Напишете програма, която чете от конзолата N (цяло число) и след това поредица от N потребителски имена и съхранява колекция само от уникалните. На първия ред ще получите цяло число N. На следващите N реда ще получите име. Отпечатайте колекцията на конзолата (редът няма значение):

Примерни Вход и изход:

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 6  Genadi  Genadi  Genadi  Peter  Genadi  Nikol | Genadi  Peter  Nikol |
| 10  Peter  Maria  Peter  Genadi  Nikol  Maria  Alex  Peter  Nikol  Genadi | Peter  Maria  Genadi  Nikol  Alex |

1. Напишете програма, която отпечатва сет от елементи. На първия ред ще получите две числа m и n, разделени с интервал – представляващи дължините на два отделни сета. На следващите n + m реда ще получавате n числа, които са числата от първия сет, и m числа, които са във втория. Намерете всички уникални елементи и в двата сета. Отпечатайте ги на отделни редове (поредността няма значение).

Например:

Сет n = 4 : {1, 5, 6, 8}

Сет m = 3 : {1, 2, 5}

Накрая отпечатайте трети сет който съдържа елементите които се повтарят от двата сета -> {1, 5}

Примерни вход и изход:

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4 3  1  3  5  7  3  4  5 | 3  5 |
| 2 2  1  3  1  5 | 1 |

1. Напишете програма, която запазва всички уникални бебешки думи. На първия ред ще ви бъде дадено число n - броят на входните редове, които ще получите. На следващите n реда ще получите бебешки думи, разделени с един интервал. Вашата задача е да отпечатате всички уникални на отделни редове (поредността няма значение):

Примерни Вход и Изход:

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  Ce O  Mo O Ce  Ee  Mo | Ce  Ee  Mo  O |
| 3  Ge Ch O Ne  Nb Mo Tc  O Ne | Ch  Ge  Mo  Nb  Ne  O  Tc |

1. Напишете програма, която чете текст от конзолата и брои срещанията на всеки знак в него. Отпечатайте резултатите по азбучен ред.

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Input** | **Output** |
| NPG po KTS - UKTC | : 4 time/s  -: 1 time/s  C: 1 time/s  G: 1 time/s  K: 2 time/s  N: 1 time/s  P: 1 time/s  S: 1 time/s  T: 2 time/s  U: 1 time/s  o: 1 time/s  p: 1 time/s | Why do you like Python so much? | : 6 time/s  ?: 1 time/s  P: 1 time/s  W: 1 time/s  c: 1 time/s  d: 1 time/s  e: 1 time/s  h: 3 time/s  i: 1 time/s  k: 1 time/s  l: 1 time/s  m: 1 time/s  n: 1 time/s  o: 4 time/s  s: 1 time/s  t: 1 time/s  u: 2 time/s  y: 3 time/s |

1. Напишете програма, която намира най-дългото сечение. Ще ви бъде дадено число N. На всеки от следващите N реда ще ви бъдат дадени два диапазона във формат: "{начало}, {край} – на първия и {начало}, {край}" на втория и т.н. Трябва да намерите пресечната точка на тези два диапазона. Началните и крайните числа в диапазоните са включени.

Накрая трябва да намерите най-дългото сечение от всички N пресечки, да отпечатате числата, които са включени, и нейната дължина във формата: „Най-дългото сечение е [{изброени числата от най-дългото сечение}] с дължина {дължина на сечението}“

Примерни входове и изходи:

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  0,3-1,2  2,10-3,5  6,15-3,10 | Longest intersection is [6, 7, 8, 9, 10] with length 5 |
| 5  0,10-2,5  3,8-1,7  1,8-2,4  4,7-2,5  1,10-2,11 | Longest intersection is [2, 3, 4, 5, 6, 7, 8, 9, 10] with length 9 |

1. Уникални имена

Напишете програма която чете цяло число N и след това N пъти поредица от имена от конзолата и сортира само уникалните.

Примерни Вход и Изход:

|  |  |
| --- | --- |
| Вход |  |
| 8  Mariq  Joro  Mariq  Joro  Ani  Ani  Peter  Toni | Ani  Toni  Mariq  Joro  Peter |
| 7  Lili  Gosho  Elena  Ivan  Sasho  Elena  Sasho | Ivan  Lili  Elena  Gosho  Sasho |
| 6  Anna  Anna  Anna  Anna  Anna  Anna | Anna |

1. Стамат има частен паркинг и иска да монтира камера, която да знаснема, номера на колата

която влиза и съответно излиза от паркинга. Помогнете му като напишете софтуер който следи колко коли има в паркинга. На първи ред ще прочетете N цяло число и за N пъти ще прочитате от конзолата номера на автомобили във формат {in}/{out}, {CA2844AA}. На изхода трябва да отпечатате номерата на автомобилите ако има такива. В противен случай „**Parking is Empty“**

**Примерни вход и изход:**

|  |  |
| --- | --- |
| Вход | Изход |
| 10  IN, CA2844AA  IN, CA1234TA  OUT, CA2844AA  IN, CA9999TT  IN, CA2866HI  OUT, CA1234TA  IN, CA2844AA  OUT, CA2866HI  IN, CA9876HH  IN, CA2822UU | CA2844AA  CA9999TT  CA2822UU  CA9876HH |
| 4  IN, CA2844AA  IN, CA1234TA  OUT, CA2844AA  OUT, CA1234TA | Parking Lot is Empty |

1. Управата на училището организира парти за ученици и учители. Всички поканени имат специален код. На първия ред ще получите N цяло число за всеки N броя ще получавате номера на поканите. След това до получаване на команда “End” ще получавате номера на покани на пристигнали гости. Поканите на учителите започват с цифра. Вашата цел е да отпечатате колко души не са дошли и номерата на поканите им, като започнете от учителите.

Примерни вход и изход:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Input** | **Output** |
| 5  7IK9Yo0h  9NoBUajQ  Ce8vwPmE  SVQXQCbc  tSzE5t0p  9NoBUajQ  Ce8vwPmE  SVQXQCbc  END | 2  7IK9Yo0h  tSzE5t0p | 6  m8rfQBvl  fc1oZCE0  UgffRkOn  7ugX7bm0  9CQBGUeJ  2FQZT3uC  2FQZT3uC  9CQBGUeJ  fc1oZCE0  END | 3  7ugX7bm0  UgffRkOn  m8rfQBvl |

1. Ще ви бъде дадена поредица от числа, разделени с интервал. Напишете числата и повторенията им във формата - "{число} - {брой} times". Числото трябва да бъде форматирано до първия десетичен знак.

Примерни входове и изходи:

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| -2.5 4 3 -2.5 -5.5 4 3 3 -2.5 3 -2.5 3 | -2.5 - 4 times  4.0 - 2 times  3.0 - 5 times  -5.5 - 1 times |
| 2 4 4 5 5 2 3 3 4 3 3 4 3 5 3 2 4 3 | 2.0 - 3 times  4.0 - 5 times  5.0 - 3 times  3.0 - 7 times |

1. Напишете програма която чете имената на вашите съученици и техни оценки, и оценките се запазват в досието им. Първо ще получите броя на учениците - n. На следващите “n” редове ще получавате име на ученик и неговата оценка.

Накрая отпечатайте всеки ученик със съответните му/и оценки и средноаритметично оценката, форматирана до втория знак след десетичната запетая. Във формат: „{име на ученик} -> {оценка1} {оценка2} ... {оценкаN} (awg: { средна оценка})". Редът в който са отпечатани различните оценки е без значение!

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 7  Pesho 5.20  Gosho 5.50  Pesho 3.20  Gosho 2.50  Stamat 2.00  Gosho 3.46  Stamat 3.00 | Gosho -> 5.50 2.50 3.46 (avg: 3.82)  Pesho -> 5.20 3.20 (avg: 4.20)  Stamat -> 2.00 3.00 (avg: 2.50) |
| 4  Sasho 4.50  Tedodora 3.00  Sasho 5.00  Tedodora 3.66 | Tedodora -> 3.00 3.66 (avg: 3.33)  Sasho -> 4.50 5.00 (avg: 4.75) |
| 5  Lubo 6.00  Lubo 5.50  Lubo 6.00  Penka 4.40  Kircho 3.30 | Penka -> 4.40 (avg: 4.40)  Lubo -> 6.00 5.50 6.00 (avg: 5.83)  Kircho -> 3.30 (avg: 3.30) |

1. Напишете програма която чете от конзолата поредица от числа разделени със (, )запетая и спейс и ги печата на конзолата в обратен ред, разделени само със спейс – използвайте стек

Пример: 1, 2, 3, 4, 5 -> 5 4 3 2 1 - използвайте стек

1. Напишете програма която извършва различни манипулации със стек, който първоначално е празен.

* На първия ред ще прочетете n-цяло число и след това за n-пъти ще получавате команди:
* 1 {число} – поставете числото най-отзад на стека
* 2 - изтрий числото което е най-отгоре на стека
* 3 – принтирай най-голямото число в стека
* 4 – принтирай най малкото число в стека
* 5 - принтирай дължината на стека

Накрая принтирайте стека като започнете от последния добавен елемент и стигнете до първия

Примерни входове

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  1 97  2  1 20  2  1 26  1 20  5  3  1 91  4 | 2  26  20  91, 20, 26 |
| 11  2  1 47  1 66  1 32  4  3  5  1 25  1 16  1 8  4 | 32  66  3  8  8, 16, 25, 32, 66, 47 |

1. Имате заведение за бързо хранене, като храната, която предлагате е предварително приготвена. Напишете програма, която проверява дали имате достатъчно храна, за да сервирате обяд на всички ваши клиенти. Също така искате да знаете кой е клиентът с най-голяма поръчка за този ден.

* Първо ще ви бъде дадено количеството храна, което имате за деня (цяло число). След това ще ви бъде дадена поредица от цели числа (разделени с един интервал), всяко от които представлява количеството храна във всяка поръчка. Съхранявайте поръчките на опашка.
* Намерете най-голямата поръчка и я отпечатайте. След това ще започнете да обслужвате клиентите си от първия до последния по ред на идване. Преди всяка поръчка проверявайте дали имате достатъчно храна, за да я изпълните:
  + Ако имате, премахнете поръчката от опашката и намалете количеството храна в ресторанта.
  + В противен случай спрете сервирането.

Входни данни:

• На първия ред ще ви бъде дадено количеството на вашата храна - цяло число

• На втория ред ще получите поредица от цели числа, представляващи всяка поръчка, разделени с един интервал

Изходни данни:

• На първия ред отпечатайте количеството на най-голямата поръчка

• На втория ред:

o Ако сте успели да обслужите всичките си клиенти, отпечатайте: „Orders complete“.

o В противен случай отпечатайте: „Оставащи поръчки: {order1}, {order2} .... {orderN}“.

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 348  20 54 30 16 7 9 | 54  Orders complete |
| 499  57 45 62 70 33 90 88 76 100 50 | 100  Orders left: 76, 100, 50 |

1. Вие притежавате моден бутик и получавате доставка едно пале с дрехи, представен като последователност от цели числа. На следващия ред ще ви бъде дадено цяло число, представляващо капацитета на един стелаж във вашия магазин. Трябва да подредите дрехите в магазина и да използвате стелажите, за да подредите всяка дреха. Започвате от най-горните дрехи на палето. Използвайте стек. От всяка дреха има количество (цяло число). Трябва да сумирате техните стойности, докато ги премахвате от палето:

• Ако сборът стане равен на капацитета на текущия стелаж, трябва да започнете нов за следващите дрехи (ако има останали на палето).

• Ако сборът стане по-голям от капацитета, не поставяйте дрехите на текущия стелаж. Вземете нов.

Накрая отпечатайте колко стелажи сте използвали за подреьдане на дрехите.

Входни данни:

• На първия ред ще ви бъде дадена поредица от цели числа, представляващи дрехите на палето, разделени с един интервал.

• На втория ред ще ви бъде дадено цяло число, представляващо капацитета на стелажа.

Изходни данни:

• Отпечатайте броя на стелажите, необходими за подредба на дрехите от палето.

Подсказка:

• Нито едно от целите числа от кутията няма да бъде по-голямо от стойността на капацитета

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5 4 8 6 3 8 7 7 9  16 | 5 |
| 1 7 8 2 5 4 7 8 9 6 3 2 5 4 6  20 | 5 |

1. Напишете програма, която получава текст и го печата като го обръща на обратно. Всяка буква, трябва да е разделена от другата с „-“ тире:

Примерни вход и изход:

* I love Python - > n-o-h-t-y-P- -e-v-o-l- -I
* I study at UKTC - > C-T-K-U- -t-a- -y-d-u-t-s- -I

1. Напишете програма която печата текста във всяка скоба на отделен ред използвайки стек

**Вход:**

**1 + (2 - (2 + 3) \* 4 / (3 + 1)) \* 5**

Изход:

(2 + 3)

(3 + 1)

(2 - (2 + 3) \* 4 / (3 + 1))

1. Гошо е мениджър в Кауфланд, и се нуждае от вашата помощ за да види добре ли се справия новият касиер Стамат на касата. Напишете програма която чете име на клиент и го добавя в края на опашката, докато не получите команда “END”. Но ако получите команда „PAID“, трябва да изпечатате името на всеки клиент на опашката от първия до последния (всеки на нов ред) и да изпразните опашката. Като получите команда „END“ трябва да изпечатате следния текст:

„{останали хора на опашката} people remaining.“

Входни данни 1:

Maria

Neven

Ognqn

PAID

Gosho

Kaloyan

Silvia

Linda

END

Изходни данни 1:

Maria

Neven

Ognqn

4 people remaining.

Входни данни 2:

Ivan

Pesho

Alexander

END

Изходни данни 2:

1. people remaining.
2. Напишете програма, която следи хората, които си взимат хапки от плато и тяхното количеството, останало накрая. На първия ред ще получите началното количество хапки (цяло число) в платото. След това на следващите редове ще ви бъдат дадени имената на някои хора, които искат да си вземат хапки(всеки на отделен ред), докато не получите командата "Start", добавяйте тези хора в опашка. Ще получавате и други две команди. Накрая ще получите командата "End":

- "{хапка}" - (цяло число), които текущият човек в опашката иска да получи. Проверете дали има достатъчно хапки в платото за този човек.

o Ако има достатъчно хапки, отпечатайте „{person\_name} take bites“ и го/я премахнете от опашката.

o В противен случай отпечатайте „{person\_name} must wait“ и премахнете лицето от опашката, без да намалявате хапките в платото.

- "refill {beets}" - добавете дадените хапки в платото.

Накрая отпечатайте колко хапки остават във формат: "остават {left\_liters} bites".

**Входни данни 1:**

2

Peter

Any

Start

2

refill 1

1

End

**Изходни данни 1:**

Peter take bites

Ay take bites

0 bites left

**Входни данни 2:**

10

Peter

George

Any

Stamat

Start

2

3

3

3

End

**Изходни данни 2:**

Peter take bites

George take bites

Any take bites

Stamat must wait

1. bites left
2. Горещ картоф е игра, в която децата образуват кръг и хвърлят горещ картоф. Броенето започва с първото дете. При всяко n-то хвърляне детето, което държи картофа, напуска играта. Когато едно дете напусне играта, то предава картофа на следващото дете(от където започва новото броене). Продължава, докато остане само едно дете.

Вашата задача е да напишете програма, която симулира играта Горещ картоф. На първия ред ще получите имената на децата, разделени с един интервал. На втория ред ще получите n-то хвърляне (цяло число), детето което „изгаря“.Отпечатайте всяко дете, което е премахнато от кръга, във формат „Removed {дете}“. Накрая отпечатайте единственото останало дете във формат „Winner is {дете}“.

Входни данни 1:

Tracy Emilia Daniel

2

Изходни данни 1:

Removed Emilia

Removed Tracy

Last is Daniel

Входни данни 2:

George Peter Michail Stamat Toni

10

Изходни данни 2:

Removed Toni

Removed Peter

Removed Michail

Removed George

Last is Stamat

Входни данни 3:

George Peter Michail Stamat Toni

1

Изходни данни 3:

Removed George

Removed Peter

Removed Michail

Removed Stamat

Last is Toni