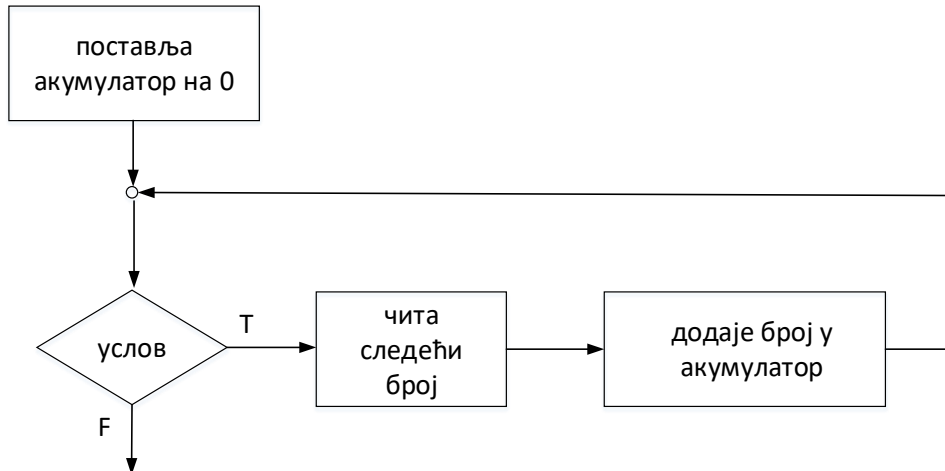


## Акумулатор у петљи

### Рачунање тренутне суме

Тренутна сума (running total) је сума бројева који се сабирају при свакој итерацији у петљи. Промењива у коју се смешта тренутна сума се назива акумулатор.

У многим петљама је потребно извести математичке операције са низом бројева.



Тада се петљом чита један по један број у низу а једна промењива (акумулатор) чува суму прочитаних бројева.

Када се петља заврши, акумулатор ће садржати суму бројева које је петља прочитала.

Иницијализација акумулатора се изводи пре уласка у петљу и обично се акумулатор поставља на 0.

То је битно пошто се врши операција сабирања а неутралан број за сабирање је 0.

### 079 Сабирање бројева помоћу петље и акумулатора

```
max = 5
```

```
#inicijalizacija akumulatora
```

```
suma = 0
```

```
print('Ovaj program racuna sumu unetih brojeva.')
```

```
for brojac in range(max):
```

```
    broj = int(input('Unesi broj: '))
```

```
    suma = suma + broj
```

```
print('Suma je', suma)
```

```
Ovaj program racuna sumu unetih brojeva.
```

```
Unesi broj: 5
```

```
Unesi broj: -3
```

```
Unesi broj: 2
```

```
Unesi broj: 0
```

```
Unesi broj: 0
```

```
Suma je 4
```

У примеру, промењива brojac је циљна промењива а suma је акумулатор.

Појачани оператори доделе

Када се промењива која је на левој страни оператора доделе (=) појави и на десној страни ( $x = x + 1$ ) то значи да се овакав израз може другачије представити са појачаним оператором доделе:  $x += 1$ .

Овакав израз се чита: стара вредност промењиве се инкрементира (повећава за 1) и постаје нова вредност исте промењиве.

Списак свих појачаних оператора доделе:  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ ,  $%=$ .

Стражари

Ако се током дизајнирања програма деси да није познат број вредности које ће се користити у низу или се зна да ће број вредности у низу бити другачији сваки пут када се програм стартује, постоји две могућности:

1. Питати корисника на крају сваке итерације да ли постоји још вредности за обраду
2. Питати корисника на почетку програма колико елемената има у низу

Обе могућности су нестабилне пошто зависе од корисника и зато је боља идеја користити стражара (sentinel).

Стражар је посебна вредност којом се означава крај низа елемената.

Када програм прочита вредност стражара, зна да је достигао крај низа, па се петља завршава.

**080** Коришћење стражара

```
zbir = 0
strazar = int(input("Vrednost za kontrolu petlje: "))
while strazar != 0:
    zbir += strazar
    strazar = int(input("Vrednost za kontrolu petlje: "))

print('Zbir je', zbir)
Vrednost za kontrolu petlje: 4
Vrednost za kontrolu petlje: 6
Vrednost za kontrolu petlje: 2
Vrednost za kontrolu petlje: 0
Zbir je 12
```

**081** Пребројавање непарних бројева у листи

```
neparni = 0
lista = [2, 5, 66, 33, 77]
for x in lista:
    if x % 2 != 0:
        neparni += 1

print("Neparni brojeva u listi ima", neparni)
```

Питања и задаци за самосталан рад

## Задаци

0185 Написати програм који приказује колико је парних, непарних, нула а колико бројева већих од 10 у листи [-23, 55, 0, 55, 32, -102, 0, 1, 8, 0, 0]

0186 За дати опсег бројева колико је бројева са цифром јединице 5?

0187 Корисник уноси речи независне једне од других, колико је пута унео реч "skola"?

0188 Компанија уноси податке о запосленима тако што се прво унесе име запосленог а затим бројчани податак о том запосленом. Написати програм који броји колико пута је запослени са именом Zoran добио бројчани податак.

0189 Деца посматрају пролазак камиона са натписима (Sokovi, Peciva, Slatkisi) и броје камионе са таквим натписима. Написати апликацију која ће им помоћи да лакше преброје камионе са датим натписима и камионе које немају те натписе. На основу узорка од првих 20 камиона, апликација даје процентуално предвиђање колико камиона носи дати натпис.