

Recomandari:

- creați câte un fișier de genul tema2.5.py pentru fiecare dintre punctele de mai jos. 2 reprezintă numărul laboratorului și 5 numărul temei. tema2.5.a.py, tema2.5.b.py dacă sunt variațiuni la tema.
- creați un zip cu toate temele și eventuale fișiere conexe și trimiteți-mi zip-ul pe email

1. Vedeți programul atașat la laboratorul curent (litere-mari.py) Scopul acelui program este să afișeze toate literele mari din lista definită la începutul programului. După cum o să vedeți, există diverse erori atât de sintaxă cât și de logică în acel program.

- identificați erorile și descrieți-le pe scurt
- corectați erorile astfel încât programul să funcționeze corect

2. Se citesc pe rând de la tastatură trei numere, a, b și c. Ele reprezintă lungimile laturilor unui triunghi. Să ne amintim puțină geometrie. Un triunghi poate fi:

- isoscel: dacă are două laturi de lungimi egale. Oricare două laturi dacă sunt egale, b și c sau a și b, etc..
- echilateral: dacă are toate trei laturile egale
- dreptunghic: dacă are un unghi drept, adică de 90 de grade. Cum se verifică dacă un triunghi este dreptunghic în funcție de dimensiunea laturilor? Avem relația (teorema lui Pitagora ☺) $a^2 = b^2 + c^2$. Dar poate fi și $b^2 = c^2 + a^2$ sau $c^2 = b^2 + a^2$. Deci oricare variantă ar fi adevărată din aceste trei egalități, triunghiul este dreptunghic.
- oarecare: dacă nu se încadrează la nicio variantă de mai sus

Programul nostru trebuie să afișeze una dintre variantele de mai jos:

"Triunghiul este isoscel" - dacă este isoscel

"Triunghiul este echilateral" - dacă este echilateral

"Triunghiul este dreptunghic" - dacă este dreptunghic

"Triunghiul este oarecare" - dacă este oarecare

3. Avem o listă care inițial este goală. Se citesc de la tastatură șiruri de caractere, pe rând (într-o buclă) până când la o citire se introduce X. Moment în care programul se încheie. La fiecare citire (diferită de X) șirul (string-ul) introdus se adaugă la lista inițială. La finalul programului se afișează lista rezultată.

Variatiune: stringurile noi citite se introduc nu la finalul listei ci la început. Sfat: căutăm în help sau Google cum putem adăuga un element într-o lista altundeva decât la final.

4. Citim de la tastatură un șir de caractere. Programul să ne spună dacă acel șir este un palindrom. Un palindrom este un șir de caractere pe care dacă-l întoarcem pe dos, este identic cu originalul. Exemple: 'cojoc', 'abc3443cba'. Cu alte cuvinte primul caracter este egal cu ultimul, al doilea cu penultimul, etc..

5. O variațiune a problemei anterioare, în care să nu se țină cont de case-ul caracterelor. Adică să nu conteze dacă folosim litere mari sau mici. De exemplu 'Cojoc' în mod normal nu e palindrom deoarece 'C' != 'c'. Dar în programul de la acest punct trebuie ignorat acest aspect și 'Cojoc' sau 'AbC3443aBc' să fie considerate palindromuri.

Sfat: cautați (help sau Google) cum convertiți în Python un string la *lowercase* (toate literele mici) sau *uppercase* (toate literele mari).