Week 1 program 230701294

R SARVESH

1. Given an input integer, you must determine which primitive data types are capable of

properly storing that input.

Input Format

The first line contains an integer, , denoting the number of test cases.

Each test case, , is comprised of a single line with an integer, , which can be arbitrarily large or

small.

Output Format

For each input variable and appropriate primitive , you must determine if the given primitives

are capable of storing it. If yes, then print: n can be fitted in:\* dataType

If there is more than one appropriate data type, print each one on its own line and order them by

size (i.e.: ).

If the number cannot be stored in one of the four aforementioned primitives, print the line:

n can&#39;t be fitted anywhere.

Sample Input

5

-150

150000

1500000000

213333333333333333333333333333333333

-100000000000000

Sample Output

-150 can be fitted in:

\* short

\* int

\* long

150000 can be fitted in:

\* int

\* long

1500000000 can be fitted in:

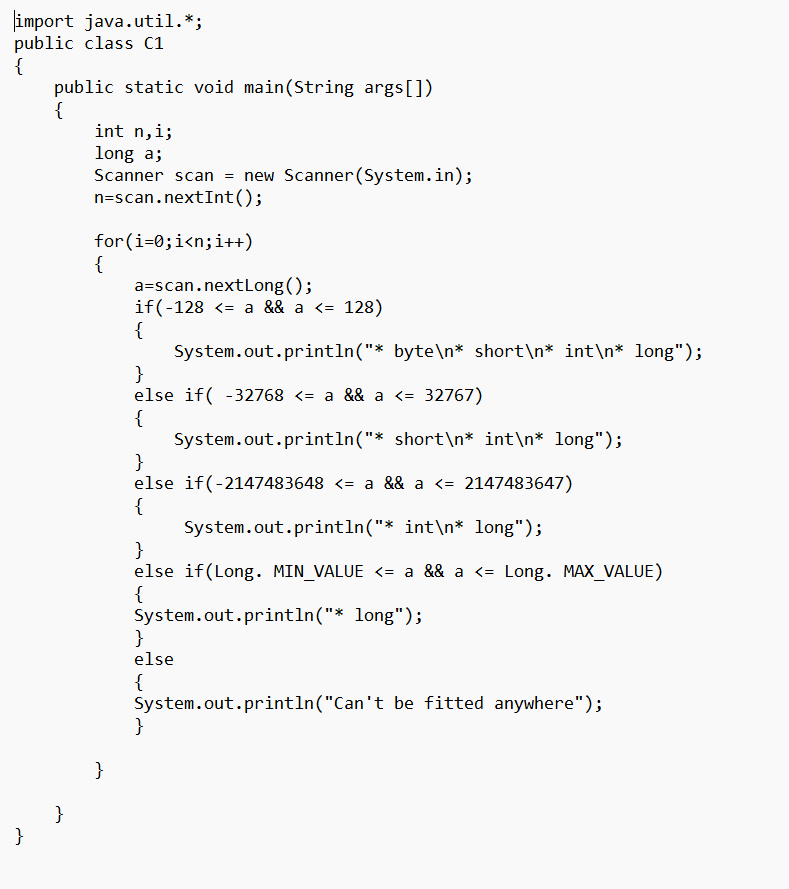
\* int

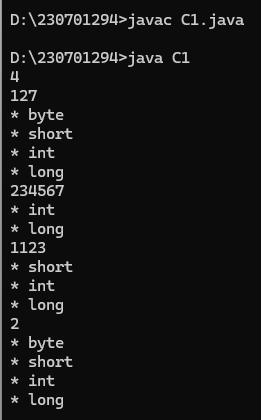
\* long

213333333333333333333333333333333333 can&#39;t be fitted anywhere.

-100000000000000 can be fitted in:

\* long





2) You are developing a financial application that needs to handle both whole numbers and

decimal values. The application takes user inputs as integers (e.g., representing amounts in cents)

and needs to convert them to double for further calculations (e.g., converting cents to dollars).

The application should:

1. Take an integer amount in cents as input.

2. Convert this integer to a double to represent the amount in dollars.

3. Ensure that the conversion is accurate and the output is properly formatted to two decimal

places.

Describe how you would implement this, and what the expected output would look like for the

following scenarios:

• Input amount: 1250 (cents)

• Input amount: 50 (cents)

Output:

Expected Output:

1. Conversion of Integer to Double:

o) Convert the integer amount in cents to double by dividing it by 100.0.

2. Formatting the Output:

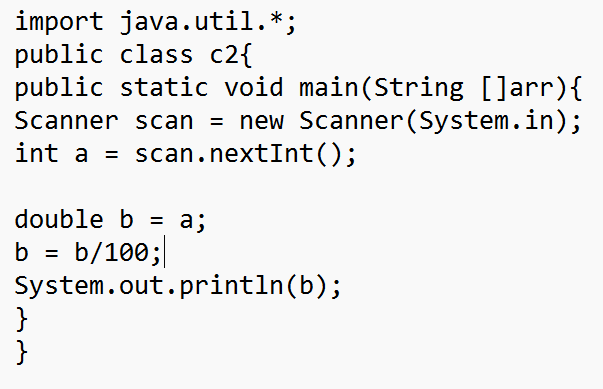
o )Format the resulting double value to two decimal places for proper representation

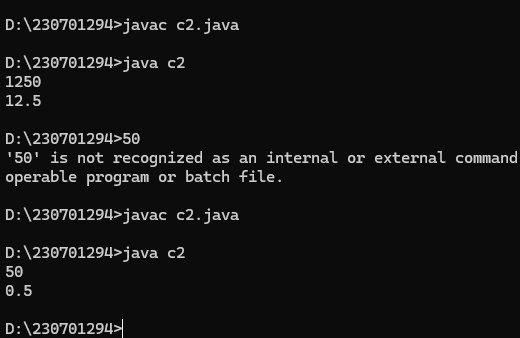
as dollars.

3. Output for Given Scenarios:

o) For an input of 1250 (cents), the output should be: 12.50 (dollars).

o) For an input of 50 (cents), the output should be: 0.50 (dollars).





3) In a game, the player's score is calculated as a double value with high precision.

However, for display purposes, you need to show the score as an integer.

Questions:

1. Input:

o A player's score is 456.89 (stored as a double).

o You need to cast this score to an integer for display on the leaderboard.

Output:

o Show how you would cast the score to an integer and what the resulting score

would be.

o Expected Output: The score after type casting to int is 456.

2. Input:

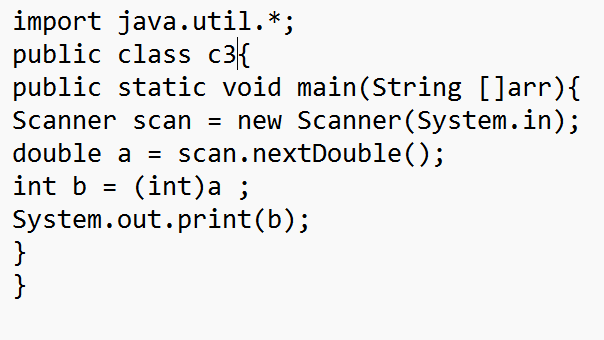
o Another player's score is 1234.56.

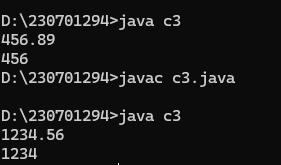
Output:

o After type casting, the score should be 1234.

o Discuss how rounding might affect the perception of the score and whether

additional logic should be implemented for rounding.





4. You are developing a payroll system where you need to calculate the adjusted salary

based on a percentage increase. The initial salary is given as an int, and the percentage increase

is given as a double.

Questions:

1. Input:

o Initial salary: 45000 (stored as int)

o Percentage increase: 7.5 (stored as double)

Output:

o Calculate the new salary after applying the percentage increase.

o Show how type promotion affects the calculation and what the resulting salary

would be.

Expected Output:

o The new salary after a 7.5% increase should be 48375.0 (as a double).

2. Input:

o Another initial salary: 32000 (stored as int)

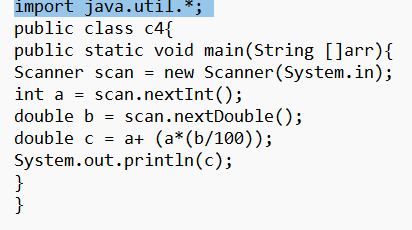
o Percentage increase: 12.3 (stored as double)

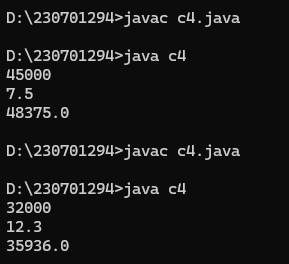
Output:

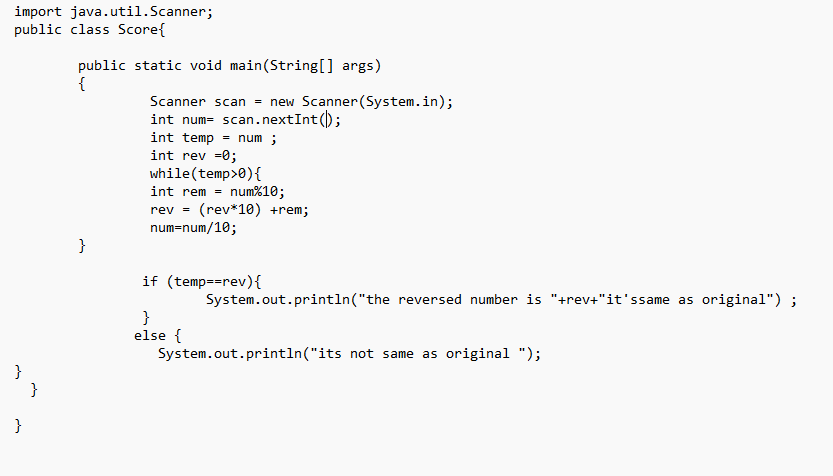
o Calculate the new salary and discuss how type promotion is applied in the

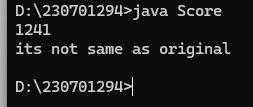
calculation.

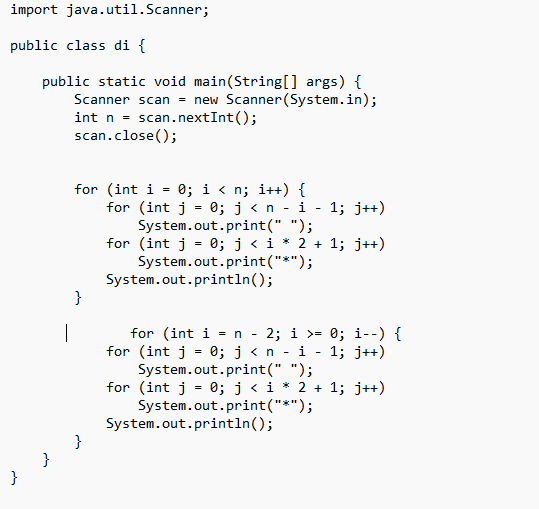
Expected Output: The new salary after a 12.3% increase should be 35976.0 (as a

double). 









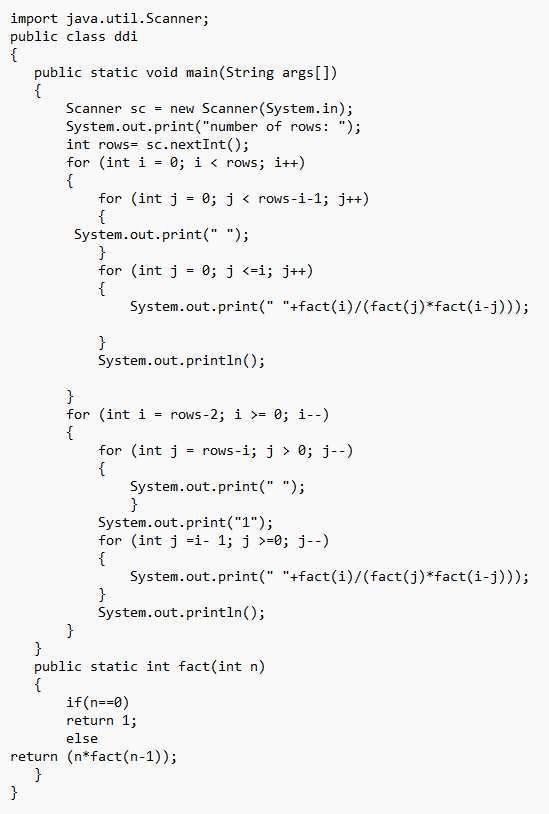


Write a Java program that prints a half-diamond pattern where each row

contains elements from Pascal's Triangle up to the middle row. For a given

integer n, generate a pattern with 2n-1 rows. The first n rows should display the

elements of Pascal's Triangle in increasing order, while the next n-1 rows should

display them in decreasing order, forming a half-diamond.

