# Упражнения: Повторения с цикли – for-цикъл

Задачи за упражнение в клас и за домашно към курса "Основи на програмирането" @ СофтУни.

## 1. Числа до 1000, завършващи на 7

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

| вход | изход |
|---|---|
| (няма) | 7<br>17<br>27<br>…<br>997 |

**Тествайте** решението си в **judge системата**: https://judge.softuni.bg/Contests/Compete/Index/1164#0

**Подсказка**: можете да завъртите **for**-цикъл от 1 до 1000 и да проверите всяко число дали завършва на 7. Едно число **num** завършва на 7, когато **(num % 10 == 7)**.

## 2. 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** (по абсолютна стойност).

### Примерен вход и изход

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

| | | |
|---|---|---|
| 3<br>1<br>1<br>1 | No<br>Diff = 1 | |

**Тествайте** решението си в **judge системата**: https://judge.softuni.bg/Contests/Compete/Index/1164#1

**Подсказка**: изчислете **сумата** на всички елементи и **най-големият** от тях и проверете търсеното условие.

## 3. 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**"}

### Примерен вход и изход

| вход | изход |
|---|---|
| 6<br>**2**<br>3<br>**5**<br>4<br>**2**<br>1 | OddSum=9,<br>OddMin=2,<br>OddMax=5,<br>EvenSum=8,<br>EvenMin=1,<br>EvenMax=4 |

| вход | изход |
|---|---|
| 2<br>**1.5**<br>-2.5 | OddSum=1.5,<br>OddMin=1.5,<br>OddMax=1.5,<br>EvenSum=-2.5,<br>EvenMin=-2.5,<br>EvenMax=-2.5 |

| вход | изход |
|---|---|
| 1<br>**1** | OddSum=1,<br>OddMin=1,<br>OddMax=1,<br>EvenSum=0,<br>EvenMin=No,<br>EvenMax=No |

| вход | изход |
|---|---|
| 0 | OddSum=0,<br>OddMin=No,<br>OddMax=No,<br>EvenSum=0,<br>EvenMin=No,<br>EvenMax=No |

| вход | изход |
|---|---|
| 5<br>**3**<br>-2<br>**8**<br>11<br>**-3** | OddSum=8,<br>OddMin=-3,<br>OddMax=8,<br>EvenSum=9,<br>EvenMin=-2,<br>EvenMax=11 |

| вход | изход |
|---|---|
| 4<br>1.5<br>**1.75**<br>1.5<br>**1.75** | OddSum=3,<br>OddMin=1.5,<br>OddMax=1.5,<br>EvenSum=3.5,<br>EvenMin=1.75,<br>EvenMax=1.75 |

| вход | изход |
|---|---|
| 1<br>**-5** | OddSum=-5,<br>OddMin=-5,<br>OddMax=-5,<br>EvenSum=0,<br>EvenMin=No,<br>EvenMax=No |

| вход | изход |
|---|---|
| 3<br>**-1**<br>-2<br>**-3** | OddSum=-4,<br>OddMin=-3,<br>OddMax=-1,<br>EvenSum=-2,<br>EvenMin=-2,<br>EvenMax=-2 |

**Тествайте** решението си в **judge системата**: https://judge.softuni.bg/Contests/Compete/Index/1164#2

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.

- Работете с **дробни числа** (не цели). Сумата, минимумът и максимумът също са дробни числа.
- Използвайте **неутрална начална стойност** при намиране на минимум / максимум, например **1000000000.0** и **-1000000000.0**. Ако получите накрая неутралната стойност, печатайте "**No**".

# 4. 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** или 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 |
|---|---|---|
| 3<br>1<br>2<br>0<br>3<br>4<br>-1 | Yes, value=3 | стойности = {3, 3, 3}<br>еднакви стойности |
| 4<br>1<br>1<br>3<br>1<br>2<br>2<br>0<br>0 | No, maxdiff=4 | стойности = {2, 4, 4, 0}<br>разлики = {2, 0, 4}<br>макс. разлика = 4 |
| 2<br>-1<br>0<br>0<br>-1 | Yes, value=-1 | стойности = {-1, -1}<br>еднакви стойности |

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

**Тествайте** решението си в **judge системата**: https://judge.softuni.bg/Contests/Compete/Index/1164#3

**Подсказки**:

- Прочитайте входните числа **по двойки**. За всяка двойка пресмятайте **сумата**.
- Докато четете входните двойки, за всяка двойка без първата пресмятайте **разликата с предходната**. За целта пазете в отделна променлива сумата на предходната двойка.
- Намерете **най-голямата разлика** между две двойки. Ако е **0**, печатайте "**Yes**" иначе "**No**" + разликата.

# Примерни изпитни задачи

## 5. Хистограма

*Тествайте решението си **тук**.*

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 |
|---|---|
| **3**<br>1<br>2<br>999 | 66.67%<br>0.00%<br>0.00%<br>0.00%<br>33.33% |

| Вход | Изход |
|---|---|
| **4**<br>53<br>7<br>56<br>999 | 75.00%<br>0.00%<br>0.00%<br>0.00%<br>25.00% |

| Вход | Изход |
|---|---|
| **7**<br>800<br>801<br>250<br>199<br>399<br>599<br>799 | 14.29%<br>28.57%<br>14.29%<br>14.29%<br>28.57% |

| Вход | Изход |
|---|---|
| **9**<br>367<br>99<br>200<br>799<br>999<br>333<br>555<br>111<br>9 | 33.33%<br>33.33%<br>11.11%<br>11.11%<br>11.11% |

| Вход | Изход |
|---|---|
| **14**<br>53<br>7<br>56<br>180<br>450<br>920<br>12<br>7<br>150<br>250<br>680<br>2<br>600<br>200 | 57.14%<br>14.29%<br>7.14%<br>14.29%<br>7.14% |

# 6. Деление без остатък

*Тествайте решението си **тук**.*

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 / 10 * 100` = **`10.00`**% |
| 4 | `680, 600, 200, 800, 128` | 5 | p3 = `5 / 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 **първият ред** - процентът на числата които **се делят на 2**
- На **вторият ред** - процентът на числата които **се делят на 3**
- На **третият ред** - процентът на числата които **се делят на 4**

## Примерен вход и изход

| Вход | Изход |
|---|---|
| **10**<br>680<br>2<br>600<br>200<br>800<br>799<br>199<br>46<br>128<br>65 | 70.00%<br>10.00%<br>50.00% |

| Вход | Изход |
|---|---|
| **3**<br>3<br>6<br>9 | 33.33%<br>100.00%<br>0.00% |

# 7. Заплата

*Тествайте решението си **тук**.*

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<br>750<br>Facebook<br>Dev.bg<br>Instagram<br>Facebook<br>Reddit<br>Facebook<br>Facebook | You have lost your salary. | Има 10 отворени таба в браузъра.<br>Заплатата е 750<br>За първия таб -> Facebook глоба 150 лв.(750 – 150 = 600)<br>За втория таб -> Dev.bg не глобяват<br>За третия таб -> Instagram глоба 100 лв.(600 – 100 = 500)<br>За четвъртия таб -> Facebook глоба 150 лв.(500 – 150 = 350)<br>За петия таб -> Reddit глоба 50 лв. (350 – 50 = 300)<br>За шестия таб -> Facebook глоба 150 лв.(300 – 150 = 150)<br>За седмия таб -> Facebook глоба 150 лв.(150 – 150 = 0)<br>Заплатата е равна на 0, следователно се изписва съответният изход и програмата приключва. |

| Вход | Изход | Вход | Изход |
|---|---|---|---|
| 3<br>500<br>Github.com<br>Stackoverflow.com<br>softuni.bg | 500 | 3<br>500<br>Facebook<br>Stackoverflow.com<br>softuni.bg | 350 |