# Изпит по "Основи на програмирането" - 28 и 29 март 2020

## Задача 2. Котешка разходка

Деси трябва да разхожда своята котка ежедневно, но не може да прецени колко минути са досатъчни на ден. Като знаете колко калории приема котката на ден, колко пъти и по колко минути Деси разхожда котката си, напишете програма, която изчислява дали разходката е достатъчна. За всяка минута от разходката, котката гори по 5 калории. Разходката е достатъчна, ако котката изграря 50% от приетите калории.

### Вход

Входът се чете от **конзолата** и съдържа **точно 3 реда**:

* На **първия** ред - минути разходка на ден - **цяло число в интервала [1...50]**
* На **втория** ред - броят на разходките дневно - **цяло число в интервала [1…10]**
* На **третия** ред - приетите от котката калории на ден – **цяло число в интервала [100…4000]**

### Изход

Да се **отпечата** на конзолата **един ред**:

* Ако изгорените калории през разходката са повече или равни на 50% от приетите през деня калории:

"Yes, the walk for your cat is enough. Burned calories per day: {общо изгорени калории от разходката}."

* Ако изгорените калории през разходката са по-малко от 50% от приетите през деня калории:

"No, the walk for your cat is not enough. Burned calories per day: {общо изгорени калории от разходката}."

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| 30  3  600 | Yes, the walk for your cat is enough. Burned calories per day: 450. | Котката се разхожда 3 пъти по 30 минути => общо минути разходка -> 30 \* 3 = 90 минути  Общо изгорени калории за един ден от разходки -> 90 \* 5 = 450  50 % от приетите калории на ден са: 50% от 600 = 300  450 > 300 => разходката е достатъчна | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 15  2  500 | No, the walk for your cat is not enough. Burned calories per day: 150. | 40  2  300 | Yes, the walk for your cat is enough. Burned calories per day: 400. |

## Задача 2. Скоростно изкачване

Георги решава да подобри рекорда за най-бързо изкачване на връх Монблан. **На конзолата се въвежда рекордът в секунди, който Георги трябва да подобри, разстоянието в метри, което трябва да изкачи и времето в секунди, за което той изкачва 1 метър.** Да се напише програма, която изчислява дали се е справил със задачата, като се има предвид, че: **наклона на терена го забавя на всеки 50 м. с 30 секунди. Да се изчисли времето в секунди, за което Георги ще изкачи разстоянието до върха и разликата спрямо рекорда.**

**Когато се изчислява колко пъти Георги ще се забави в резултат на наклона на терена, резултатът трябва да се закръгли надолу до най-близкото цяло число.**

### Вход

От конзолата се четат **3 реда**:

1. **Рекордът в секунди – реално число в интервала [0.00 … 100000.00]**
2. **Разстоянието в метри – реално число в интервала [0.00 … 100000.00]**
3. **Времето в секунди, за което изкачва 1 м.** **– реално число в интервала [0.00 … 1000.00]**

### Изход

Отпечатването на конзолата зависи от резултата:

* Ако **Георги е подобрил рекорда** отпечатваме:
  + **"** **Yes! The new record is {времето на Георги} seconds."**
* Ако **НЕ е подобрил рекорда** отпечатваме:
  + **"No! He was {недостигащите секунди} seconds slower."**

**Резултатът трябва да се форматира до втория знак след десетичната запетая**.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10164  1400  25 | No! He was 25676.00 seconds slower. | Георги трябва да изкачи **1400 м**.: **1400 \* 25** = **35000 сек.**  **На всеки** **50 м. към времето му се добавят 30 сек**.:  (**1400 / 50**) \* 30 = **840 сек.**  **Общо време:** 35000 + 840 = **35840 сек.**  **Но понеже** 10164 < 35840, значи не е подобрил рекорда.  **Времето, което не му е стигнало за да подобри рекорда:**  35840 - 10164 = **25676 сек.** |
| **Вход** | **Изход** | **Обяснения** |
| 5554.36  1340  3.23 | Yes! The new record is 5108.20 seconds. | Георги трябва да изкачи **1340 м**.: 1340 \* 3.23 = **4328.20 сек.**  **На всеки** **50 м. към времето му се добавят 30 сек**.:  1340/ 50 = 26.8 -> **закръглено надолу** = 26 пъти се добавят 30 секунди: 26 \* 30 = **780 сек.**  **Общо време:** 4328.20 + 780 = **5108.20 сек.**  **Рекордът е подобрен, понеже** 5108.20 < 5554.36 |
| 1377  389  3 | No! He was 0.00 seconds slower. |  |

## Задача 3. Енергийни гелове

Сезона за изкачване на алпийски върхове започва и всички алпинисти, се запасяват с енергийни гелове за изкачването. Фирма предлага малки и големи разфасовки с по **2 бр. и 5 бр.** енергийни гела,като **цената на един гел зависи от плодовете, от които е направен. В зависимост от размера на разфасовката, цената за брой енергиен гел е различна. От конзолата се четат плодовете, размерът на опаковката ((малка) 2 бр. или (голяма) 5 бр.), както и колко разфасовки са поръчани.** Да се напише програма, **която изчислява сумата, която трябва да се плати** за поръчката.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Диня** | **Манго** | **Ананас** | **Малина** |
| **2 броя (small)** | 56 лв./бр. | 36.66 лв./бр. | 42.10 лв./бр. | 20 лв./бр. |
| **5 броя (big)** | 28.70 лв./бр. | 19.60 лв./бр. | 24.80 лв./бр. | 15.20 лв./бр. |

**При поръчки:**

* **от 400 лв. до 1000 лв. включително има 15% отстъпка**
* **над 1000 лв. има 50% отстъпка**

### Вход

От конзолата се четат **3 реда**:

1. **Плод -** **текст с възможности:** **"Watermelon"**, **"Mango", "Pineapple"** или **"Raspberry"**
2. **Размерът на сета** - **текст с възможности:** **"small"** или **"big"**
3. **Брой на поръчаните сетове** - **цяло число в интервала [1 … 10000]**

### Изход

На конзолата се отпечатва **1 ред**:

* **Цената, която трябва да се заплати**, **форматирана до втория знак след десетичната запетая**,   
  в следния формат:

**"**{цената} lv."

### Примерен вход и изход

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | | | |
| Watermelon  big  4 | 487.90 lv. | **Цената на голям пакет гелове с диня** е 5 \* 28.70 = **143.50 лв.**  **Цената на 4 сета**: **574 лв.**  400 <= 574 <= 1000 -> клиентът получава **15% отстъпка**  15 % от 574 = **86.1 лв. отстъпка**  **Крайна цена**: 574 - 86.1 = **487.9 лв.** | | | |
| **Вход** | **Изход** | **Вход** | **Изход** | **Вход** | **Изход** |
| Pineapple  small  1 | 84.20 lv. | Raspberry  small  50 | 1000.00 lv. | Mango  big  8 | 666.40 lv. |

## Задача 3. Карта за фитнес

Да се напише програма, която проверява дали първоначално налична сума е достатъчна, за да се заплати карта за месечен достъп във фитнес.

**Цената на картата зависи от пола на клиента и спорта**, който практикува:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Пол** | **Gym** | **Boxing** | **Yoga** | **Zumba** | **Dances** | **Pilates** |
| **мъж** | $42 | $41 | $45 | $34 | $51 | $39 |
| **жена** | $35 | $37 | $42 | $31 | $53 | $37 |

Всички цени на карти за ученици (възраст **под 19 години вкл.**) са с **20% намаление**.

**Вход**

От конзолата се прочитат **4 реда**:

* **Сумата, с която разполагаме** - **реално число** в интервала **[10.00…1000.00]**
* **Пол** - **символ** (**'m'** за мъж и **'f'** за жена)
* **Възраст** - **цяло число** в интервала **[5…105]**
* **Спорт - текст** (една от възможностите в таблицата)

**Изход**

На конзолата се отпечатва **1 ред**:

* Ако сумата **е достатъчна**:

**"You purchased a 1 month pass for {sport}."**

където **{sport}** е въведения тип спорт

* Ако сумата **не е достатъчна** трябва да се пресметне **колко още пари** са необходими, за да се закупи карта:

**"You don't have enough money! You need ${money} more."**

където **{money}** e оставащата сума нужна, за да се закупи картата, форматирана до **втория знак след десетичната запетая**.

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 50  m  23  Gym | You purchased a 1 month pass for Gym. | Клиентът е мъж, който иска да закупи карта за Gym, която струва $42, а разполага с $50. Той е на 23 години 🡪 не получава намаление.  $42 <= $50 🡪 Той успява да закупи карта, тъй като има достатъчно пари. |
| 20  f  15  Yoga | You don't have enough money! You need $13.60 more. |  |
| 10  m  50  Pilates | You don't have enough money! You need $29.00 more. |  |

## Задача 4. Храна за домашни любимци

Ани има два домашни любимеца - куче и котка. Напишете програма, която изготвя статистика за **храната на домашните любимци за определен брой дни**. Всеки ден кучето и котката изяждат **различно** количество от общата им храна. На всеки **трети** ден получават награда - **бисквитки**. Количеството на бисквитките е **10%** от общо **изядената** храна **за деня**.

Вашата програма трябва да отпечатва **статистика** за **количеството бисквитки, които са изяли**, **колко процента от първоначалното количество** обща храна са **изяли** и колко процента **от изядената храна** е изяло кучето и колко е изяла котката.

### Вход

Първоначално се чете **един** ред:

* Брой дни – **цяло число** в диапазона **[1…30]**
* Общо количество храна– **реално число** в диапазона **[0.00…10000.00]**

След това **за всеки ден** се чете**:**

* Количество изядена храна от кучето – **цяло число** в диапазона **[10…500]**
* Количество изядена храна от котката – **цяло число** в диапазона **[10…500]**

### Изход

На конзолата да се отпечатват **четири** реда:

* **"Total eaten biscuits: {количество изядени бисквитки}gr."**
* **"{процент изядена храна}% of the food has been eaten."**
* **"{процент изядена храна от кучето}% eaten from the dog."**
* **"{процент изядена храна от котката}% eaten from the cat."**

**Количеството изядени бисквитки** трябва да бъде закръглено до **най – близкото цяло число**, а **процентът** храна трябва да бъде форматиран до **втората цифра след десетичния знак.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  1000  300  20  100  30  110  40 | Total eaten biscuits: 15gr.  60.00% of the food has been eaten.  85.00% eaten from the dog.  15.00% eaten from the cat. | Имаме 3 дена и общо количество храна 1000гр.  На първия ден:  Кучето изяжда 300гр, котката изяжда 20гр.  На втория ден:  Кучето изяжда 100гр, котката изяжда 30гр.  На третия ден:  Кучето изяжда 110гр, котката изяжда 40гр.  На този ден трябва да получат и награда – бисквитки:  10% от 110 + 40 – 15гр.  Общо изядена храна: 600гр.  Изядена храна от кучето: 510гр, от котката: 90гр.  600гр от 1000гр = 60% от храната е изядена.  510гр от 600гр = 85% е изяло кучето.  90гр от 600гр = 15% е изяла котката. |
| 3  500  100  30  110  25  120  35 | Total eaten biscuits: 16gr.  84.00% of the food has been eaten.  78.57% eaten from the dog.  21.43% eaten from the cat. |  |

## Задача 4. 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**– **Мусала**
* Група **от 6 до 12** – **Монблан**
* Група **от 13 до 25** – **Килиманджаро**
* Група **от 26 до 40** – **К2**
* Група **от 41 или повече** – **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** | **Вход** | **Изход** |
| 10  10  5  1  100  12  26  17  37  40  78 | 1.84%  6.75%  5.21%  31.60%  54.60% | Всички хора 10 + 5 + 1 + .... + 78 = 326  6/326\*100 = 1.84% изкачващи Мусала  22/326\*100 = 6.75% изкачващи Монблан  17/326\*100 = 5.21% изкачващи Килиманджаро  103/326\*100 = 31.60% изкачващи К2  178/326\*100 = 54.60% изкачващи Еверест | 5  25  41  31  250  6 | 0.00%  1.70%  7.08%  8.78%  82.44% |

## Задача 5. Грижи за кученце

Ани намира кученце, за което ще се грижи, докато се намери някой да го осинови. То изяжда дневно **определено количество храна**. Да се напише програма, която проверява **дали количеството храна, което е закупено за кученцето, ще е достатъчно докато кученцето бъде осиновено.**

**Вход**

От конзолата се прочитат:

* **Закупеното количество храна за кученцето в килограми** – **цяло число** в интервала **[1 …100]**
* **На всеки следващ ред** до получаване на команда **Adopted** ще получавате **колко грама изяжда кученцето на всяко хранене** - **цяло число** в интервала **[10 …1000]**

**Изход**

На конзолата се отпечатва **1 ред**:

* Ако количеството храна **е достатъчно** да се отпечата:

**"Food is enough! Leftovers: {останала храна} grams."**

* Ако количеството храна **не е достатъчно** да се отпечата:

**"Food is not enough. You need {нужно количество храна} grams more."**

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 4  130  345  400  180  230  120  **Adopted** | Food is enough! Leftovers: 2595 grams. | Закупеното количество храна е 4 кг = 4 \* 1000 = 4000 грама.  **Общото количество храна**, което е изяло кученцето преди да бъде осиновено е:  130 + 345 + 400 + 180 + 230 + 120 = 1405 грама. Това количество е **по-малко** от първоначалното закупено (4000 грама) => остава храна, която е 4000 – 1405 = 2595 грама. |
| 3  1000  1000  1000  **Adopted** | Food is enough! Leftovers: 0 grams. |  |
| 2  999  456  999  999  123  456  **Adopted** | Food is not enough. You need 2032 grams more. |  |

## Задача 5. Товарене на багажи

Напишете програма, която ви помага при товаренето на куфари в багажника на самолет. Всеки самолет има определен **капацитет на багажника**. До получаване на команда **"End"** ще получавате **обем на куфар**. Обемът на **всеки трети куфар** трябва да **се увеличава с 10%,** поради загубата на пространство при начина на подреждане. Ако **свободното пространството** в даден момент **е по-малко от обема на куфар** товаренето трябва да прекъсне.

### Вход

Първоначално се чете **един ред**:

* Капацитетът на багажника – **реално число** в диапазона **[100.0…6000.0]**

След това до получаване на команда **"End" или до запълване на багажника, се чете по един ред:**

* Обем на куфар – **реално число** в диапазона **[100.0…6000.0]**

### Изход

На конзолата да се отпечатат следните **редове** според случая:

* При получаване на командата **"End"** се печата:

**"Congratulations! All suitcases are loaded!"**

* Ако обемът на куфара е по-голям от оставащото пространство в багажника:

**"No more space!"**

* Накрая винаги се отпечатва статистика – колко багажа са натоварени:

**"Statistic: {брой натоварени багажи} suitcases loaded."**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 550  100  252  72  End | Congratulations! All suitcases are loaded!  Statistic: 3 suitcases loaded. | Капацитетът на багажника е 550.  На следващия ред получаваме обем на куфар 100.  От общия обем вадим обема на първия куфар и го товарим. Вече имаме един натоварен куфар. Останало пространство: 550 – 100 = 450.  На следващия ред получаваме обем на куфар 252.  От общия обем вадим обема на втория куфар и го товарим. Имаме втори куфар, който е натоварен успешно.  Останало пространство: 450 – 252= 198.  На следващия ред получаваме обем на куфар 72, тъй като това е **трети куфар трябва да увеличим обема му с 10%** и той става 79.2.  От общия обем вадим обема на третия куфар и го товарим. Имаме трети успешно натоварен куфар. Останало пространство: 198 – 79.2 = 118.8  Получаваме команда **"End"** и принтираме съответния изход. |
| 700.5  180  340.6  126  220 | No more space!  Statistic: 3 suitcases loaded. | Капацитетът на багажника е 700.5  На следващия ред получаваме обем на куфар 180.  От общия обем вадим обема на първия куфар и го товарим.  Останало пространство: 700.5 – 180 = 520.5  Вече имаме един натоварен куфар.  На следващия ред получаваме обем на куфар 340.6  От общия обем вадим обема на втория куфар и го товарим. Останало пространство: 520.5 – 340.6 = 179.9  Имаме втори куфар, който е натоварен успешно.  На следващия ред получаваме обем на куфар 126, тъй като това е **трети куфар трябва да увеличим обема му с 10%** и той става 138.6  От общия обем вадим обема на третия куфар и го товарим.  Останало пространство: 179.9 – 138.6 = 41.3  Имаме трети успешно натоварен куфар.  На следващия ред получаваме обем на куфар 220.  Обемът на куфара е по-голям от оставащото място, съответно приключваме товаренето на куфари.  Имаме 3 успешно натоварени куфара. |
| 1200.2  260  380.5  125.6  305  End | Congratulations! All suitcases are loaded!  Statistic: 4 suitcases loaded. |  |

## Задача 6. Коледен турнир

Напишете програма, която проследява представянето на вашия отбор на благотворителен коледен турнир. Всеки ден получавате **имена на игри** до команда **"Finish"**. Със **спечелването** на всяка една игра печелите по **20лв**. за благотворителност. Трябва да **изчислите колко пари сте спечелили на края на деня**. Ако имате **повече спечелени игри, отколкото загубени** – вие сте победители този ден и **увеличавате** парите от него с **10%**. При **приключване на турнира** ако през повечето дни сте били **победители** печелите турнира и **увеличавате** всичките спечелени пари с **20%**.

**Никога няма да имате равен брой спечелени и загубени игри.**

### Вход

**Първоначално от конзолата се прочита броя дни на турнира – цяло число в интервала [1… 20]**

До получаване на командата **"Finish"** се чете:

* Спорт – **текст**

За всеки спорт се прочита:

* + Резултат – **текст с възможности: "win" или "lose"**

### Изход

Накрая се отпечатва един ред:

* Ако сте **спечелили** турнира:

**"You won the tournament! Total raised money: {спечелените пари}"**

* Ако сте **загубили** на турнира:

**"You lost the tournament! Total raised money: {спечелените пари}"**

**Парите да бъдат форматирани до втората цифра след десетичния знак.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| **2**  volleyball  win  football  lose  basketball  win  **Finish**  golf  win  tennis  win  badminton  win  **Finish** | You won the tournament! Total raised money: 132.00 | Турнирът е 2 дена.  Първият ден:  Играем волейбол и печелим -> 20лв.  Играем футбол и губим -> 0лв.  Играем баскетбол и печелим -> 20лв.  Получаваме команда Finish и игрите за деня приключват. Спечелените пари 20 + 0 + 20 = 40лв. Имаме повече спечелени игри, отколкото загубени съответно увеличаваме парите от деня с 10% -> 44лв.  Втори ден:  Играем голф и печелим -> 20лв.  Играем тенис и печелим -> 20лв.  Играем бадминтон и печелим -> 20лв.  Получаваме команда Finish и игрите за деня приключват. Спечелените пари 20 + 20 + 20 = 60лв. Имаме само спечелени игри съответно увеличаваме парите от деня с 10% -> 66лв.  Спечелените пари от двата дена: 44 + 66 = 110лв.  Тъй като имаме повече победи, отколкото загуби, печелим турнира и увеличаваме парите с 20% -> 132лв. |
| **3**  darts  lose  handball  lose  judo  win  **Finish**  snooker  lose  swimming  lose  squash  lose  table tennis  win  **Finish**  volleyball  win  basketball  win  **Finish** | You lost the tournament! Total raised money: 84.00 |  |