**Линейни алгоритми – подготовка за практическо изпитване**

1. Да се състави програма, която изчислява лицето на пълната повърхнина (S) и обема (V) на цилиндър по зададени радиус на основата и височина на цилиндъра. Формулите необходими за изчисленията са следните:, , където е лицето на основата на цилиндъра.
2. Да се намери стойността на променливата x1,x2), където x1 и x2 са различни числа по формулата: maxx1,x2) = 0.5 \* x1  + x2 + | x1 - x2|).
3. Да се състави алгоритъм за изчисляване на лице на триъгълник по Хероновата формула:

1. Да се напише програма , която въвежда положително трицифрено число и извежда на отделни редове цифрите на стотиците, десетиците и на единиците на числото.
2. Да се напише програма, която въвежда положително седемцифрено число и извежда неговата 4– та цифра.
3. Да се напише програма, която въвежда положително седемцифрено число и извежда неговата ***k***– та цифра(1<=k<=7), като цифрите се броят: а/ в посока отляво -надясно; б/ в посока отдясно наляво;
4. **Задача: учебна зала**

**Учебна зала** има правоъгълен размер **l** на **w** метра, без колони във вътрешността си. Залата е разделена на две части – лява и дясна, с коридор - приблизително по средата. В лявата и в дясната част има **редици с бюра**. В задната част на залата има голяма **входна врата**. В предната част на залата има **катедра** с подиум за преподавателя. Едно **работно място** заема **70 на 120 cm** (маса с размер 70 на 40 cm + място за стол и преминаване с размер 70 на 80 cm). **Коридорът** е широк поне **100 cm**. Изчислено е, че заради **входната врата** (която е с отвор 160 cm) **се губи точно 1 работно място**, а заради **катедрата** (която е с размер 160 на 120 cm) се губят точно **2 работни места**. Напишете програма, която въвежда размери на учебната зала и изчислява **броя работни места в нея** при описаното разположение (вж. фигурата).

**Входни данни**

От конзолата се четат **2 числа**, по едно на ред: **l** (дължина в метри) и **w** (широчина в метри).

Ограничения: **3 ≤ w ≤ l ≤ 100**.

**Изходни данни**

Да се отпечата на конзолата едно цяло число: **броят места** в учебната зала.

**Примерен вход и изход**

| **Вход** | **Изход** | **Чертеж** |
| --- | --- | --- |
| 15 8.9 | 129 | Картина, която съдържа текст, екранна снимка, Правоъгълник, линия  Описанието е генерирано автоматично |
| 8.4 5.2 | 39 | Картина, която съдържа текст, екранна снимка, квадрат, Правоъгълник  Описанието е генерирано автоматично |

**Пояснения към примерите**

В първия пример залата е дълга 1500 cm. В нея могат да бъдат разположени **12 реда** (12 \* 120 cm = 1440 + 60 cm остатък). Залата е широка 890 cm. От тях 100 cm отиват за коридора в средата. В останалите 790 cm могат да се разположат по **11 бюра на ред** (11 \* 70 cm = 770 cm + 20 cm остатък). **Брой места = 12 \* 11 - 3** = 132 - 3 = **129** (имаме 12 реда по 11 места = 132 минус 3 места за катедра и входна врата).

Във втория пример залата е дълга 840 cm. В нея могат да бъдат разположени **7 реда** (7 \* 120 cm = 840, без остатък). Залата е широка 520 cm. От тях 100 cm отиват за коридора в средата. В останалите 420 cm могат да се разположат по **6 бюра на ред** (6 \* 70 cm = 420 cm, без остатък). **Брой места = 7 \* 6 - 3** = 42 - 3 = **39** (имаме 7 реда по 6 места = 42 минус 3 места за катедра и входна врата).

#### Идея за решение

Както при всяка една задача по програмиране, е **важно да си изградим идея за решението ѝ**, преди да започнем да пишем код. Да разгледаме внимателно зададеното ни условие. Изисква се да напишем програма, която да изчислява броя работни места в една зала, като този брой е зависим от дължината и височината ѝ. Забелязваме, че те ще ни бъдат подадени като входни данни **в метри**, а информацията за това колко пространство заемат работните места и коридорът, ни е дадена **в сантиметри**. За да извършим изчисленията, ще трябва да използваме еднакви мерни единици, няма значение дали ще изберем да превърнем височината и дължината в сантиметри, или останалите данни в метри. За представеното тук решение е избрана първата опция.

Следва да изчислим **колко колони и колко редици** с бюра ще се съберат. Колоните можем да пресметнем като **от широчината извадим необходимото място за коридора (100 cm)** и **разделим остатъка на 70 cm** (колкото е дължината на едно работно място). Редиците ще намерим като разделим **дължината на 120 cm**. И при двете операции може да се получи **реално число** с цяла и дробна част, **в променлива трябва да запазим обаче само цялата част**. Накрая умножаваме броя на редиците по този на колоните и от него изваждаме 3 (местата, които се губят заради входната врата и катедрата). Така ще получим исканата стойност.

#### Избор на типове данни

От примерните входни данни виждаме, че за вход може да ни бъде подадено реално число с цяла и дробна част, затова не е подходящо да избираме тип **int**, нека за тях използваме **double**. Изборът на тип за следващите променливи зависи от метода за решение, който изберем. Както всяка задача по програмиране, тази също има **повече от един начин на решение**. Тук ще бъдат показани два такива.

1. **Задача: зеленчукова борса**

Градинар продава реколтата от градината си на зеленчуковата борса. Продава **зеленчуци за N лева на килограм** и **плодове за M лева за килограм**. Напишете програма, която **да пресмята приходите от реколтата в евро** (ако приемем, че **едно евро** е равно на **1.94 лв.**).

**Входни данни**

От конзолата се четат **4 числа**, по едно на ред:

* Първи ред – Цена за килограм зеленчуци – число с плаваща запетая.
* Втори ред – Цена за килограм плодове – число с плаваща запетая.
* Трети ред – Общо килограми на зеленчуците – цяло число.
* Четвърти ред – Общо килограми на плодовете – цяло число.

**Ограничения: Всички числа ще са в интервала от 0.00 до 1000.00**

**Изходни данни**

Да се отпечата на конзолата **едно число с плаваща запетая: приходите от всички плодове и зеленчуци в евро**.

**Примерен вход и изход**

| **Вход** | **Изход** |
| --- | --- |
| 0.194 19.4 10 10 | 101 |

**Пояснения към първия пример:**

* Зеленчуците струват: 0.194 лв. \* 10 кг. = **1.94 лв.**
* Плодовете струват: 19.4 лв. \* 10 кг. = **194 лв.**
* Общо: **195.94 лв. = 101 евро**.

| **Вход** | **Изход** |
| --- | --- |
| 1.5 2.5 10 10 | 20.6185567010309 |

**Насоки и подсказки**

Първо ще дадем няколко разсъждения, а след това и конкретни насоки за решаване на задачата, както и съществената част от кода.

**Идея за решение**

Нека първо разгледаме зададеното ни условие. В случая, от нас се иска да пресметнем колко е **общият приход** от реколтата. Той е равен на **сбора от печалбата от плодовете и зеленчуците**, а тях можем да изчислим като умножим **цената на килограм по количеството им**. Входните данни са дадени в лева, а за изхода се изисква да бъде в евро. По условие 1 евро е равно на 1.94 лева, следователно за да получим исканата **изходна стойност, трябва да разделим сбора на 1.94**.

**Избор на типове данни**

След като сме изяснили идеята си за решаването на задачата, можем да пристъпим към избора на подходящи типове данни. Да разгледаме **входа**: дадени са **две цели числа** за общия брой килограми на зеленчуците и плодовете, съответно променливите, които декларираме, за да пазим техните стойности, ще бъдат от тип **int**. За цените на плодовете и зеленчуците е указано, че ще бъдат подадени **две числа с плаваща запетая**, т.е. променливите ще бъдат от тип **double**.

Може да декларираме също две променливи, в които да пазим стойността на печалбата от плодовете и зеленчуците поотделно. Тъй като умножаваме променлива от тип **int** (общо килограми) с такава от тип **double** (цена), резултатът също трябва да бъде от тип **double**. Нека поясним това: по принцип **операторите работят с аргументи от един и същи тип**. Следователно, за да извършим операция като умножение върху два различна типа данни, ни се налага да ги преобразуваме към един и същ такъв. Когато в един израз има типове с различен обхват, преобразуването винаги се извършва към този с най-голям обхват, в този случай това е **double**. Тъй като няма опасност от загуба на данни, **преобразуването е неявно** (implicit) и става автоматично от компилатора.

Като **изход** се изисква също **число с плаваща запетая**, т.е. резултата ще пазим в променлива от тип **double**.

1. **задача: ремонт на плочки**

На площадката пред жилищен блок трябва **да се поставят плочки**. Площадката е с форма **на квадрат със страна N метра**. Плочките са **широки „W“ метра** и **дълги „L“ метра**. На площадката има една пейка с **ширина M метра и дължина O метра**. Под нея не е нужно да се слагат плочки. Всяка плочка се поставя за **0.2 минути**.

Напишете програма, която **чете от конзолата размерите** на **площадката, плочките и пейката** и пресмята **колко плочки са необходими** да се покрие площадката и пресмята **времето за поставяне на всички плочки**.

**Пример: площадка с размер 20 м.** има **площ 400 кв.м.**. **Пейка**, широка **1 м.** и дълга **2 м.**, заема площ **2 кв.м.** Една **плочка** е **широка 5 м.** и **дълга 4 м.** и има **площ = 20 кв.м.** **Площта**, която трябва да се покрие, е **400 – 2 = 398 кв.м.** Необходими са **398 / 20 = 19.90 плочки**. Необходимото **време** е **19.90 \* 0.2 = 3.98 минути.**

**Входни данни**

От конзолата се четат **5 числа**:

* **N – дължината** на **страна** от **площадката** в интервала [**1 … 100**].
* **W – широчината** на една **плочка** в интервала [**0.1 … 10.00**].
* **L – дължината** на една **плочка** в интервала [**0.1 … 10.00**].
* **М – широчината** на **пейката** в интервала [**0 … 10**].
* **О – дължината** на **пейката** в интервала [**0 … 10**].

**Изходни данни**

Да се отпечатат на конзолата **две числа**: **броя плочки**, необходим за ремонта и **времето за поставяне**, всяко на нов ред.

**Примерен вход и изход**

| **Вход** | **Изход** |
| --- | --- |
| 20 5 4 1 2 | 19.9 3.98 |

**Обяснение към примера:**

* **Обща площ** = 20 \* 20 = 400.
* **Площ на пейката** = 1 \* 2 = 2.
* **Площ за покриване** = 400 – 2 = 398.
* **Площ на плочки** = 5 \* 4 = 20.
* **Необходими плочки** = 398 \/ 20 = 19.9.
* **Необходимо време** = 19.9 \* 0.2 = 3.98.

| **Вход** | **Изход** |
| --- | --- |
| 40 0.8 0.6 3 5 | 3302.08333333333 660.416666666667 |

**Насоки и подсказки**

Картина, която съдържа линия, текст, екранна снимка, Паралелен

Описанието е генерирано автоматичноНека да си направим чертеж, за да поясним условието на задачата. Той може да изглежда по следния начин:

**Идея за решение**

Изисква се да пресметнем **броя плочки**, който трябва да се постави, както и **времето, за което това ще се извърши**. За да **изчислим броя**, е необходимо да сметнем **площта, която трябва да се покрие**, и да я **разделим на лицето на една плочка**. По условие площадката е квадратна, следователно общата площ ще намерим като умножим страната ѝ по стойността й **N \* N**. След това пресмятаме **площта, която заема пейката**, също като умножим двете ѝ страни **M \* O**. Като извадим площта на пейката от тази на цялата площадка, получаваме площта, която трябва да се ремонтира.

Лицето на единична плочка изчисляваме като **умножим едната ѝ страна по другата** **W \* L**. Както вече отбелязахме, сега трябва да **разделим площта за покриване на площта на една плочка**. По този начин ще разберем какъв е необходимият брой плочки. Него умножаваме по **0.2** (времето, за което по условие се поставя една плочка). Така вече ще имаме исканите изходни стойности.

**Избор на типове данни**

Дължината на страна от площадката, широчината и дължината на пейката ще бъдат дадени като **цели числа**, следователно, за да запазим техните стойности може да декларираме **променливи от тип int**. За широчината и дължината на плочките ще ни бъдат подадени реални числа (с цяла и дробна част), затова за тях ще използваме **double**. Изходът на задачата отново ще е реално число, т.е. променливите ще бъдат също от тип **double**