

Uzdevums par burkāniem

Sarakstu
veidošana

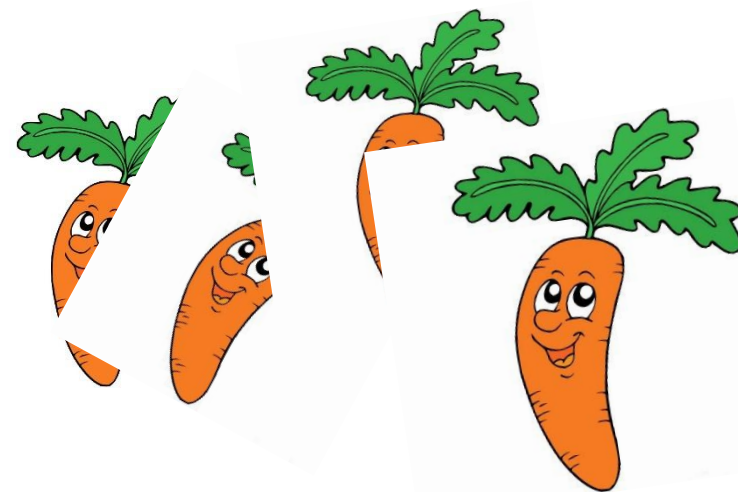
20.09. Uzdevums

Veikals «Jautrā vāga» strādā dažas nedēļas (sākot no pirmdienas). Katru dienu veikalā piegādā svaigus burkānus un nākošās nedēļas sākuma veic iepriekšējās nedēļas kopsavilkumu.

Ievadi katras dienas piegādātu burkānu svaru.

Analizē «Jautrās vāgas» darbu:

1. Cik nedēļas un dienas strādāja veikals.
2. Kāds ir kopējais piegādāto burkānu svars.
3. Aprēķini mazāko un lielāko burkānu piegādes svaru.
4. Aprēķini dienas vidējo piegādes svaru.
5. Cik dienas piegādāja burkānus vairāk pa 5kg.
6. Cik dienas nepiegādāja burkānus.
7. Sakārto burkānu svarus augošā secībā.
8. Cik dienas burkānus piegādāja vairāk nekā dienas vidējais svars.



7.Sakārto burkanu *kg* augošā secībā: **varianti / kļūdas**

```
ArrayList<Float> piegade = new ArrayList<>();
Scanner sc = new Scanner(System.in);
System.out.println("Ievadi katru dienu piegādāto burkānu svarus:");
while(sc.hasNextFloat()) {
    piegade.add(sc.nextFloat());
}
sc.close();
System.out.println(piegade);
```

Collections.sort(piegade);

```
piegade.sort((Float m1, Float m2)->{
    return (int) (m1 - m2);
});
```

```
piegade.sort((Float m1, Float m2)->{
    return m1.compareTo(m2);
});
```

```
piegade.sort((Integer m1, Integer m2)->{
    return m1.compareTo(m2);
});
```

```
piegade.sort((String m1, String m2)->{
    return m1.compareTo(m2);
});
```

26.09. Turpinājums

Veikals «Jautrā vāga» strādā dažas nedēļas (sākot no pirmdienas). Katru dienu veikalā piegādā svaigus burkānus un nākošās nedēļas sākuma veic iepriekšējās nedēļas kopsavilkumu.

9. Sakārtot burkānu *kg* dilstošā secībā.

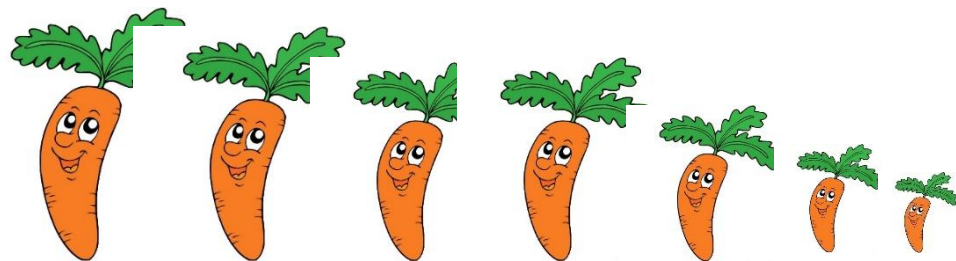
10.* Cik burkānu *kg* piegādāja veikalā katru nedēļu ?

10.1. Katras nedēļas burkānu *summaro kg* sakārtot dilstošā secībā.

10.2. Nedēļas burkānu maksimālā piegāde.

10.3. Cik burkānu vidēji piegādāja katrā nedēļā ?

11. Sakārtot burkānu *kg* tā, lai saraksta sākumā būtu $kg > 10$, beigās – pārējie *kg*.



10. Cik burkānu *kg* piegādāja veikalā katru nedēļu ?

```
ArrayList <ArrayList<Float>> nedelas = new ArrayList<>();  
  
int nedSkaitis = (int) (Math.ceil(piegade.size()/7f));  
  
for(int i=0; i<nedSkaitis; i++) {  
    nedelas.add(new ArrayList<>());  
}  
  
//System.out.println(nedelas);
```

Izveidots jauns sarakstu saraksts *nedelas* (tukšs)

```
[[], [], []]
```

```
System.out.println(nedelas);  
  
for(int i=0; i<piegade.size(); i++) {  
    nedelas.get(i/7).add(piegade.get(i));  
}  
  
System.out.println(nedelas);
```

Sarakstu saraksts *nedelas* aizpildam ar saraksta *piegāde* elementiem pa 7

```
[[0.0, 3.0, 4.0, 4.0, 5.0, 5.0, 5.0], [6.0, 6.0, 7.0, 7.0, 7.0, 8.0, 8.0], [9.0]]
```

nedelas.get(0)

nedelas.get(1)

nedelas.get(2)

```
ArrayList<Float> nedSvars = new ArrayList<>();  
for(ArrayList<Float> vienaNed: nedelas ) {  
    float summa = 0;  
    for(Float svars: vienaNed) {  
        summa +=svars;  
    }  
    nedSvars.add(summa);  
}  
  
System.out.println(nedSvars);
```

summa

```
[26.0, 49.0, 9.0]
```

10.1. Katras nedēļas burkānu *summaro kg* sakārtot dilstošā secībā