# Programming Fundamentals with Python: Exam Preparation

## Computer Store

**Submit your solutions in the SoftUni judge system at** [**https://judge.softuni.org/Contests/Practice/Index/2517#0**](https://judge.softuni.org/Contests/Practice/Index/2517#0)**.**

Write a program that **prints you a receipt** for your new computer. You will receive the **parts' prices (without tax)** until you receive what type of customer this is - **special** or **regular**. Once you receive the type of customer you should print the receipt.

The **taxes are 20%** of each part's price you receive.

If the customer is **special**, he has a 10% discount on the total price with taxes.

If a given price is not a positive number, you should print **"Invalid price!"** on the console and continue with the next price.

If the total price is equal to zero, you should print **"Invalid order!"** on the console.

### Input

* You will receive numbers representing **prices (without tax)** until command **"special"** or **"regular":**

### Output

* The receipt should be in the following format:

**"Congratulations you've just bought a new computer!**

**Price without taxes: {total price without taxes}$**

**Taxes: {total amount of taxes}$**

**-----------**

**Total price: {total price with taxes}$"**

**Note: All prices should be displayed to the second digit after the decimal point! The discount is applied only on the total price. Discount is only applicable to the final price!**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1050  200  450  2  18.50  16.86  special | Congratulations you've just bought a new computer!  Price without taxes: 1737.36$  Taxes: 347.47$  -----------  Total price: 1876.35$ |
| **Comment** | |
| 1050 – valid price, total 1050  200 – valid price, total 1250  …  16.86 – valid price, total 1737.36  We receive **special**  Price is positive number, so it is valid order  Price without taxes is 1737.36  Taxes: 20% from 1737.36 = 347.47  Final price = 1737.36 + 347.47 = 2084.83  Additional 10% discount for special customers  2084.83 – 10% = 1876.35 | |
| **Input** | **Output** |
| 1023  15  -20  -5.50  450  20  17.66  19.30  regular | Invalid price!  Invalid price!  Congratulations you've just bought a new computer!  Price without taxes: 1544.96$  Taxes: 308.99$  -----------  Total price: 1853.95$ |
| regular | Invalid order! |

## Array Modifier

**Submit your solutions in the SoftUni judge system at** [**https://judge.softuni.org/Contests/Practice/Index/2474#1**](https://judge.softuni.org/Contests/Practice/Index/2474#1)**.**

You are given **an array with integers**. Write a program to **modify the elements** after **receiving the following commands**:

* "swap {index1} {index2}" takes **two elements** and **swap their places**.
* "multiply {index1} {index2}" takes **element at the 1st index** and **multiply** **it** **with the element at 2nd index**. **Save the product at the 1st index.**
* "decrease" **decreases** **all elements** in the array **with 1**.

### Input

On the **first input line,** you will be given **the initial array values** separated by a single space.

On the **next lines** you will receive commands **until** you receive the **command "end"**. The **commands are** as follow:

* "swap {index1} {index2}"
* "multiply {index1} {index2}"
* "decrease"

### Output

**The output** should be printed on the console and consist of **elements** **of the** **modified array** – **separated by a comma and a single space** "**,** ".

### Constraints

* **Elements of the array** will be **integer numbers** in the range **[-231...231]**
* **Count of the array elements** will be in the range **[2...100]**
* **Indexes** **will be always in the range of the array**

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** | | **Comments** | |
| 23 -2 321 87 42 90 -123  swap 1 3  swap 3 6  swap 1 0  multiply 1 2  multiply 2 1  decrease  end | 86, 7382, 2369942, -124, 41, 89, -3 | | 23 -2 321 87 42 90 -123 – initial values  swap 1(-2) and 3(87) ▼  23 87 321 -2 42 90 -123  swap 3(-2) and 6(-123) ▼  23 87 321 -123 42 90 -2  swap 1(87) and 0(23) ▼  87 23 321 -123 42 90 -2  multiply 1(23) 2(321) = 7383 ▼  87 7383 321 -123 42 290 -2  multiply 2(321) 1(7383) = 2369943 ▼  87 7383 2369943 -123 42 90 -2  decrease – all - 1 ▼  86 7383 2369942 -124 41 89 -3 | |
| 1 2 3 4  swap 0 1  swap 1 2  swap 2 3  multiply 1 2  decrease  end | 1, 11, 3, 0 | |  | |
| [  '1 2 3 4',  'swap 0 1',  'swap 1 2',  'swap 2 3',  'multiply 1 2',  'decrease',  'end'  ] | | 1, 11, 3, 0 | |  |

## Moving Target

**Submit your solutions in the SoftUni judge system at** [**https://judge.softuni.org/Contests/Practice/Index/2305#2**](https://judge.softuni.org/Contests/Practice/Index/2305#2)**.**

You are at the shooting gallery again, and you need a program that helps you keep track of moving targets. On the first line, you will receive a **sequence of targets with their integer values**, split by a **single space**. Then, you will start receiving **commands for manipulating the targets** until the **"End"** command. The commands are the following:

* **"Shoot {index} {power}"**
  + Shoot the target at the index **if it exists** by **reducing** its **value** by the **given** **power** (**integer value**).
  + Remove the target **if it is shot**. A target is considered **shot** when **its value reaches 0**.
* **"Add {index} {value}"**
  + Insert a target with the received value at the received **index if it exists**.
  + If not, print: **"Invalid placement!"**
* **"Strike {index} {radius}"**
  + **Remove** the target at the given **index** and **the ones before and after it** depending on the **radius**.
  + If **any of the indices** in the range is **invalid**, print: **"Strike missed!"** and **skip** this command.

**Example:** **"Strike 2 2"**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | {radius} | {radius} | {strikeIndex} | {radius} | {radius} |  |  |

* **"End"**
  + **Print** the sequence with targets in the following format and **end** **the program**:

**"{target1}|{target2}…|{targetn}"**

### Input / Constraints

* On the **first line,** you will receive **the sequence of targets** – **integer values [1-10000]**.
* On the **following lines,** until the **"End"** will be receiving the command described above – **strings**.
* There will never be a case when the **"Strike"** command would empty the whole sequence.

### Output

* Print the appropriate message in case of any command if necessary.
* In the end, print the sequence of targets in the format described above.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 52 74 23 44 96 110  Shoot 5 10  Shoot 1 80  Strike 2 1  Add 22 3  End | Invalid placement!  52|100 | The first command is "**Shoot**", so we reduce the target on **index** **5**, which is valid, with the given **power** – **10**.  Then we receive the same command, but we need to reduce the target on the 1st index, with power 80. The value of this target is 74, so it is considered shot, and we **remove** it.  Then we receive the "**Strike**" command on the 2nd index, and we need to check if the range with radius 1 is valid:  **52 23 44 96 100**  And it is, so we **remove** the targets.  At last, we receive the "**Add**" command, but the index is **invalid**, so we print the appropriate **message**, and in the end, we have the following result:  **52|100** |
| 1 2 3 4 5  Strike 0 1  End | Strike missed!  1|2|3|4|5 |  |