# Exercise: Basic Syntax, Conditional Statements and Loops

Problems for exercise and homework for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3204/python-fundamentals-january-2021). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1719>

## Jenny's Secret Message

Jenny studies programming with Python and wants to create a program that **greets a user** when he/she gives his/her **name**. Jenny however is in love with **Johnny**, and would like to **greet him differently**. Can 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**, **Adults** drink **whisky**.

Make a program that receives an age, and returns what they drink.

**Rules:**

**Kids** **under 14** years old.

**Teens under 18** years old.

**Young adults under 21** years old.

**Adults above 21**.

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

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

Given a string, you have to print a string in which each character (case-sensitive) is repeated.

### Examples

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

## Can't Sleep? Count Sheep

If you can't sleep, just count sheep! Given a non-negative integer, 3 for example, return a string with a murmur: **"1 sheep...2 sheep...3 sheep..."** Input will always be valid, i.e. no negative integers.

### Examples

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

## Next Happy Year

You're saying good-bye your best friend, "***See you next happy year"***. Happy Year is the year with only **distinct digits**, (e.g) 2018. Write a program that receives an integer number and finds the next happy year.

### Examples

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

## Maximum Multiple

Given a **Divisor** and a **Bound**, find the largest integer **N**, such that:

N is **divisible by divisor**

N is **less than or equal to bound**

N is **greater than 0**.

***Notes:*** The **divisor** and **bound** are only **positive values**. It's guaranteed that a **divisor 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, **one letter at a time** and **print it.** 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 Cozonacs

*Since it’s Easter you have decided to make some cozonacs and exchange them for eggs.*

Create a program that **calculates** how much **cozonacs** 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** cozonac:

|  |  |
| --- | --- |
| **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 price for **1 kg flour**. Notice, that you need **0.250l milk** for **one** cozonac and the calculated price is for **1l**.

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

* For **every** cozonac that you make, you will receive **3 colored eggs**.
* For **every** **3rd** cozonac that you make, you will lose some of your **colored** eggs **after** you have **received** the usual **3 colored eggs** for your cozonac. The count of eggs you will lose is calculated when you **subtract** **2** from your **current** **count** of **cozonacs** – **({currentCozonacsCount} – 2)**

In the end, print the cozonacs 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 {countOfCozonacs} cozonacs! Now you have {coloredEggs} eggs and {moneyLeft}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 right 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 **count** of **cozonacs** 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 cozonacs! Now you have 16 eggs and 2.45BGN left. |
| **Comments** | |
| We start by calculating the price for a **pack of eggs**, which is **75%** of the price for **1 kg** floor, which in this case is **1.25**. The pack of eggs price is **0.9375**. The price for **1l milk** is **25%** more than the price for **1kg** floor and in this case it is – **1.5625**, but we need the price for 0.250ml, which is - **0.390625**. The total price for one cozonac is:  **1.25** + **0.9375 + 0.390625** = **2.578125**.  And we start subtracting the **price** for a **single** cozonac **from the budget**, and **for each cozonac** we receive **3** eggs. So after the first **subtraction** we will have **17.921875** budget, **1** cozonac and **3** eggs.After the second **- 15.34375** budget, **6** eggs, and on the **third** - **12.765625 budget** and **9 eggs** and since it’s the **third**, we need to **subtract** the **lost eggs**, which will be 3 – 2 = **1**, so we subtract 1 from 9 and our **eggs** become **8**. We continue **subtracting** money from the **budget** until the money **aren't enough** for us to make a cozonac. In the end we have 2.45BGN left. | |
|  | |
| 15.75  1.4 | You made 5 cozonacs! Now you have 14 eggs and 1.31BGN left. |

## \* Christmas Spirit

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

You will receive **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$ a piece
* Tree Skirt – 5$ a piece
* Tree Garlands – 3$ a piece
* Tree Lights – 15$ a 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** quantity of times and **increase** your Christmas spirit by **17**. If you have bought Tree Skirts and Tree Garlands at the **same day** you **additionally increase** your spirit by **30**.

Every **tenth day** you **lose 20 spirit**, because your cat ruins all tree decorations and you have to 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 decides to demolish even more and ruins the Christmas turkey and you **lose** additional **30 spirit**.

At 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 range [1…100]**
* days **– integer in range [1…100]**

### Output

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

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

### Examples

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