2*n:`
15. `l[i][j]="*"`
16. `for i in l:`
17. `for j in i:`
18. `print(j,end="")`
19. `print()`

3-Question : Check Armstrong

[Send Feedback](#)

Write a Program to determine if the given number is Armstrong number or not. Print true if number is armstrong, otherwise print false.

An Armstrong number is a number (with digits n) such that the sum of its digits raised to nth power is equal to the number itself.

For example,

371, as $3^3 + 7^3 + 1^3 = 371$

1634, as $1^4 + 6^4 + 3^4 + 4^4 = 1634$

Input Format :Integer n

Output Format :true or false

Sample Input 1 :1

Sample Output 1 :true

Sample Input 2 :103

Sample Output 2 :false

1. ## Read input as specified in the question.
2. ## Print output as specified in the question.
3. ## Print output as specified in the question.
4. n=input()
5. l=list(n)
6. n=int(n)
7. x=len(l)
8. s=0
9. for i in l:
10. s+=int(i)**x
11. if s==n:
12. print("true")
13. else:
14. print("false")

L9 : Arrays and Lists Practice Questions

1-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
li = ['abcd', 'def']  
li.insert(4,5)  
print(li)
```

Note : li.insert(index,value) : if index out of range default it will be added after the last available index

Options

Index Error

['abcd', 'def']

['abcd', 'def', 5]

None of the above

Correct Answer : C

2-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
li = ['abcd',5,'def',5]  
li.remove(5)  
print(li)
```

Note : remove(5) will remove only the first occurrence of 5 from the list

Options

Error

['abcd', 5, 'def']

['abcd', 'def']

['abcd', 'def', 5]

Correct Answer : D

3-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [5,2,6,8]  
li.pop(2)  
print(li)
```

Options

[5, 6, 8]
[5, 2, 8]
[5, 2, 6, 8]
Error

Correct Answer : B

Solution Description

The pop() method returns the item present at the given index.

Note : some more operations on list : 1. Slicing of a list, 2. li.append(), 3.li.extend()

4-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [1,2,3,4,5]
for i in li[1:4]:
    print(i,end= " ")
```

Options

1 2 3 4 5
2 3 4 5
2 3 4
None of the above

Correct Answer : C

Negative Indexing:(index start from -1 till -length , -1 represents last index, -length is for 0th index) and Sequencing (list [start : end : step] start is included end is excluded)

Line separated input of list

5-Tut : Predict the Output

[Send Feedback](#)

What will be the output of code if the following input is provided?

```
5
1
2
3
4
5
n = int(input())
li = []
for i in range(n):
    li.append(input())

print(li)
```

Options

[1, 2, 3, 4, 5]

['1', '2', '3', '4', '5']

['12345']

None of the above

Correct Answer : B

Space Separated Input of list : input() function by default deal with string, string.split(delimiter) is used to split string into list on basis of a specified delimited : (default delimiter is : ' ' space) there are lots of other ways to take space separated input for list but python have something amazing that we can do everything in just single line of code

Ex : li = [int(x) for x in input().split()]

6-Tut : Predict the Output

Send Feedback

What will be the output of code if the following input is provided?

1 3 6 8 9

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

```
print(li)
```

Options

['1', '3', '6', '8', '9']

[1, 3, 6, 8, 9]

['1 3 6 8 9']

None of the above

Correct Answer : A

7-Tut : Array Sum

Send Feedback

Given an array of length N, you need to find and print the sum of all elements of the array.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Output Format : Sum

Constraints :

$1 \leq N \leq 10^6$

Sample Input : 3

9 8 9

Sample Output : 26

1. `## Read input as specified in the question.`
2. `## Print output as specified in the question.`
3. `N = int(input())`
4. `list = [int(x) for x in input().split()]`
5. `sum = 0`
6. `for ele in list:`
7. `sum = sum + ele`
8. `print(sum)`

8-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
def change(li):
```

```
    li[1] = li[1] + 2
```

```
li = [1,2,3,4,5]
```

```
change(li)
```

```
print(li)
```

Options

`[3, 2, 3, 4, 5]`

`[1, 4, 3, 4, 5]`

`[1, 2, 3, 4, 5]`

`None of the above`

Correct Answer : B

9-Tut : Predict the Output

[Send Feedback](#)

What will be the output of the following code?

```
def change(li):
```

```
    li[1] = li[1] + 2
```

```
    li = [3,3,3,4,5]
```

```
li = [1,2,3,4,5]
```

```
change(li)
```

```
print(li)
```

Options

[3, 2, 3, 4, 5]

[1, 4, 3, 4, 5]

[3, 3, 3, 4, 5]

None of the above

Correct Answer : B

Reversing a list in python :

for i in range(len(list)):

```
    list [i] , list[ length- i - 1 ] = list [ length - i - 1 ], list[ i ]
```

method 2 : using negative indexing : li [i], li [-i - 1] = li [-i - 1], li [i]

Swap in python in 1 line : a,b = b,a

Reverse list without loop, in 1 line in python using Slicing Concept :

List = List [: : -1]

10-Tut : Swap Alternate

[Send Feedback](#)

You have been given an array/list(ARR) of size N. You need to swap every pair of alternate elements in the array/list.

You don't need to print or return anything, just change in the input array itself.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the resulting array in a single row separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^5$

Time Limit: 1sec

Sample Input 1:

1

6

9 3 6 12 4 32

Sample Output 1 :

3 9 12 6 32 4

Sample Input 2:

2

9

9 3 6 12 4 32 5 11 19

4

1 2 3 4

Sample Output 2 :

3 9 12 6 32 4 11 5 19

2 1 4 3

```

1. def swapAlternate(arr, n) :
2.     #Your code goes here
3.     if n % 2 == 0:
4.         for i in range(0,n,2):
5.             arr[i],arr[i+1] = arr[i+1],arr[i]
6.     else:
7.         for i in range(0,n-1,2):
8.             arr[i],arr[i+1] = arr[i+1],arr[i]

```

11-Ass : Find Unique

Send Feedback

You have been given an integer array/list (ARR) of size N. Where N is equal to $[2M + 1]$. Now, in the given array/list, 'M' numbers are present twice and one number is present only once. You need to find and return that number which is unique in the array/list.

Note:

Unique element is always present in the array/list according to the given condition.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format : For each test case, print the unique element present in the array.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

Time Limit: 1 sec

Sample Input 1:

1

7

2 3 1 6 3 6 2

Sample Output 1: 1

Sample Input 2:

2

5

2 4 7 2 7

9

1 3 1 3 6 6 7 10 7

Sample Output 2:

4

10

Method : 1 : TC : $O(n^2)$: pick each element one by one and check its occurrence

Method : 2 : TC : $O(n)$: take the benefit of properties of XOR

```
1. def findUnique(arr, n) :
2.     #Your code goes here
3.     uniq = 0
4.     for ele in arr:
5.         uniq = uniq ^ ele
6.     return uniq
```

12-Ass : Find Duplicate

[Send Feedback](#)

You have been given an integer array/list (ARR) of size N which contains numbers from 0 to (N - 2). Each number is present at least once. That is, if N = 5, the array/list constitutes values ranging from 0 to 3 and among these, there is a single integer value that is present twice. You need to find and return that duplicate number present in the array.

Note : Duplicate number is always present in the given array/list.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format : For each test case, print the duplicate element in the array/list.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

Time Limit: 1 sec

Sample Input 1:

```
1
9
0 7 2 5 4 7 1 3 6
```

Sample Output 1:

```
7
```

Sample Input 2:

```
2
5
0 2 1 3 1
7
0 3 1 5 4 3 2
```

Sample Output 2:

```
1
3
```

```
1. def duplicateNumber(arr, n) :
2.     #Your code goes here
3.     for i in range(0,n-1,1):
4.         for j in range(i+1,n,1):
5.             if(arr[i] == arr[j]):
6.                 return arr[i]
```

13-Ass : Array Intersection

Send Feedback

You have been given two integer arrays/list (ARR1 and ARR2) of size N and M, respectively. You need to print their intersection; An intersection for this problem can be defined when both the arrays/lists contain a particular value or to put it in other words, when there is a common value that exists in both the arrays/lists.

Note : Input arrays/lists can contain duplicate elements.

The intersection elements printed would be in the order they appear in the first array/list(ARR1)

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements of the first array/list.

Third line contains an integer 'M' representing the size of the second array/list.

Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output format :

For each test case, print the intersection elements in a row, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$$1 \leq t \leq 10^2$$

$$0 \leq N \leq 10^5$$

$$0 \leq M \leq 10^5$$

Time Limit: 1 sec

Sample Input 1 :

2

6

2 6 8 5 4 3

4

2 3 4 7

2

10 10

1

10

Sample Output 1 :

2 4 3

10

Sample Input 2 :

1

4

2 6 1 2

5

1 2 3 4 2

Sample Output 2 :

2 1 2

Explanation for Sample Output 2 :

Since both input arrays have two '2's, the intersection of the arrays also have two '2's. The first '2' of the first array matches with the first '2' of the second array. Similarly, the second '2' of the first array matches with the second '2' of the second array.

```
1. def intersections(arr1, n, arr2, m) :  
2.     #Your code goes here  
3.     for i in range(0,n,1):  
4.         for j in range(0,m,1):  
5.             if (arr1[i] == arr2[j]):  
6.                 print(arr1[i],end=" ")  
7.                 arr2[j] = -1000000007  
8.                 break
```

14-Ass : Pair Sum

[Send Feedback](#)

You have been given an integer array/list (ARR) and a number X. Find and return the total number of pairs in the array/list which sum to X.

Note: Given array/list can contain duplicate elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains an integer 'X'.

Output format :

For each test case, print the total number of pairs present in the array/list.

Output for every test case will be printed in a separate line.

Constraints :

$$1 \leq t \leq 10^2$$

$$0 \leq N \leq 10^3$$

$$0 \leq X \leq 10^9$$

Time Limit: 1 sec

Sample Input 1:

1

9

1 3 6 2 5 4 3 2 4

7

Sample Output 1: 7

Sample Input 2:

2

9

1 3 6 2 5 4 3 2 4

12

6

2 8 10 5 -2 5

10

Sample Output 2:

0

2

Explanation for Input 2:

Since there doesn't exist any pair with a sum equal to 12 for the first query, we print 0.

For the second query, we have 2 pairs in total that sum up to 10. They are, (2, 8) and (5, 5).

```
1. def pairSum(arr, n, x):
2.     #Your code goes here
3.     count = 0
4.     for i in range(0,n-1,1):
5.         for j in range(i+1,n,1):
6.             if(arr[i] + arr[j] == x):
7.                 count += 1
8.     return count
```

15-Ass : Triplet Sum

[Send Feedback](#)

You have been given a random integer array/list (ARR) and a number X. Find and return the number of triplets in the array/list which sum to X.

Note :

Given array/list can contain duplicate elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains an integer 'X'.

Output format :

For each test case, print the total number of triplets present in the array/list.

Output for every test case will be printed in a separate line.

Constraints :

$$1 \leq t \leq 50$$

$$0 \leq N \leq 10^2$$

$$0 \leq X \leq 10^9$$

Time Limit: 1 sec

Sample Input 1:

1

7

1 2 3 4 5 6 7

12

Sample Output 1: 5**Sample Input 2:**

2

7

1 2 3 4 5 6 7

19

9

2 -5 8 -6 0 5 10 11 -3

10

Sample Output 2:

0

5

Explanation for Input 2:

Since there doesn't exist any triplet with sum equal to 19 for the first query, we print 0.

For the second query, we have 5 triplets in total that sum up to 10. They are, (2, 8, 0), (2, 11, -3), (-5, 5, 10), (8, 5, -3) and (-6, 5, 11)

```
1. def findTriplet(arr, n, x):
2.     #Your code goes here
3.     count = 0
4.     for i in range(0,n-2,1):
5.         for j in range(i+1,n-1,1):
6.             for k in range(j+1,n,1):
7.                 if (arr[i] + arr[j] + arr[k] == x):
8.                     count += 1
9.     return count
```

Note : Above time complexity is cubic : $O(n^3)$: will see other optimised version later

16-Ass : Sort 0 1

[Send Feedback](#)

You have been given an integer array/list (ARR) of size N that contains only integers, 0 and 1. Write a function to sort this array/list. Think of a solution which scans the array/list only once and don't require use of an extra array/list.

Note:

You need to change the given array/list itself. Hence, no need to return or print anything.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers(all 0s and 1s) representing the elements in the array/list.

Output format :

For each test case, print the sorted array/list elements in a row separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^5$

Time Limit: 1 sec

Sample Input 1:

1

7

0 1 1 0 1 0 1

Sample Output 1:

0 0 0 1 1 1 1

Sample Input 2:

2

8

1 0 1 1 0 1 0 1

5

0 1 0 1 0

Sample Output 2:

0 0 0 1 1 1 1 1

0 0 0 1 1


```
1. def sortZeroesAndOne(arr, n) :  
2.     #Your code goes here  
3.     i = 0  
4.     j = n-1  
5.     while(i <= j):  
6.         if(arr[i] == 0):  
7.             i += 1  
8.         elif(arr[j] == 0):  
9.             arr[i],arr[j] = arr[j],arr[i]  
10.            i += 1  
11.            j -= 1  
12.        else:  
13.            j -= 1
```

L10 : Searching and Sorting Practice Questions

1-Tut : Code Binary Search

[Send Feedback](#)

You have been given a sorted(in ascending order) integer array/list(ARR) of size N and an element X.

Write a function to search this element in the given input array/list using 'Binary Search'. Return the index of the element in the input array/list. In case the element is not present in the array/list, then return -1.

Input format :

The first line contains an Integer 'N' which denotes the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow..

All the 't' lines henceforth, will take the value of X to be searched for in the array/list.

Output Format :

For each test case, print the index at which X is present, -1 otherwise.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^4$

$0 \leq N \leq 10^6$

$0 \leq X \leq 10^9$

Time Limit: 1 sec

Sample Input 1:

7

1 3 7 9 11 12 45

1

3

Sample Output 1: 1

Sample Input 2:

7

1 2 3 4 5 6 7

2

9

7

Sample Output 2:

-1

6

```
1. def binarysearchhelper(arr,si,ei,x):
2.     while(si <= ei):
3.         mid = (si + ei)//2
4.         if(arr[mid] == x):
5.             return mid
6.         elif(x < arr[mid]):
```

```

7.         ei = mid-1
8.         else:
9.             si = mid+1
10.    return -1
11. def binarySearch(arr, n, x) :
12.     #Your code goes here
13.    return binarysearchhelper(arr,0,n-1,x)

```

2-Tut : Code Selection Sort

[Send Feedback](#)

Provided with a random integer array/list (ARR) of size N, you have been required to sort this array using 'Selection Sort'.

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output format :

For each test case, print the elements of the array/list in sorted order separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

Time Limit: 1 sec

Sample Input 1:

```

1
7
2 13 4 1 3 6 28

```

Sample Output 1:

```

1 2 3 4 6 13 28

```

Sample Input 2:

```

2
5
9 3 6 2 0
4
4 3 2 1

```

Sample Output 2:

```

0 2 3 6 9

```

```

1 2 3 4

```

```

1. def selectionSort(arr, n) :
2.     #Your code goes here
3.     for i in range(0,n-1,1):

```

```

4.     min_idx = i
5.     for j in range(i+1,n,1):
6.         if(arr[j] < arr[min_idx]):
7.             min_idx = j
8.     arr[i],arr[min_idx] = arr[min_idx],arr[i]

```

3-Tut : Code Bubble Sort

[Send Feedback](#)

Provided with a random integer array/list(ARR) of size N, you have been required to sort this array using 'Bubble Sort'.

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output format :

For each test case, print the elements of the array/list in sorted order separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

Time Limit: 1 sec

Sample Input 1:

```

1
7
2 13 4 1 3 6 28

```

Sample Output 1:

```

1 2 3 4 6 13 28

```

Sample Input 2:

```

2
5
9 3 6 2 0
4
4 3 2 1

```

Sample Output 2:

```

0 2 3 6 9

```

```

1 2 3 4

```

```

1. def bubbleSort(arr, n) :
2.     #Your code goes here
3.     for i in range(0,n-1,1):
4.         for j in range(0,n-i-1,1):
5.             if(arr[j] > arr[j+1]):
6.                 arr[j],arr[j+1] = arr[j+1],arr[j]

```

4-Tut : Code Insertion Sort

[Send Feedback](#)

Provided with a random integer array/list (ARR) of size N, you have been required to sort this array using 'Insertion Sort'.

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the array/list in sorted order separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

Time Limit: 1 sec

Sample Input 1:

```
1
7
2 13 4 1 3 6 28
```

Sample Output 1:

```
1 2 3 4 6 13 28
```

Sample Input 2:

```
2
5
9 3 6 2 0
4
4 3 2 1
```

Sample Output 2:

```
0 2 3 6 9
1 2 3 4
```

```
1. def insertionSort(arr, n) :
2.     #Your code goes here
3.     for i in range(1,n):
4.         key = arr[i]
5.         j = i-1
6.         while( (arr[j] > key) and (j >= 0) ):
7.             arr[j+1] = arr[j]
8.             j = j - 1
9.         arr[j+1] = key
```

Note : Dry Run on : 9, 8, 5, 6, 7, 1 to get feel of insertion sort working

5-Tut : Code Merge Two Sorted Arrays

[Send Feedback](#)

You have been given two sorted arrays/lists (ARR1 and ARR2) of size N and M respectively, merge them into a third array/list such that the third array is also sorted.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements of the first array/list.

Third line contains an integer 'M' representing the size of the second array/list.

Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output Format :

For each test case, print the sorted array/list (of size N + M) in a single row, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^5$

$0 \leq M \leq 10^5$

Time Limit: 1 sec

Sample Input 1 :

```
1
5
1 3 4 7 11
4
2 4 6 13
```

Sample Output 1 :

```
1 2 3 4 4 6 7 11 13
```

Sample Input 2 :

```
2
3
10 100 500
7
4 7 9 25 30 300 450
4
7 45 89 90
0
```

Sample Output 2 :

```
4 7 9 10 25 30 100 300 450 500

7 45 89 90
```

1. `def merge(arr1, n, arr2, m) :`
2. `#Your code goes here`
3. `arr3 = []`
4. `i = 0`

```

5.     j = 0
6.     while(i < n and j < m):
7.         if(arr1[i] < arr2[j]):
8.             key = arr1[i]
9.             arr3.append(key)
10.            i += 1
11.        else:
12.            key = arr2[j]
13.            arr3.append(key)
14.            j += 1
15.
16.    while(i < n):
17.        arr3.append(arr1[i])
18.        i += 1
19.    while(j < m):
20.        arr3.append(arr2[j])
21.        j += 1
22.    return arr3

```

6-Ass : Push Zeros to end

[Send Feedback](#)

You have been given a random integer array/list (ARR) of size N. You have been required to push all the zeros that are present in the array/list to the end of it. Also, make sure to maintain the relative order of the non-zero elements.

Note:

Change in the input array/list itself. You don't need to return or print the elements.

You need to do this in one scan of the array only. Don't use extra space.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the array/list in the desired order separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^5$

Time Limit: 1 sec

Sample Input 1:

```

1
7
2 0 0 1 3 0 0

```

Sample Output 1:

```

2 1 3 0 0 0 0

```

Explanation for the Sample Input 1 :

All the zeros have been pushed towards the end of the array/list. Another important fact is that the order of the non-zero elements have been maintained as they appear in the input array/list.

Sample Input 2:

```
2
5
0 3 0 2 0
5
9 0 0 8 2
```

Sample Output 2:

```
3 2 0 0 0
9 8 2 0 0
```

```
1. def pushZerosAtEnd(arr, n) :
2.     #Your code goes here
3.     nonZero = 0
4.     for i in range(n):
5.         if arr[i] != 0 :
6.             arr[i], arr[nonZero] = arr[nonZero], arr[i]
7.             nonZero += 1
```

7-Ass : Rotate array

[Send Feedback](#)

You have been given a random integer array/list (ARR) of size N. Write a function that rotates the given array/list by D elements (towards the left).

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains the value of 'D' by which the array/list needs to be rotated.

Output Format :

For each test case, print the rotated array/list in a row separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^4$

$0 \leq N \leq 10^6$

$0 \leq D \leq N$

Time Limit: 1 sec

Sample Input 1:

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


Sample Output 1:

3 4 5 6 7 1 2

Sample Input 2:

2

7

1 2 3 4 5 6 7

0

4

1 2 3 4

2

Sample Output 2:

1 2 3 4 5 6 7

3 4 1 2

Method 1 :- TC : $O(d*n)$

```

1. def rotate(arr, n, d):
2.     #Your code goes here
3.     i = 1
4.     while(i <= d):
5.         temp = arr[0]
6.         for j in range(n-1):
7.             arr[j] = arr[j+1]
8.         arr[n-1] = temp
9.         i += 1

```

Method :- 2 : TC : $O(d + n-d + d)$, space : $O(d)$

```

1. def rotate(arr, n, d):
2.     #Your code goes here
3.     arrtemp = []
4.     for i in range(d):
5.         arrtemp.append(arr[i])
6.     for i in range(n-d):
7.         arr[i] = arr[i+d]
8.     i = n - d
9.     k = 0
10.    while(i < n):
11.        arr[i] = arrtemp[k]
12.        i += 1
13.        k += 1

```

Method 3 : 1. Reverse the whole, 2. then reverse until $n - d$ of the reversed list 3. then reverse last d of the reversed list

```

1. def rotate(arr, n, d):
2.     #Your code goes here
3.     i, j = 0, n-1
4.     while(i <= j):
5.         arr[i], arr[j] = arr[j], arr[i]

```

```

6.     i,j = i+1,j-1
7.
8.     i,j = 0,n-d-1
9.     while(i <= j):
10.        arr[i],arr[j] = arr[j],arr[i]
11.        i,j = i+1,j-1
12.    i,j = n-d,n-1
13.    while(i <= j):
14.        arr[i],arr[j] = arr[j],arr[i]
15.        i,j = i+1,j-1

```

Method - 4 : Using Slicing (Here slicing will not work because we have to do change in original array itself , but slicing does the manipulation onto the new list so space complexity will be $O(n)$ also we have to return the address of the array which is manipulated using slicing)

```

1.  def rotate(arr, n, d):
2.      #Your code goes here
3.      arr = arr[::-1]
4.      arr = arr[:n-d:-1]
5.      arr = arr[n-d:n:-1]
6.      return arr

```

Note : here return method is necessary bcz slicing doesn't do any change in the original array rather it creates a copy and then does the manipulation onto the new array.

8-Ass : Second Largest in array

[Send Feedback](#)

You have been given a random integer array/list (ARR) of size N. You are required to find and return the second largest element present in the array/list.

If $N \leq 1$ or all the elements are same in the array/list then return -2147483648 or -2^{31} (It is the smallest value for the range of Integer)

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains an integer 'N' representing the size of the array/list.

The second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the second largest in the array/list if exists, -2147483648 otherwise.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^5$

Time Limit: 1 sec

Sample Input 1:

```

1
7
2 13 4 1 3 6 28

```

Sample Output 1: 13

Sample Input 2:

1
5
9 3 6 2 9

Sample Output 2: 6

Sample Input 3:

2
2
6 6
4
90 8 90 5

Sample Output 3:

-2147483648

8

```
1. def secondLargestElement(arr, n):
2.     #Your code goes here
3.     int_min = -2147483648
4.     fl,sl = int_min,int_min
5.     if n < 2:
6.         return sl
7.
8.     for i in range(n):
9.         if (arr[i] > fl):
10.            sl = fl
11.            fl = arr[i]
12.         elif(arr[i] > sl and arr[i] != fl):
13.            sl = arr[i]
14.     return sl
```

9-Ass : Check Array Rotation

[Send Feedback](#)

You have been given an integer array/list (ARR) of size N. It has been sorted (in increasing order) and then rotated by some number 'K' in the right hand direction.

Your task is to write a function that returns the value of 'K', that means, the index from which the array/list has been rotated.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the value of 'K' or the index from which the array/list has been rotated.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
Time Limit: 1 sec

Sample Input 1:

1
6
5 6 1 2 3 4

Sample Output 1: 2

Sample Input 2:

2
5
3 6 8 9 10
4
10 20 30 1

Sample Output 2:

0
3

```
1. def arrayRotateCheck(arr, n):
2.     #Your code goes here
3.     k = 0
4.     for i in range(1,n):
5.         if(arr[i-1] > arr[i]):
6.             k = i
7.             return k
8.     return k
```

10-Ass : Sort 0 1 2

[Send Feedback](#)

You are given an integer array/list (ARR) of size N. It contains only 0s, 1s and 2s. Write a solution to sort this array/list in a 'single scan'.

'Single Scan' refers to iterating over the array/list just once or to put it in other words, you will be visiting each element in the array/list just once.

Note:

You need to change the given array/list itself. Hence, no need to return or print anything.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers (all 0s, 1s and 2s) representing the elements in the array/list.

Output Format :

For each test case, print the sorted array/list elements in a row separated by a single space.

Output for every test case will be printed in a separate line.

Constraints : $1 \leq t \leq 10^2$ $0 \leq N \leq 10^5$

Time Limit: 1 sec

Sample Input 1:

```
1
7
0 1 2 0 2 0 1
```

Sample Output 1:

```
0 0 0 1 1 2 2
```

Sample Input 2:

```
2
5
2 2 0 1 1
7
0 1 2 0 1 2 0
```

Sample Output 2:

```
0 1 1 2 2
0 0 0 1 1 2 2
```

```

1. def sort012(arr, n) :
2.     #Your code goes here
3.     nextZero = 0
4.     nextTwo = (n - 1)
5.     i = 0
6.     while i <= nextTwo :
7.         if arr[i] == 0 :
8.             arr[nextZero],arr[i] = arr[i],arr[nextZero]
9.             i += 1
10.            nextZero += 1
11.        elif arr[i] == 2 :
12.            arr[nextTwo],arr[i] = arr[i],arr[nextTwo]
13.            nextTwo -= 1
14.        else :
15.            i += 1
```

11-Ass : Sum of Two Arrays[Send Feedback](#)

Two random integer arrays/lists have been given as ARR1 and ARR2 of size N and M respectively. Both the arrays/lists contain numbers from 0 to 9(i.e. a single digit integer is present at every index). The idea here is to represent each array/list as an integer in itself of digits N and M.

You need to find the sum of both the input arrays/list treating them as two integers and put the result in another array/list i.e. output array/list will also contain only a single digit at every index.

Note:

The sizes N and M can be different.

Output array/list(of all 0s) has been provided as a function argument. Its size will always be one more than the size of the bigger array/list. Place 0 at the 0th index if there is no carry.

No need to print the elements of the output array/list.

Using the function "sumOfTwoArrays", write the solution to the problem and store the answer inside this output array/list. The main code will handle the printing of the output on its own.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements of the first array/list.

Third line contains an integer 'M' representing the size of the second array/list.

Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output Format :

For each test case, print the required sum of the arrays/list in a row, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^5$

$0 \leq M \leq 10^5$

Time Limit: 1 sec

Sample Input 1:

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

Sample Output 1:

```
1 3 8 0
```

Sample Input 2:

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

Sample Output 2:

```
0 8 6 5
1 0 2 2 0
```

```
1. def sumOfTwoArrays(arr1, n, arr2, m, output) :
2.     #Your code goes here
3.     i=n-1
4.     j=m-1
5.     carry=0
6.     k=max(n,m)+1
7.     while i>=0 and j>=0:
8.         num=arr1[i]+arr2[j]+carry
9.         s=num%10
10.        carry=num//10
11.        output[k-1]=s
12.        k=k-1
13.        i=i-1
14.        j=j-1
15.    while i>=0:
16.        num=arr1[i]+carry
17.        s=num%10
18.        carry=num//10
19.        output[k-1]=s
20.        k=k-1
21.        i=i-1
22.    while j>=0:
23.        num=arr2[j]+carry
24.        s=num%10
25.        carry=num//10
26.        output[k-1]=s
27.        k=k-1
28.        j=j-1
29.    if carry!=0:
30.        output[0]=carry
```

L11 : Strings Practice Questions

In python string is sequence of characters that is stored in machine language using unicode encoding technique whereas in c++ / java ASCII encoding is used.

Python does the auto optimization for smaller strings, meaning no extra space will be allocated to the same string (ex : a = 'Parikh', b = 'Parikh') then id(a) will be the same as id(b) means both will point to the same memory where 'Parikh' is stored.

We can enclose string in single quote (' '), double quote (" "), Triple quote ("'' "'' "''): multiline string allowed only in triple quote.

1-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code ?

```
s = "abcd"
print(s[-2])
```

Options

d
c
b

[None of these](#)

[Correct Answer : B](#)

[How Strings are Stored](#)

2-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
s="abcd"
s[0]='c'
print(s)
```

Note : in Python Strings are immutable means change in string is not allowed whereas in c++ we can make changes in the string.

Options

cbcd
abcd
Error

[None of these](#)

[Correct Answer : C](#)

3-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
s="abcd"
a="abcd"
if id(s) == id(a):
    print("They are same")
else:
    print("They are not same")
```

Options

- They are same
- They are not same
- Error
- None of the above

Correct Answer : A

[Concatenation of Strings](#)

4-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
s="abcd"
b=s+"ef"
print(s)
```

Options

- abcdef
- abcd
- Error
- None of the above

Correct Answer : B

5-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
s="abcd"
b=s+2
print(b)
```

Options

abcd2
abcd
Error
None of the above

Correct Answer : C

Slicing of Strings

6-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
s = "abcdef"
print (s[2:])
```

Options

bcdef
cdef
abcdef
None of the above

Correct Answer : B

7-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
s = "abcdef"
print (s[4:2:-1])
```

Options

edc
ed
bcd
None of the above

Correct Answer : B

[Iterating on strings](#), [Comparison Operator on strings](#)

8-Tut : Check Palindrome

[Send Feedback](#)

Given a string, determine if it is a palindrome, considering only alphanumeric characters.

Palindrome

A palindrome is a word, number, phrase, or other sequences of characters which read the same backwards and forwards.

Example:

If the input string happens to be, "malayalam" then as we see that this word can be read the same as forward and backwards, it is said to be a valid palindrome.

The expected output for this example will print, 'true'.

From that being said, you are required to return a boolean value from the function that has been asked to implement.

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

Output Format:

The only line of output prints either 'true' or 'false'.

Note:

You are not required to print anything. It has already been taken care of.

Constraints:

$0 \leq N \leq 10^6$

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1 :

abdcdba

Sample Output 1 :

true

Sample Input 2:

coding

Sample Output 2:

false

```
1. def isPalindrome(string) :  
2.     # Your code goes here  
3.     i,j = 0,len(string)-1  
4.     ans = True  
5.     while(i <= j):  
6.         if(string[i] != string[j]):  
7.             ans = False  
8.             break  
9.         i += 1  
10.        j -= 1  
11.    return ans
```

9-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
a = "abcdef" == "abcd"
```

```
print(a)
```

Options

False

True

Error

None of the above

Correct Answer : A

10-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
a = "abcdef" >= "abcd"
```

```
print(a)
```

Options

False

True

Error

None of the above

Correct Answer : B

11-Tut : Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
a = "abce" >= "abcdef"
```

```
print(a)
```

Options

False

True

Error

None of the above

Correct Answer : B

Some Commonly used functions of strings :

1. List = String.split ("delimiter", k): Default value : space, default k = max possible split based on delimiter , split function returns the list
2. String = string.replace("to be replaced", "to by Replaced",k) : default k = 1(how many occurrence to be replaced)

3. `Index = string.find("str to be find", start , end) : default : start = 0, end = len-1+1 = len of original string` in which we will perform find operation it returns the first starting index of char if string is available otherwise return -1
4. `String = str.lower(), string = str.upper(),`
5. `ans = str.startswith("string", startindex, endindex)`

12-Ass : Check Permutation

[Send Feedback](#)

For a given two strings, 'str1' and 'str2', check whether they are a permutation of each other or not.

Permutations of each other

Two strings are said to be a permutation of each other when either of the string's characters can be rearranged so that it becomes identical to the other one.

Example:

```
str1= "sinrtg"
```

```
str2 = "string"
```

The character of the first string(str1) can be rearranged to form str2 and hence we can say that the given strings are a permutation of each other.

Input Format:

The first line of input contains a string without any leading and trailing spaces, representing the first string 'str1'.

The second line of input contains a string without any leading and trailing spaces, representing the second string 'str2'.

Note:

All the characters in the input strings would be in lower case.

Output Format:

The only line of output prints either 'true' or 'false', denoting whether the two strings are a permutation of each other or not.

You are not required to print anything. It has already been taken care of. Just implement the function.

Constraints:

$0 \leq N \leq 10^6$

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1:

```
abcde
```

```
baedc
```

Sample Output 1:

```
true
```

Sample Input 2:

```
abc
```

```
cbd
```

Sample Output 2:

```
false
```

```

1. def isPermutation(string1, string2) :
2.     #Your code goes here
3.     n1 = len(string1)
4.     n2 = len(string2)
5.     if (n1 != n2):
6.         return False
7.     else:
8.         ans = True
9.         for char in string1:
10.            if char not in string2:
11.                ans = False
12.                break
13.         return ans

```

Note : Some test case will fail by above code ex : (string1 = aa, string2 = ab) one optimisation is as below

```

1. def isPermutation(string1, string2) :
2.     #Your code goes here
3.     n1 = len(string1)
4.     n2 = len(string2)
5.     if (n1 != n2):
6.         return False
7.     else:
8.         ans = True
9.         for char in string1:
10.            if char not in string2:
11.                ans = False
12.                break
13.         ans1 = True
14.         for char in string2:
15.            if char not in string1:
16.                ans1 = False
17.                break
18.         return ans and ans1

```

Note : still some cases will not pass ex: (aba, bab) expected o/p should be false but it gives true

Idea : other than length, Frequency of each character should also be same

```

1. def isPermutation(string1, string2) :
2.     #Your code goes here
3.     n1 = len(string1)
4.     n2 = len(string2)
5.     if (n1 != n2):
6.         return False
7.     else:
8.         arr = [0] * 26
9.         for char in string1:

```

```
10.     arr[ord(char)-ord('a')] += 1
11.     for char in string2:
12.         arr[ord(char)-ord('a')] -= 1
13.     for i in arr:
14.         if i != 0:
15.             return False
16.     else:
17.         return True
```

13-Ass : Remove Consecutive Duplicates

[Send Feedback](#)

For a given string(str), remove all the consecutive duplicate characters.

Example:

Input String: "aaaa"

Expected Output: "a"

Input String: "aabbbcc"

Expected Output: "abc"

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

Output Format:

The only line of output prints the updated string.

Note:

You are not required to print anything. It has already been taken care of.

Constraints:

$0 \leq N \leq 10^6$

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1: aabccbaa

Sample Output 1: abcba

Sample Input 2: xxyyzxx

Sample Output 2: xyzx

```
1. def removeConsecutiveDuplicates(string) :
2.     # Your code goes here
3.     n = len(string)
4.     i = 0
5.     string2 = ""
6.     while(i < n):
7.         x = string[i]
8.         j = i + 1
9.         while(j < n and string[j] == x):
10.            j += 1
11.        string2 += x
12.        i = j
13.    return string2
```

Method 2 :- using list operation bcz of string += k in python always creates a new string whereas in c++ strings are mutable hence no need to create another string

Method 3 : inbuilt :

```
1. import itertools
2. def removeConsecutiveDuplicates(string) :
3.     # Your code goes here
4.     return "".join(i for i, _ in itertools.groupby(string) ) )
```

14-Ass : Reverse Each Word

Send Feedback

Aadil has been provided with a sentence in the form of a string as a function parameter. The task is to implement a function so as to print the sentence such that each word in the sentence is reversed.

Example:

Input Sentence: "Hello, I am Aadil!"

The expected output will print, ",olleH I ma !lidaA".

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. The input string represents the sentence given to Aadil.

Output Format: The only line of output prints the sentence(string) such that each word in the sentence is reversed.

Constraints:

$0 \leq N \leq 10^6$

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1:

Welcome to Coding Ninjas

Sample Output 1:

emocleW ot gnidoC sajniN

Sample Input 2:

Always indent your code

Sample Output 2:

syawIA tnedni ruoy edoc

```
1. def reverseEachWord(string) :
2.     # Your code goes here
3.     w = string.split(" ")
4.     nw = [i[::-1] for i in w]
5.     ns = " ".join(nw)
6.     return ns
```

15-Ass : Remove character

[Send Feedback](#)

For a given string(str) and a character X, write a function to remove all the occurrences of X from the given string.

The input string will remain unchanged if the given character(X) doesn't exist in the input string.

Input Format:

The first line of input contains a string without any leading and trailing spaces.

The second line of input contains a character(X) without any leading and trailing spaces.

Output Format: The only line of output prints the updated string.

Note: You are not required to print anything explicitly. It has already been taken care of.

Constraints:

$0 \leq N \leq 10^6$

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1:

aabccbaa

a

Sample Output 1:

bccb

Sample Input 2:

xyyzxx

y

Sample Output 2:

Xxzx

```
1. def removeAllOccurrencesOfChar(string, ch) :  
2.     # Your code goes here  
3.     list1 = string.split(ch)  
4.     string2 = "".join(list1)  
5.     return string2
```

16-Ass : Highest Occurring Character

[Send Feedback](#)

For a given string(str), find and return the highest occurring character.

Example:

Input String: "abcdeapapqarr"

Expected Output: 'a'

Since 'a' has appeared four times in the string which happens to be the highest frequency character, the answer would be 'a'.

If there are two characters in the input string with the same frequency, return the character which comes first.

Consider:

Assume all the characters in the given string to be in lowercase always.

Input Format:

The first and only line of input contains a string without any leading and trailing spaces.

Output Format:

The only line of output prints the updated string.

Note:

You are not required to print anything explicitly. It has already been taken care of.

Constraints:

$0 \leq N \leq 10^6$

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1:

abdefgbabfba

Sample Output 1:

b

Sample Input 2:

xy

Sample Output 2:

x

```
1. def highestOccuringChar(string) :  
2.     #Your code goes here  
3.     ASCII_SIZE = 256  
4.     ctr = [0] * ASCII_SIZE  
5.     max = -1  
6.     ch = "
```

```
7.     for i in string:
8.         ctr[ord(i)] += 1
9.     for i in string:
10.        if max < ctr[ord(i)]:
11.            max = ctr[ord(i)]
12.        ch = i
13.    return ch
```

17-Ass : Compress the String

Send Feedback

Write a program to do basic string compression. For a character which is consecutively repeated more than once, replace consecutive duplicate occurrences with the count of repetitions.

Example:

If a string has 'x' repeated 5 times, replace this "xxxxx" with "x5".

The string is compressed only when the repeated character count is more than 1.

Note:

Consecutive count of every character in the input string is less than or equal to 9.

Input Format:

The first and only line of input contains a string without any leading and trailing spaces.

Output Format:

The output contains the string after compression printed in a single line.

Note:

You are not required to print anything. It has already been taken care of. Just implement the given function.

Constraints:

$0 \leq N \leq 10^6$

Where 'N' is the length of the input string.

Time Limit: 1 sec

Sample Input 1:

aaabbccdsa

Sample Output 1:

a3b2c2dsa

Explanation for Sample Output 1:

In the given string 'a' is repeated 3 times, 'b' is repeated 2 times, 'c' is repeated 2 times and 'd', 's' and 'a' and occurring 1 time hence no compression for last 3 characters.

Sample Input 2:

aaabbcddeeeee

Sample Output 2:

a3b2cd2e5

Explanation for Sample Output 2:

In the given string 'a' is repeated 3 times, 'b' is repeated 2 times, 'c' is occurring single time, 'd' is repeating 2 times and 'e' is repeating 25times.

```
1. def getCompressedString(s) :  
2.     # Write your code here.  
3.     i=0  
4.     y = ""  
5.     while i < len(s):  
6.         x = s[i]  
7.         j = i+1  
8.         c=1  
9.         while j < len(s) and s[j] == x:  
10.            j=j+1  
11.            c=c+1  
12.         if c>1:  
13.            y=y+s[i]+str(c)  
14.         else:  
15.            y=y+s[i]  
16.            i = j  
17.     return y
```

L12: Two Dimensional List Practice Questions

1-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [[1,2,3],[4,5,6],[7,8,9]]  
print(li[2][1])
```

Options

- 5
- 8
- 2
- Error

Correct Answer : B

2-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code ?

```
li = [[1,2,3],[4,5,6],[7,8,9]]  
print(li[1][3])
```

Options

- 6
- 9
- 3
- Error

Correct Answer: D

[Jagged Lists](#)

3-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code ?

```
li = [[1,2,3,4],[5,6],[7,8,9]]  
print(li[2])
```

Options

- [5,6]
- Error
- [7,8,9]
- None of the above

Correct Answer : C

4-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [[1,2,3,4],[5,6],[7,8,9]]  
print(li[1][3])
```

Options

4

6

Error

None of the above

Correct Answer : C

[List Comprehension](#)

5-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [ele**2 for ele in range(5)]  
print(li)
```

Options

[1,2,3,4,5]

[1,4,9,16,25]

[0,1,4,9,16]

[0,1,4,9,16,25]

Correct Answer : C

6-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [ele**2 for ele in range(10) if ele%3 ==0]  
print(li)
```

Options

[1,3,9]

[1,9,36,81]

[0,3,6,9]

[0,9,36,81]

Correct Answer : D

7-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
li = [[ i*j for j in range(4)] for i in range(3)]  
print(li)
```

Options

[\[\[0, 0, 0, 0\], \[0, 1, 2, 3\], \[0, 2, 4, 6\]\]](#)

[\[\[0, 0, 0\], \[0, 1, 2\], \[0, 2, 4\], \[0, 3, 6\]\]](#)

[\[\[1, 2, 3, 4\], \[2, 4, 6, 8\], \[3, 6, 9, 12\]\]](#)

[None of above](#)

[Correct Answer : A](#)

[Input of 2d list](#)

8-Tut : Row Wise Sum

[Send Feedback](#)

For a given two-dimensional integer array/list of size (N x M), find and print the sum of each of the row elements in a single line, separated by a single space.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space.

They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output Format :

For each test case, print the sum of every ith row elements in a single line separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1:

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

Sample Output 1:

```
3 7 11 15
```

Sample Input 2:


```
2
2 5
4 5 3 2 6
7 5 3 8 9
4 4
1 2 3 4
9 8 7 6
3 4 5 6
-1 1 -10 5
```

Sample Output 2:

```
20 32
10 30 18 -5
```

```
1. def rowWiseSum(mat, nRows, mCols):
2.     #Your code goes here
3.     for i in range(nRows):
4.         sum = 0
5.         for j in range(mCols):
6.             sum = sum + mat[i][j]
7.         print(sum,end=" ")
```

9-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
li=[[1,2,3,4],[5,6,7,8],[9,10,11,12]]
for j in range(4):
    for ele in li:
        print(ele[j],end = " ")
```

Options

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

Correct Answer : B

10-Tut : Largest Row or Column

[Send Feedback](#)

For a given two-dimensional integer array/list of size (N x M), you need to find out which row or column has the largest sum(sum of all the elements in a row/column) amongst all the rows and columns.

Note :

If there are more than one rows/columns with maximum sum, consider the row/column that comes first. And if ith row and the jth column have the same largest sum, consider the ith row as the answer.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output Format :

For each test case, If row sum is maximum, then print: "row" <row_index> <row_sum>

OR

If column sum is maximum, then print: "column" <col_index> <col_sum>

It will be printed in a single line separated by a single space between each piece of information.

Output for every test case will be printed in a separate line.

Consider :

If there doesn't exist a sum at all then print "row 0 -2147483648", where -2147483648 or -2^{31} is the smallest value for the range of Integer.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1 :

```
1
2 2
1 1
1 1
```

Sample Output 1 :

```
row 0 2
```

Sample Input 2 :

```
2
3 3
3 6 9
1 4 7
2 8 9
4 2
1 2
90 100
3 40
-10 200
```

Sample Output 2 :

```
column 2 25
```

```
column 1 342
```

```
1. def largestRowCol(arr,nrows,mcols):
2.     rows = nrows
3.     cols = mcols
4.     if(rows == 0 or cols == 0):
5.         return ['row', 0, -2147483648]
```

```

6.
7.     sumRow = [0] * rows
8.     sumCol = [0] * cols
9.     for i in range(rows):
10.        for j in range(cols):
11.            sumRow[i] += arr[i][j]
12.            sumCol[j] += arr[i][j] # Assume row 0 has maximum sum
13.     l = ['row', 0, sumRow[0]]
14.     for i in range(rows):
15.         if sumRow[i] > l[2]:
16.             l[2] = sumRow[i]
17.             l[1] = i
18.     for j in range(cols):
19.         if sumCol[j] > l[2]:
20.             l[2] = sumCol[j]
21.             l[1] = j
22.             l[0] = 'column'
23.     return l
24.
25. def findLargest(arr, nRows, mCols):
26.     #Your code goes here
27.     l=largestRowCol(arr,nRows,mCols)
28.     print(*l)

```

11-Tut : Wave Print

[Send Feedback](#)

For a given two-dimensional integer array/list of size (N x M), print the array/list in a sine wave order, i.e, print the first column top to bottom, next column bottom to top and so on.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space.

They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output format :

For each test case, print the elements of the two-dimensional array/list in the sine wave order in a single line, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1:

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

Sample Output 1:

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

Sample Input 2:

```
2
5 3
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
3 3
10 20 30
40 50 60
70 80 90
```

Sample Output 2:

```
1 4 7 10 13 14 11 8 5 2 3 6 9 12 15
10 40 70 80 50 20 30 60 90
```

```
1. def wavePrint(arr, nRows, mCols):
2.     #Your code goes here
3.     for j in range(mCols):
4.         if j % 2 == 0:
5.             for i in range(nRows):
6.                 print(arr[i][j], end = " ")
7.         else:
8.             for i in range(nRows-1, -1, -1):
9.                 print(arr[i][j], end = " ")
```

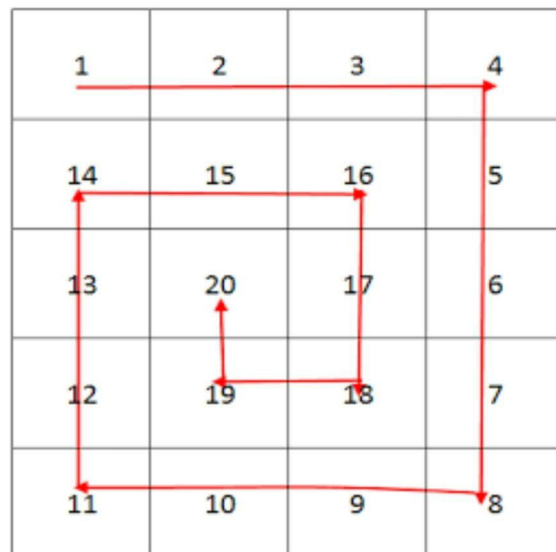
12-Tut : Spiral Print

[Send Feedback](#)

For a given two-dimensional integer array/list of size (N x M), print it in a spiral form. That is, you need to print in the order followed for every iteration:

- First row(left to right)
 - Last column(top to bottom)
 - Last row(right to left)
 - First column(bottom to top)
- Mind that every element will be printed only once.

Refer to the Image:



Output : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output format :

For each test case, print the elements of the two-dimensional array/list in the spiral form in a single line, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1:

```
1
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Sample Output 1:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Sample Input 2:

2

3 3

1 2 3

4 5 6

7 8 9

3 1

10

20

30

Sample Output 2:

1 2 3 6 9 8 7 4 5

10 20 30

```
1. def spiralPrint(mat, nRows, mCols):
2.     #Your code goes here
3.     rs,re = 0,nRows-1
4.     cs,ce = 0,mCols-1
5.     total = nRows * mCols
6.     n = 1
7.     while(n <= total):
8.         for i in range(cs,ce+1):
9.             print(mat[rs][i],end= " ")
10.            n += 1
11.            rs += 1
12.
13.        for i in range(rs,re+1):
14.            print(mat[i][ce],end = " ")
15.            n += 1
16.            ce = ce - 1
17.
18.        for i in range(ce,cs-1,-1):
19.            print(mat[re][i],end = " ")
20.            n += 1
21.            re = re - 1
22.
23.        for i in range(re,rs-1,-1):
24.            print(mat[i][cs],end = " ")
25.            n += 1
26.            cs = cs + 1
```

L13: Tuples Dictionary and Sets Practice Questions

1-Tut : Predict The Output

[Send Feedback](#)

What will be the output of following code?

```
a = 5,6,7  
print(a[1:])
```

Options

[\[6,7\]](#)

[\(6,7\)](#)

[\(5,6,7\)](#)

[\[5,6,7\]](#)

Correct Answer : B

2-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
a = 5,6,7  
a[2] = 9  
print(a)
```

Options

[\(5,6,7\)](#)

[\(5,6,9\)](#)

[\(5,9,7\)](#)

[Error](#)

Correct Answer : D

3-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
a = 1,2  
b = (4,5)  
d = (a,b)  
print(d[0])
```

Options

- 1
- 2
- (1,2)
- Error

Correct Answer : C

4-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
a = 1,2  
b = (4,5)  
d = a+b
```

```
print(d[2])
```

Options

- 2
- (4,5)
- Error
- 4

Correct Answer : D

5-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code ?

```
a = ("ab","abc","def")
```

```
print(min(a))
```

Options

- abc
- ab
- def
- Error

Correct Answer : B

[Variable Length Input / Output](#)

6-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
def multiply(a,b,c,*more):
```



```
    value = a*b*c
    for i in more:
        value = value * i
    return value
V = multiply(1,2,3,4,5)

print(V)
```

Options

- 6
- 0
- 120
- None of the above

Correct Answer : C

7-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
def sum_multiply(a,b,*more):
    sum_value = a+b
    m_value = a*b
    for i in more:
        sum_value += i
        m_value*=i
    return sum_value,m_value
s_m = sum_multiply(2,3,4)
print(s_m)
```

Note : if we use one variable to store the o/p of a function which returns multiple values it will be stored in the form of tuple , if we try to get in more than one variable then it should be exactly the same number of variables that function is returning otherwise it will throw an error..

Options

- 9,24
- (9,24)
- Error
- (5,6)

Correct Answer : B

Dictionary

8-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
d = {1:2, "abc":5, "def":7}
print(d[0])
```

Note : indexing doesn't work for dictionaries, rather we should enter the key like d[key].

Options

1:2

1

2

Error

Correct Answer : D

9-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code ?

```
d = {1:2, "abc":5, "def":7}
print(d.get(0,5))
```

d.get(key,v) is same as d [key] when key is present in the dictionary otherwise d[key] gives error and d.get(key) returns None and d.get(key,v) returns v the second argument is returned when key is not present in the dictionary.

Options

Error

1

2

5

Correct Answer : D

10-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
d = {1:2, "abc":5, "def":7}
if 2 in d:
    print('Present')
else:
    print('Not Present')
```

Options

[Present](#)

[Not Present](#)

[Error](#)

Correct Answer : B

11-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
a = {1:2,'list':[1,2],3:5}
b = {4:5,3:7}
a.update(b)
print(a[3])
```

Options

[5](#)

[7](#)

[Error](#)

[None Of The Above](#)

Correct Answer : B

12-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code ?

```
a = {1:2,'list':[1,2],3:5}
a.pop('list')
a['list'] = [3,5]
print(a['list'])
```

Options

[None](#)

[\[3,5\]](#)

[\[1,2\]](#)

[Error](#)

Correct Answer : B

Sets

Learn About some interesting functions of sets 😊

13-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code ?

```
s = {1,2,3,5,4,2,3,1}
print(len(s),end= " ")
s.add(4)
s.add(3)

print(len(s))
```

Options

8 10

5 7

5 5

None Of The Above

Correct Answer : C

14-Tut : Predict The Output

[Send Feedback](#)

What will be the output of the following code?

```
s = {}
s.add(4)
s.add(4)
print(len(s))
```

Note : to initialize the set we need to use s = set() otherwise it will create a empty dictionary

Options

2

1

Error

None Of The Above

Correct Answer : C

Question 1 :

Print 2D Array

[Send Feedback](#)

Given a 2D integer array with n rows and m columns. Print the 0th row from input n times, 1st row n-1 times.....(n-1)th row will be printed 1 time.

Input format :

Line 1 : No of rows (n) and no of columns (m) (separated by single space)

Line 2 : Row 1 elements (separated by space)

Line 3 : Row 2 elements (separated by space)

Line 4 : and so on

Sample Input 1:

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

Sample Output 1 :

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

1. `n,m = map(int,input().split())`
2. `for i in range(n):`
3. `a = list(map(int,input().split()))`
4. `for j in range(n-i):`
5. `print(*a)`

Question : 2

Minimum Length Word

[Send Feedback](#)

Given a string S (that can contain multiple words), you need to find the word which has minimum length.

Note : If multiple words are of the same length, then the answer will be the first minimum length word in the string.

Words are separated by a single space only.

Input Format :

String S

Output Format :

Minimum length word

Constraints :

$1 \leq \text{Length of String } S \leq 10^5$

Sample Input 1 :

this is test string

Sample Output 1 :

is

Sample Input 2 :

abc de ghijk a uvw h j

Sample Output 2 :

a

```
1. s=input().split()
2. l=[]
3. for i in s:
4.     l.append(len(i))
5. print(s[l.index(min(l))])
```

Question : 3

Leaders in array

[Send Feedback](#)

Given an integer array A of size n. Find and print all the leaders present in the input array. An array element A[i] is called Leader, if all the elements following it (i.e. present at its right) are less than or equal to A[i].

Print all the leader elements separated by space and in the same order they are present in the input array.

Input Format :

Line 1 : Integer n, size of array

Line 2 : Array A elements (separated by space)

Output Format :

leaders of array (separated by space)

Constraints :

$1 \leq n \leq 10^6$

Sample Input 1 :

6

3 12 34 2 0 -1

Sample Output 1 :

34 2 0 -1

Sample Input 2 :

5

13 17 5 4 6

Sample Output 2 :

17 6

```
1. n=int(input())
2. l=list(map(int,input().split()))
3. ma=l[-1]
4. l=l[:-1]
5. k=[l[0]]
6. for i in range(1,n):
7.     if l[i]>=ma:
8.         ma=l[i]
9.         k.append(l[i])
10. k=k[:-1]
11. print(*k)
```