**Упражнениe: Повторения с цикли – For-цикъл**

Задачи за упражнение в клас и за домашно към курса ["Основи на програмирането" @ СофтУни](https://softuni.bg/courses/programming-basics).

Тествайте решенията си в Judge системата: <https://judge.softuni.bg/Contests/2394>

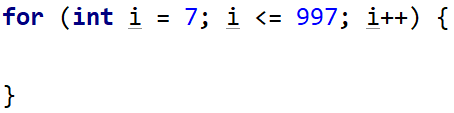
## Числа до 1000, завършващи на 7

Напишете програма, която отпечатва числата в диапазона **от 1 до 1000**, които **завършват на 7**.

|  |  |
| --- | --- |
| **вход** | **изход** |
| (няма) | 7  17  27  …  997 |

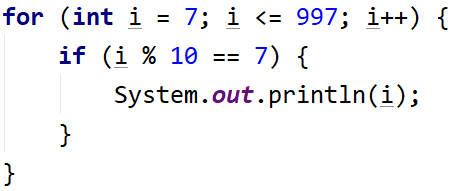
### Насоки

1. Направете **for цикъл**, за да изпълните **многократно принтирането на число**, ако то **завършва на 7**. Началната стъпка, от която може **да тръгнете е 7**, **а крайната 997**, т.е. **първото число от поредицата ви е 7, а последното 997**.



1. За да принтирате числата, **завършващи на 7**, трябва да направите **проверка, дали текущото число завършва на 7**. За целта използвайте **модулно деление на 10** и принтирайте числото, ако **резултатът от това деление е 7**.

**\*Забележка: С модулно деление на 10 може да намерите последната цифра, на което и да е цяло число, независимо каква е дължината му.**



## Елемент, равен на сумата на останалите

Да се напише програма, която чете n-на брой цели числа, въведени от потребителя,и проверява дали сред тях съществува число, което е равно на сумата на всички останали.

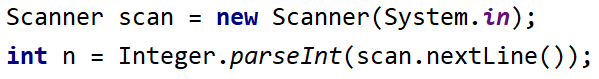
* Ако има такъв елемент печата "Yes" и на нов ред "Sum = " + **неговата стойност**
* Ако няма такъв елемент печата "No" и на нов ред "Diff = " + **разликата между най-големия елемент и сумата на останалите** (по абсолютна стойност)

### Примерен вход и изход

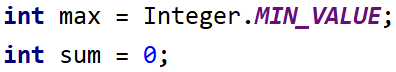
|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **коментари** |
| 7  3  4  1  1  2  **12**  1 | Yes  Sum = 12 | 3 + 4 + 1 + 2 + 1 + 1 = 12 |
| 4  **6**  1  2  3 | Yes  Sum = 6 | 1 + 2 + 3 = 6 |
| 3  1  1  **10** | No  Diff = 8 | |10 - (1 + 1)| = 8 |
| 3  5  5  1 | No  Diff = 1 | |5 - (5 + 1)| = 1 |
| 3  1  1  1 | No  Diff = 1 |  |

### Насоки

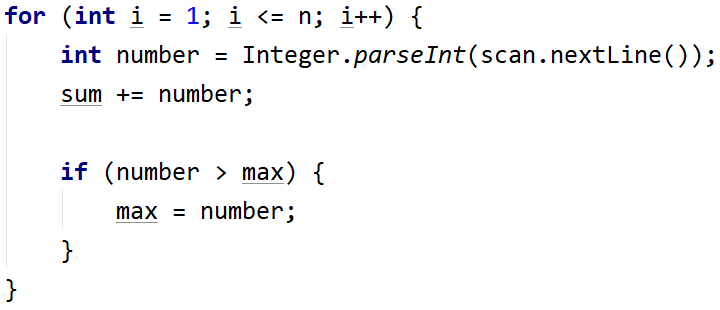
1. Прочетете входните данни **(n – броя на числата)**:



1. Създайте си **две нови променливи**. В едната **изчислявайте максималното число**, в другата **сумата на всички числа**. Задайте **начални стойности на двете променливи**. На променливата за **максимума задайте стойност Integer.MIN\_VALUE**, а на тази за **сумата задайте стойност 0**.



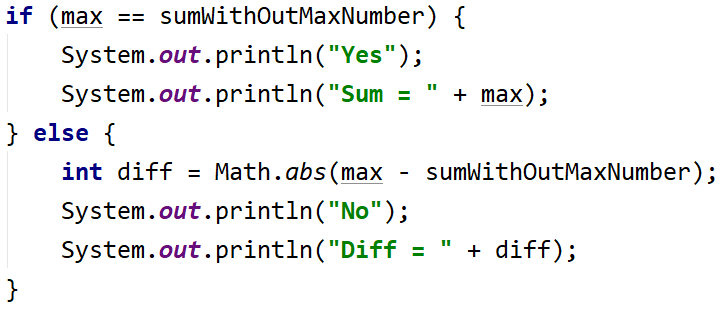
1. Направете **for цикъл, за да прочетете n на брой числа**. При всяко прочитане на число **го добавяйте към сумата.** Направете проверка **дали въведеното число е по-голямо от моментния** **максимум**. Ако е **по-голямо, максимумът става равен на това число**.



1. След като вече имате **сумата на всички числа и максималното** **число** сред тях, намерете **колко е сборът на числата без максималното** **число**. Създайте си **нова променлива**, в която да изчислите **сумата без максималното число**.



1. Направете проверка, **ако сумата без максималното число е равна на максималното число, принтирайте на два реда изхода (Yes…) и максималното число**, в противен случай **принтирайте на два реда изхода (No…) и абсолютната стойност от разликата между максималното число и сумата на всички числа без максималното** (използвайте **Math.abs**, за да намерите **абсолютната стойност**).



## Четна / нечетна сума

Да се напише програма, която чете n-на брой цели числа, подадени от потребителя и проверява дали **сумата от числата на четни позиции** е равна на **сумата на числата на нечетни позиции**.

* Ако сумите са равни да се отпечатат два реда: "Yes" и на нов ред "Sum = " + **сумата**;
* Ако сумите не са равни да се отпечат два реда: "No" и на нов ред "Diff = " + **разликата**.

Разликата се изчислява по **абсолютна стойност**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |
| 4  10  50  60  20 | Yes  Sum = 70 | 10+60 = 50+20 = 70 | 4  3  5  1  -2 | No  Diff = 1 | 3+1 ≠ 5-2  Diff = |4-3| = 1 | 3  5  8  1 | No  Diff = 2 | 5+1 ≠ 8  Diff = |6-8| = 2 |

# Примерни изпитни задачи

## 4.Хистограма

Дадени са n **цели числа** в интервала [**1**…**1000**]. От тях някакъв процент p1 са под 200, друг процент p2 са от 200 до 399, друг процент p3 са от 400 до 599, друг процент p4 са от 600 до 799 и останалите p5 процента са от 800 нагоре. Да се напише програма, която изчислява и отпечатва процентите p1, p2, p3, p4 и p5.

**Пример**: имаме n = **20** числа: 53, 7, 56, 180, 450, 920, 12, 7, 150, 250, 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. Получаваме следното разпределение и визуализация:

|  |  |  |  |
| --- | --- | --- | --- |
| **Диапазон** | **Числа в диапазона** | **Брой числа** | **Процент** |
| < 200 | 53, 7, 56, 180, 12, 7, 150, 2, 199, 46, 128, 65 | 12 | p1 = 12 / 20 \* 100 = **60.00**% |
| 200 … 399 | 250, 200 | 2 | p2 = 2 / 20 \* 100 = **10.00**% |
| 400 … 599 | 450 | 1 | p3 = 1 / 20 \* 100 = **5.00**% |
| 600 … 799 | 680, 600, 799 | 3 | p4 = 3 / 20 \* 100 = **15.00**% |
| ≥ 800 | 920, 800 | 2 | p5 = 2 / 20 \* 100 = **10.00**% |

### Вход

На първия ред от входа стои цялото число n (1 ≤ n ≤ 1000) – брой числа. На следващите n **реда** стои **по едно** **цяло число** в интервала [**1**…**1000**] – числата върху които да бъде изчислена хистограмата.

### Изход

Да се отпечата на конзолата **хистограмата** – **5 реда**, всеки от които съдържа число между 0% и 100%, с точност две цифри след десетичната точка, например 25.00%, 66.67%, 57.14%.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| **3**  1  2  999 | 66.67%  0.00%  0.00%  0.00%  33.33% | **4**  53  7  56  999 | 75.00%  0.00%  0.00%  0.00%  25.00% | **7**  800  801  250  199  399  599  799 | 14.29%  28.57%  14.29%  14.29%  28.57% | **9**  367  99  200  799  999  333  555  111  9 | 33.33%  33.33%  11.11%  11.11%  11.11% | **14**  53  7  56  180  450  920  12  7  150  250  680  2  600  200 | 57.14%  14.29%  7.14%  14.29%  7.14% |

## Заплата

**Шеф на компания забелязва че все повече служители прекарват време в сайтове, които ги разсейват.**

**За да предотврати това, той въвежда изненадващи проверки на отворените табове на браузъра на служителите си.**

**Според отворения сайт в таба се налагат следните глоби:**

* **"Facebook" -> 150 лв.**
* **"Instagram" -> 100 лв.**
* **"Reddit" -> 50 лв.**

**От конзолата се четат два реда:**

* **Брой отворени табове в браузъра** **n -** **цяло число в интервала [1...10]**
* **Заплата - число в интервала [500...1500]**

**След това n – на брой пъти се чете име на уебсайт – текст**

### Изход

* **Ако по време на проверката заплатата стане по-малка или равна на 0 лева, на конзолата се изписва   
  "You have lost your salary." и програмата приключва.**
* **В противен случай след проверката на конзолата се изписва остатъкът от заплатата (да се изпише като цяло число).**

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| **10**  **750**  Facebook  Dev.bg  Instagram  Facebook  Reddit  Facebook  Facebook | You have lost your salary. | Има **10** отворени таба в браузъра.  Първоначалната заплата е: **750**  За първия таб -> Facebook -> глоба **150** лв. (заплата: **750** – **150** = **600**)  За втория таб -> Dev.bg -> няма глоба  За третия таб -> Instagram -> глоба **100** лв. (заплата: **600** – **100** = **500**)  За четвъртия таб -> Facebook -> глоба **150** лв. (заплата: **500** – **150** = **350**)  За петия таб -> Reddit -> глоба **50** лв. (заплата: **350** – **50** = **300**)  За шестия таб -> Facebook -> глоба **150** лв. (заплата: **300** – **150** = **150**)  За седмия таб -> Facebook -> глоба **150** лв. (заплата: **150** – **150** = **0**)  Заплатата е равна на **0** => изписва съответният изход и програмата приключва | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 3  500  Github.com  Stackoverflow.com  softuni.bg | 500 | 3  500  Facebook  Stackoverflow.com  softuni.bg | 350 |