**PROIECT SDA: PROBLEMA 3**

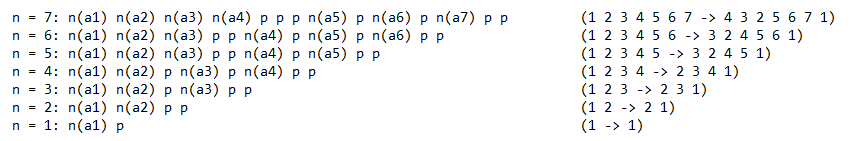
***(Condrat Mihai, Guiţa Bianca Oana, Mihalea Andreas, GRUPA 211)***

1. **FORMULAREA CERINŢEI:**

* ***CERINŢA INIŢIALĂ***: Gasiţi o succesiune de mutări a.i. introducând: **1 2 3 4 5** în stiva (în aceasta ordine) la final să se afişeze: **3 2 4 5 1**.
* ***REFORMULAREA CERINŢEI***: Se citesc ‘n’ numere naturale (a1, a2, …, an) care sunt adăugate într-o stiva (s = [an, a(n-1), …, a1]). Gasiţi o metodă generală (pentru oricare ‘n’ număr natural citit) astfel încât secvenţa obţinută să respecte patternul dat pentru n = 5 (din exemplul primit). Implemenţati metoda gasită în Python.

1. **IDEEA PROBLEMEI:**

Verificăm cum ar trebui sa arate secvenţa pentru n = 1, …, 7 si observăm urmatorul pattern:



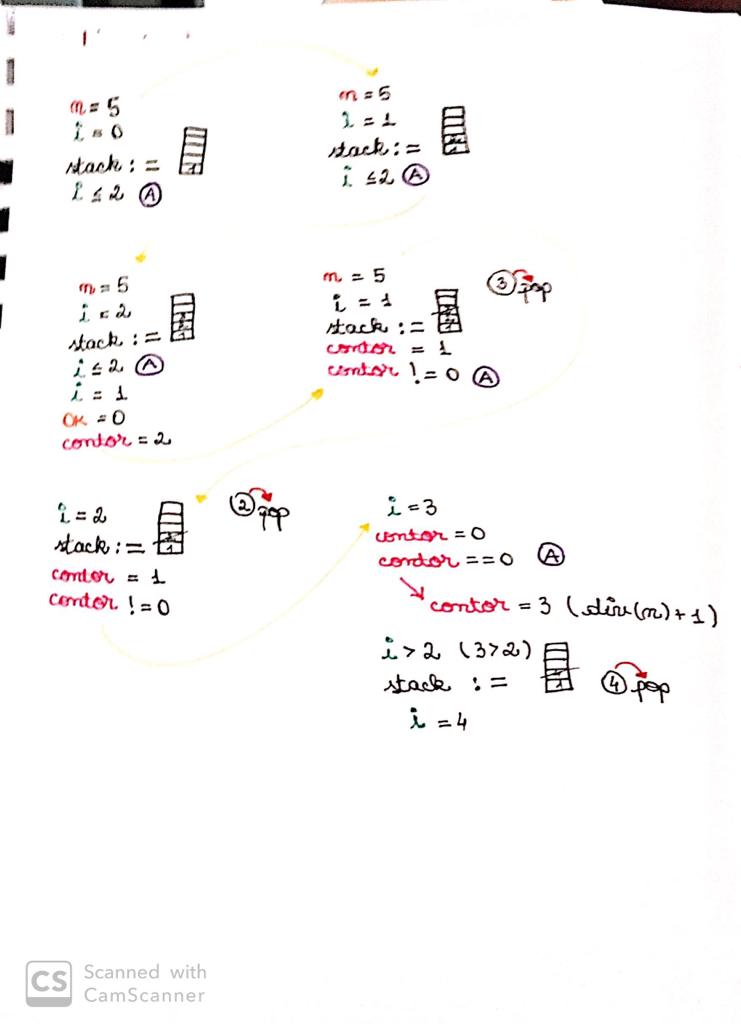
1. **STRUCTURĂ SI FUNCŢIONALITATE:**

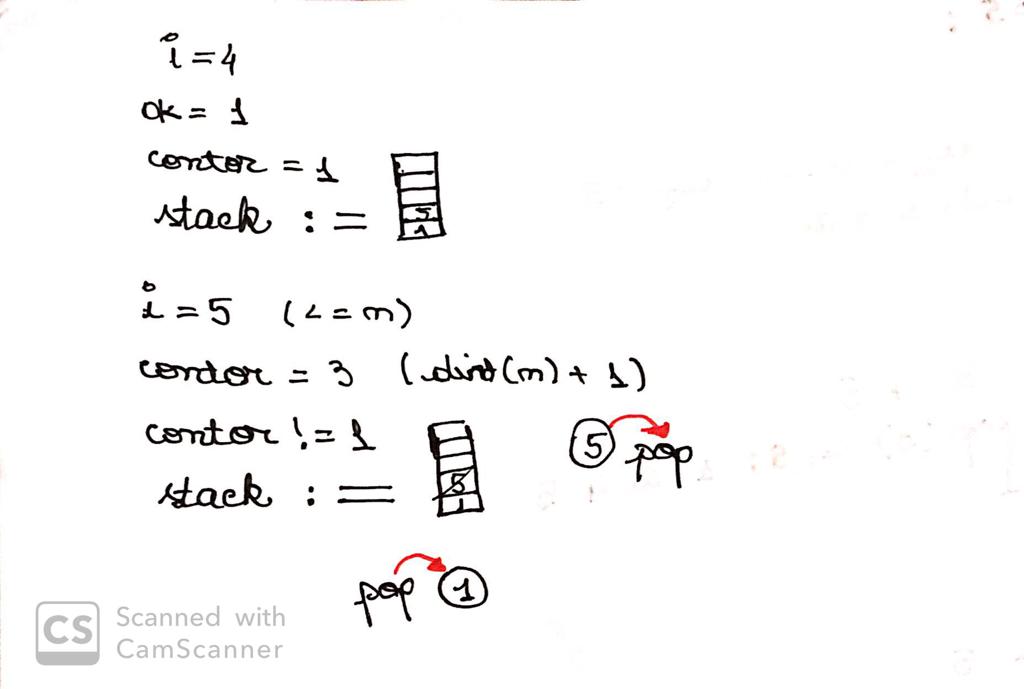
Rezolvarea propusă de noi se bazează pe partiţionarea stivei în două jumătăţi pe care operăm în moduri diferite. Pentru prima jumatate, încărcăm primele n/2 numere şi apoi apelăm un pop. Pentru cea de-a doua jumătate, încărcăm numerele rămase până la al n-lea şi, imediat după introducerea lor în stiva, le dăm pop.

Variabilele necesare pentru înţelegerea exemplului prezentat în continuare:

* n := lungime stack
* i := iteraţia curentăîn parcurgerea stackului
* e := reţine jumătatea indicilor
* contor := partiţioneaza stackul in două parţi (pentru a controla pop-urile)

1. **EXEMPLU REZOLVAT:**





1. **IMPLEMENTARE IN PYTHON:**

Codul complet poate fi găsit pe acest cont: [github.com/condratmihai](https://github.com/condratmihai/Universitate/blob/master/An1/Sem1/SDA/proiect_problema3.py).

