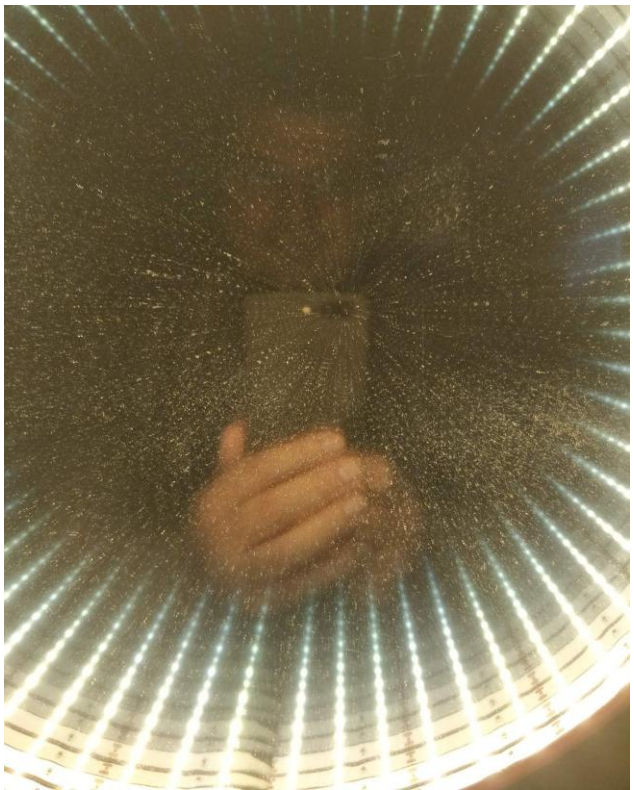
ABORDĂRI ITERATIVE ȘI RECURSIVE

CE ESTE RECURSIA?

Recursia este procesul în care funcția se autoapelează, apelul poate fi direct sau indirect. Folosind recursia, anumite probleme pot fi rezolvate cu ușurință.

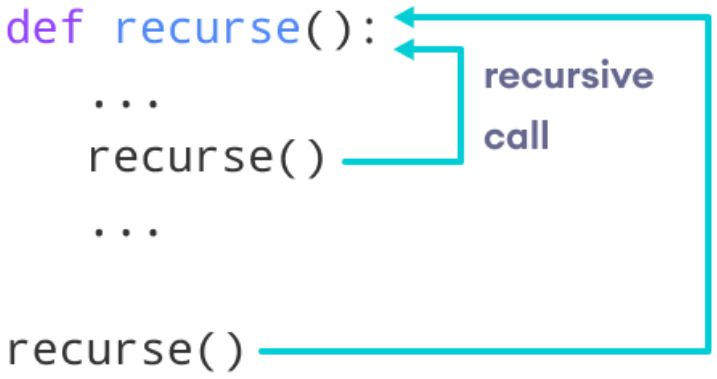
Cel mai cunoscut exemplu de recursivitate în fizică sunt imaginile infinite formate din două oglinzi paralele. Imaginea formată acționează ca un obiect și formează o nouă imagine în acest fel procesul este definit în termeni de sine și, prin urmare, arată recursivitate.



FUNCȚIA RECURSIVĂ în PYTHON

În Python, știm că o funcție poate apela alte funcții. Este chiar posibil ca funcția să se autoapeleze singură. Aceste tipuri de construcții și reprezintă funcțiile recursive.

Următoarea imagine arată funcționarea unei funcții recursive numită recurs.



ITERATIV sau RECURSIV

Provocarea pe care ne vom concentra este să definim o funcție care returnează rezultatul 1+2+3+4+….+n, unde n este un parametru citit de la tastatură.

ABORDAREA ITERATIVĂ

main.py

Run

Shell

```
1  # Funcţia iterativă care va întoarce rezultatul: 1 +2+3+4+5+...+n)
2  # Pentru n citit de la tastatură
3  def iterativeSum(n):
4      s=0
5      for i in range(1,n+1):
6          s+= i
7      return s
8  n=int(input('n='))
9  print('Suma 1+2+3+ ... =',iterativeSum(n))
```

```
n=3
Suma 1+2+3+ ... = 6
>
```

ITERATIV sau RECURSIV

Provocarea pe care ne vom concentra este să definim o funcție care returnează rezultatul 1+2+3+4+….+n, unde n este un parametru citit de la tastatură.

ABORDAREA RECURSIVĂ

main.py

Run

Shell

```
1  # Funcția recursivă care va întoarce rezultatul: 1 +2+3+4+5+...+n)
2  # Pentru n citit de la tastatură
3  def recursiveSum(n):
4      if (n > 1):
5          return n + recursiveSum(n - 1)
6      else:
7          return n
8  n=int(input('n='))
9  print('Suma 1+2+3+ ... =',recursiveSum(n))
```

```
n=3
Suma 1+2+3+ ... = 6
>
```

AVANTAJE și DEZAVANTAJE

AVANTAJELE RECURSIEI

- Funcțiile recursive fac codul să arate curat și elegant.
- O sarcină complexă poate fi împărțită în sub-probleme mai simple folosind recursia.
- Generarea secvenței este mai ușoară cu recursia decât folosind o iterație imbricată.

DEZAVANTAJELE RECURSIEI

- Uneori, logica din spatele recursiei este greu de urmărit (de înțeles).
- Apelurile recursive sunt costisitoare (ineficiente), deoarece ocupă multă memorie și timp.
- Funcțiile recursive sunt greu de depanat.

REZOLVĂ INDEPENDENT

Modificați ambele funcții (varianta iterativă și cea recursivă) de mai sus pentru a:

- aduna numai numere pare: de ex. 2+4+6+8+….
- aduna numai numere impare: de ex. 1+3+5+…
- aduna numere, numărând în 5: de ex. 5+10+15+…

EXEMPLE DE FUNCȚII RECURSIVE

EXEMPLUL 1

Elaborați funcție recursivă pentru a calcula factorialul unui număr întreg.
Factorialul unui număr este produsul tuturor numerelor întregi de la 1 la acel număr.
De exemplu, factorialul lui 6 (notat cu 6!) este 1*2*3*4*5*6 = 720.

main.py

Run

Shell

```
1 def factorial(x):
2     if x == 1:
3         return 1
4     else:
5         return (x * factorial(x-1))
6 num =int(input('numarul='))
7 print(num,"!=", factorial(num))
```

```
numarul=3
3 != 6
>
```

EXEMPLE DE FUNCȚII RECURSIVE

EXEMPLUL 2

Elaborați funcție recursivă care va afişa şirul lui FIBONACCI: 0, 1, 1, 2, 3, 5, 8….

De exemplu, pentru n=5 se va afișa: 0, 1, 1, 2, 3

main.py

Run

Shell

```
1  # fincția recursivă
2  def recursive_fibonacci(n):
3    if n <= 1:
4        return n
5    else:
6        return(recursive_fibonacci(n-1) + recursive_fibonacci(n-2))
7
8  n_terms = int(input('Dati nr='))
9
10 # verificam daca numarul de termeni este valid
11 if n_terms <= 0:
12   print("Intrare nevalidă ! Vă rugăm să introduceți o valoare pozitivă")
13 else:
14   print("Șirul lui Fibonacci este:")
15 for i in range(n_terms):
16     print(recursive_fibonacci(i))
```

```
Dati nr=5
Șirul lui Fibonacci este:
0
1
1
2
3
>
```

REZOLVĂ INDEPENDENT

1. Modificați funcțiile recursive FACTORIAL și FIBONACCI din recursivă în iterativă.

2. Elaborați o funcție iterativă și una recursivă care va calcula A la puterea B, pentru A și B – citit de la tastatură.