**Structură Slideuri .ppt:**

1. Cerinţă prof. + Cerinţa pe care am compus-o noi
2. Modul prin care am dedus cerinţa
3. Explicaţii variabile plus funcţionalitate program
4. Rezolvarea pas cu pas a exemplului dat de prof. (n = 5)
5. Implementarea rezolvării în Python

+ Numele nostru, grupă, etc (ce mai trebuie trecut)

Slide 1:

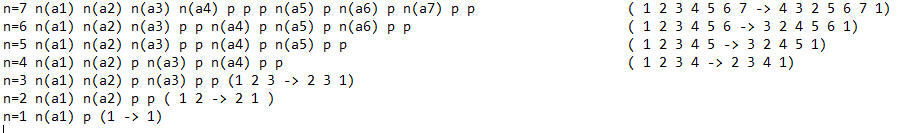
Cerinţă prof. - Aia de pe excel: Primiţi o stivă...

Cerinţa compusă de noi: Se citesc n numere naturale (a1, a2,... ,an) care sunt adăugate într-o stivă (s = ((a1, a2,... ,an)).

1. Găsiţi o metodă generală pentru orice n număr natural citit astfel încât secvenţa obţinută să respecte patternul dat pt. n = 5 (din exemplul primit).
2. Implemetaţi metoda găsită în Python.

Slide 2:

Verificăm cum ar trebui să arate secvenţa pentru n = 1, ..., 7 şi observăm următorul pattern:



(poate îl organizăm pe ăsta ceva mai frumos, da' aceasta este informaţia de bază care trebuie să apară)

Slide 3:

Explicaţia variabilelor:

n = lungime stack

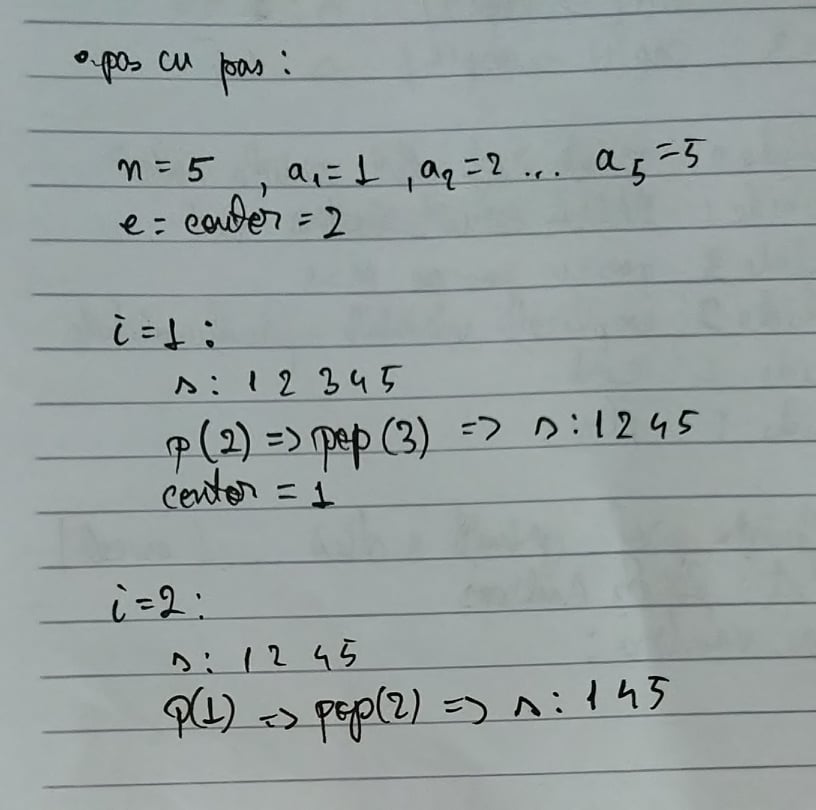
i = iteraţia curentă în parcurgerea stackului (observaţie: începem cu i = 1, nu i = 0, pentru a respecta patternul observat; dacă începeam cu i = 0, ordinea operaţiilor era "shiftată" cu o unitate)

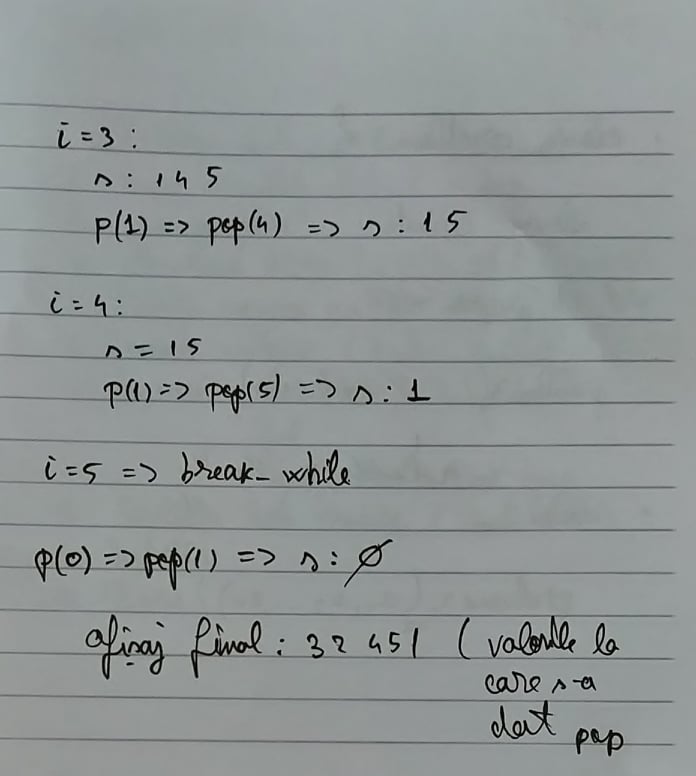
e = reţine jumătatea indicilor

contor = partiţionează stackul în două părţi (pentru a controla popurile)

//Trebuie adăugată funcţionalitatea efectivă a programului; o descriere a modului în care operăm

Slide 4:





Slide 5:

1. **def** numar(x):
2. stack.append(x);
4. **def** p(x):
5. **print**(stack.pop(x), end= " ");
7. **def** div2(a):
8. **if** (a%2==0):
9. **return** int(a/2)-1;
10. **else**:
11. **return** int(a/2);

14. **print**('Introduceti numere pana la introducerea unui string in ordine**\n**');
15. **try**:
16. stack = [];
17. **while** True:
18. numar(int(input()))
20. **except**:
21. **print**('Stiva s-a incarcat cu succes!**\n**');
23. **print**('Stiva s-a ordonat astfel: **\n**');
24. i=1;
25. n=len(stack);
26. e=div2(n);
27. contor=e;
28. **while**(i<n):
29. p(contor);
30. **if**(i<e):
31. contor-=1;
32. i+=1;
34. p(contor-1);