Lab 3 Solutions
Groups 9-16

Action Game 1

Ada has health x and Adi has health y. Ada's character is tired and can no longer fight. On the other hand, Adi's character is on fury and is attacking Ada continuously. Each attack that Adi does imparts a damage of 7 on Ada's health. Moreover, each time Adi attacks, Adi heals which increases their health by 6.
You need to find the first time Adi's health surpasses Ada's health and print the value of Ada's and Adi's health at that time.

Input Format

The first line contains the health of Ada which is x
The first line contains the health of Adi which is y

Constraints

$0 < x < 10^4$
$0 < y < 10^4$

Output Format

The first line contains the final health of Ada
The second line contains the final health of Adi

Sample Input 0

```
40
20
```

Sample Output 0

```
26
32
```

Explanation 0

Initially, $x = 40$ and $y = 20$
When, Adi attacks Ada, Ada suffers a damage of 7 and Adi heals by 6
Implying, after the first hit $x = 40 - 7 = 33$ and $y = 20 + 6 = 26$
Now, after the second hit $x = 33 - 7 = 26$ and $y = 26 + 6 = 32$
We see Adi's health just surpassed Ada's health. So, we print the current value of x and y which is 26 and 32 respectively.

Solution:

```
x=int(input())
y=int(input())
while x>=y:
    x=x-7
    y=y+6

print(x)
print(y)
```

Two Two Za?

Given two numbers x and y. Print the multiplication table of x from 1 to y.

Input Format

The first line contains x
The second line contains y

Constraints

$0 < x, y \leq 100$

Output Format

Print the multiplication table of x from 1 to y.

Sample Input 0

```
9
7
```

Sample Output 0

```
9
18
27
36
45
54
63
```

Solution:

```
x = int(input())
y = int(input())
for i in range(1,y+1):
    print(x*i)
```

Positives and Negatives

Ask the user for number of inputs n. Then, proceed to input n integers in separate lines, i.e. both positive and negative values. Return two integers in separate lines: where the first element is the count of positive numbers and the second element is the sum of negative numbers

Input Format

1st line contains the value of integer n

Next n lines contain an integer value

Constraints

$-100000 <= eachvalue <= 100000$
$eachvalue \neq 0$

Output Format

1st line contains the count of positive numbers

2nd line contains the sum of negative numbers

Sample Input 0

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

Sample Output 0

```
10
-65
```

Explanation 0

Count of positive numbers = count (1,2,3,4,5,6,7,8,9,10) = 10

Sum of negative numbers = (-11 -12 -13 -14 -15) = -65

Solution:

```
n=int(input())
count_neg = 0
sum_pos = 0
for i in range(n):
    x = int(input())
    if x>=0:
        count_neg += 1
    elif x<0:
        sum_pos += x
print(count_neg)
print(sum_pos)
```

Parallel Resistors

If two or more resistors are connected in parallel, the overall resistance of the circuit reduces. It is possible to calculate the total resistance of a parallel circuit by using this formula:

R1 in Parallel with R2, R3, and So On

$$\frac{1}{R_{Total}} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \ldots + \frac{1}{Rn}$$

Ask the user for input of resistance values and keep on taking inputs till user enters an invalid resistance value (value <= 0). Calculate the total resistance of the circuit given the resistance values.

Input Format

Take 1 by 1 input from the user, each integer value being a resistance value. Inputs should be in separate lines, and the process of input should get terminated for invalid value (value <= 0)

Constraints

$1 \leq R1, R2, R3, \ldots, Rn \leq 1000$

Note: Non zero valid resistance values will be entered by user

Output Format

Output a float value, which is the total resistance of the parallel circuit

Sample Input 0

```
3
3
3
0
```

Sample Output 0

```
1.0
```

Explanation 0

$\frac{1}{3} + \frac{1}{3} + \frac{1}{3} = \frac{1}{R}$

$\frac{1}{3} = \frac{1}{R}$

$R = 1$

Solution:

```
R = 0
while True:
    k = int(input())
    if k>0:
        R = R+(1/k)
    else:
        break
print(1/R)
```

Less than 10

Given a positive integer n, sum all of its digits to get a new number. Repeat this operation until the new number is less than 10 and print it.

Input Format

The first line contains the integer n

Constraints

$1 \leq n < 10^8$

Output Format

Print the new number you get after repeating the operation until it is less than 10.

Sample Input 0

```
8486
```

Sample Output 0

```
8
```

Explanation 0

8 + 4 + 8 + 6 = 26
2 + 6 = 8
So, we print 8

Solution:

```
n=int(input())
while n>=10:
```

```
    sum_of_digits=0
    while n>0:
        r=n%10
        n=n//10
        sum_of_digits=sum_of_digits+r
    n=sum_of_digits
print(n)
```

Practice Questions

FizzBuzz

Given a number N. Iterate from 1 to N and print according to the following rules:

1. For multiples of three, print "Fizz" instead of the number.
2. For multiples of five,print "Buzz".
3. For numbers which are multiples of both three and five, print "FizzBuzz".
4. If none of the rules above apply, print the number

Input Format

The first and only line contains a single integer N

Constraints

$0 < N \leq 100$

Output Format

The program should output N lines.
Each line should be either an integer, or "Fizz" or "Buzz" or "FizzBuzz" according to the rules.

Sample Input 0

```
5
```

Sample Output 0

```
1
2
Fizz
4
Buzz
```

Sample Input 1

```
16
```

Sample Output 1

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
```

Solution:

```python
N = int(input())
for i in range(1,N+1):
    if i%3 == 0:
        if i%5 == 0:
            print("FizzBuzz")
        else:
            print("Fizz")
    elif i%5 == 0:
        print("Buzz")
    else:
        print(i)
```

Ramanujan Numbers 1

Ramanujan Numbers are the numbers that can be expressed as the sum of two cubes in two different ways.

For example, 1729 is a Ramanujan number which can be expressed as $12^3 + 1^3$ and $10^3 + 9^3$.

Your task is to determine if a given number N is a Ramanujan number. If N is a Ramanujan number then print True. Otherwise, print False

Input Format

Input contains an Integer "N"

Constraints

$0 < N \leq 10000000$

Output Format

Output returns a boolean value, True if it's a Ramanujan number, else False

Sample Input 0

```
1729
```

Sample Output 0

```
True
```

Explanation 0

$1729 = 12^3 + 1^3 = 10^3 + 9^3$

Solution:

```
N = int(input())
cnt = 0
for i in range(1, int(N**(1/3))+1):
    for j in range(i+1, int(N**(1/3))+1):
        if((i**3) + (j**3) == N):
            cnt+=1
        if(cnt==2):
            break
    if(cnt==2):
```

```
        break
if(cnt==2):
    print(True)
else:
    print(False)
```