# Exercise: Basic Syntax, Conditional Statements and Loops

Problems for exercise and homework for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3450/programming-fundamentals-with-python-september-2021).

Submit your solutions in the SoftUni judge system at <https://judge.softuni.org/Contests/1719>.

## Jenny's Secret Message

Jenny studies programming with Python and wants to create a program that **greets users** when they give their **names.** The greeting should be in a format **"Hello, {name}!**". However, Jenny is in love with **Johnny** and would like to **greet him differently: "Hello, my love!"**. Could you help her?

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter | Hello, Peter! |
| Amy | Hello, Amy! |
| Johnny | Hello, my love! |

## Drink Something

**Kids** drink **toddy**, **teens** drink **coke**, **young adults** drink **beer**, and **adults** drink **whisky**. Create a program that receives an age and prints what they drink.

**Rules:**

**A kid** is defined as someone **under** **the age of** **14**.

**A teen** is defined as someone **under** **the age of 18**.

**A young adult** is defined as someone **under** **the age of 21**.

**An adult** is defined as someone **above the age of 21**.

***Note:*** All the values are **inclusive** except the **last** **one**!

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 13 | drink toddy |
| 17 | drink coke |
| 21 | drink beer |
| 30 | drink whisky |

## Leonardo DiCaprio Oscars

Write a program that receives a **single integer** number and prints **different messages** depending on the number:

* If the number **is** **88** - **"Leo finally won the Oscar! Leo is happy".**
* If the number **is** **86** - **"Not even for Wolf of Wall Street?!"**
* If the number is **not** **88** **nor** **86** (and **below 88**) - **"When will you give Leo an Oscar?"**
* If the number is **over 88** - **"Leo got one already!"**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 88 | Leo finally won the Oscar! Leo is happy |
| 86 | Not even for Wolf of Wall Street?! |
| 81 | When will you give Leo an Oscar? |
| 89 | Leo got one already! |

## Double Char

You will be given a **string**. You should **print** a **string** in which **each character** (case-sensitive) is **repeated twice**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Hello World | HHeelllloo WWoorrlldd |
| 1234! | 11223344!! |

## Can't Sleep? Count Sheep

If you can't fall asleep, try counting sheep! You will be given a **positive integer**, 3, for example. You should return a string with a murmur: **"1 sheep...2 sheep...3 sheep..."**. Input will always be valid, i.e., integers greater than 0.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1 sheep...2 sheep...3 sheep...4 sheep...5 sheep... |
| 1 | 1. sheep... |

## Next Happy Year

You are saying goodbye to your best friend: "***See you next happy year"***. Happy Year is the year with only **distinct digits**, for example, 2018. Write a program that receives an integer number and finds the next happy year.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8989 | 9012 |
| 1001 | 1023 |

## Maximum Multiple

On the first line, you will be given a positive number, which will serve as a **divisor**. On the second line, you will receive a positive number that will be the **boundary**. You should find the **largest** integer **N**, that is:

* **divisible by the given divisor**
* **less than or equal to the given bound**
* **greater than 0**

***Note:*** it is guaranteed that **N** **is found**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  7 | 6 |
| 10  50 | 50 |
| 37  200 | 185 |

## \* Mutate Strings

You will be given **two strings**. **Transform the first** string into **the second** one, **letter** by letter.Print only the **unique** strings.

***Note:*** the strings will have the same lengths.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| bubble gum  turtle hum | tubble gum  turble gum  turtle gum  turtle hum |
| Kitty  Doggy | Ditty  Dotty  Dogty  Doggy |

## \* Easter Bread

*Since it is Easter, you have decided to make some loaves of Easter* *bread and exchange them for eggs.*

Create a program that **calculates** how many **loaves** you can make with the **budget** you **have**.

**First**, you will **receive** your **budget**. Then, you will **receive** the **price** for **1 kg flour**. Here is the **recipe** for **one** bread:

|  |  |
| --- | --- |
| **Eggs** | **1 pack** |
| **Flour** | **1 kg** |
| **Milk** | **0.250 l** |

The **price for 1 pack of eggs** is **75%** of the **price** **for 1 kg flour**. The **price** for **1l** **milk** is **25%** **more** than the price for **1 kg flour**. Notice that you need **0.250l milk** for **one** bread, and the calculated price is for **1l**.

**Start** cooking the loaves and **keep making** them until you have **enough budget**. Keep in mind that:

* For **every** bread that you make, you will receive **3 colored eggs**.
* For **every** **3rd** bread you make, you will lose some of your **colored** eggs **after receiving** the usual **3 colored eggs** for your bread. The count of eggs you will lose is calculated when you **subtract** **2** from your **current** **count** of loaves – **({current\_bread\_count} – 2)**

In the end, print the loaves of bread you made, the eggs you have gathered, and the money you have **left**, **formatted** to the **2nd decimal place**, in the following format:

**"You made {number\_of\_bread} loaves of Easter bread! Now you have {colored\_eggs} eggs and {money\_left}BGN left."**

### Input / Constraints

* On the **1st line**, you will receive the budget – a **real number** in the range [0.0…100000.0]
* On the **2nd line**, you will receive the price for **1 kg flour** – a **real number** in the range [0.0…100000.0]
* The input will always be in the correct format
* You will **always** have a **remaining** **budget**
* There will **not** be a case in which the **eggs** become a **negative** **count**

### Output

* In the end, print the **number** of **Easter bread** you have made, the colored **eggs** you have gathered, and the **money** **formatted** to **the 2nd** decimal place in the format described above.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 20.50  1.25 | You made 7 loaves of Easter bread! Now you have 16 eggs and 2.45BGN left. |
|  | |
| 15.75  1.4 | You made 5 loaves of Easter bread! Now you have 14 eggs and 1.31BGN left. |

## \* Christmas Spirit

*It is time to get in a Christmas mood. You need to decorate the house in time for the big event, but you have limited days to do so.*

You will receive an **allowed quantity** for **one type** of decoration and **days** **left** until Christmas day to decorate the house.

There are **4 types** of decorations, and each piece costs a **price**

* Ornament Set – 2$ per piece
* Tree Skirt – 5$ per piece
* Tree Garlands – 3$ per piece
* Tree Lights – 15$ per piece

Every **second day** you buy an **Ornament Set** quantity of times and **increase** your Christmas spirit by **5**.

Every **third day** you buy **Tree Skirts** and **Tree Garlands** (both quantity of times) and **increase** your spirit by **13**.

Every **fifth day** you buy **Tree Lights** and **increase** your Christmas spirit by **17**. If you have bought Tree Skirts and Tree Garlands on the **same day,** you **additionally increase** your spirit by **30**.

Every **tenth day** you **lose 20 points of the spirit** because your cat ruins all tree decorations, and you should rebuild the tree and buy **one** piece of tree **skirt**, **garlands,** and **lights**. That is why you are forced to **increase** the allowed **quantity with 2** at the **beginning** of every **eleventh day**.

Also, if the **last day** is a **tenth day**, the cat demolishes even more and ruins the Christmas turkey, and you **lose** an additional **30 points of spirit**.

It the end, you must print the **total cost** and the **gained spirit**.

### Input / Constraints

The input will consist of **exactly 2 lines**:

* quantity – **integer in the range [1…100]**
* days **– integer in the range [1…100]**

### Output

It the end, print the **total cost** and the total gained **spirit** in the following format:

* **"Total cost: {budget}"**
* **"Total spirit: {totalSpirit}"**

### Examples

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Input** | **Output** |
| 1  7 | Total cost: 37  Total spirit: 58 | 3  20 | Total cost: 558  Total spirit: 156 |