Introduction of Functions in Python

Q 1: Introduction to basic functions with pass by value. What will be the output?

```
def CovidTimeline( x ):
    if (x < 2020):
        print("Welcome to Before Covid life")
    else:
        print("Welcome to After Covid life")
CovidTimeline(2019)
CovidTimeline(2022)
```

```
Sol: Welcome to Before Covid life
     Welcome to After Covid life
```

Q 2: Functions with pass by reference. Guess the output.

```
def myFun(x):
  x[0] = 20
list = [10, 11, 12, 13, 14, 15]
myFun(list);
print(list)
```

Sol: `[20, 11, 12, 13, 14, 15]`

Scope of variables. Local vs global.

Q 3: Predict the output.

```
def myfunc():
  x = 300
  print(x)

myfunc()
```

Sol:300

```
def myfunc():
  x = 30
  def myinnerfunc():
    print(x)
  myinnerfunc()

myfunc()
```

Sol:30

x = 250
def myfunc():
  print(x)
myfunc()
print(x)
	x = 300
	def myfunc():
	  x = 200
	  print(x)
	myfunc()
	print(x)

def myfunc():
  global x
  x = 100
myfunc()
print(x)
	x = 400
	def myfunc():
	  global x
	  x = 50
	myfunc()
	print(x)

Sol:250, 250 Sol:100	Sol:200, 300 Sol:50

Functions with arguments.

Q 4: Predict the output

def my_function(fname, lname):
  print(fname + " " + lname)
 	
my_function("Akash", "Yadav")
	def my_function(fname, lname):
	  print(fname + " " + lname)
	
	my_function("Akash")

Sol: Akash Yadav	Sol: **TypeError**: my_function() missing 1 required positional argument: 'lname'

Q 5: Predict the output

Functions with arbitrary arguments	Functions with Keyword arguments

def my_function(*avenger):
  print("My fav avenger is " + avenger[2])
 	
my_function("Cap", "Natasha", "Tony")
	def my_function(child3, child2, child1):
	  print("The youngest child is " + child3)
	
	my_function(child1 = "Amar", child2 = "Akbar", child3 = "Anthoni")

Sol: My fav avenger is Tony	Sol: The youngest child is Anthoni

Q 6: Predict the output

Function with arbitrary keyword args	Function with default argument

def my_function(**Diwar):
  print("I have money, fame and power. What do you have " + Diwar["option1"])
 	
my_function(option1 = "Mother", option2 = "Father")
	def my_function(**country** = "Norway"):
	  print("I am from " + country)
	
	my_function("Sweden")
	my_function("India")
	my_function() # no arg given
	my_function("Brazil")

```
Sol: I have money, fame and power.
     What do you have Mother
```

```
Sol: I am from Sweden
     I am from India
     I am from Norway
     I am from Brazil
```

Q 7: Predict the output

Passing list as argumets	Return values

def my_function(food):
  for x in food:
  print(x)
 	
fruits = ["apple", "banana", "cherry"]
 	
my_function(fruits)
	def my_function(x):
	  return 5 * x
	
	print(my_function(3))
	print(my_function(5))
	print(my_function(9))

```
Sol: apple
     Banana
     Cherry
```

```
Sol: 15
     25
     45
```

Q 8:

There is a new mobile game that starts with consecutively numbered clouds. Some of the clouds are thunderheads and others are cumulus. The player can jump on any cumulus cloud having a number that is equal to the number of the current cloud plus 1 or 2 . The player must avoid the thunderheads. Write a python program with function to determine the minimum number of jumps it will take to jump from the starting position to the last cloud. It is always possible to win the game.

For each game, you will get a list of clouds numbered 0 if they are safe or 1 if they must be avoided.

Example

C=[0, 1, 0, 0, 0, 1, 0]

Index the list from 0 to 6. The number on each cloud is its index in the list so the player must avoid the clouds at indices 1 and 5. They could follow these two paths: 0--2--4--6 or 0--2--3--4--6 . The first path takes 3 jumps while the second takes 4. Return 3 .

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

Input Format

The first line contains an integer , the total number of clouds.
The second line contains space-separated binary integers describing clouds C[i], where $0 \leq i < n$

Constraints

$2 \leq n \leq 100$

$C[i] \in \{0, 1\}$

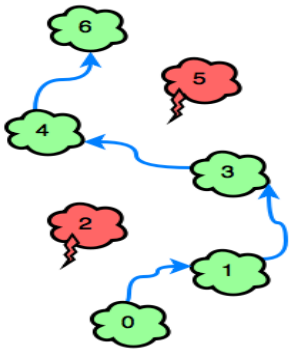
Output Format

Print the minimum number of jumps needed to win the game.

Test Case 1

Sample Input
7
0 0 1 0 0 1 0
Sample Output 0
4

Explanation for Test case 1:
The player must avoid C[2] and C[5] . The game can be won with a minimum of 4 jumps.



Solution:

```
def jumps_on_clouds(n, C):
    C.append(0)
    ans = 0
    position = 0
    while position < n-1:
        position += (C[position+2] == 0) + 1
        ans += 1
    return ans
n = int(input())
C = list(map(int, input().rstrip().split()))
print(jumps_on_clouds(n, C))
```

Q 9:

There is a string S, of lowercase English letters that is repeated infinitely many times. Given an integer N , find and print the number of letter a's in the first N letters of the infinite string.

Example
S = ‘abcac’
N=10

The substring we consider is ‘abcacabcac’, the first 10 characters of the infinite string. There are 4 occurrences of ‘a’ in the substring.

Input Format
The first line contains a single string, S
The second line contains an integer, N

Output Format
Print the number of letter a's in the first N letters

Constraints
$1 \leq |S| \leq 100$
$1 \leq N \leq 50$

Test Case 1:
Input
aba
10
Output
7

Explanation of *Test Case 1*:

The first letters of the infinite string are ‘abaabaabaa’. Because there are 7 a's, we return 7 .

Solution:

```
def repeated_str(S, N):
    p = S[0: N % len(S)].count('a')
    q = S.count('a')
    r = N // len(S)
    y = r *q + p
    return y

S = input()
N = int(input())
print(repeated_str(S, N))
```