

Exerciții - Săptămâna 3

Formatori:

Tutor: [Militaru Mihai-Adrian](#) 

Tutor: [Dragomir Titian-Cornel](#) 

+8



Data de începere a cursului:

 25.09.2023

 [Utilizatori înscriși](#)

 [Calendar](#)

 [Note](#)

 [Cursurile mele](#) [S1-L-AC-CTIRO1-LSD](#) [Săptămâna 3: Recursivitate](#) [Exerciții - Săptămâna 3](#)

Exerciții - Săptămâna 3

! PENTRU PROBLEMELE DE MAI JOS NU AVEȚI VOIE SĂ FOLOSIȚI STRUCTURI ITERATIVE (for/while), FUNCȚIA range().

Exercițiul 1: Progresie aritmetică

Implementați în Python o funcție recursivă pentru a calcula valoarea termenului de rang $n \in \mathbb{N}$, pentru progresia aritmetică definită de relația:

$$a_n = 2 \times a_{n-1} - 3, \forall n \in \mathbb{N}$$

Se consideră valoare termenului $a_0 = 2$.

Funcția va primi ca unic parametru numărul natural n .

Exercițiul 2: Fibonacci

Implementați în Python o funcție recursivă pentru a calcula termenul de rang n din șirul lui Fibonacci:

$$fibonacci(n) = \begin{cases} fibonacci(n-1) + fibonacci(n-2) & \text{pentru } n \neq 1 \text{ sau } n \neq 0 \\ 1 & \text{pentru } n = 1 \\ 0 & \text{pentru } n = 0 \end{cases}$$

Exercițiul 3: Suma primelor N numere naturale

Implementați în Python o funcție recursivă pentru a calcula suma primelor N numere naturale.

Exercițiul 4: Cifrele unui număr

- Implementați în Python o funcție recursivă pentru a calcula produsul cifrelor unui număr dat ca parametru.
- Implementați în Python o funcție recursivă pentru a număra câte cifre are numărul dat ca parametru.
- Implementați în Python o funcție recursivă care returnează cifra maximă a numărului dat ca parametru.
- Implementați în Python o funcție recursivă care returnează numărul de cifre pare a numărului dat ca parametru.

Exercițiul 5: Exponentul unui număr natural

Implementați în Python o funcție recursivă care calculează a^n , $a \geq 1$, $n \geq 0$, $a, n \in \mathbb{N}$.

Exercițiul 6: Număr prim

Implementați în Python o funcție care returnează True dacă un număr n este prim, altfel False.

Exercițiul 7: Cel mai mare divizor comun

Implementați în Python o funcție recursivă care calculează cel mai mare divizor comun.

Hint: $cmmdc(a, b) = cmmdc(b, a \bmod b)$, pentru $b \neq 0$

Exercițiul 8: Funcția my_reverse

Implementați în Python o funcție recursivă care inversează un șir de caractere.

Hint: Puteți extrage părți dintr-un șir de caractere folosind sintaxa `nume[start:end]`. Dacă vreți să extrageți un subșir de la o anumită poziție până la finalul șirului inițial puteți folosi `nume[start:]`.

Exercițiul 9: Interval

Implementați în Python o funcție recursivă care primește două valori, `min_value`, respectiv `max_value` și tipărește toate numerele naturale care se regăsesc în interiorul intervalului ale cărui extremități sunt cele două valori.

Exercițiul 10: Apariții

- a) Implementați în Python o funcție recursivă ce verifică dacă o cifră este prezentă sau nu într-un număr.
 b) Implementați în Python o funcție recursivă care returnează numărul de apariții ale unei cifre într-un număr.

Exercițiul 11: Palindrom

Implementați în Python o funcție care determină dacă un număr este sau nu palindrom. Un număr este considerat palindrom dacă este egal cu răsturnatul său.

Exemple: 121, 34543, 1111 sunt astfel de numere

Exercițiul 12: Compunere de funcții

Implementați în Python o funcție recursivă care primește ca parametri o funcție f , o valoare reală x și un număr natural n mai mare sau egal cu 2. Funcția va returna valoarea rezultată în urma compunerii funcției f de n ori aplicată în punctul x .

Exercițiul 13: Sume remarcabile

Implementați în Python funcții recursive care calculează termenul n al următoarelor sume:

$$a) S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n+1}$$

$$b) S = e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots \text{ (serie Taylor)}$$

Observație: Încercați pentru seria Taylor să găsiți o metodă de calcul ce nu necesită recalcularea inutilă a puterii, respectiv a factorialului la fiecare pas (puteți fie să găsiți o relație recurentă, fie să vă folosiți de faptul că în Python puteți returna mai multe valori)

$$c) S = \underbrace{\sqrt{1 + \sqrt{1 + \sqrt{1 + \dots + \sqrt{1 + 1}}}}}_{\text{de } n \text{ ori}}$$

Exercițiul 14: Conversie zecimal-binar

Scrieți o funcție recursivă în Python care primește ca parametru un număr natural și întoarce un șir de caractere ce reprezintă conversia în format binar a numărului dat ca parametru.

Exemplu: Pentru $n=5$ se va returna "101".

Procedeu: Pentru a face conversia din sistemul zecimal în sistemul binar, mai întâi se împarte numărul ales la 2; restul reprezintă cifra cea mai puțin semnificativă (cea mai din dreapta) a rezultatului conversiei. Câtul se reîmparte la 2, se notează restul, și procedura se repetă cu noul cât. Operația se sfârșește când câtul devine nul.

Exercițiul 15: Triunghi

Tipăriți următorul model triunghiular cu n linii folosind funcții recursive.

Exemplu: Pentru $n = 5$

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Exercițiul 16: Resturi modulo p

În matematică știm că dacă p e un număr prim, și a nu se divide cu p , atunci șirul a, a^2, a^3, \dots va ajunge la 1, luând numerele modulo p (adică resturile la împărțirea cu p).

Scrieți o funcție care ia ca parametru un număr întreg a și un număr p (presupus prim) și returnează cea mai mică putere n pentru care $a^n \equiv 1 \pmod{p}$ (sau returnează 0 dacă a se divide cu p).

Hint: Scrieți o funcție auxiliară care mai are ca parametri și exponentul k respectiv valoarea $a^k \pmod{p}$, și care se apelează recursiv până când $a^k \equiv 1 \pmod{p}$.

Exemplu: Fie $p = 7$ și $a = 4$. Atunci $a^2 = 16 \equiv 2 \pmod{7}$, și $a^3 = a^2 * a \equiv 2 * 4 \equiv 1 \pmod{7}$. Funcția va returna 3.

◀ Laboratorul 3

Sari la...

Laborator 3 - Notebook (Subgrupele 1.1C, 1.2D) ►

✉ Contactați serviciul de asistență

Sunteți conectat în calitate de Ciobanu Daria-Andreea (Delogare)

S1-L-AC-CTIRO1-LSD

Meniul meu

Profil

Preferinte

Calendar

 ZOOM

Română (ro)

English (en)

Română (ro)

Rezumatul păstrării datelor

Politici utilizare site