

1. Stapanje realnih brojeva

https://petlja.org/biblioteka/r/problemi/Zbirka/prosek_skijasa

Na takmičenju u skijaškim skokovima pored udaljenosti koju skakač preskoči ocenjuje se i stil skakača. Pet sudija ocenjuju stil skakača ocenama od 0 do 20. Zatim se od svih dobijenih ocena izbrišu jedna najmanja i jedna najveća. Konačna ocena stila skakača računa se kao prosek preostalih ocena. Napiši program kojim se na osnovu ocena 5 sudija određuje konačna ocena stila skakača. U svakoj od pet linija standardnog ulaza nalazi se po jedna ocena stila skakača (ceo broj između 0 i 20) koju je skijaš dobio od pet sudija. Konačna ocena stila skakača prikazana na dve decimale.

Primer

Ulaz

3

5

0

1

5

Izlaz

3.00

```
# Ucitavanje ocena
```

```
ocena1 = int(input())
```

```
ocena2 = int(input())
```

```
ocena3 = int(input())
```

```
ocena4 = int(input())
```

```
ocena5 = int(input())
```

```
# Izracunavanje trazenog proseka
```

```
zbirSvihOcena = ocena1 + ocena2 + ocena3 + ocena4 + ocena5
```

```
najmanjaOcena = min(ocena1, ocena2, ocena3, ocena4, ocena5)
```

```
najvecaOcena = max(ocena1, ocena2, ocena3, ocena4, ocena5)
```

```
konacnaOcena = (zbirSvihOcena - najmanjaOcena - najvecaOcena) / 3.0
```

```
# Prikaz rezultata
```

```
print("%.2f" % konacnaOcena)
```

2. Ucitavanje i ispis float podataka

https://petlja.org/biblioteka/r/problemi/Zbirka/najblizi_proseku

Napisati program koji za četiri data broja određuje koji je od ta četiri broja najbliži aritmetičkoj sredini datih brojeva (ako su dva broja jednako blizu ispisati prvi). U četiri linije standardnog ulaza nalaze se četiri realna broja. U jednoj liniji standardnog izlaza prikazati jedan od unetih realnih brojeva, na dve decimale, najbliži njihovoj aritmetičkoj sredini.

Primer

Ulaz

24.3

20.2

23.5

22.7

Izlaz

22.70

```
# data cetiri broja
```

```

broj = [float(input()) for i in range(4)]
# njihova aritmeticka sredina
aS = sum(broj) / 4.0
# najmanje rastojanje od aritmetickie sredine i broj za koji je ono
# postignuto
razlika = [abs(broj[i] - aS) for i in range(4)]
# pozicija najblizeg elementa je pozicija najmanje razlike
pozicijaNajblizeg = razlika.index(min(razlika))
najblizi = broj[pozicijaNajblizeg]
print("%.2f" % najblizi)

```

3. učitavanje i sumiranje niza brojeva

Napisati program kojim se određuje zbir n datih celih brojeva. U prvoj liniji standardnog ulaza nalazi se prirodan broj n . U svakoj od narednih n linija nalazi se po jedan ceo broj. U prvoj liniji standardnog izlaza prikazati zbir unetih n celih brojeva.

Primer

Ulaz

5

10

-3

2

4

-1

Izlaz

12

https://petlja.org/biblioteka/r/problemi/Zbirka/zbir_n_brojeva

```

n = int(input())
a = (int(input()) for i in range(n))
zbir = sum(a)
print(zbir)

```

C++ rešenje

```
#include <iostream>
```

```
#include <vector>
```

```
#include <numeric>
```

```
using namespace std;
```

```

int main() {
    int n;
    cin >> n;
    vector<int> a(n);
    for(int i = 0; i < n; i++)
        cin >> a[i];
    int zbir = accumulate(begin(a), end(a), 0);
    cout << zbir << endl;
    return 0;
}

```

4. Učitavanje do 0

https://petlja.org/biblioteka/r/problemi/Zbirka/citanje_do_nule

Unose se celi brojevi dok se ne unese nula. Napisati program kojim se prikazuje koliko je uneto brojeva, ne računajući nulu.

Ulaz

Svaka linija standardnog ulaza, izuzev poslednje, sadrži ceo broj različit od nule. Poslednja linija sadrži nulu.

Izlaz

Na standardnom izlazu u prvoj liniji prikazati koliko je učitano brojeva, ne računajući nulu.

Primer 1

Ulaz

5

12

-5

675

-123

0

Izlaz

5

Један програм, више решења

Решење 1	Решење 2	Решење 3 - Ц++
<pre>ukupno = 0 broj = int(input()) while broj != 0: ukupno = ukupno + 1 broj = int(input()) print(ukupno)</pre>	<pre>ukupno = 0 while True: broj = int(input()) if broj == 0: break ukupno = ukupno + 1 print(ukupno)</pre>	<pre>#include <iostream> using namespace std; int main() { int broj; int ukupno = 0; while(true) { cin >> broj; if (broj == 0) break; ukupno++; } cout << ukupno << endl; return 0; }</pre>

5. Učitavanje do kraja ulaza

Sa ulaza se unose celi brojevi sve dok se ne dođe do kraja ulaza. Napisati program kojim se prikazuje koliko je uneto brojeva.

https://petlja.org/biblioteka/r/problemi/Zbirka/citanje_do_kraja_ulaza

Решење 1	Решење 2 - Ц++	Решење 3 – Ц#
<pre>import sys broj = 0 for linija in sys.stdin: broj = broj + 1 print(broj)</pre>	<pre>#include <iostream> using namespace std; int main() {</pre>	<pre>using System; class Program {</pre>

	<pre>int x; int br = 0; while (cin >> x) br++; cout << br << endl; return 0; }</pre>	<pre>static void Main(string[] args) { int br = 0; while (Console.ReadLine() != null) br++; Console.WriteLine(br); }</pre>
--	--	--

Sablonska klasa u C++ za učitavanje do kraja ulaza

```
#include <iostream>
```

```
#include <iterator>
```

```
using namespace std;
```

```
int main() {
    int br = 0;
    istream_iterator<int> it(cin), kraj;
    while (it != kraj) {
        br++;
        it++;
    }
    cout << br << endl;
    return 0;
}
```

Korišćenje iteratora je naročito korisno u kombinaciji sa bibliotečkom STL funkcijom distance

```
#include <iostream>
```

```
#include <iterator>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
int main() {
    cout << distance(istream_iterator<int>(cin), istream_iterator<int>()) <<
endl;
    return 0;
}
```

Python modul sys

sys — system-specific parameters and functions

Ovaj modul omogućava pristup promenljivama koje koristi ili održava interpreter. Ovaj Ovaj modul omogućava pristup funkcijama koje interaguju sa interpreterom. Primer: rad sa argumentima komandne linije,....

6. Iscitavanje do kraja ulaza I ispis realnih brojeva

https://petlja.org/biblioteka/r/problemi/Zbirka/prosek_svih_brojeva_do_kraja_ulaza

Решење 1	Решење 2	Решење 3 – Ц++
<pre>import sys zbir = 0 broj = 0 for linija in sys.stdin: x = int(linija) zbir = zbir + x broj = broj + 1 prosek = zbir / broj print("%.5f" % prosek)</pre>	<pre>import sys a = [] for linija in sys.stdin: a.append(int(linija)) prosek = sum(a) / len(a) print("%.5f" % prosek)</pre>	<pre>#include <iostream> #include <iomanip> using namespace std; int main() { int zbir = 0, broj = 0; int x; while (cin >> x) { zbir += x; broj++; } double prosek = (double)zbir/(double)broj; cout << fixed << showpoint << setprecision(5) << prosek << endl; return 0; }</pre>

7. Zadatak za ocenu za samostalni rad

https://petlja.org/biblioteka/r/problemi/Zbirka/citanje_do_minus_1_ili_do_ntog

Enumeracija – zadatak za vezbu

- <https://petlja.org/biblioteka/r/problemi/Zbirka/berza>

Обновимо петље

- https://petlja.org/biblioteka/r/problemi/Zbirka/brojevi_od_a_do_b

Један задатак, три решења

Решење 1	Решење 2	Решење 3
<pre>a = int(input()) b = int(input()) for i in range(a, b+1): print(i)</pre>	<pre>a = int(input()) b = int(input()) i = a while i <= b: print(i) i = i + 1</pre>	<pre>a = int(input()) b = int(input()) r = range(a, b + 1) for x in r: print(x)</pre>

- https://petlja.org/biblioteka/r/problemi/Zbirka/brojevi_trocifreni_parni

```
# ucitavamo interval [a, b]
a = int(input())
b = int(input())
# odredjujemo presek sa intervalom [100, 999]
Od = max(a, 100)
```

```

Do = min(b, 999)
# vrsimo iteraciju i ispisujemo sve trazene brojeve
for i in range(Od, Do+1):
    if i % 2 == 0:
        print(i)

```

3.

Lambda u Python-u

https://petlja.org/biblioteka/r/problemi/Zbirka/magacin_sokova

- Tvorac Python-a se opirao uvodjenju lambda operatora odnosno lambda funkcija (ako vec postoji concept list comprehension): Having both list comprehension and "Filter, map, reduce and lambda" is transgressing the Python motto "There should be one obvious way to solve a problem".

Решење 1	Решење 2
<pre> # broj polica i broj flasa u gajbici n = int(input()) k = int(input()) ukupanBrojGajbica = 0 for i in range(n): # broj flasa na trenutnoj polici f = int(input()) # ukupan broj gajbica se uvecava za broj gajbica # potreban da se # smeste flase na trenutnoj polici ukupanBrojGajbica = ukupanBrojGajbica + f // k if (f % k != 0): ukupanBrojGajbica = ukupanBrojGajbica + 1 # ispisujemo ukupan broj gajbica print(ukupanBrojGajbica) </pre>	<pre> # broj polica i broj flasa u gajbici n = int(input()) k = int(input()) # ucitavamo broj flasa na svakoj polici brojFlasa = (int(input()) for i in range(n)) # izracunavamo broj gajbica potrebnih za flase na svakoj # polici brojGajbica = map(lambda f: (f + k - 1) // k, brojFlasa) # izracunavamo i ispisujemo ukupan broj gajbica ukupanBrojGajbica = sum(brojGajbica) print(ukupanBrojGajbica) </pre>

Mapiranje u drugim programskim jezicima

U jeziku C++ možemo učitati sve brojeve flaša na policama u niz ili vektor, preslikati ga u novi niz tj. vektor koji sadrži broj gajbica za svaku policu pomoću funkcije `transform`, a zatim sve elemente sabrati pomoću funkcije `accumulate` koju smo opisali u ovom materijalu u zadatku [Zbir n brojeva](#).

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <numeric>

using namespace std;

int main() {
    // broj polica i broj flasa u gajbici
    int n, k;
    cin >> n >> k;

    // ucitavamo broj flasa na svakoj polici
    vector<int> brojFlasa(n);
    for (int i = 0; i < n; i++)

```

```

    cin >> brojFlasa[i];

    // izracunavamo broj gajbica potrebnih za flase na svakoj polici
    vector<int> brojGajbica(n);
    transform(begin(brojFlasa), end(brojFlasa), begin(brojGajbica),
              [k](int f) { return (f + k - 1) / k; });

    // izracunavamo ukupan broj gajbica
    int ukupanBrojGajbica =
        accumulate(begin(brojGajbica), end(brojGajbica), 0);

    // ispisujemo ukupan broj gajbica
    cout << ukupanBrojGajbica << endl;

    return 0;
}

```

U jeziku C# možemo učitati sve brojeve flaša na policama u niz, preslikati ga metodom `Select`, a zatim sabrati rezultat metodom `Sum`. Pošto se ova kombinacija često javlja (preslikavanje, pa sabiranje) metoda `Sum` može da primi funkciju kojom se serija preslikava pre sabiranja, čime dobijamo veoma jednostavno rešenje u jednom redu.

```
int brojGajbica = brojFlasa.Sum(f => (f + k - 1) / k);
```

```

using System;
using System.Linq;

class Program
{
    static void Main(string[] args)
    {
        // broj polica i broj flasa u gajbici
        int n = int.Parse(Console.ReadLine());
        int k = int.Parse(Console.ReadLine());

        // ucitavamo broj flasa na svakoj polici
        int[] brojFlasa = new int[n];
        for (int i = 0; i < n; i++)
            brojFlasa[i] = int.Parse(Console.ReadLine());

        // izracunavamo broj gajbica potrebnih za flase na svakoj polici
        var brojGajbica = brojFlasa.Select(f => (f + k - 1) / k);

        // izracunavamo ukupan broj gajbica
        int ukupanBrojGajbica = brojGajbica.Sum();

        // ispisujemo ukupan broj gajbica
        Console.WriteLine(ukupanBrojGajbica);
    }
}

```