Operators in Python

An operator in a programming language is a symbol that perform specific mathematical / relational / logical operation and produce final result.

Arithmetic operators

- Addition
- Subtraction
- Division
- mod (%)
- Multiplication
- Floor Division (//)
- To the power (**)

Arithmetic operators are used with numeric values to perform common mathematical operations

```
a = 5
b = 3
a + b
8
a - b
2
```

\# Division operator will always yield float value

```
a / b
1.6666666666666667
5 / 5
1.0
a * b
15
b - a
-2
1.5 + 1
2.5
1.0 + 1.0
2.0
```

\# string concatenation

```
first = "Uvaraj."
last = "Thillai"
```

```
print (first + last)
```
Uvaraj Thillai
```
type ("1")
```
str
```
print ("1" + "1")
```
11
```
first - last
```
Type Error : unsupported operand type(s) for - : 'str' and 'str'

```
first * 3
```
' Uvaraj Uvaraj Uvaraj '

```
first * first
```
Type Error : Can't multiply sequence by non-int of type 'str'

Modulus % : It will give remainder
```
print (a, b)
```
5 3
```
print (a % b)
```
2
```
print (10 % 10)
```
0

In Floor Division, the result is floored to the nearest
                                smaller integer. (left side)
```
print (a, b)
```
5 3
```
print (a // b)
```
1
```
print (-3 // 2)
```
-2

$\frac{5}{3} = 1.666..$

Nearest smaller integer is 1.

$\frac{-3}{2} = -1.5$

Nearest smaller integer is -2.

To the power accepts two values : Base and Power
```
print (a, b)
```
5 3
```
print (a ** b)
```
125
```
print (2 ** 3)
```
8

$5^3 = 5 \times 5 \times 5 = 125$

$2^3 = 2 \times 2 \times 2 = 8$

Comparison Operators

- $==$ (True if equal)
- $!=$ (True if not equal)
- $<$ (less than)
- $>$ (greater than)
- $<=$ (less than or equal to)
- $>=$ (greater than or equal to)

Comparison operators are used to compare two values.

```
a = 6
b = 4
print (a == b)
False
print (3 == 3)
True
print (a != b)
True
print (3 != 3)
False
print (a > b)
True
print (b > a)
False
print (6 >= 3)
True
print (3 >= 3)
True
print (2 >= 3)
False
print (6 <= 3)
False
```

Assignment operators

- $=$
- $-=$
- $+=$
- $/=$
- $*=$

Assignment operators are used to assign values to variables.

```
a = 5
a += 3
print (a)
8
a -= 4
print (a)
4
a /= 2
print (a)
2.0
```

Logical operators

- and
- or
- not

Logical operators are used to combine conditional statements

```
2 > 3    and    3 > 2
False

4 > 3    and    3 > 2
True

3 > 2    or    2 > 3
True

2 > 3    or    3 > 5
False

not    2 > 3
True

not    3 > 2
False
```

Special operators

- in Operator (Membership Operator)
- is Operator

The in Operator checks if a specified value exists within a sequence, such as a string, list, tuple or dictionary.

name = "Uvaraj Thillai"

"U" in name

True

"u" in name

False

"Uvaraj" in name

True

· The is operator checks if two variables refer to the same object in memory. It returns True if they point to the same object, and False otherwise.

```
a = 5
b = 5
print (id(a) , id(b))
```
140349867747760   140349867747760

a is b

True

```
a = 34
b = 45
print (id(a) , id(b))
```
140349867748688   140349867749040

a is b

False

Challenge

① What is the output of the following Program ?
```
print (type (6/3))
```
< class 'float' >

② What is the output of the following Program ?
```
print (6 % 2)
```
0

③ What is the output of the following Program ?
```
print (2 > 3 and 4 < 2)
```
False

④ What is the output of the following Program ?

```
print( not  2 < 4)
```

False

⑤ what is the output of the following Program ?

```
print (0 or 1)
```

1

⑥ Problem Description :

You are given a positive integer r denoting the radius of a sphere as a parameter. Write a program to calculate the volume of the sphere. The volume of a sphere having radius R is given by $(4 * \pi * R^3) / 3$.

__Note 1__ : Return the volume of the sphere up to two decimal places. You can use round ().

__Note 2__ : Use pi as 22/7 (not math.pi)

__Input Format__ : The first line indicates the No. of Test cases. For each Test case, there will be one line of input. The one line contains r in integer format.

__Output Format__ : The volume of Sphere in float format is printed for each Test case in new line.

```
def volume_sphere (r) :
    ''' input : r = Input in Integer format
        output : return volume of sphere upto two decimals
    '''
    vol = None
    pi = 22/7
    vol = (4 * pi * r ** 3) / 3

    return   round (vol, 2)

# Read the No. of Test cases
num_test_cases = int (input ())

# Process each Test case
for _ in range (num_test_cases) :
    # Read the radius for the current test case
    r = int (input ())
    # Calculate and print the volume of sphere
    print (volume_sphere (r))
```

Sample Input : 1
                8

Sample Output : 2145.52

Sample Input : 2
output   8.
              2145.52
              4
              268.19

⑦ Problem Description :

You have 2 strings (A and B). You have to concatenate two strings and return the concatenated string.

Problem constraints :.

1 ≤ A.size() ≤ 1000

1 ≤ B.size() ≤ 1000

Input format.

First argument containing a lower case string A
Second argument containing a lower case string B

output Format

Concatenated String.

Sample input

A = "uvaraj" , B = "thillai"

Sample output

uvaraj thillai

```
def concatenate_strings (A : str, B : str) -> str :

    return A + B
```

Example Input   output : uvarajthillai

A = "uvaraj"
B = "thillai"

Example output

result = concatenate_strings (A, B)
print (result)