# **Exam preparation**

Exam preparation for the practical exam to the course "**Programming Basics**" course [@ SoftUni Global](https://softuni.org).

**Submit** your solutions to the **SoftUni** **Judge** system at: <https://judge.softuni.org/Contests/3699>

## 1. Fruit Market

Sophie decides to go on a diet and goes to a nearby market to buy strawberries, bananas, oranges, and raspberries. **On the console, the price of strawberries in USD/kg and the amount of bananas, oranges, raspberries, and strawberries she needs to buy are entered. Write a program that calculates how much money she needs to pay the bill,** knowing that:

* **The price of the raspberries is half the price of the strawberries**.
* **The price of the oranges is** **40% lower than the price of the raspberries**.
* **The price of bananas is** **80% lower than the price of raspberries**.

### Input Data

Read **5 lines** from the console:

1. **Price of strawberries in USD – a floating-point number in the range [0.00 … 10000.00]**
2. **The number of bananas in kilograms – a floating-point number in the range [0.00 … 1 0000.00]**
3. **The number of oranges in kilograms – a floating-point number in the range [0.00 … 10000.00]**
4. **The number of raspberries in kilograms – a floating-point number in the range [0.00 … 10000.00]**
5. **The number of strawberries in kilograms – a floating-point number in the range [0.00 … 10000.00]**

### Output Data

Print **one number** to the console:

* **The needed money,** from Sophie for the bill

Format the result to the second digit after the decimal point.

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 48  10  3.3  6.5  1.7 | 333.12 | **Price of the raspberries** **per kilogram**: **24** USD.  **Price of the oranges per kilogram**: 24 – (0.4 \* 24) = **14.4** USD.  **Price of the bananas per kilogram**: 24 – (0.8 \* 24) = **4.8** USD.  **Sum for the raspberries**: 6.5 \* 24 = **156** USD.  **Sum for the oranges**: 3.3 \* 14.4 = **47.52** USD.  **Sum for the bananas**: 4.8 \* 10 = **48** USD.  **Sum for the strawberries**: 1.7 \* 48 = **81.6** USD.  **Total sum**: 156 + 47.52 + 48 + 81.6 = **333.12** USD. |
| **Input** | **Output** |  |
| 63.5  3.57  6.35  8.15  2.5 | 561.15 |  |

### Testing in the Judge System

**Test** your solution in the online **Judge** system: <https://judge.softuni.org/Contests/Compete/Index/3699#0>

## 2. Family Trip

The Smiths family is planning their family vacation. Your task is to write a program that **calculates** whether their **budget** will be **enough**, knowing how many **nights** they have planned, what is the **price per night** and what **percentage of the budget** they have provided for **additional costs** You also know that if the **number** of nights is **more than 7**, the **price per night** is **reduced by 5%.**

### Input Data

Read **4 lines** from the console:

* **The budget they have – a floating-point number in the range [1.00 … 10000.00]**
* **The number of nights – integer in the range [0 … 1000]**
* **Price per night – a floating-point number in the range [1.00 … 500.00]**
* **The percentage for additional costs – integer in the range [0 … 100]**

### Output Data

Printing the console depends on the result:

* If the **sum is enough**:
  + **"Smiths will be left with {remaining money after the trip} USD after vacation."**
* If the **sum is NOT enough** :
  + **"{the money needed for the trip} USD needed."**

Format the sum to the second digit after the decimal point.

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 800.50  8  100  2 | Smiths will be left with 24.49 USD after vacation. | 8 nights > 7, therefore they have 5% reductions of the price per night, which is 100 USD. 5% out of 100 is 5 USD. -> 100 – 5 -> 95 USD. per night after the discount, 8 nights times 95 USD. -> 760 USD.  2% of the budget is provided for additional costs. 2% out of 800.50 is 16.01 USD.  760 + 16.01 = 776.01 <= 800.50 therefore the money is enough for them and after the trip they return with 800.50 – 776.01 = 24.49 USD. |
| **Input** | **Output** | **Comments** |
| 500  7  66  15 | 37.00 USD needed. | 7 nights, no discount  7 nights times 66 USD -> 462 USD.  15% of the budget is provided for additional costs. 15% out of 500 is 75 USD.  462 + 75 = 537 > 500, therefore the money is not enough.  The needed amount is: 537 – 500 = 37 USD. |

### Testing in the Judge System

**Test** your solution in the online **Judge** system: <https://judge.softuni.org/Contests/Compete/Index/3699#1>

## 3. Movie Destination

The director of a major film production wants to know if the budget allocated to him will be enough to shoot the film. Help him by **writing a program** that calculates **how much it will cost him to make the film**, knowing **how much a day of filming costs**. The price for one day is determined by the **season** and the **destination**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Destination**  **Season** | **Dubai** | **Washington** | **London** |
| **Winter** | 45 000 USD. | 17 000 USD. | 24 000 USD. |
| **Summer** | 40 000 USD. | 12 500 USD. | 20 250 USD. |

**The following taxations/reliefs exist:**

* **If the destination is Dubai – 30% discount from the total price**
* **If the destination is Washington – the price increases by 25%**

### Input Data

Read **4 lines** from the console:

1. **Movie budget** – **a floating-point number in the range [100 000.0… 2 000 000.0]**
2. **Destination** – **text**, can be "**Dubai", "** **Washington" and "London"**
3. **Season** – **text**, can be "**Summer"** or "**Winter"**
4. **Number of days** – **integer** in the range**[1… 40]**

### Output Data

Print **one line** to the console:

* If the budget **is enough**:

"**The budget for the movie is enough! We have {remaining budget} USD left!**"

* If the budget **is NOT enough**:

"**The director needs {needed sum} USD more!**"

Format the result to the second digit after the decimal point.

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 400000  Washington Winter  20 | The director needs 25000.00 USD more! | The destination in **Washington** in **Winter =>** the price per day is **17000 USD.**  **20** days \* **17000** = **34000 USD.**  The destination is **Washington, which means that we have a 25% price increase=> 34000 + 25% = 425000 USD.**  Budget 400000 - 425000 => **is NOT enough**  **The needed money is 25000 USD.** |
| 1000000  Dubai  Summer  5 | The budget for the movie is enough! We have 860000.00 USD left! |  |
| 200000  London  Summer  7 | The budget for the movie is enough! We have 58250.00 USD left! |  |

### Testing in the Judge System

**Test** your solution in the online **Judge** system: <https://judge.softuni.org/Contests/Compete/Index/3699#2>

## 4. Food for Pets

Sophie has two pets - a dog and a cat. Write a program that prepares statistics on **pet food for a certain number of days**. Every day the dog and the cat eat a **different amount** of their shared food. Every **third** day they receive a prize - **cookies**. The amount of cookies is **10%** of the total food **eaten** for the **day**. Your program should print **statistics** on the **number of cookies they ate, what percentage of the original amount** of total food they **ate**, and what **percentage of the food** the dog ate, and how much the cat ate.

### Input Data

At the start, read **one line** from the console:

* Number of days – **integer** in the range **[1…30]**
* The total amount of food– **a floating-point number** in the range **[0.00…10000.00]**

After that, for **each day** read:

* The amount of eaten food by the dog – **integer** in the range **[10…500]**
* The amount of eaten food by the cat – **integer** in the range **[10…500]**

### Output Data

Print **four lines** to the console:

* **"Total eaten biscuits: {amount of eaten biscuits}gr."**
* **"{percentage of eaten food}% of the food has been eaten."**
* **"{percentage of eaten food by the dog}% eaten from the dog."**
* **"{percentage of eaten food by the cat}% eaten from the cat."**

The **number of cookies eaten** must be rounded to the **nearest whole number**, and the **percentage** of food must be formatted to the **second digit after the decimal** **point**.

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 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. | We have 3 days and total food of 1000gr.  On the first day:  The dog eats 300gr, the cat eats 20gr.  On the second day:  The dog eats 100gr, the cat eats 30gr.  On the third day:  The dog eats 110gr, the cat eats 40gr.  On this day they should also receive a reward - cookies  10% out of 110 + 40 – 15gr.  Total eaten food: 600gr.  Eaten food by the dog: 510gr, by the cat: 90gr.  600gr out of 1000gr = 60% of the food is eaten.  510gr out of 600gr = 85% was eaten by the dog.  90gr out 600gr = 15% was eaten by the cat. |
| 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. |  |

### Testing in the Judge System

**Test** your solution in the online **Judge** system: <https://judge.softuni.org/Contests/Compete/Index/3699#3>

## 5. Easter Eggs Battle

On Easter, Sophie’s family gathers and she decides to organize a "battle" between Easter eggs. The rules of the "battle" are as follow:

* Two players participate
* Each of them starts with a certain number of eggs
* **When receiving the command "one" -> the first player wins => the eggs of the second decrease by one**
* **When receiving the command "two" -> the second player wins => the eggs of the first decrease by one**
* **The game ends if one of the players runs out of eggs or until the "End" command is received**

### Input Data

At the start, read **two lines** from the console:

1. **Number of eggs the first player has - integer in the range [1 … 99]**
2. **Number of eggs the second player has - integer in the range [1 … 99]**

Then, until the **"End" command** is received, **one line** is read repeatedly:

1. **Winner – text - "one" or "two"**

### Output Data

**If the first player runs out of eggs:**

* **"Player one is out of eggs. Player two has {number of remaining eggs of the second player} eggs left."**

**If the second player runs out of eggs:**

* **"Player two is out of eggs. Player one has {number of remaining eggs of the first player} eggs left."**

**When the "End" is received, print two lines to the console:**

* **"Player one has {number of remaining eggs of the first player} eggs left."**
* **"Player two has {number of remaining eggs of the second player} eggs left."**

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 5  4  one  two  one  two  two  End | Player one has 2 eggs left.  Player two has 2 eggs left. | **Player 1 has 5 eggs. Player 2 has 4 eggs.**  We receive one -> The first one wins and the eggs of the second one are 4 – 1 = 3  We receive two -> The second wins, and the eggs of the first are 5 – 1 = 4  We receive one -> The first one wins and the eggs of the second one are 3 – 1 = 2  We receive two -> The second wins, and the eggs of the first are 4 – 1 = 3  We receive two -> The second wins, and the eggs of the first are 3 – 1 = 2  **End of the battle => The first player has 2 eggs. The second player has 2 eggs.** |
| 2  6  one  two  two | Player one is out of eggs. Player two has 5 eggs left. | **Player 1 has 2 eggs. Player 2 has 6 eggs.**  We receive one -> The first one wins and the eggs of the second one are 6 – 1 = 5  We receive two -> The second wins, and the eggs of the first are 2 – 1 = 1  We receive two -> The second wins, and the eggs of the first are 1 – 1 = 0  **End of the battle => The first player has no eggs. The second player has 5 eggs.** |
| 6  3  one  two  two  one  one | Player two is out of eggs. Player one has 4 eggs left. | **Player 1 has 6 eggs. Player 2 has 3 eggs.**  We receive one -> The first one wins and the eggs of the second one are 3 – 1 = 2  We receive two -> The second wins, and the eggs of the first are 6 – 1 = 5  We receive two -> The second wins, and the eggs of the first are 5 – 1 = 4  We receive one -> The first one wins and the eggs of the second one are 2 – 1 = 1  We receive one -> The first one wins and the eggs of the second one are 1 – 1 = 0  **End of battle => The second player has no eggs. The first player has 4 eggs.** |

### Testing in the Judge System

**Test** your solution in the online **Judge** system: <https://judge.softuni.org/Contests/Compete/Index/3699#4>

## 6. Barcode Generator

The equipment in a Christmas decoration shop has broken down. Items that contain even numbers in their barcode cannot be marked by the cashiers. Your task is to write a program that generates all barcodes that do NOT contain even numbers.

### Input Data

Two four-digit numbers, which show the range of the barcodes, which you’ll need to change.

* **First line** – a four-digit number – the start of the range. **Integer in the range [1000…9999]**
* **Second line** – a four-digit number – end of the range. **Integer in the range [1000…9999]**

### Output Data

**All "barcodes"** that do **NOT contain an even number**, separated by a **space**, must be printed on the console.

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 2345  6789 | 3355 3357 3359 3375 3377 3379 3555 3557 3559 3575 3577 3579 3755 3757 3759 3775 3777 3779 5355 5357 5359 5375 5377 5379 5555 5557 5559 5575 5577 5579 5755 5757 5759 5775 5777 5779 | We take the **first, second, third, and fourth** digits of the two **four-digit numbers**, respectively from the first as the initial and the second as the final value. We check in the intervals between each of the digits of the first four-digit number and each of the digits of the second whether there is an even digit and if it exists we ignore it in the result. |
| 3256  6579 | 3357 3359 3377 3379 3557 3559 3577 3579 5357 5359 5377 5379 5557 5559 5577 5579 | |
| 1365  5877 | 1375 1377 1575 1577 1775 1777 3375 3377 3575 3577 3775 3777 5375 5377 5575 5577 5775 5777 | |

### Testing in the Judge System

**Test** your solution in the online **Judge** system: <https://judge.softuni.org/Contests/Compete/Index/3699#5>