

Java Question for absolute beginner

(Part 2 - loops)

INDEX

INDEX	1
1. Print Range from 1 to n	3
2. Print n to m	4
3. Multiple Of n	5
4. Print Even	5
5. Print 3 7 11 15	6
6. Print odd from n to m	6
7. Print n, n-3, n-6	7
8. Print n, n-k, n-2k, n-3k	8
9. print a to z	9
10. Print a, B, c, D, e, F, g..... 26 characters	9
11. Print a, C, e, G, i, K... till 'z' or 'Z'	9
12. Print series AAA,bb,CCC,dd,EEE,ff till 26 lines	10
13. Print even or odd from a list of integers	11
14. Print n/3	12
15. Multiples of 3, 5 and Both 3 and 5	13
16. FizzBuzz problem	14
17. Running sum and average	15
18. Running Sum for loop	16
19. Fibonacci number	17
20. Print fibonacci series alternatively	17
21. Nth Fibonacci Number	18
22. Print nth Tribonacci number	19
23. Print from n to 3 using while loop	20
24. Find Permutation	21
25. Steps till n greater than 0	22
26. Print all digits from end	23
27. Number of Digits	24

28.	Print total steps when $n/2$	25
29.	Print steps and update maximum	26
30.	Find GCD	27
31.	Calculate LCM	28
32.	Reverse Digits	29
33.	Print the final number xyzw	29
34.	Reverse n-digit number	30
35.	Print Primes - 1	31
36.	Prime checker 2	31
37.	Print all factors of a number	32
38.	Print all unique prime factors	33
39.	Check if an Armstrong number or not	34
40.	Print Armstrong in a range	35
41.	Rotate 7-digit number to right by three	36
42.	Divide n by 2 3 5 and tell steps	37
43.	Odd Even and Divisibility by 3	38
44.	Even Odd 2	39

Print Range from 1 to n

Given an integer **n**, print all integers in *range 1 to n*.

Input Format

Single line of input An integer **n**

Output Format

A range/series of numbers from 1 till n, with each number in one line

Sample Input

5

Sample Output

1
2
3
4
5

Explanation

Since **n** is 5, output is numbers from 1 to 5

Print n to m

You will be given an input **n** and **m** of integer data-type.

You have to print numbers from **n** to **m** and **m** to **n** in different lines.

for eg. **n** = 5, and **m** = 10 so the output should be something like

```
5 6 7 8 9 10
```

```
10 9 8 7 6 5
```

Input Format

For each test case, you will be given the value of **n** and **m** as an integer-input.

Output Format

You have to print the output in n different lines.

Sample Input

```
4
11
```

Sample Output

```
4 5 6 7 8 9 10 11
11 10 9 8 7 6 5 4
```

Multiple Of n

Take an integer **n** and **m** as input, and print all the multiples of n till m.

Input Format

Single line Input **n** and **m** integer value

Output Format

Single line Output Integer values

Sample Input

7
98

Sample Output

0 7 14 21 28 35 42 49 56 63 70 77 84 91 98

Print Even

Given a integer **n**, print all even numbers from 0 till **n** (*including, if even*)

Input Format

Single line of Input An integer **n**

Output Format

A range of even numbers from 0 till n

Sample Input

10

Sample Output

0
2
4
6
8
10

Print 3 7 11 15

You will be given an integer input **n**, and you have to print the series 3, 7, 11, 15 till the integer is just less than **n**, in n different lines.

Input Format

For each test case, you will be given an integer input n.

Output Format

You have to print the series 3, 7, 11, 15 in n different lines.

Sample Input

51

Sample Output

3 7 11 15 19 23 27 31 35 39 43 47

Explanation

Here 51 although a part of the series is still not printed as we have to print the series just less than 51.

Print odd from n to m

You will get an integer input **n** and **m**. You have to print all the odd numbers from **n** to **m**

Input Format

For each test case, you will get integer input n and m.

Output Format

You have to print all the odd numbers in an integer format from n to m.

Sample Input

30

10

Sample Output

29 27 25 23 21 19 17 15 13 11

Print n, n-3, n-6

You will be given an input n of integer data type.

You have to print the series n, n-3, n-6....

Important points:

1. You have to print each number in a different line
2. Also you have to print till the time the printed value is greater than 0.

Take a look at the sample test cases.

Input Format

For each test case, you will be given n as an integer input.

Output Format

Print each number as an integer output in a different line.

Sample Input 0

20

Sample Output 0

20

17

14

11

8

5

2

Print n , $n-k$, $n-2k$, $n-3k$

"Print the series n , $n-k$, $n-2k$ in separate lines until the printed integer is greater than or equal to zero, using the integers n and k as input."

Input Format

For each test case, n will be given as an integer input in the first line,

k will be given as an integer input in the second line.

Output Format

Print the numbers as integer outputs in the separate line.

Sample Input 0

30

4

Sample Output 0

30

26

22

18

14

10

6

2

print a to z

You have to print characters from **a** till **z** using for or while loop

Input Format

No input will be given

Output Format

Print characters as given in the problem statement.

Print a, B, c, D, e, F, g..... 26 characters

Print a, B, c, D, e, F, g..... 26 characters where each character should be printed in a separate line.

Input Format

No input will be given

Output Format

Print as mentioned in the problem statement.

Sample Output 0

a B c D e F g H i J k L m N o P q R s T u V w X y Z

Print a, C, e, G, i, K... till 'z' or 'Z'

Print a, C, e, G, i, K, m, O, q, S till the last character is less than 'z' or 'Z' accordingly whether 'z' or 'Z' is a part of the series or not.

Input Format

No input will be given

Output Format

Print as given in the problem statement, where each character should be printed in a new line.

Sample Output 0

a C e G i K m O q S u W y

Print series AAA,bb,CCC,dd,EEE,ff till 26 lines

Write a program to print **AAA,bb,CCC,dd,EEE,ff**,.....uptil **26** lines ,where each String printed in a separate line.

Output Format

Print each String in a separate line.

Sample Output 0

```
AAA
bb
CCC
dd
EEE
ff
GGG
hh
III
jj
KKK
ll
MMM
nn
OOO
pp
QQQ
rr
SSS
tt
UUU
vv
WWW
xx
YYY
zz
```

Print *even* or *odd* from a list of integers

First take n as an integer input.

Then you will be given n integers as integer inputs and each time you have to print "even" if the number is an even number and "odd" if the number is an odd number.

Input Format

For each test case, You will be given an integer n of int data-type in the first line, After this you will be given n integers each of int data-type in separate lines.

Output Format

Print the string "even" or "odd" accordingly in separate lines

Sample Input

```
3
10
13
14
```

Sample Output

```
even
odd
even
```

Print $n/3$

Can you write a program for Alice, a computer science student, that takes an integer input from the user, divides it by 3 and prints the result, continuing until the result is greater than 0? Bob, a computer science professor, has given her this problem to help her understand a concept related to loops.

Note: Start printing from n , keep on updating n by dividing n by 3 each time, and print the updated value of n each time.

Input Format

The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows. For each test case, n will be given as an integer input.

Output Format

You have to print an integer each time in a separate line.

Sample Input

```
3
78
35
24
```

Sample Output

```
78 26 8 2
35 11 3 1
24 8 2
```

Multiples of 3, 5 and Both 3 and 5

Can you help Maria, a math teacher, write a program that takes an integer n as input and returns a list of all the multiples of 3, 5, and both 3 and 5 within a range of 1 to n for her lesson plan activity for her students?

Input Format

The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows. The first and the only line of each test case contains an integer n .

Output Format

Single line Output

Integer values (Multiples of 3, 5 or Both 3 and 5)

Sample Input 0

```
2
10
15
```

Sample Output 0

```
3 5 6 9 10
3 5 6 9 10 12 15
```

FizzBuzz problem

The rules of the FizzBuzz game are given below:

1. If a given number is divisible by **3** then the program should print "**Fizz**".
2. If a given number is divisible by **7** then the program should print "**Buzz**".
3. If a given number is divisible by both **3** and **7** then the program should print "**FizzBuzz**".
4. Otherwise print the given number as it is.

Input Format

For each test case, you will get **n** as an integer input.

Output Format

Print the output in a single line.

Sample Input

21

Sample Output

1 2 Fizz 4 5 Fizz Buzz 8 Fizz 10 11 Fizz 13 Buzz Fizz 16 17 Fizz 19 20 FizzBuzz

Explanation

Print the output according to the given condition.

Running sum and average

You are given an integer **n**.

Your task is to write a program to print the **running sum** from **1** to **n** and its **average**.

Input Format

For each test case, you will get **n** as an integer input.

Output Format

First line print the **sum**.

The second line prints its **average**.

Sample Input 0

5

Sample Output

15

3

Explanation

First line sum from 1 to 5 is 15.

Second line average is 3.

Running Sum for loop

"Maria, a student struggling with understanding how to find the running sum of a series of integers, is given a problem by her math teacher: Given a series of n integers as input, print the sum after taking input of each integer one at a time."

With guidance from her math teacher, Maria is able to successfully understand and solve a problem of finding the running sum of a series of integers. The problem gives her an example where the series of integers is 3, 4, 5, 6 and the output should be 3, 7, 12, 18. Initially, Maria is a little confused but with practice, she is able to understand the concept.

NOTE: Initially the sum is zero.

Input Format

The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows

For each test case, You will get the value n as an integer input in the first line, and n integers as integer input in different lines.

Output Format

You have to print the running sum, each time in a different line. You will be given a number n of integer data-type.

Sample Input

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

Sample Output

```
1 3 6 10 15
2 7 10
```

Explanation

In the first line we receive 5, meaning five integer inputs will be given as input.

Initially before taking any integer input the sum is zero.

When we take in the first integer input which is 3, the sum becomes 3.

When we take in the second integer input which is 2, the sum becomes 5.

After 3rd input, the sum becomes 7.

After 4th input, sum becomes 10.

After the 5th input, the sum becomes 15.

Fibonacci number

You have given an integer n , you have to print first n numbers of the fibonacci series till n .

Input Format

First and only line of input contains an integer n denoting the number.

Output Format

print fibonacci series till n

Sample Input

10

Sample Output

0 1 1 2 3 5 8 13 21 34

Print fibonacci series alternatively

you are given an input n as an integer input, Write a program to print the **alternate** fibonacci numbers starting from the **first** fibonacci till the **nth** fibonacci numbers accordingly, if the n th fibonacci number is part of the series or not.

Input Format

For each test case, you will get n as an integer input.

Output Format

Print the output in a single line.

Sample Input

10

Sample Output

0 1 3 8 21

Explanation 0

Alternate fibonacci till n th fibonacci are **0 1 3 8 21**

Nth Fibonacci Number

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

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

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'.

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

Output Format

For each test case, return its equivalent Fibonacci number.

Sample Input 0

6

Sample Output 0

8

Explanation 0

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.

Sample Input 1

5

Sample Output 1

5

Print nth Tribonacci number

nth term T_n of The Tribonacci sequence is defined as follows:

T_0 (0th term) = 0, T_1 (1st term) = 1, T_2 (2nd term) = 1, and $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ for $n \geq 0$.

Take n as an integer input, print the value of T_n (nth term) as an integer output.

Input Format

For each test case, n will be given as an integer input.

Output Format

Print the value of the Tribonacci number.

Sample Input	Sample Output	Sample Input	Sample Output
0	0	1	1
2	1	7	24
10	149	11	247
20	66012		

Print from n to 3 using while loop

Print numbers from n to 3 using a while loop where n is taken as input from the user.

Input Format

Int Given as Input 'N'

Output Format

Print all the numbers from N to 3

Sample Input

-9

Sample Output

-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2

Explanation

Printing from n=-9 to 3

Find Permutation

Given n and r , find the value of nPr . (formula of $nPr = \frac{n!}{(n-r)!}$)

Input Format

Take 2 inputs n and r as integers.

Output Format

Print an integer as output.

Sample Input

5
2

Sample Output

20

Explanation

Take $n = 5$ and $r = 2$.

Output should be 20 by the formulae mentioned above.

Steps till n greater than 0

Can you help Jake, a data analyst, come up with a solution for a problem of simulating the operation of a machine learning model by writing a program which takes an integer input n , performs a series of steps until the value of n becomes 0, subtract 1 from n if n is even, subtract 3 from n if n is odd and keep track of the total number of steps that the program performs and print this value at the end of the operation for his project?

Input Format

The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows. For each test case, you will be given the value of n as an integer data-type.

Output Format

Print the total steps as an integer data-type

Sample Input

```
2
20
37
```

Sample Output

```
10
19
```

Print all digits from end

Imagine Charlie is a high school student who is learning programming as a hobby. One day, he comes across the following problem:

"Write a program that takes an integer input from the user. The program should print the digits of the number starting from the end, going towards the first digit, where each digit should be printed on a separate line. Can you write a solution for this problem?"

Input Format

The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows. For each test case, print n as an integer input

Output Format

Print digits as an integer output as given in the problem statement, where each digit should be printed in a separate line.

Sample Input	Sample Output
7654	4 5 6 7
87543	3 4 5 7 8
987651	1 5 6 7 8 9

Number of Digits

Take an **integer N** as input.

Print the number of **digits** present in **N**.

Input Format

Single Line Input. An integer **N**.

Output Format

Single Line Output. An integer **N**.

Sample Input

523

Sample Output

3

Explanation

There are **three** digits in number **523**.

Print total steps when $n/2$

Take an integer input n and then keep on dividing n by 2, till the time n is greater than equal to 1.

Each time you divide n by 2, increment steps by 1.

Print the total number of steps in the end.

Input Format

For each test case, take an integer input n .

Output Format

Print the number of steps as an integer output.

Sample Input 0

32

Sample Output 0

6

Sample Input 1

20

Sample Output 1

5

Sample Input 2

34

Sample Output 2

6

Sample Input 3

109

Print steps and update maximum

Take n as input from the user. Then you will be given a list of n positive integers, each time you find a new maximum value, you have to increment the steps by 1.

Take steps as 0 initially and maximum value as -100 in the starting.

In the end print the number of steps performed.

Input Format

Take n as an integer input from the user.

Output Format

Print the total steps in the end.

Sample Input 0

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

Sample Output 0

```
6
```

Sample Input 1

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

Sample Output 1

```
5
```

Find GCD

Take two integer inputs x and y. Then print the gcd of these two numbers as an integer output.

Input Format

For each test case,

x will be given as an integer input in the first line,

y will be given as an integer input in the second line.

Output Format

Print the gcd as an integer output.

Sample Input 0

100

35

Sample Output 0

5

Sample Input 1

300

20

Sample Output 1

20

Sample Input 2

500

37

Sample Output 2

1

Calculate LCM

Take **x** and **y** as input. Write a function that takes in **x** and **y** as integer parameters. The function should return the **lcm** of these two numbers. In the end print the final **lcm**.

Input Format

The first line takes integer input as **x**.

The second line takes integer input as **y**.

Output Format

print the lcm of given numbers.

Sample Input

15
20

Sample Output

60

Explanation

LCM of 15 and 20 is 60

Reverse Digits

You are given a number **n** , Write a program to **reverse** digits of **n**.

Input Format

For each test case, you will get an integer input.

Output Format

Print the reverse

Sample Input

1234

Sample Output

4321

Print the final number

Take **n** as an integer input. Then take **n** digits as integer inputs and form a number from it and print that number as an integer output.

Input Format

For each test case, **n** will be given as an integer input in the first line,

then **n** digits will be given as integer inputs in each line.

Output Format

Print the final number as an integer output.

Sample Input

4
1
2
3
6

Sample Output

1236

Reverse n-digit number

Take a number n greater than or equal to zero as an integer input.

Then you will be given n digits as integer inputs and you have to form a number from it. Print the number formed.

Then you have to reverse the digits of this number. And then print the final reversed number in the next line.

Input Format

For each test case, n will be given as an integer input, then you will be given n digits as integer inputs in each line.

Output Format

Print the number formed from the digits in the first line.

Print the reversed number formed in the second line.

Sample Input 0

3
2
5
6

Sample Output 0

256
652

Sample Input 1

4
2
5
6
1

Sample Output 1

2561
1652

Print Primes - 1

You are given two integer inputs **x** and **y**. Make a **function** that takes in **x** and **y** as parameters. Then print all the **prime** numbers which lie between **x** and **y** (**x** and **y** both **inclusive** and $y > x$).

Input Format

First line takes an Integer input from the user as **x**.

Second line takes an Integer input from the user as **y**.

Output Format

Print all the prime numbers between given intervals.

Sample Input 0

10
20

Sample Output 0

11 13 17 19

Explanation 0

All prime numbers between **10** to **20** are **11 13 17 19** .

Prime checker 2

Write a Java program to check whether a number is a Prime number or not.

Input Format

First line contains a number A.

Output Format

Print Yes or No.

Sample Input 0

5

Sample Output 0

Yes

Print all factors of a number

Take a whole number n as an integer input and print all the factors of it such that each factor should be printed in a separate line.

Input Format

n will be given as an integer input.

Output Format

Print all the factors of the number where each factor should be printed in a separate line.

Sample Input 0

12

Sample Output 0

1
2
3
4
6
12

Sample Input 1

30

Sample Output 1

1
2
3
5
6
10
15
30

Print all unique prime factors

Take a whole number n as an integer input and then print all the unique prime factors of n such that each prime factor is printed in a separate line.

Input Format

n will be given as an integer input

Output Format

Print the prime factors as an integer value where each prime number should be printed in a separate line.

Sample Input	Sample Output	Sample Input	Sample Output
45	3 5	100	2 5
240	2 3 5	350	2 5 7

Check if an Armstrong number or not

Take n as an integer input.

Then take n as an integer parameter.

In the end, print "true" if n is armstrong and "false" if n is not armstrong.

Note: An armstrong number is a number which is equal to the sum of the cube of its digits.

Input Format

For each test case, n will be given as an integer input.

Output Format

Print "true" or "false" accordingly.

Sample Input 0

782

Sample Output 0

false

Sample Input 1

153

Sample Output 1

true

Sample Input 2

370

Sample Output 2

true

Print Armstrong in a range

Take x and y as integer inputs.

Print all the Armstrong numbers in separate line which lie in the range x to y (both x and y inclusive)

Use the function `isArmstrong()` which checks if a number is an Armstrong number or not and returns true or false accordingly.

Input Format

For each test case,

x will be given in the first line

y will be given in the second line.

Output Format

Print the numbers as integer outputs where each number is printed in a separate line.

Sample Input 0

```
1
200
```

Sample Output 0

```
1
153
```

Rotate 7-digit number to right by three

Take n as an integer input, you have to pick the last 3 digits of the number of and put them in the starting.

eg. 1234567 is given, then this number should transform to 5671234.

Input Format

A 7 -digit number will be given as an integer input.

Constraints

Only a 7-digit number will be given as input.

Output Format

Print the rotated number

Sample Input 0

2345678

Sample Output 0

6782345

Sample Input 1

1236789

Sample Output 1

7891236

Divide n by 2 3 5 and tell steps

Take a natural number n as an integer input, and variable steps of integer type as input. Then perform the following operations on it.

1. If the number is divisible by 2, then keep on dividing the number n by 2, till the time the number is divisible by 2 and also increment the variable steps by 2, each time you divide the number by 2.
2. Also, check If the number is divisible by 3, then keep on dividing the number n by 3, till the time the number is divisible by 3 and also increment the variable steps by 3, each time you divide the number by 3.
3. Also, If the number is divisible by 5, then keep on dividing the number n by 5, till the time the number is divisible by 5 and also increment the variable steps by 5, each time you divide the number by 5.

In the end print the value of the variable steps in the first line and final value of number n in the second line.

Input Format

For each test case, n will be given in the first line,

steps will be given in the second line.

Output Format

Print the final value of steps in the first line as an integer output,

and print the final remaining value of n in the second line as an integer output.

Sample Input	Sample Output	Sample Input	Sample Output
30 0	10 1	100 20	34 1
210 7	17 7	243 0	15 1

Odd Even and Divisibility by 3

Take n as an integer input. After this you will be given n numbers as integer inputs and you have to print each time if the number is **Even** or **Odd**. And Print **“Divisible by 3”** if the number is a multiple of 3 and print **“Not Divisible by 3”** if the number is not a multiple of 3

Input Format

- N as an Integer Input
- N numbers as an Integer Input

Output Format

As Described in Problem Statement

Sample Input 0

5
9
11
2
6
15

Sample Output 0

Odd Divisible by 3
Odd Not Divisible by 3
Even Not Divisible by 3
Even Divisible by 3
Odd Divisible by 3

Even Odd 2

First Take an integer t as an input. Then you will be given t number test cases.

In each test case, " First take an integer n as an input.

Then take n integer inputs, and for each one of these inputs, print "Even" if the given integer is an even number and print "Odd" if the given number is an odd number. "

Make sure to write the complete output for each test case in a separate line.

Also, within the same line give a space between two words.

Input Format

In the first line an integer input t will be given.

From the next line onward for each test case, first you will get an integer input n .

Then for the same test case, you will get n integer numbers.

Output Format

Make sure to write the complete output for each test case in a separate line.

****Also, within the same line give a space between two words.**

Sample Input 0

```
2
4
10 21 33 35
3
27 31 32
```

Sample Output 0

```
Even Odd Odd Odd
Odd Odd Even
```

Explanation 0

Here t is 2. So we will be having 2 test cases. Then in the first test case, we have $n=4$, so we have 4 numbers in total. Then we are given 4 numbers. 10 is even so we print Even. 21 is odd so we print Odd. 33 is odd so we print Odd. 35 is odd so we print Odd.

Then for the second test case we have $n=3$, so we have 3 numbers in total. So we print Odd Odd Even

Sample Input 1

4
1
11
2
12 14
5
22 33 44 57 99
4
12 15 17 19

Sample Output 1

Odd
Even Even
Even Odd Even Odd Odd
Even Odd Odd Odd

Sample Input 2

2
2
12 13
3
11 98 97

Sample Output 2

Even Odd
Odd Even Odd

Sample Input 3

3
2
17 29
3
12 39 45
4
3 2 1 -480

Sample Output 3

Odd Odd
Even Odd Odd
Odd Even Odd Even