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

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

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**

* 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**.



* 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**

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



* 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**.



* 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**.



* 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**.



* 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**).



**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 / 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% |

* **Умната Лили**

Лили вече е на **N години**. За всеки свой **рожден ден** тя получава подарък.

* За **нечетните** рождени дни (**1, 3, 5...n**) получава **играчки.**
* За **четните** рождени дни (**2, 4, 6...n**) получава **пари**.

За **втория рожден ден** получава **10.00 лв**, като **сумата се увеличава с 10.00 лв., за всеки следващ четен рожден ден** (**2 -> 10**, **4 -> 20**, **6 -> 30**...и т.н.). През годините Лили тайно е спестявала парите. **Братът** на Лили, **в годините**, които тя **получава пари**, **взима по 1.00 лев** от тях. Лили **продала играчките** получени през годините, **всяка за P лева** и **добавила** сумата **към спестените пари**. С парите искала да си **купи пералня за X лева**. Напишете програма, която да пресмята, **колко пари е събрала** и дали ѝ **стигат да си купи пералня**.

**Вход**

Програмата прочита **3 числа**, въведени от потребителя, на отделни редове:

* **Възрастта** на Лили - **цяло число** в интервала **[1...77]**
* **Цената на пералнята** - число в интервала **[1.00...10 000.00]**
* **Единична цена** **на играчка** - **цяло** **число** в интервала **[0...40]**

**Изход**

Да се отпечата на конзолата един ред:

* Ако парите на Лили са достатъчни:
* **"Yes! {N}"** - където **N** е остатъка пари след покупката
* Ако парите не са достатъчни:
* **"No! {М}"** - където **M** е сумата, която не достига

Числата **N** и **M** трябва да за **форматирани до вторият знак след десетичната запетая**.

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **Коментари** |
| 10  **170.00**  **6** | Yes! 5.00 | **Първи рожден ден** получава **играчка**  **Втори рожден ден** получава **пари (10 лв.)**  **Трети рожден ден** получава **играчка**  **Четвърти рожден ден** получава **пари (20 лв.)**  **Пети рожден ден** получава **играчка**  **Шести рожден ден** получава **пари (30 лв.)**  **Седми рожден ден** получава **играчка**  **Осми рожден ден** получава **пари (40 лв.)**  **Девети рожден ден** получава **играчка**  **Десети рожден ден** получава **пари (50 лв.)**  **Спестила е**: 10 + 20 + 30 + 40 + 50 = **150 лв**.  Продала е **5 броя играчки \* 6 лв.** = **30 лв**.  **Брат ѝ взел 5 пъти \* 1 лев** = **5 лв**.  **Общо спестени пари**: (**150** + **30**) – **5** = **175 лв**.  **175 >= 170** (цената на пералнята)  => **успяла** е да я купи  **Остават:** **175** - **170** = **5 лв.** |
| 21  **1570.98**  **3** | No! 997.98 | **Спестила е 550 лв**.  **Продала** е **11** играчки \* **3 лв.** = **33 лв**.  Брат ѝ **взимал 10** години \* **1 лев** = **10 лв**.  **Общо спестени пари:** (**550** + **33**) – **10** = **573 лв.**  **573 < 1570.98** – **не е успяла** да купи пералня  **Нужни пари:** **1570.98** – **573** = **997.98 лв.** |

* **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 |

* **6. Оскари**

Поканени сте от академията да напишете софтуер, който да пресмята точките за актьор/актриса. Академията ще ви даде първоначални **точки за актьора**. След това **всеки** **оценяващ** ще дава **своята** **оценка**. Точките, които **актьора** **получава** се формират от: **дължината** на **името** на **оценяващия умножено** по **точките,** които **дава делено** на **две.**

**Ако резултатът в някой момент надхвърли 1250.5 програмата трябва да прекъсне и да се отпечата, че дадения актьор е получил номинация.**

**Вход**

* Име на актьора - **текст**
* Точки от академията - **реално число в интервала [2.0... 450.5]**
* Брой оценяващи **n** - **цяло число в интервала[1… 20]**

На следващите **n-на брой реда**:

* Име на оценяващия - **текст**
* Точки от оценяващия - **реално число в интервала [1.0... 50.0]**

**Изход**

Да се отпечата на конзолата един ред:

* Ако точките са над **1250.5**:

**"Congratulations, {име на актьора} got a nominee for leading role with {точки}!"**

* Ако точките **не са** достатъчни:

**"Sorry, {име на актьора} you need {нужни точки} more!"**

**Резултатът да се форматирана до първата цифра след десетичния знак!**

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| Zahari Baharov  **205**  **4**  Johnny Depp  **45**  Will Smith  **29**  Jet Lee  **10**  Matthew Mcconaughey  39 | Sorry, Zahari Baharov you need 247.5 more! | Zahari Baharov започва с **205** точки, като **4** човека ще го оценяват.  Първи е Johnny Depp  => **205** + ((11 \* **45**) / 2) = **452.5**  Втори е Will Smith  => **452.5** + ((10 \* **29**) / 2) = **597.5**  Трети е Jet Lee  => **597.5** + ((7 \***10**) / 2) = **632.5**  Четвърти е Matthew Mcconaughey  => **632.5** + ((19 \* **39**) / 2) = **1003.0**  **1003.0** < **1250.5**  => Toчките не са достатъчни.  Нужни точки: **1250.5** - **1003.0** = 247.5 |
| Sandra Bullock  340  5  Robert De Niro  50  Julia Roberts  40.5  Daniel Day-Lewis  39.4  Nicolas Cage  29.9  Stoyanka Mutafova  33 | Congratulations, Sandra Bullock got a nominee for leading role with 1268.5! |  |

* **7. Трекинг мания**

Катерачи от цяла България се събират на групи и набелязват следващите върхове за изкачване. Според **размера на групата**, катерачите ще изкачват **различни върхове**.

* Група **до 5 човека** – изкачват **Мусала**
* Група **от 6 до 12** **човека** – изкачват **Монблан**
* Група **от 13 до 25** **човека** – изкачват **Килиманджаро**
* Група **от 26 до 40** **човека** – изкачват **К2**
* Група **от 41 или повече** **човека** – изкачват **Еверест**

Да се **напише програма**, която **изчислява процента на катерачите изкачващи всеки връх.**

**Вход**

От конзолата се четат **поредица от числа, всяко на отделен ред**:

* На **първия ред** – **броя на групите** от катерачи – **цяло число в интервала [1...1000]**
* За **всяка една група** на отделен ред – **броя на хората в групата** – **цяло число в интервала [1...1000]**

**Изход**

Да се отпечатат на конзолата **5 реда**, всеки от които съдържа **процент между 0.00% и 100.00%** **с точност до втората цифра след десетичната запетая**.

* **Първи ред** - **процентът** изкачващи **Мусала**
* **Втори ред** – **процентът** изкачващи **Монблан**
* **Трети ред** – **процентът** изкачващи **Килиманджаро**
* **Четвърти ред** – **процентът** изкачващи **К2**
* **Пети ред** – **процентът** изкачващи **Еверест**

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10  10  5  1  100  12  26  17  37  40  78 | 1.84%  6.75%  5.21%  31.60%  54.60% | Общ брой хора: 10 + 5 + 1 + 100 + 12 + 26 + 17 + 37 + 40 + 78 = 326  Изкачващи Мусала: 6 / 326 \* 100 = 1.84%  Изкачващи Монблан: 22/326\*100 = 6.75%  Изкачващи Килиманджаро: 17/326\*100 = 5.21%  Изкачващи К2: 103/326\*100 = 31.60%  Изкачващи Еверест: 178/326\*100 = 54.60% |
| **Вход** | **Изход** | **Обяснения** |
| 5  25  41  31  250  6 | 0.00%  1.70%  7.08%  8.78%  82.44% | Общ брой хора: 25 + 41 + 31 + 250 + 6 = 353  Изкачващи Мусала: 0 / 353 \* 100 = 0.00%  Изкачващи Монблан: 6 / 353 \* 100 = 1.78%  Изкачващи Килиманджаро: 25 / 353 \* 100 = 7.08%  Изкачващи К2: 31 / 353 \* 100 = 8.78%  Изкачващи Еверест: 291 / 353 \* 100 = 82.44% |

* **8. Световна ранглиста по тенис**

Григор Димитров е тенисист, чиято следваща цел е изкачването в световната ранглиста по тенис за мъже.

През годината Гришо участва в **определен брой турнири**, като за всеки турнир получава точки, които зависят от **позицията, на която е завършил в турнира**. Има **три варианта** за завършване на турнир:

* **W - ако е победител получава 2000 точки**
* **F - ако е финалист получава 1200 точки**
* **SF - ако е полуфиналист получава 720 точки**

Напишете програма, която **изчислява колко ще са точките на Григор след изиграване на всички турнири**, като знаете **с колко точки стартира сезона**. Също изчислете **колко точки средно печели** от всички изиграни турнири и **колко процента** от турнирите е спечелил.

**Вход**

От конзолата първо се четат **два реда**:

* **Брой турнири, в които е участвал – цяло число в интервала [1…20]**
* **Начален брой точки в ранглистата - цяло число в интервала [1...4000]**

**За всеки турнир се прочита отделен ред:**

* **Достигнат етап от турнира – текст –** "**W**"**,** "**F**" **или** "**SF**"

**Изход**

Отпечатват се **три реда** в следния формат:

* "**Final points: {брой точки след изиграните турнири}**"
* "**Average points: {средно колко точки печели за турнир}**"
* "**{процент спечелени турнири}%**"

**Средните точки да бъдат закръглени към най-близкото цяло число надолу, а процентът да се форматира до втората цифра след десетичния знак.**

**Примерен вход и изход**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | | |
| **5**  **1400**  **F**  **SF**  **W**  **W**  **SF** | Final points: 8040  Average points: 1328 40.00% | **5 турнира и начален брой точки: 1400**  **1-ви турнир -> финал (F) -> точки = 1400 + 1200 = 2600**  **2-ри турнир -> полуфинал (SF) -> точки = 2600 + 720 = 3320**  **3-ти турнир -> победител (W) -> точки = 3320 + 2000 = 5320**  **4-ти турнир -> победител (W) -> точки = 5320 + 2000 = 7320**  **5-ти турнир -> полуфинал (SF) -> точки = 7320 + 720 = 8040**  **Точки след изиграване на турнирите: 8040**  **Средно спечелени точки за турнир:**  **(1200 + 720 + 2000 + 2000 + 720) / 5 = 6640 / 5 = 1328**  **Брой спечелени турнири: 2**  **Процент спечелени турнири: (2 / 5) \* 100 = 40 %** | | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 4  750  SF  W  SF  W | Final points: 6190  Average points: 1360  50.00% | 7  1200  SF  F  W  F  W  SF  W | Final points: 11040  Average points: 1405  42.86% |