# Упражнения: Повторения с цикли

Тествайте решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/3899>

## Четни степени на 2

Да се напише програма, която чете число n, въведено от потребителя, и **печата четните степени на 2** **≤** **2n**:

* **20**, **22**, **24**, **26**, …, **2n**.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 3 | 1  4 | 4 | 1  4  16 | 5 | 1  4  16 | 6 | 1  4  16  64 | 7 | 1  4  16  64 |

## Редица цели числа

TODO: да се премести задачата в Loops Basics упражненията.

Напишете програма, която чете **n на брой цели числа**. Принтирайте **най-голямото** и **най-малкото** число измежду въведените, както е показано в примера по-долу.

На **първия ред** от входа е дадено числото **n**, a на **следващите n реда** е дадено по едно **цяло число**.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| **5**  10  20  304  0  50 | Max number: 304  Min number: 0 | **6**  250  5  2  0  100  1000 | Max number: 1000  Min number: 0 |

## Лява и дясна сума

Да се напише програма, която чете **2\***n на брой цели числа, подадени от потребителя, и проверява дали **сумата на първите** n **числа** (лява сума) е равна на **сумата на вторите** n **числа** (дясна сума). При равенство печата "Yes, sum = " + **сумата**; иначе печата " No, diff = " + **разликата**. Разликата се изчислява като положително число (по абсолютна стойност).

### Примери

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Коментар** |  | **Вход** | **Изход** | **Коментар** |
| **2**  10  90  60  40 | Yes, sum = 100 | 10+90 = 60+40 = 100 | **2**  90  9  50  50 | No, diff = 1 | 90+9 ≠ 50+50  Difference = |99-100| = 1 |

## 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**. При равенство да се отпечатат два реда: "Yes" и на нов ред "Sum = " + **сумата**; иначе да се отпечата "No" и на нов ред "Diff = " + **разликата**. Разликата се изчислява по абсолютна стойност.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

## Сума от числа

TODO: да се премести задачата в Loops Basics упражненията.

Напишете програма, която чете цяло число от конзолата и на всеки следващ ред цели числа, докато тяхната сума стане по-голяма или равна на първоначалното число. След приключване да се отпечата **сумата на въведените числа**.

### Примери

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Коментар** |  | **Вход** | **Изход** | **Коментар** |
| **100**  10  20  30  40 | 100 | 10+20+30+40 = 100 |  | **20**  1  2  3  4  5  6 | 21 | 1+2+3+4+5+6 = 21 |

## Редица числа 2K+1

Напишете програма, която чете число n, въведено от потребителя,и отпечатва **всички числа ≤ n от редицата**: 1, 3, 7, 15, 31, …. Всяко следващо число се изчислява като умножим **предишното** с **2** и добавим **1**.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 3 | 1  3 | 8 | 1  3  7 | 17 | 1  3  7  15 | 31 | 1  3  7  15  31 |

### Насоки

1. Прочетете от конзолата цяло число.



1. **Създайте** променлива от тип **цяло число**, която ще е брояч и има **първоначална стойност 1**.



1. **Създайте while** цикъл, който се повтаря докато **брояча е по-малък** или равен на числото, което сте прочели от конзолата.

Logo, company name

Description automatically generated

1. При всяко повторение на цикъла **принтирайте стойността на брояча** и му **прибавяйте дадената стойност**.

Text

Description automatically generated

## Най-голямо число

TODO: да се премести задачата в Loops Basics упражненията.

Напишете програма, която до получаване на **командата "Stop"**, чете **цели числа**, въведени от потребителя, и намира **най-голямото** измежду тях. Въвеждат се по едно число на ред.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 100  99  80  70  Stop | 100 | -10  20  -30  Stop | 20 | 45  -20  7  99  Stop | 99 | 999  Stop | 999 | -1  -2  Stop | -1 |

## Най-малко число

TODO: да се премести задачата в Loops Basics упражненията.

Напишете програма, която до получаване на **командата "Stop"**, чете **цели числа**, въведени от потребителя, и намира **най-малкото** измежду тях. Въвежда се по едно число на ред.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 100  99  80  70  Stop | 70 | -10  20  -30  Stop | -30 | 45  -20  7  99  Stop | -20 | 999  Stop | 999 | -1  -2  Stop | -2 |

## Баланс по сметка

Напишете програма, която пресмята колко общо пари има в сметката, след като направите определен брой вноски. На всеки ред ще получавате сумата, която трябва да внесете в сметката, **до получаване на команда "NoMoreMoney"** . При всяка получена сума на конзолата трябва да се извежда **"Increase: "** + сумата и тя да се **прибавя в сметката**. Ако получите число **по-малко от 0** на конзолата трябва да се изведе **"Invalid operation!"** и **програмата да приключи**. Когато програмата приключи трябва да се принтира **"Total: "** + общата сума в сметката закръглена до втория знак след десетичната запетая.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 5.51  69.42  100  NoMoreMoney | Increase: 5.51  Increase: 69.42  Increase: 100  Total: 174.93 | 120  45.55  -150 | Increase: 120  Increase: 45.55  Invalid operation!  Total: 165.55 |

## Завършванe

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е по-голяма или равна на 4.00**. **Ако ученикът бъде скъсан повече от един път, то той бива изключен и програмата приключва,** като се отпечатва **името на ученика и в кой клас бива изключен**.

При успешно завършване на **12-ти** клас да се отпечата :

"{**име на ученика**} graduated. Average grade: {**средната оценка от цялото обучение**}"

**В случай, че ученикът е изключен от училище, да се отпечата:**

"{**име на ученика**} has been excluded at {**класа, в който е бил изключен**} grade"

**Стойността трябва да бъде форматирана до втория знак** след десетичната запетая.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| Gosho  5  5.5  6  5.43  5.5  6  5.55  5  6  6  5.43  5 | Gosho graduated. Average grade: 5.53 | Mimi  5  6  5  6  5  6  6  2  3 | Mimi has been excluded at 8 grade |

## Преместване

На осемнадесетия си рожден ден на Хосе взел решение, че ще се изнесе да живее на квартира. Опаковал багажа си в **кашони** и намерил подходяща обява за апартамент под наем. Той започва да пренася своя багаж **на части**, защото не може да пренесе целия наведнъж. Има ограничено **свободно пространство** в новото си жилище, където може да разположи вещите, така че мястото да бъде подходящо за живеене.

Напишете **програма, която изчислява свободния обем от жилището на Хосе, който остава след като пренесе багажа си.**

**Бележка: Един кашон е с точни размери: 1m. x 1m. x 1m.**

### Вход

Потребителят въвежда следните данни на отделни редове:

* **Широчина на свободното пространство - цяло число в интервала [1...1000]**
* **Дължина на свободното пространство - цяло число в интервала [1...1000]**
* **Височина на свободното пространство - цяло число в интервала [1...1000]**
* На следващите редове **(до получаване на команда "Done") -** **брой кашони, които се пренасят в квартирата - цели числа в интервала [1...10000];**

**Програмата трябва да приключи прочитането на данни при команда "Done"** **или ако свободното място свърши.**

### Изход

Да се **отпечата** на конзолата **един** от следните редове:

* Ако стигнете до командата **"Done"** и има още свободно място:

**"{брой свободни куб. метри} Cubic meters left."**

* Ако свободното място свърши преди да е дошла команда **"**Done**":**

**"No more free space! You need {брой недостигащи куб. метри} Cubic meters more."**

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснение** |
| 10  10  2  20  20  20  20  122 | No more free space! You need 2 Cubic meters more. | 10 \* 10 \* 2 = **200 кубични метра.** 20 + 20 + 20 + 20 + 122 = **202** **кубични метра.** 200 - 202 = **2 недостигащи** кубични метра |
| 10  1  2  4  6  **Done** | 10 Cubic meters left. | 10 \* 1 \* 2 = **20 кубични метра.** 4 + 6 = **10** **кубични метра.** 20 - 10 = **10 кубични метра.** |

## Сума от две числа

Напишете програма която проверява **всички възможни комбинации от двойка числа** **в интервала от две дадени числа**. На изхода се отпечатва, **коя поред е комбинацията** чиито **сбор от числата** **е равен** на дадено **магическо число**. Ако няма **нито една комбинация** отговаряща на условието се отпечатва **съобщение, че не е намерено**.

### Вход

Входът се чете от конзолата и се състои от **три реда**:

* **Първи ред** – **начало на интервала** – **цяло число** в интервала **[1...999]**
* **Втори ред** – **край на интервала** – **цяло число** в интервала **[по-голямо от първото число...1000]**
* **Трети ред – магическото число – цяло число** в интервала **[1...10000]**

### Изход

На конзолата трябва да се отпечата **един ред,** според резултата:

* Ако **е намерена** **комбинация** чиито **сбор на числата е равен на магическото число**
  + "Combination N:{пореден номер} ({първото число} + {второ число} = {магическото число})"
* Ако **не е намерена** комбинация отговаряща на условието
  + "{броят на всички комбинации} combinations - neither equals {магическото число}"

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | **Вход** | **Изход** |
| 1  10  5 | Combination N:4 (1 + 4 = 5) | Всички комбинации от две числа между 1 и 10 са:  1 1, 1 2, 1 3, **1 4**, 1 5, ... 2 1, 2 2, ... 4 9, 4 10, 5 1 ... 10 9, 10 10  Първата комбинация, чиито сбор на числата е равен на магическото число 5 е **четвъртата (1 и 4)** | 88  888  1000 | Combination N:20025 (112 + 888 = 1000) |
| **Вход** | **Изход** | **Обяснения** | **Вход** | **Изход** |
| 23  24  20 | 4 combinations - neither equals 20 | Всички комбинации от две числа между 23 и 24 са: 23 23, 23 24, 24 23, 24 24 (общо 4)  Няма двойки числа, чиито сбор е равен на магическото 20 | 88  888  2000 | 641601 combinations - neither equals 2000 |

## Сграда

Напишете програма, която извежда на конзолата номерата на стаите в една сграда (в низходящ ред), като са изпълнени следните условия:

* На **всеки четен етаж има само офиси**
* На **всеки нечетен етаж има само апартаменти**
* Всеки **апартамент** се означава по следния начин : **А**{**номер на етажа**}{**номер на апартамента**}, **номерата на апартаментите започват от 0.**
* Всеки **офис** се означава по следния начин : **О**{**номер на етажа**}{**номер на офиса**}, **номерата на** **офисите** **също започват от 0.**
* **На последният етаж винаги има апартаменти** и те са по-големи от останалите, за това **пред номера им пише 'L', вместо 'А'. Ако има само един етаж, то има само големи апартаменти!**

От конзолата се прочитат две **цели числа - броят на етажите и броят на стаите за един етаж.**

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| 6  4 | L60 L61 L62 L63  A50 A51 A52 A53  O40 O41 O42 O43  A30 A31 A32 A33  O20 O21 O22 O23  A10 A11 A12 A13 | Имаме общо 6 етажа, с по 4 стаи на етаж. Нечетните етажи имат само апартаменти, а четните само офиси. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 9  5 | L90 L91 L92 L93 L94  O80 O81 O82 O83 O84  A70 A71 A72 A73 A74  O60 O61 O62 O63 O64  A50 A51 A52 A53 A54  O40 O41 O42 O43 O44  A30 A31 A32 A33 A34  O20 O21 O22 O23 O24  A10 A11 A12 A13 A14 | 4  4 | L40 L41 L42 L43  A30 A31 A32 A33  O20 O21 O22 O23  A10 A11 A12 A13 |

1. Проверете дали сумата е равна на max и принтирайте съответният изход.

Text

Description automatically generated

## 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 [500...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).**

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

## Еднакви суми на четни и нечетни позиции

Напишете програма, която чете от конзолата **две шестцифрени цели числа** в диапазона от 100000 до 300000. Винаги **първото** въведено число ще бъде **по малко от второто**. На конзолата да се отпечатат на **1 ред** **разделени с интервал** всички числа, които се намират **между двете**, прочетени от конзолата числа и отговарят на следното **условие**:

* **сумата** от цифрите на **четни** и **нечетни** позиции да са **равни**. Ако няма числа, отговарящи на условието на конзолата не се извежда резултат.

### Примери

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | | | |
| 100000  100050 | 100001 100012 100023 100034 100045 | Първото число, което генерираме е числото 100000. Сумата от цифрите на четни позиции (жълто) е 0+0+0=0. Сумата от цифрите на нечетни позиции (зелено) е 0+0+1=1. Тъй като двете суми са различни числото не се отпечатва.  Следващото, число е 100001. Сумата на четни позиции е 1+0+0=1, а на нечетни 0+0+1=1. Двете суми са равни и числото се отпечатва.  Следващото число за проверка е 100002. То не отговаря на условието и не се отпечатва.  ……  При числото 100045 сумата от четните позиции е 5+0+0=5, а на нечетни 4+0+1=5. Двете суми са равни числото се отпечатва. И т.н. | | | |
| **Вход** | **Изход** | **Вход** | **Изход** | **Вход** | **Изход** |
| 123456  124000 | 123464 123475 123486 123497 123530 123541 123552 123563 123574 123585 123596 123640 123651 123662 123673 123684 123695 123750 123761 123772 123783 123794 123860 123871 123882 123893 123970 123981 123992 | 299900  300000 | 299970 299981 299992 | 100115  100120 | *Няма изход* |

### Насоки

1. Прочетете входните данни от потребителя:



1. За да преминете през всички числа от интервала, направете for цикъл. След като сте прочели входните числа, задайте първото число за начална стойност на контролната променлива**.** Итерирайте до достигане на **второто число** като **увеличавате** стойността на контролната променлива с **1**:

Graphical user interface, text

Description automatically generated

1. Вземете числото на текущата позиция **като текст, като използвате метода** .ToString():

A picture containing chart

Description automatically generated

1. За да обходите всяка цифра от числото, направете **for** цикъл. След като е взето като текст, вземете неговата дължина с .Length. Итерирайте до достигане на **дължината на числото** като **увеличавате** стойността на контролната променлива с **1**:

Text

Description automatically generated

1. Продължете към дописване на логиката за намиране на сумата на четна и нечетна позиция за всяко число. Декларирайте по една променлива за четната и нечетна сума. За да получите точната числова стойност на цифрите използвайте метода int.Parse(). Text

   Description automatically generated
2. За да намерите цифрите, които се намират на четна позиция, използвайте условна **if** конструкция, проверите дали индекса му е четно число като го разделите модулно на 2(**index % 2**), ако е четно, добавете го към сумата на четните, ако не е, към сумата на нечетните.

Graphical user interface, text, application

Description automatically generated

1. След като сте намерили сумата на цифрите на четни и нечетни позиции, проверете дали са равни, ако са, принтирайте числото. Програмата ви трябва да изглежда по следния начин: Graphical user interface, text, application, email

   Description automatically generated