Pentru următoarele probleme nu se vor folosi secvențe (vectori, șiruri de caractere)

1. Sa se verifice dacă un număr dat de la tastatură este pătrat perfect.
2. Se citesc două numere naturale a și b cu cel mult două cifre. Să se afișeze toate numerele naturale pozitive de cel mult două cifre care se divid cu 5 și nu se află în intervalul [a,b] (numerele se vor afișa pe aceeași linie, ordonate crescător, apoi descrescător).
3. Se citește un șir de n >1 numere. Să se afișeze numărul de valori de tip deal (mai mari decât elementele vecine; primul și ultimul element din șir au doar un vecin)
4. Ce afișează următoare :

```
for i in range(10):
    print(i, end=" ")
print(i)
for i in range(10):
    print(i, end=" ")
    i="ab"
    print(i, end=" ")
for i in range(10):
    print(i, end=" ")
    i+=2
    print(i)
print(i)
```

5. Se dă un număr natural pozitiv n. Sa se afișeze o descompunere a lui n ca sumă de termeni distincți din șirul lui Fibonacci care nu conține ca termeni două numere Fibonacci consecutive. Există mereu o astfel de descompunere?
https://en.wikipedia.org/wiki/Zeckendorf%27s_theorem

Șiruri de caractere

5. Ce valoare au pentru șirul s= "acest exemplu" felierile: s[0:6:2], s[0:6:-2], s[0::2], s[::100], s[::-100], s[-3:-1], s[0:4], s[-1:4], s[len(s):]? Cum se poate accesa prin feliere subsecvența “exe” folosind indici pozitivi? Dar folosind indici negativi?
6. Scriere echivalentă pentru s[::-1] de forma s[i:j:-1].
7. Se citește un cuvânt s. Să se verifice dacă s este palindrom sau semipalindrom (format din două jumătăți egale)
8. Se citește un șir de caractere și un număr natural k. Să se șteargă din s caracterul de pe poziția k (pozițiile numerotate de la 0) și să se afișeze șirul nou obținut.
9. Se citește un cuvânt s. Să se afișeze șirul obținut prin ștergerea primei vocale.
10. Se citește un cuvânt s de cel mult 10 de caractere. Sa se afișeze pe câte o linie cuvintele obținute succesiv din s tăind prima și ultima literă (afișate centrat pe 10 de caractere):

```
 algoritm
  lgorit
   gori
    or
```

11. a)Se citeşte un cuvânt w, un număr natural nenul p, un număr natural n şi un şir format din n cuvinte, date fiecare pe o linie. Să se afişeze toate cuvintele care sunt p-rime cu w (adică ultimele p caractere din cuvânt coincid cu ultimele p caractere ale lui w) şi au lungime cel puţin p+2. De exemplu, pentru w = "mere", p = 2 , n=4 şi cuvintele "pere", "teste", “are” şi "programare", trebuie să fie afişate cuvintele "pere" şi "programare" (“are” rimează cu “mere”, dar are lungime mai mică decât p+2.

    b) Aceeaşi cerinţă, dar pentru cuvintele dintr-o propoziţie în care cuvintele sunt despărţite între ele prin spaţii. De exemplu, pentru w = "cere", p = 2 , n=4 şi propoziţia "Ana are mere si pere si banane de mancare", trebuie să se afişeze următorul rezultat: 2-rimele cuvântului 'cere' sunt: are mere pere mancare

12. **Sir periodic.** Se citeşte un şir de caractere *s*. Să se verifice dacă există un şir *t*, diferit de *s*, astfel încât *s* să se poată obţine prin concatenarea de un număr arbitrar de ori (*k*>1) a şirului t (adică să se verifice dacă şirul s este periodic). Dacă există mai multe astfel de şiruri t se va determina cel mai lung. Exemplu: şirul s = abbaabbaabbaabba se obţine prin concatenarea şirului t= abbaabba de două **ori.**

13. Se citeşte un text codificat după regula: în faţa fiecărui caracter este scris un număr care reprezintă numărul de apariţii consecutive ale acestui caracter în text. Scrieţi un program care citeşte textul codificat şi îl decodifică, ştiind că textul iniţial nu conţinea cifre. De exemplu textul codificat “1v3a1c1a1n1t5a2 1m13a1r4e” se va decodifica “vaaacantaaaaa maaaaaaaaaaaareeee”

    b) Scrieţi un program care citeşte un text dat pe o linie şi îl codifică după regula de la a).