# Упражнение: 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/Compete/Index/1164](https://judge.softuni.bg/Contests/Compete/Index/1164#0)

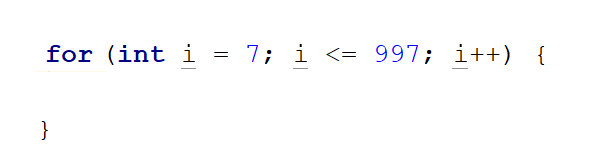
## Числа до 1000, завършващи на 7

Напишете програма, която отпечатва числата в диапазона **[1…1000]**, които **завършват на 7**.

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

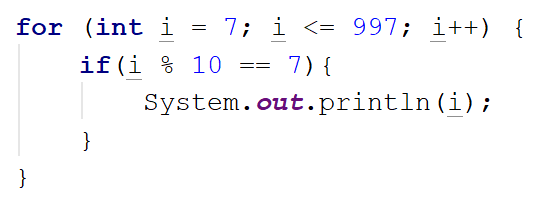
### Насоки

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 наприте **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 намиарте послендата цифра, на което и да е цяло число, независимо каква е дължината му.**



## Елемент, равен на сумата на останалите

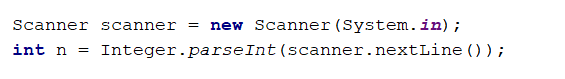
Да се напише програма, която чете n-на брой цели числа, въведени от потребителя, и проверява дали сред тях съществува число, което е равно на сумата на всички останали. Ако има такъв елемент, печата "Yes", "Sum = " + **неговата стойност**; иначе печата "No", "Diff = " + **разликата между най-големия елемент и сумата на останалите** (абсолютна стойност).

### Примерен вход и изход

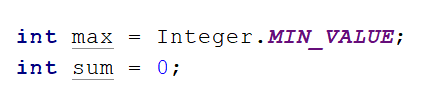
|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **коментари** |
| 7  3  4  1  1  2  **12**  1 | Yes  Sum = 12 | Намираме сбора на всички **7** числа:  3 + 4 + 1 + 1 + 2 + **12** + 1 = **24**  Най-голямото число в поредицата е: **12**  Следователно търсим дали сбора на всички числа без максималното е равен на максималното число.  Сбор на числата, без най-голямото: **24** – **12** = 12, който е равен на максималното число **12**. |
| 4  **6**  1  2  3 | Yes  Sum = 6 | Намираме сбора на всички **4** числа:  **6** + 1 + 2 + 3 = 12  Най-голямото число в поредицата е: **6**  Следователно търсим дали сбора на всички числа без максималното е равен на максималното число.  Сбор на числата, без най-голямото: **12** – **6** = 6, който е равен на максималното число **6.** |
| **3**  1  1  10 | No  Diff = 8 | Намираме сбора на всички **3** числа:  1 + 1 + 10 = 12  Най-голямото число в поредицата е: 10  Следователно търсим дали сбора на всички числа без максималното е равен на максималното число.  Сбор на числата, без най-голямото: **12** – 10= 2, който **НЕ е** равен на максималното число 10**.**  Разликата между най-големия елемент и сумата на останалите :  |10 - 2| = 8 |
| 3  5  5  1 | No  Diff = 1 | Намираме сбора на всички **3** числа:  5 + 5 + 1 = 11  Най-голямото число в поредицата е: **5**  Следователно търсим дали сбора на всички числа без максималното е равен на максималното число.  Сбор на числата, без най-голямото: **11** – **5** = 6, който **НЕ е** равен на максималното число **5.**  Разликата между най-големия елемент и сумата на останалите :  |5 - 6| = |-1| = **1** |
| 3  1  1  1 | No  Diff = 1 | Намираме сбора на всички **3** числа:  1 + 1 + 1 = 3  Най-голямото число в поредицата е: 1  Следователно търсим дали сбора на всички числа без максималното е равен на максималното число.  Сбор на числата, без най-голямото: **3** – 1= 2, който **НЕ е** равен на максималното число 1**.**  Разликата между най-големия елемент и сумата на останалите :  |1 - 2| = |-1| = **1** |

### Насоки

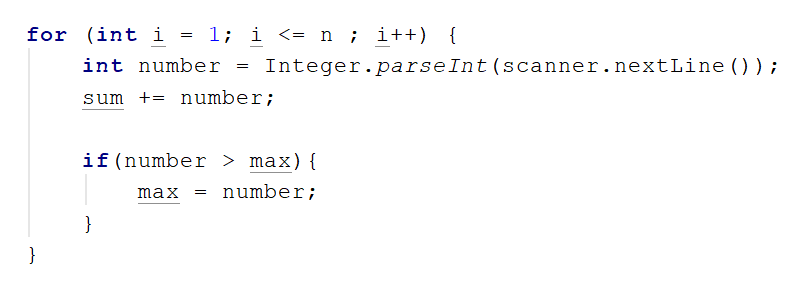
1. Прочетете входните данни **(n – броя на числата)**:



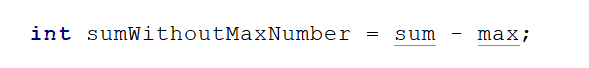
1. Създайте си **две нови променливи**. В едната **изчислявайте максималното число**, в другата **сумата на всички числа**. Задайте **начални стойности на двете променливи**. На променливата за **максимума задайте стойност Integer.MIN\_VALUE**, а на тази за **сумата задайте стойност 0**.



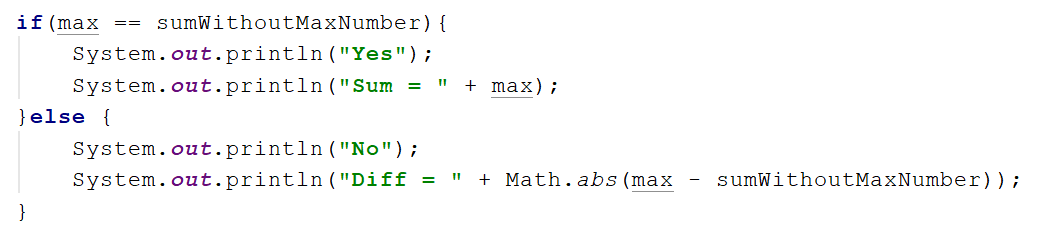
1. Направете **for цикъл, за да прочетете n на брой числа**. При всяко прочитане на число **го добавяйте към сумата.** Направете проверка **дали въведеното число е по-голямо от моментния** **максимум**. Ако е **по-голямо, максимумът става равен на това число**.



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 максиамлното 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** (изплозвайте **Math.abs**, за да намерите **абсолютната стойност**).



## Четни / нечетни позиции

Напишете програма, която чете n-на брой **числа**, въведени от потребителя, и пресмята **сумата**, **минимума** и **максимума** на числата на **четни** и **нечетни** позиции (броим от 1). Когато няма минимален / максимален елемент, отпечатайте "No".

Изходът да се форматира в следния вид:

"OddSum=" + {**сума** на числата на **нечетни** позиции},

"OddMin=" + { **минимална** стойност на числата на **нечетни** позиции } / {“No”},

"OddMax=" + { **максимална** стойност на числата на **нечетни** позиции } / {“No”},

"EvenSum=" + { **сума** на числата на **четни** позиции },

"EvenMin=" + { **минимална** стойност на числата на **четни** позиции } / {“No”},

"EvenMax=" + { **максимална** стойност на числата на **четни** позиции } / {“No”}

**Всяко число трябва да е форматирано до втория знак след десетичната запетая.**

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 6  **2**  3  **5**  4  **2**  1 | OddSum=9.00, OddMin=2.00, OddMax=5.00, EvenSum=8.00, EvenMin=1.00, EvenMax=4.00 | 2  **1.5**  -2.5 | OddSum=1.50, OddMin=1.50, OddMax=1.50, EvenSum=-2.50, EvenMin=-2.50, EvenMax=-2.50 | 1  **1** | OddSum=1.00, OddMin=1.00, OddMax=1.00, EvenSum=0.00, EvenMin=No, EvenMax=No | 0 | OddSum=0.00, OddMin=No, OddMax=No, EvenSum=0.00, EvenMin=No, EvenMax=No |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **Изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5  **3**  -2  **8**  11  **-3** | OddSum=8.00, OddMin=-3.00, OddMax=8.00, EvenSum=9.00, EvenMin=-2.00, EvenMax=11.00 |  | 4  1.5  **1.75**  1.5  **1.75** | OddSum=3.00, OddMin=1.50, OddMax=1.50, EvenSum=3.50, EvenMin=1.75, EvenMax=1.75 |  | 1  **-5** | OddSum=-5.00, OddMin=-5.00, OddMax=-5.00, EvenSum=0.00, EvenMin=No, EvenMax=No |  | 3  **-1**  -2  **-3** | OddSum=-4.00, OddMin=-3.00, OddMax=-1.00, EvenSum=-2.00, EvenMin=-2.00, EvenMax=-2.00 |

Задача обединява няколко предходни задачи: намиране на **минимум**, намиране на **максимум**, намиране на **сума** и обработка на елементите от **четни и нечетни позиции**. Припомнете си ги.

### Насоки

1. Работете с **реални числа** (не цели). Сумата, минимумът и максимумът също са реални числа.
2. Използвайте **неутрална начална стойност** при намиране на минимум / максимум, например **1000000000.0** и **-1000000000.0**. Ако получите накрая неутралната стойност, печатайте “**No**”.
3. Завъртете for цикъл до числото, което ви се въвежда като на всеки нов ред прочитате ново число num.
4. Проверете дали **позицията на числото** е **четна** или **нечетна**, като променливата инициализирана в цикъла i отговаря на **позицията на числото**.
5. Ако позицията на числото е **четно**, увеличете **сумата на четните числа** и проверете дали числото е   
   **по-голямо** от **най-голямото четно**, и му презапишете стойността. Също така проверете дали числото е **по-малко** от **най-малкото четно число** и му презапишете стойността.
6. Аналогично направете същото и за **нечетните числа**.

## Еднакви двойки

Дадени са 2 \* n - на брой числа. Първото и второто формират **двойка**, третото и четвъртото също и т.н. Всяка двойка има **стойност** – сумата от съставящите я числа. Напишете програма, която проверява **дали всички двойки имат еднаква стойност**. В противен случай печата **максималната разлика** между две последователни двойки. Ако всички двойки имат еднаква стойност, отпечатайте "**Yes, value={value}**" + **стойността**. В противен случай отпечатайте "**No, maxdiff={difference}**" + **максималната разлика**.

### Примерен вход и изход

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **коментари** |  | **вход** | **изход** | **коментари** |
| 3  1  2  0  3  4  -1 | Yes, value=3 | стойности = {3, 3, 3}  еднакви стойности | 2  1  2  2  2 | No, maxdiff=1 | стойности = {3, 4}  разлики = {1}  макс. разлика = 1 |
| 4  1  1  3  1  2  2  0  0 | No, maxdiff=4 | стойности = {2, 4, 4, 0}  разлики = {2, 0, 4}  макс. разлика = 4 | 1  5  5 | Yes, value=10 | стойности = {10}  една стойност  еднакви стойности |
| 2  -1  0  0  -1 | Yes, value=-1 | стойности = {-1, -1}  еднакви стойности |  | 2  -1  2  0  -1 | No, maxdiff=2 | стойности = {1, -1}  разлики = {2}  макс. разлика = 2 |

# Примерни изпитни задачи

## 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 = 12.0 / 20 \* 100 = **60.00**% |
| 200 … 399 | 250, 200 | 2 | p2 = 2.0 / 20 \* 100 = **10.00**% |
| 400 … 599 | 450 | 1 | p3 = 1.0 / 20 \* 100 = **5.00**% |
| 600 … 799 | 680, 600, 799 | 3 | p4 = 3.0 / 20 \* 100 = **15.00**% |
| ≥ 800 | 920, 800 | 2 | p5 = 2.0 / 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

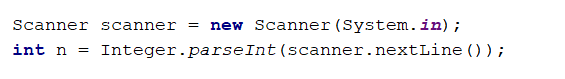
Да се отпечата на конзолата **хистограмата** – **5 реда**, всеки от които съдържа число между 0% и 100%, с точност **две цифри след десетичната запетая**, например 25.00%, 66.67%, 57.14%.

### Примерен вход и изход

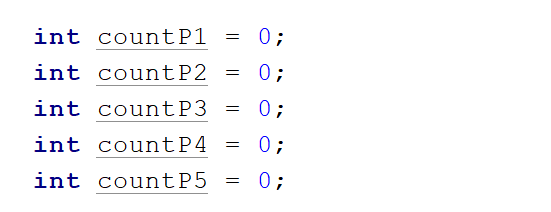
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| **3**  1  2  999 | 66.67%  0.00%  0.00%  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% |

### Насоки

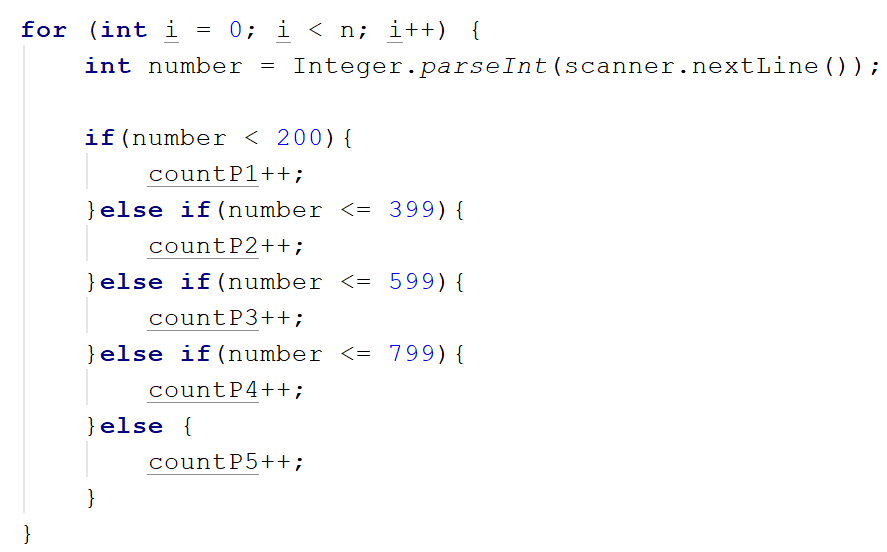
1. Прочетете входните данни **(n – броя на числата)**:



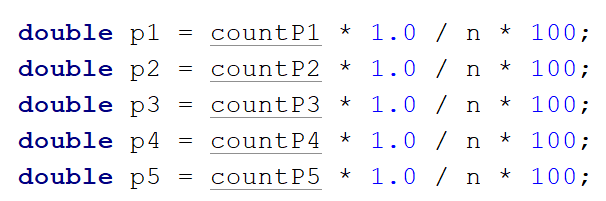
1. За да изчислите какъв **процент са тези числа**, трябва да знаете броя на числата **във всеки един диапазон**. Създайте си **5 нови променливи**, като във всяка една от тях ще пазите **броя на числата във всеки диапазон**. Задайте на всяка една от тях **начална стойност 0**.



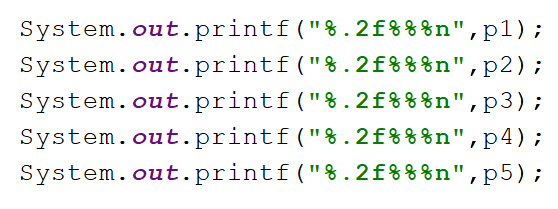
1. Направете **for цикъл, за да прочетете n на брой числа**. След това **проверете** **в кой диапазон попада въведеното число**. Ако е по-малко от 200, увеличете променливата, в която държите броя на числата в първия диапазон, с едно. Ако е между 200 и 399, увеличете променливата за числата във втория диапазон и т.н. за всички останали интервали.



1. След като знаете какъв **брой числа има във всеки един диапазон**, трябва да намерите какъв **процент** са тези числа от всички. **Процентът се изчислява като разделите броя на числата в даден диапазон на общия брой числа и го умножите по 100**. За целта си **създайте 5 нови променливи** и в тях изчислявайте процента числа във всеки диапазон.



1. Принтирайте получените проценти на **5 отделни реда** и форматирайте процентите до **втория знак след десетичната запетая**. Използвайте **printf** и шаблоните **%.2f и %n**, за да отпечатате нов ред.



## Деление без остатък

Дадени са n-на брой **цели числа** в интервала [**1**…**1000**]. От тях някакъв **процент** p1 **се делят без остатък на 2**, друг **процент** p2 се **делят без остатък на 3**, друг **процент** p3 се **делят без остатък на 4**. Да се напише програма, която изчислява и отпечатва процентите p1, p2 и p3.

**Пример**: имаме n = **10** числа: 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. Получаваме следното разпределение и визуализация:

|  |  |  |  |
| --- | --- | --- | --- |
| **Деление без остатък на:** | **Числа в диапазона** | **Брой числа** | **Процент** |
| 2 | 680, 2, 600, 200, 800, 46, 128 | 7 | p1 = 7.0 / 10 \* 100 = **70.00**% |
| 3 | 600 | 1 | p2 = 1.0 / 10 \* 100 = **10.00**% |
| 4 | 680, 600, 200, 800, 128 | 5 | p3 = 5.0 / 10 \* 100 = **50.00**% |

### 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, които да бъдат проверени на колко се делят.

### Изход

Да се отпечатат на конзолата **3 реда**, всеки от които съдържа процент между 0% и 100%, с точност две цифри след десетичната точка, например 25.00%, 66.67%, 57.14%.

* На **първия ред** - процентът на числата, които **се делят на 2**
* На **втория ред** - процентът на числата, които **се делят на** **3**
* На **третия ред** - процентът на числата, които **се делят на 4**

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| **10**  680  2  600  200  800  799  199  46  128  65 | 70.00%  10.00%  50.00% | **3**  3  6  9 | 33.33%  100.00%  0.00% |

## 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 сайта се налагат различни 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 [700...1500]**

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

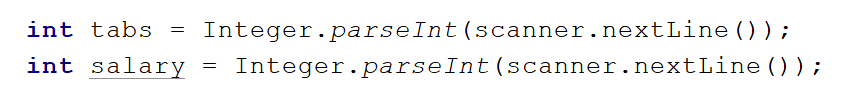
**Ако по време на проверката заплатата стане по-малка или равна на 0 лева, на конзолата се изписва   
"You have lost your salary." и програмата приключва. В противен случай, след проверката на конзолата се изписва остатъкът от заплатата (да се изпише като цяло число).**

### 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.  Заплатата е 750  За първия таб -> Facebook глоба 150 лв.(750 – 150 = 600)  За втория таб -> Dev.bg не глобяват  За третия таб -> Instagram глоба 100 лв.(600 – 100 = 500)  За четвъртия таб -> Facebook глоба 150 лв.(500 – 150 = 350)  За петия таб -> Reddit глоба 50 лв. (350 – 50 = 300)  За шестия таб -> Facebook глоба 150 лв.(300 – 150 = 150)  За седмия таб -> Facebook глоба 150 лв.(150 – 150 = 0)  Заплатата е равна на 0, следователно се 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 |

### Насоки

1. Прочетете входните данни (**брой отворени табове и заплата**):

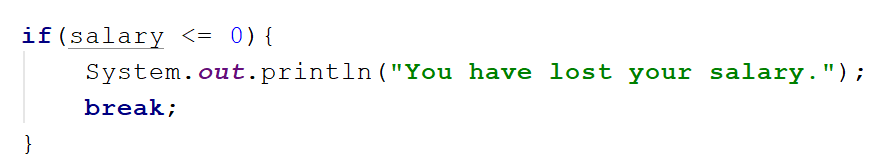


1. Направете **for цикъл**, за да въведете **n на брой пъти (броя на табовете) името на посетения сайт**.

След това направете **проверка за името на сайта**, ако сайтът е **Facebook, намалете заплатата със 150**, ако е **Instagram, намалете заплатата със 100**, а ако е **Reddit, намалете с 50**.



1. След всяка модификация на заплатата, **направете проверка дали не е по-малка или равна на 0**. Направете проверката във **for цикъла**. Ако е **по-малка или равна на 0**, принтирайте **You have lost your salary** и прекъснете цикъла с **break**.



1. Ако след всяко въвеждане на сайт, **все още има останала заплата**, т.е. **заплатата е по-голяма от 0**, принтирайте **остатъка от нея** (**извън for цикъла**).

