

For - цикъл – допълнително упражнение

Задачи за допълнително упражнение към курса "[Основи на програмирането](#)" @ СофтУни.

Тествайте решенията си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/1680#0>

1. Завръщане в миналото

Иванчо е на **18 години** и получава наследство, което се състои от **X сума пари** и **машина на времето**. Той решава да се върне до **1800 година**, но не знае дали парите ще са достатъчни, за да живее без да работи. Напишете програма, която пресмята, дали Иванчо ще има достатъчно пари, за да не се налага да работи до дадена година включително. Като приемем, че за всяка четна (1800, 1802 и т.н.) година ще харчи **12 000 лева**. За всяка нечетна (1801, 1803 и т.н.) ще харчи **12 000 + 50 * [годините, които е навършил през дадената година]**.

Вход

Входът се чете от конзолата и съдържа точно 2 реда:

- Наследените пари – реално число в интервала [1.00 ... 1 000 000.00]
- Годината, до която трябва да живее (включително) – цяло число в интервала [1801 ... 1900]

Изход

Да се отпечата на конзолата **1 ред**. Сумата трябва да е форматирана до два знака след десетичната запетая:

- Ако парите са достатъчно:
 - "Yes! He will live a carefree life and will have {N} dollars left." – където N са парите, които ще му останат.
- Ако парите НЕ са достатъчно:
 - "He will need {M} dollars to survive." – където M е сумата, която НЕ достига.

Примерен вход и изход

Вход	Изход	Обяснения
50000 1802	Yes! He will live a carefree life and will have 13050.00 dollars left.	1800 → четна → Харчи 12000 лева → Остават 50000 – 12000 = 38000 1801 → нечетна → Харчи 12000 + 19*50 = 12000 + 950 = 12950 → Остават 38000 – 12950 = 25050 1802 → четна → Харчи 12000 лева → Остават 25050 – 12000 = 13050
Вход	Изход	Обяснения

100000. 15 1808	He will need 12399.85 dollars to survive.	1800 → четна → Остават 100000.15 – 12000 = 88000.15 1801 → нечетна → Остават 100000.15 – 12950 = 87050.15 ... 1808 → четна → -399.85 – 12000 = -12399.85 12399.85 не достигат
-----------------------	---	--

2. Болница

За даден период от време, всеки ден в болницата пристигат пациенти за преглед. Тя разполага първоначално със **7 лекари**. Всеки лекар може да преглежда **само по един пациент на ден**, но понякога има недостиг на лекари, затова **останалите пациенти се изпращат в други болници**. Всеки **трети ден**, болницата прави изчисления и **ако броят на непрегледаните пациенти е по-голям от броя на прегледаните, се назначава още един лекар**. Като назначаването става преди да започне приемът на пациенти за деня.

Напишете програма, която изчислява за дадения период броя на прегледаните и непрегледаните пациенти.

Вход

Входът се чете от **конзолата** и съдържа:

- На **първия ред** – **периода**, за който трябва да направите изчисления. **Цяло число** в интервала [1 ... 1000]
- На следващите **редове(равни на броя на дните)** – **броя пациенти**, които пристигат за преглед за текущия ден. **Цяло число** в интервала [0...10 000]

Изход

Да се **отпечата** на конзолата **2 реда** :

- На **първия ред**: "Treated patients: {брой прегледани пациенти}."
- На **втория ред**: "Untreated patients: {брой непрегледани пациенти}."

Примерен вход и изход

Вход	Изход	Обяснения	
4 7 27 9 1	Treated patients: 23. Untreated patients: 21.	1 ден : 7 прегледани и 0 непрегледани пациента за деня 2 ден : 7 прегледани и 20 непрегледани пациента за деня 3 ден : До момента прегледаните пациенти са общо 14, а непрегледаните – 20 → Назначава се нов лекар → 8 прегледани и 1 непрегледан пациент за деня 4 ден : 1 прегледан и 0 непрегледани пациента за деня Общо : 23 прегледани и 21 непрегледани пациенти.	
Вход	Изход	Вход	Изход
6 25 25 25	Treated patients: 40. Untreated patients: 87.	3 7 7 7	Treated patients: 21. Untreated patients: 0.

25			
25			
2			

3. Логистика

Отговаряте за логистиката на различни товари. В зависимост от теглото на товара е нужно различно превозно средство. Цената на тон, за която се превозва товара е различна за всяко превозно средство:

- До 3 тона – микробус (200 лева на тон)
- От 4 до 11 тона – камион (175 лева на тон)
- 12 и повече тона – влак (120 лева на тон)

Вашата задача е да изчислите **средната цена на тон превозен товар**, както и **процента на тоновете превозвани с всяко превозно средство**, спрямо **общото тегло(в тонове)** на всички товари.

Вход

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

- На **първия ред** – броя на товарите за превоз – цяло число в интервала [1...1000]
- За **всеки един товар** на отделен ред – **тонажа на товара** – цяло число в интервала [1...1000]

Изход

Да се отпечата на конзолата **4 реда**, както следва:

- **Първи ред** – **средната цена на тон превозен товар** (закръглена до втория знак след дес. запетая);
- **Втори ред** – **процентът тона превозвани с микробус** (процент между 0.00% и 100.00%);
- **Трети ред** – **процентът тона превозвани с камион** (процент между 0.00% и 100.00%);
- **Четвърти ред** – **процентът тона превозвани с влак** (процент между 0.00% и 100.00%).

Примерен вход и изход

Вход	Изход	Обяснения	Вход	Изход
4	143.80	Всички товари 1 + 5 + 16 + 3 = 25 тона	5	149.38
1	16.00%	(4 * 200 + 5 * 175 + 16 * 120)/25 = 143.80 (средно на тон)	2	7.50%
5	20.00%		10	42.50%
16	64.00%	С микробус – 4/25*100 = 16.00%	20	50.00%
3		С камион – 5/25*100 = 20.00%	1	
		С влак – 16/25*100 = 64.00%	7	

4. Оценки

Напишете програма, която да **пресмята статистика на оценки** от изпит. В началото програмата получава **броят на студентите** явили се на изпита и за **всеки студент неговата оценка**. На края програмата трябва да **изпечата процента на студенти** с оценка между **2.00 и 2.99**, между **3.00 и 3.99**, между **4.00 и 4.99**, **5.00 или повече**. Също така и **средният успех** на изпита.

Вход

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

- На **първия ред** – броя на студентите явили се на изпит – цяло число в интервала [1...1000]
- За **всеки един студент** на отделен ред – оценката от изпита – реално число в интервала [2.00...6.00]

Изход

Да се отпечата на конзолата **5 реда**, които съдържат следната информация:

Ред 1 - "Top students: {процент студенти с успех 5.00 или повече}%"

Ред 2 - "Between 4.00 and 4.99: {между 4.00 и 4.99 включително}%"

Ред 3 - "Between 3.00 and 3.99: {между 3.00 и 3.99 включително}%"

Ред 4 - "Fail: {по-малко от 3.00}%"

Ред 5 - "Average: {среден успех}%"

Всички числа трябва да са **форматирани до вторият знак след десетичната запетая**.

Примерен вход и изход

Вход	Изход	Обяснения
10 3.00 2.99 5.68 3.01 4 4 6.00 4.50 2.44 5	Top students: 30.00% Between 4.00 and 4.99: 30.00% Between 3.00 and 3.99: 20.00% Fail: 20.00% Average: 4.06	5 и повече – трима = 30% от 10 Между 4 и 4.99 – трима = 30% от 10 Между 3 и 3.99 – двама = 20% от 10 Под 3 – двама = 20% от 10 Средният успех: $3 + 2.99 + 5.68 + 3.01 + 4 + 4 + 6 + 4.50 + 2.44 + 5 = 40.62 / 10 = 4.062$
Вход	Изход	
6 2 3 4 5 6 2.2	Top students: 33.33% Between 4.00 and 4.99: 16.67% Between 3.00 and 3.99: 16.67% Fail: 33.33% Average: 3.70	

5. Игра на интервали

Напишете програма, която да **пресмята резултата от игра**. Първо получавате **число**, което **показва колко хода ще продължи играта**. После за всеки ход на играта ще получавате **по едно ново число**. Според интервала в който попада числото **се прибавят точки**. Ако числото е **отрицателно или по-голямо 50**, тогава то е **невалидно**.

В началото на играта резултата е **0** и на всеки ход се **прибавят точки** по следният начин:

- От 0 до 9 → 20 % от числото
- От 10 до 19 → 30 % от числото
- От 20 до 29 → 40 % от числото
- От 30 до 39 → 50 точки
- От 40 до 50 → 100 точки
- Невалидно число → резултата се дели на 2



Освен резултата програмата трябва да изкарва статистика за проценти числа в дадените интервали.

Вход

Входът се чете от конзолата:

- **Първи ред** - колко хода ще има по време на играта – цяло число в интервала [1...100]
- **За всеки ход** – числата, които се проверяват в кой интервал са – цели числа в интервала [-100...100]

Изход

Да се отпечата на конзолата 7 реда:

- 1ви ред: "{Краен резултат}"
- 2ри ред: "From 0 to 9: {процент в интервала}%"
- 3ти ред: "From 10 to 19: {процент в интервала}%"
- 4ти ред: "From 20 to 29: {процент в интервала}%"
- 5ти ред: "From 30 to 39: {процент в интервала}%"
- 6ти ред: "From 40 to 50: {процент в интервала}%"
- 7ми ред: "Invalid numbers: {процент в интервала}%"

Всички числа трябва да са форматирана до вторият знак след запетаята.

Примерен вход и изход

Вход	Изход	Обяснения
10 43 57 -12 23 12 0 50 40 30 20	295.80 From 0 to 9: 10.00% From 10 to 19: 10.00% From 20 to 29: 20.00% From 30 to 39: 10.00% From 40 to 50: 30.00% Invalid numbers: 20.00%	10 хода; Начален резултат = 0 точки 1 ви ход: $40 \leq 43 \leq 50 \rightarrow$ към резултата се добавят 100 точки 2 ри ход: $57 > 50 \rightarrow$ невалидно число $\rightarrow 100/2 = 50$ точки 3 ти ход: $-12 < 0 \rightarrow$ невалидно число $\rightarrow 50/2 = 25$ точки 4 ти ход: $20 \leq 23 \leq 29 \rightarrow$ към резултата се добавят 40% от 23 = 25 + 9.2 = 34.2 и така до 10тият ход..... От 0 до 9: 1 число (0) = 10% От 10 до 19: 1 число (10) = 10% От 20 до 29: 2 числа (20 и 23) = 20% От 30 до 39: 1 число (30) = 10% От 40 до 50: 3 числа (40, 43 и 50) = 30% Невалидни: 2 числа (57 и -12) = 20%

6. Месечни разходи

Напишете програма която да пресмята средният разход за месец на семейство за даден период време. За всеки месец разходите са следните:

- За ток – всеки месец е различен, ще се чете от конзолата
- за вода – 20 лв.
- за интернет – 15 лв.
- за други – събират се тока, водата и интернета за месеца и към сумата се прибавят 20%.

За всеки разход трябва да се пресметне колко общо е платено за всички месеци.

Вход

Входът се чете от конзолата:

- **Първи ред** – месеците за които се търси средният разход – цяло число в интервала [1...100]
- **За всеки месец** – сметката за ток – реално число в интервала [1.00...1000.00]



Изход

Да се отпечата на конзолата 5 реда:

- 1ви ред: "Electricity: {ток за всички месеци} lv"
- 2ри ред: "Water: {вода за всички месеци} lv"
- 3ти ред: "Internet: {интернет за всички месеци} lv"
- 4ти ред: "Other: {други за всички месеци} lv"
- 5ти ред: "Average: {средно всички разходи за месец} lv"

Всички числа трябва да са форматирана до вторият знак след запетаята.

Примерен вход и изход

Вход	Изход	Обяснения
5 68.63 89.25 132.53 93.53 63.22	Electricity: 447.16 lv Water: 100.00 lv Internet: 75.00 lv Other: 746.59 lv Average: 273.75 lv	За 5 месеца: Ток → $68.63 + 89.25 + 132.53 + 93.53 + 63.22 = 447.16$ лв. Вода → 5 месеца по 20 лв. = 100 лв. Интернет → 5 месеца по 15 = 75 лв. Други → $(68.63+20+15) + 20\% = 124.356$ → общо 746.592 лв. $(89.25+20+15) + 20\% = 149.1$ $(132.53+20+15) + 20\% = 201.036$ $(93.53+20+15) + 20\% = 154.236$ $(63.22+20+15) + 20\% = 117.864$ Средно на месец → $(447.16+100+75+746.592)/5 = 273.7504$ лв.
Вход	Изход	
8 123.54 231.54 140.23 100 122.4 430 178.52 64.2	Electricity: 1390.43 lv Water: 160.00 lv Internet: 120.00 lv Other: 2004.52 lv Average: 459.37 lv	

7. Футболен турнир

Екипът на СофтУни си организира футболен турнир. Първоначално прочитаме от конзолата **капацитета на стадиона и броят на всички фенове**. След това за всеки фен се чете **секторът, в който се намира**. Феновете на първия отбор са в сектор А и Б, а на втория – В и Г. Да се напише програма, която **изчислява процентите на феновете във всеки сектор, спрямо общия брой фенове на стадиона, както и общият процент на феновете за двата отбора, спрямо капацитета на стадиона**. **Общият брой на феновете НЕ надвишава капацитета на стадиона.**

Вход

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

1. Капацитетът на стадиона – цяло число в интервала [1 ... 10000];
2. Броят на всички фенове – цяло число в интервала [1 ... 10000].

За всеки един фен, на отделен ред се прочита:

1. Секторът, на който се намира – текст – "А", "В", "V" и "G".

Изход

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

1. Процентът на феновете в сектор А
2. Процентът на феновете в сектор Б
3. Процентът на феновете в сектор В
4. Процентът на феновете в сектор Г
5. Процентът на всички фенове, спрямо капацитета на стадиона.

Примерен вход и изход

Вход	Изход	Обяснения	
76 10 A V V V G B A V B B	20.00% 30.00% 40.00% 10.00% 13.16%	<p>Феновете в сектор A са 2. Изчисляваме какъв процент са от общия брой фенове – 10 на стадиона. $2 / 10 * 100 = 20.00\%$</p> <p>Феновете в сектор B са 3. По същия начин изчисляваме техния процент спрямо общия брой на феновете. $3 / 10 * 100 = 30.00\%$</p> <p>Феновете в сектор V са 4. По същия начин изчисляваме техния процент спрямо общия брой на феновете. $4 / 10 * 100 = 40.00\%$</p> <p>В сектор G има само 1 фен. По същия начин изчисляваме какъв процент е спрямо общия брой на феновете. $1 / 10 * 100 = 10.00\%$</p> <p>Изчисляваме процентът на всички фенове, спрямо капацитетът на стадиона: $10 / 76 * 100 = 13.16\%$</p>	
Вход	Изход	Вход	Изход
93 16 A V G G B B G B A B B B A B B A	25.00% 50.00% 6.25% 18.75% 17.20%	1000 12 A A V V A G A A V G V A	50.00% 0.00% 33.33% 16.67% 1.20%

8. 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 съставлящите 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	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

Задачи за шампиони

9. Часовник

Напишете програма, която отпечатва часовете в денонощието от 0:0 до 23:59, всеки на отделен ред. Часовете трябва да се изписват във формат "{час} : {минути}".

Примерен вход и изход

Вход	Изход
(няма вход)	0 : 0 0 : 1 0 : 2 0 : 3 0 : 4 0 : 5 0 : 6 0 : 7 0 : 8 0 : 9 0 : 10 ... 23 : 50 23 : 51 23 : 52

	23 : 53
	23 : 54
	23 : 55
	23 : 56
	23 : 57
	23 : 58
	23 : 59

Насоки: потърсете информация за **вложени цикли**.

10. Часовник – част 2

Напишете програма, която да отпечата **часовете в денонощието от 0:0:0 до 23:59:59**, всеки на отделен ред. Часовете да се изписват във формат "{час} : {минути} : {секунди} ".

Примерен вход и изход

Вход	Изход
(няма вход)	0 : 0 : 0 0 : 0 : 1 0 : 0 : 2 0 : 0 : 3 0 : 0 : 4 0 : 0 : 5 0 : 0 : 6 0 : 0 : 7 0 : 0 : 8 0 : 0 : 9 0 : 0 : 10 ... 23 : 59 : 50 23 : 59 : 51 23 : 59 : 52 23 : 59 : 53 23 : 59 : 54 23 : 59 : 55 23 : 59 : 56 23 : 59 : 57 23 : 59 : 58 23 59 : 59

11. 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
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

вход	изход
2 1.5 -2.5	OddSum=1.50, OddMin=1.50, OddMax=1.50, EvenSum=-2.50, EvenMin=-2.50, EvenMax=-2.50

вход	изход
1 1	OddSum=1.00, OddMin=1.00, OddMax=1.00, EvenSum=0.00, EvenMin=No, EvenMax=No

вход	изход
0	OddSum=0.00, OddMin=No, OddMax=No, EvenSum=0.00, EvenMin=No, EvenMax=No

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

вход	изход
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

вход	изход
1 -5	OddSum=-5.00, OddMin=-5.00, OddMax=-5.00, EvenSum=0.00, EvenMin=No, EvenMax=No

вход	изход
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**.