Zoznam (pole/postupnosť)

je indexovaná a meniteľná postupnosť hodnôt, prvá hodnota je pod indexom 0. Počet hodnôt v poli je obmedzena len veľkosťou operačnej pamäte.

deklarácia zoznamu:

```
zoznam=[]
```

príklad:

```
p=[1,2,3,8,7,6,1,9]
```

Vytvorili sme zoznam s názvom `p` v ktorom je 8 hodnôt. Index (pozícia) prvej hodnoty je na 0 a poslednej je na 7

```
print(p[0]) - 1
print(p[7]) - 9
print(p[2]+p[6]) - 4
```

pridávanie prvkov na koniec zoznamu:

```
p.append(hodnota)
```

zistenie počtu hodnôt v poli:

```
len(p)
```

Pozor počet členov zoznamu sa nerovná indexu posledného člena ale počet hodnôt -1 = poslednému indexu.

Príklad:

Vytvorte n prvkový zoznam dvojciferných náhodných čísel - vytvoríme zoznam, kde počet prvkov (n) načítate od užívateľa a pomocou náhodných čísel random vygenerujete náhodné čísla 10 - 99 a pridáte ich do zoznamu

vytvorenie zoznamu:

```
import random
n=int(input('zadaj pocet prvkov zoznamu'))
z=[]
for i in range(n):
    z.append(random.randint(10,99))
```

vypísanie zoznamu:

```
print(z)
```

vypísanie zoznamu pomocou pozícií (indexov) pozícia sa udáva v []:

```
for i in range(n):
     print(z[i])
```

vypísanie zoznamu priamo po prvkoch:

```
for prvok in z:
     print(prvok)
```

Ukážky programov:

Pole s DEF maximum

```
import random

def vytvor(n,t):
    for i in range(t):
        n.append(random.randrange(100))
    return(n)

def maximalny(n):
    najvac=pole[0]
    poz=0
    for i in range(1,len(pole)):
        if najvac<pole[i]:
            najvac=pole[i]
            poz=i
    return(najvac,poz)

p=int(input('zadaj pocet prvkov'))
pole=[]
pole=vytvor(pole,p)

maxim=maximalny(pole)

print(pole)

print(maxim)
```

Pole, DEF, NSD

```
import random

def nsd(x,y):
    while x!=y:
        if x>y:
            x=x-y
        else:
            y=y-x
    return(x)

def napln(a,n):
    for i in range(n):
```

```
        a.append(random.randrange(100))
    return(a)

pole=[]
napln(pole,10)

print(pole)


hcc=nsd(pole[0],pole[1])

for i in range(2,len(pole)-1):
    hcc=nsd(hcc,pole[i])

print(hcc)
```

- Vytvorte n-prvkovu množinu náhodných čísel
- Zistite minimum a maximum
- Zistite aritmetický priemer postupnosti
- Zistite a pamätajte si párne čísla
- Zistite či sa zadaný prvok nachádza v postupnosti a ak nie, nech o tom informuje užívateľa vhodným spôsobom
- Zistite či sa prvok nachádza a vymeňte ho s novým prvkom načítaním od užívateľa
- Zistite počet výskytov zadaného čísla, pamätajte si kde sa nachádzajú
- Nahraďte všetky výskyty novou hodnotou
- Nájdite najmenší a najväčší prvok a zistite ich rozdiel
- Vypíšte prvky ktoré spĺňajú podmienky: väčšie, menšie ako; delitelné; prvočísla;
- Zistite NSD celej postupnosti
- Zistite všetky prvočísla postupnosti

Funkcie

- funkcia `len(postupnosť)` -> vráti počet prvkov postupnosti
- funkcia `sum(postupnosť)` -> vypočíta číselný súčet prvkov postupnosti
- funkcia `max(postupnosť)` -> vráti maximálny prvok postupnosti (t.j. jeho hodnotu)
- funkcia `min(postupnosť)` -> vráti minimálny prvok postupnosti

Metódy:

metóda `count()`

Volanie metódy: `zoznam.count(hodnota)`

vráti počet výskytov danej hodnoty v zozname. Táto metóda je **immutable** lebo nemení obsah zoznamu.

metóda `index()`

Volanie metódy: `zoznam.index(hodnota)`

vráti index prvého výskytu danej hodnoty v zozname. Táto metóda je **immutable** lebo nemení obsah zoznamu. Funkcia spadne na chybe, ak sa daná hodnota v zozname nenachádza.

metóda `append()`

Volanie metódy: `zoznam.append(hodnota)`

pridá na koniec zoznamu nový prvok - zoznam sa takto predĺži o 1. Táto metóda je **mutable** lebo mení obsah zoznamu.

metóda `pop()`

Volanie metódy: `zoznam.pop()`

odoberie z konca zoznamu posledný prvok - zoznam sa takto skráti o 1. Táto metóda je **mutable** lebo mení obsah zoznamu. Funkcia vracia hodnotu odobratého prvku. Ak bol zoznam prázdny, funkcia nič nevracia ale spadne na chybe.

metóda `pop()` s indexom

Volanie metódy: `zoznam.pop(index)`

doberie zo zoznamu príslušný prvok (daný indexom) - zoznam sa takto skráti o 1. Táto metóda je **mutable** lebo mení obsah zoznamu. Funkcia vracia hodnotu odobratého prvku. Ak bol zoznam prázdny, funkcia nič nevracia ale spadne na chybe.

metóda `insert()`

Volanie metódy: `zoznam.insert(index, hodnota)`

pridá na dané miesto zoznamu nový prvok - zoznam sa takto predĺži o 1. Táto metóda je **mutable** lebo mení obsah zoznamu. Funkcia nič nevracia, preto nemá zmysel priraďovať jej volanie do nejakej premennej (teda vracia hodnotu `None`).

metóda `remove()`

Volanie metódy: `zoznam.remove(hodnota)`

odoberie zo zoznamu prvý výskyt prvku s danou hodnotou - zoznam sa takto skráti o 1. Táto metóda je **mutable** lebo mení obsah zoznamu. Funkcia nič nevracia (teda vracia `None`). Ak sa daná hodnota v zozname nenachádza, funkcia spadne na chybe.

metóda `sort()`

Volanie metódy: `zoznam.sort()`

zmení poradie prvkov zoznamu tak, aby boli **usporiadané vzostupne** - zoznam takto nemení svoju dĺžku. Táto metóda je **mutable** lebo mení obsah zoznamu. Funkcia nič

nevracia (teda vracia `None`). Ak sa prvky v zozname nedajú navzájom porovnávať (napríklad sú tam čísla aj reťazce), funkcia spadne na chybe.

metóda `split()`

Keďže je to **reťazcová metóda**, má tvar: `reťazec.split()`

Metóda rozbije daný reťazec na samostatné reťazce a uloží ich do zoznamu (teda vracia **zoznam reťazcov**). Predpokladáme, že tieto podreťazce sú navzájom oddelené „medzerovými“ znakmi (medzera, znak konca riadku, tabulátor). V helpe (napríklad `help(''.split)`) sa môžete dozvedieť ďalšie možnosti tejto funkcie.

metóda `join()`

Opäť je to **reťazcová metóda**. Má tvar: `oddeľovač.join(zoznam_reťazcov)`

Metóda zlepí všetky reťazce z daného **zoznamu reťazcov** do jedného, pričom ich navzájom oddelí uvedeným `oddeľovačom`, t. j. nejakým zadaným reťazcom. Ako zoznam môžeme uviesť ľubovoľnú postupnosť (iterovateľný objekt) reťazcov.