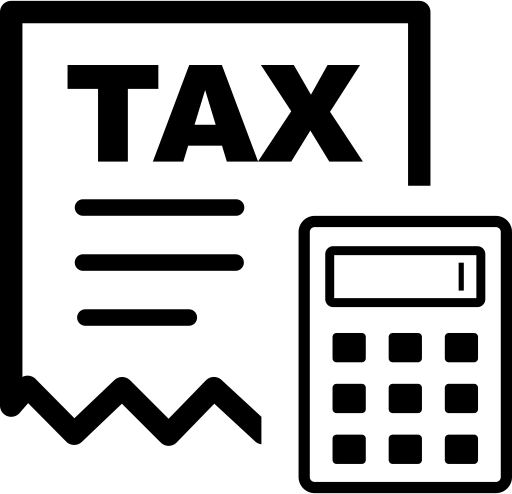
# Problem 2 - Tax Calculator



*The National Revenue Agency hired you to create software, which will help them to calculate the vehicle taxes.*

You will be given **a string representing vehicles that will be taxed**. Each vehicle is separated by **">>",** where the first element is a string representing the **car type**, the second element is an integer representing the **years the tax should be paid**, and the third element is an integer representing the **kilometers** traveled.

There are three valid types of vehicles:

* **"family" –** the initial tax for a family car is **50 euros**
* **"** **heavyDuty" –** the initial tax for a heavy-duty is **80 euros**
* **"sports" –** the initial tax for a sports car is **100 euros**

If the car is not valid print **"Invalid car type."** and continue to the next vehicle.

When calculating tax keep in mind the following rules:

* For a **family** car, the tax declines by **5 euros** for every year in use. Also, the tax increases by **12 euros** for every **3000 km.** traveled.
* For a **heavyDuty** car, the tax declines by **8 euros** for every year in use. Also, the tax increases by **14 euros** for every **9000 km.** traveled.
* For a **sports** car, the tax declines by **9 euros** for every year in use. Also, the tax increase by **18 euros** for every **2000 km.** Traveled.

### Input

You receive a **string** representing the vehicles, separated with "**>>"**: **"vehicle1>>vehicle2>>vehicle3…"**.

### Output

* Upon every successful taxed car print: **"A {car type} car will pay {total tax to pay} euros in taxes."** Format the **total tax to pay** to the **second** digit after the decimal point.
* On the last line, print how much the National Revenue Agency will collect: **"The National Revenue Agency will collect {total tax collected} euros in taxes."** Formatted to the **second digit** after the decimal point.

### Examples

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **family 3 7210>>van 4 2345>>heavyDuty 9 31000>>sports 4 7410** | **A family car will pay 59.00 euros in taxes.**  **Invalid car type.**  **A heavyDuty car will pay 50.00 euros in taxes.**  **A sports car will pay 118.00 euros in taxes.**  **The National Revenue Agency will collect 227.00 euros in taxes.** |
| ****Comment**** | |
| We start looping through the array, the first car is a **family** car, which is **3 years** in use and has **traveled 7210 km**.  3000 is contained 2 times in 7210.  The taxes are calculate as follows: 2 \* 12 + (50 - 3 \* 5) = **59.00 euros**  **The family car must pay 59.00 euros in taxes.**  **The next car is a van, which is an invalid car type.**  **Next, we have heavyDuty car, with is 9 years in use and** has **traveled 31000 km. The tax which heavyDuty car should pay is 50.00 euros.**  **On the last iteration, we have a sports car that is 4 years in use and** has **traveled 7410 km. The tax which the sports car should pay is 118.00 euros.**  **At the end the National Revenue Agency collected 59.00 + 50.00 + 118.00 = 227.00 euros in taxes.** | |
| **Input** | **Output** |
| **family 5 3210>>pickUp 1 1345>>heavyDuty 7 21000>>sports 5 9410>>family 3 9012** | **A family car will pay 37.00 euros in taxes.**  **Invalid car type.**  **A heavyDuty car will pay 52.00 euros in taxes.**  **A sports car will pay 127.00 euros in taxes.**  **A family car will pay 71.00 euros in taxes.**  **The National Revenue Agency will collect 287.00 euros in taxes.** |

**JS Examples**

The input will be an array with a **string**.

|  |  |
| --- | --- |
| **Input** | **Output** |
| ([ 'family 3 7210>>van 4 2345>>heavyDuty 9 31000>>sports 4 7410' ]) | **A family car will pay 59.00 euros in taxes.**  **Invalid car type.**  **A heavyDuty car will pay 50.00 euros in taxes.**  **A sports car will pay 118.00 euros in taxes.**  **The National Revenue Agency will collect 227.00 euros in taxes.** |
| **Comments** | |
| We start looping through the array, the first car is a **family** car, which should pay taxes for **3 years** in use and has **traveled 7210 km**.  3000 is contained 2 times in 7210.  The taxes are calculate as follows: 2 \* 12 + (50 - 3 \* 5) = **59.00 euros**  **The family car must pay 59.00 euros in taxes.**  **The next car is a van, which is an invalid car type.**  **Next, we have heavyDuty car, with 9 years in use, and** has **traveled 31000 km. The tax which heavyDuty car should pay is 50.00 euros.**  **On the last iteration, we have a sports car that is 4 years in use and** has **traveled 7410 km. The tax which the sports car should pay is 118.00 euros.**  **At the end the National Revenue Agency collected 59.00 + 50.00 + 118.00 = 227.00 euros in taxes.** | |
| **Input** | **Output** |
| ([ 'family 5 3210>>pickUp 1 1345>>heavyDuty 7 21000>>sports 5 9410>>family 3 9012' ]) | **A family car will pay 37.00 euros in taxes.**  **Invalid car type.**  **A heavyDuty car will pay 52.00 euros in taxes.**  **A sports car will pay 127.00 euros in taxes.**  **A family car will pay 71.00 euros in taxes.**  **The National Revenue Agency will collect 287.00 euros in taxes.** |