Exercises: C# Conditional Statements and Loops

Problems for exercises and homework for the <u>"Programming Fundamentals" course @ SoftUni</u>.

Problem 1. Choose a Drink

Write a program, which receives a **profession** (as a **string**) and chooses the perfect **drink** for the person. The possible combinations are:

- "**Water**" – for "**Athlete**"
- "**Coffee**" – for "**Businessman**" or "**Businesswoman**"
- "**Beer**" – for "**SoftUni Student**"
- "**Tea**" – for all **other professions**.

Examples

Input	Output
Athlete	Water

Input	Output
Doctor	Tea

Problem 2. Choose a Drink 2.0

Your program needs to get smarter. Now you will receive on the second line the quantities of the drink and you have to print the calculated the price. You can see the prices of the drinks in the table below:

	Water	Coffee	Beer	Tea
Price	0.70	1.00	1.70	1.20

Input

The **input** will be on two lines:

- On the **first line**, you will receive the **profession**
- On the **second** line, you will receive the **quantity** as an **integer**.

Output

Print the output in the following format:

The {profession} has to pay {totalPrice}.

Format the **price** to the **2nd decimal place**.

Examples

Input	Output
Athlete 1	The Athlete has to pay 0.70.
SoftUni Student 8	The SoftUni Student has to pay 13.60.
Chef 3	The Chef has to pay 3.60.

SoftUni Foundation

Follow us:

Problem 3. Restaurant Discount

A restaurant want to automate their reservation process. They need a program that reads the **hall** and the **count of people** from the console and calculates **how much** the customer should **pay** to book the place.

Different halls have different prices:

	Small Hall	Terrace	Great Hall
Price	2500$	5000$	7500$
Capacity	50	100	120

The restaurant has **discounts** depending on the **service package,** which the group wants.

You can see the discounts in the table below:

	Normal	Gold	Platinum
Discount	5%	10%	15%
Price	500$	750$	1000$

You should **add** the **price** of the **package** to the **price** of the **hall**. The discount is calculated based on the **hall's price + package's price**.

Example: The group has **10 people** and wants the **gold package** ➔ **$292.50 per person**:

- **10 people** <= 50 ➔ they get the **Small Hall** ➔ $2500
- Gold package ➔ **$750**, **10%** discount on the entire purchase
- Total price: **$2500 + $750** = **$3250**
- Discount: $3250 - **10% discount** = $2925
- Price per person: $2925 / **10 people** = **$292.50 per person**

You should print **which hall** is the **most suitable** for the group and the **price per person**. If the group is **bigger than 120** people – print "**We do not have an appropriate hall.**".

Input

- First line: the **group size** as an **integer**
- Second line: the **package** as a **string**

Output

Print the output in the following format:

```
We can offer you the {hallName}
The price per person is {price}$
```

Format the **price** to the **2ⁿᵈ decimal place**.

Examples

Input	Output
20 Gold	We can offer you the **Small Hall** The price per person is **146.25$**

90 Platinum	We can offer you the **Terrace** The price per person is **56.67$**
150 Normal	We do not have an appropriate hall.

Problem 4. Hotel

A hotel has three types of rooms: **studio**, **double** and **master suite**. The prices are different for the different months:

May and October	June and September	July, August and December
Studio - **50** leva per night	Studio - **60** leva per night	Studio - **68** leva per night
Double - **65** leva per night	Double - **72** leva per night	Double - **77** leva per night
Suite - **75** leva per night	Suite - **82** leva per night	Suite - **89** leva per night

They have also the following discounts:

- For **studio** and **more than 7** nights in **May and October**: **5% discount**
- For **double** and **more than 14** nights in **June and September**: **10% discount**
- For **suite** and **more than 14** nights in **July, August and December**: **15% discount**
- For **studio** and **more than 7** nights in **September and October**: **one night is free**

Input

The input consists of exactly **2 lines**:

- First line: **Month – May, June, July, August, September, October** or **December**
- Second line: **Nights Count – an integer between [0 ... 200]**

Output

Print **3 lines** on the console:

- On **the first**: "**Studio: {price for the stay} lv.**"
- On **the second**: "**Double: {price for the stay} lv.**"
- On **the third: "Suite: {price for the stay} lv.**"

Format the **prices** to the **2nd decimal place**.

Examples

Input	Output	Comment
June 5	Studio: 300.00 lv. Double: 360.00 lv. Suite: 410.00 lv.	The **nights are not enough** for getting a discount, so the studio is 60 lv, **double room = 72** and apartment = 82. We **multiply the prices by the nights** and get the total prices.

Input	Output	Comment
May 10	Studio: 475.00 lv. Double: 650.00 lv. Suite: 750.00 lv.	In May, we have a discount of **5%**, when the nights are **more than 7.** That means the **price for night** in the studios is **50 * 0.95 = 47.5**. Again, we **multiply the prices by the nights** and get the total prices.

Input	Output		Input	Output
July 16	Studio: 1088.00 lv. Double: 1232.00 lv. Suite: 1210.40 lv.		September 10	Studio: 540.00 lv. Double: 720.00 lv. Suite: 820.00 lv.

Problem 5. * Word in Plural

Write a program, which receives a **noun** and prints the **noun** in **plural**. You will receive one of the following cases:

- A noun that ends in **y** – remove the **y** and add **ies**
- A noun that ends in **o**, **ch**, **s**, **sh**, **x** or **z** – add **es** at the end of the word
- In all other cases – just add **s** at the end

Input

You will receive a **single** word, which you have to **pluralize.**

Output

Print only the **word in plural.**

Examples

Input	Output		Input	Output		Input	Output
couch	couches		butterfly	butterflies		door	doors

Hints

- You can use the method **String.EndsWith(…)** and **String.Remove(…)** – search for more information on how to use this methods in internet. Do not forget that **strings are immutable**.

Problem 6. Interval of Numbers

Write a program, which takes **two numbers** as input and prints the **interval of numbers between them**, **starting** from the **smaller one** and **ending** with the **larger** one.

Input

You will receive **two lines**. Each of them will contain **one integer**.

Output

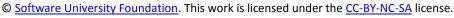Print all the numbers separated on **new lines**.

Constraints

- The numbers, which you receive will be in the interval **[0…100]**.
- The two numbers, which you take as an input will **not be equal**.

Examples

Input	Output		Input	Output
42 48	42 43		100 14	14 15

	44		16
	45		*continues...*
	46		98
	47		99
	48		100

Problem 7. Cake Ingredients

Write a baking program, which takes as an input ingredients and writes a message when the ingredient is in the system. For every given ingredient, you should write: "Adding ingredient {name of the ingredient}.". When you receive the command "Bake!" from the console you should stop the program and write "Preparing cake with {number of given ingredients} ingredients.".

Input

You will receive **ingredients until the command** "**Bake!**" is given.

Output

For every given ingredient write on a **new line** the message: "**Adding ingredient {name of the ingredient}.**". At the end print the message: "**Preparing cake with {number of given ingredients} ingredients.**".

Constraints

- You will receive maximum **20** ingredients.
- Every ingredient will be between **1** and **50 characters**.

Examples

Input	Output
Flour Bread Sugar Butter Bake!	Adding ingredient Flour. Adding ingredient Bread. Adding ingredient Sugar. Adding ingredient Butter. Preparing cake with 4 ingredients.

Problem 8. Calories Counter

You have to write a program, which **counts the calories**, which can be found in your **pizza recipe**. In your recipe, there are only **four** ingredients – **cheese**, **tomato sauce**, **salami** and **pepper**. Each ingredient contains a **fixed amount** of calories:

- **Cheese – 500** calories
- **Tomato sauce – 150** calories
- **Salami – 600** calories
- **Pepper – 50** calories

If you **receive** one of these ingredients **more than once**, you should **add** the calories to the **total amount again**. You should **not process any other ingredients**. Ingredients are **case-insensitive**.

Input

On the next **n lines**, you will receive different **ingredients**. **Add** the calories **only** from the ones, **which are in your recipe**.

Output

Print the answer in the following format:

`Total calories: {totalCaloriesAmount}`

Constraints

- You will receive maximum **20** ingredients.
- Every ingredient will be between **1** and **50 characters**.

Examples

Input	Output
5 cheese toMatO sauce flour salami pepper	Total calories: 1300

Input	Output
3 Cheese Cucumber cheese	Total calories: 1000

Problem 9. Count the Integers

Write a program, which can receive **any type of input**, but upon receiving anything **other than an integer** – **stops the execution** of the program and prints **how many numbers were read**.

Input

You can receive **any type of data** as **input** from the console.

Output

Print only the **total count** of the numbers.

Constraints

- You will receive **no more than 100 lines**.
- Every number will not be longer than **7 digits**.

Examples

Input	Output
1 2 3 4 5 6 PF is the best!	6

Input	Output
12312 End the input	1

Problem 10. Triangle of Numbers

Write a program, which receives a number – **n**, and prints a triangle from **1 to n** as in the examples.

Constraints

- **n** will be in the interval [**1...20**].

Examples

Input	Output
3	1 2 2 3 3 3

Input	Output
5	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

Input	Output
6	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6

Problem 11. 5 Different Numbers

You will be given two numbers – **a** and **b**. Generate **five** numbers - n_1, n_2, n_3, n_4, n_5, for which the following **conditions** are true: $a \le n_1 < n_2 < n_3 < n_4 < n_5 \le b$. If there is **no number** in the given interval, which **satisfies the conditions** – print "**No**".

Input

The input contains **two integers**, each on a **new line**.

Output

Print all numbers in **increasing order** and on a **new line**.

Constraints

- **a** and **b** will be integers in the interval **[-100...100]**

Examples

Input	Output
3 8	3 4 5 6 7 3 4 5 6 8 3 4 5 7 8 3 4 6 7 8 3 5 6 7 8 4 5 6 7 8

Input	Output
40 46	40 41 42 43 44 40 41 42 43 45 40 41 42 43 46 40 41 42 44 45 40 41 42 44 46 40 41 42 45 46 40 41 43 44 45 40 41 43 44 46 40 41 43 45 46 40 41 44 45 46 40 42 43 44 45 40 42 43 44 46 40 42 43 45 46 40 42 44 45 46 40 43 44 45 46 41 42 43 44 45 41 42 43 44 46

Input	Output
13 16	No

		41 42 43 45 46 41 42 44 45 46 41 43 44 45 46 42 43 44 45 46		

Problem 12. Test Numbers

Write a program, which finds all the possible combinations between two numbers – **N** and **M**.

The first digit **decreases** from **N to 1**, and the second digit **increases** from **1 to M**. The two digits form a **number**. **Multiply the two digits**, then **multiply** their **product** by **3**. **Add** the **result** to the **total sum**.

You will also be given a **third number**, which will be the **maximum boundary of the sum**. If the **sum is equal or greater than this number** you should **stop the program**. See the examples for further information.

Input

The input is read from the console and consists of three lines:

- **First line** – **N** – **integer** in the interval **[1…100]**
- **Second line** – **M** – **integer** in the interval **[1…100]**
- **Third line** – **maximum sum boundary** – **integer** in the interval **[1…1000000]**

Output

The output depends on the result:

- If the **sum is equal or greater** than the **maximum sum:**
  - "{count of combinations} combinations"
  - "Sum: {sum from the combinations} >= {maximum sum}"
- If the sum is **less than** the **maximum sum**:
  - "{count of combinations} combinations"
  - "Sum: {sum from the combinations}"

Examples

Input	Output	Comments
3 4 123	7 combinations Sum: 126 >= 123	Total **12** combinations: 3 1; 3 2; 3 3; 3 4; 2 1; 2 2; 2 3; 2 4; 1 1; 1 2; 1 3; 1 4; 1st: 3 * (3 * 1) = 9; 2nd: 9 + 3 * (3 * 2) = 27;…; 7th: 108 + 3 * (2 * 3) = 126; 126 >= 123 – we have to stop our program and print the result.
2 2 50	4 combinations Sum: 27	Total 4 combinations: 2 1; 2 2; 1 1; 1 2 1st: 3 * (2 * 1) = 6; 2nd: 6 + 3 * (2 * 2) = 18; 3rd: 18 + 3 * (1 * 1) = 21; 4th: 21 + 3 * (1 * 2) = 27 Sum: 27 < 50 -> total 4 combinations

Problem 13. Game of Numbers

Write a program, which finds all possible combinations in the interval **between two numbers**. The program should also find the **last combination**, in which a **number's digits are equal to a given magical number**.

Input

The input is read from the console and consists of three lines:

- **First line** – **N** – **integer** in the interval **[1…999]**
- **Second line** – **M** – **integer** in the interval **[N…1000]**
- **Third line** – **magical number** – **integer** in the interval **[1…10000]**

Output

The output depends on the result:

- If there is a number with **digits equal to the magical number:**

    o **"Number found! {first number} + {second number} = {magical number}"**

- If such **combination does not exist**:

    o **"{total number of combinations} combinations - neither equals {magical number}"**

Examples

Input	Output	Comments
1 10 5	Number found! 4 + 1 = 5	All combinations between 1 and 10 are: 1 1, 1 2, 1 3, 1 4, 1 5, … 2 1, 2 2, … **4 1**, 4 2, 4 3 … 10 9, 10 10. Last combination with sum of the digits equal to the magical number (**5**) is **4 1**
23 24 100	4 combinations - neither equals 100	Total 4 combinations: 23 23; 23 24; 24 23; 24 24 Neither of them has a sum of **100**.

Problem 14. * Magic Letter

Write a program, which prints all **3-letter combinations**, **using only the letters from a given interval**. You will also receive a **third letter**. Every **combination**, which **contains** this letter **should not** be printed.

Input

The input is read from the console and consists three lines:

- **First line** – lower case **English letter** from 'a' to 'z'
- **Second line** – lower case **English letter** from 'a' to 'z'
- **Third line** – lower case **English letter** from 'a' to 'z' – combinations, containing this letter should not be printed

Output

Print **all combinations on a single line**.

Examples

Input	Output	Comments
a c b	aaa aac aca acc caa cac cca ccc	All combinations with **a**, **b**, and **c** are: aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc

---

SoftUni Foundation

		Combinations containing **b are invalid**.

Input	Output
f k h	fff ffg ffi ffj ffk fgf fgg fgi fgj fgk fif fig fii fij fik fjf fjg fji fjj fjk fkf fkg fki fkj fkk gff gfg gfi gfj gfk ggf ggg ggi ggj ggk gif gig gii gij gik gjf gjg gji gjj gjk gkf gkg gki gkj gkk iff ifg ifi ifj ifk igf igg igi igj igk iif iig iii iij iik ijf ijg iji ijj ijk ikf ikg iki ikj ikk jff jfg jfi jfj jfk jgf jgg jgi jgj jgk jif jig jii jij jik jjf jjg jji jjj jjk jkf jkg jki jkj jkk kff kfg kfi kfj kfk kgf kgg kgi kgj kgk kif kig kii kij kik kjf kjg kji kjj kjk kkf kkg kki kkj kkk

Input	Output
a c z	aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc

Problem 15. * Neighbour Wars

Gosho and Pesho are neighbours, but they don't like each other very much. They don't like violence as well, so they decided they need a save environment where they can fight each other. They hired you to write a program, which calculates who would win the fight.

Gosho and Pesho have their own signature attacks – **Gosho** attacks with "**Thunderous fist**" **on every even turn** of the game and **Pesho** attacks with "**Roundhouse kick**" **on every odd turn**. You will receive **how much damage these attacks do from the console**.

Both players start with 100 Health points. On **every third turn** Pesho and Gosho **restore 10 Health Points**. Health points are restored **after the attack is made**.

When one of the **player's health is below or equal to zero** you should **stop any other further operations** and **print who the winner is**, **how much health points** he has and in **which turn** he won. Since Mike Tyson is the judge of the match, the winning round is always printed with "**th**" in the end.

Input

The input is read from the console and consists of two lines:

- **First line – Pesho's damage**
- **Second line – Gosho's damage**

Output

Print on every turn who is attacking and how much health the opponent is after the attack:

"{name of the attacker} used {name of the attack} and reduced {name of the defending player} to {health of the defending player} health."
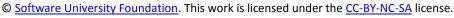
When one of the players is **dead** print:

"{name of the winner} won in {number of the round}th round."

Constraints

- **Pesho's damage** and **Gosho's damage** will be **integers** in the interval **[0…100]**
- There will **always** be a **winner**

Examples

Input	Output	Comments
30 40	Pesho used Roundhouse kick and reduced Gosho to 70 health. Gosho used Thunderous fist and reduced Pesho to 60 health. Pesho used Roundhouse kick and reduced Gosho to 40 health. Gosho used Thunderous fist and reduced Pesho to 30 health. Pesho used Roundhouse kick and reduced Gosho to 20 health. Gosho won in 6th round.	1st round -> **Pesho** attacks in **odd rounds** -> so he does **30 damge to Gosho**. 2nd round -> it is **Gosho's** turn and he **does 40 damage to Pesho**. 3rd round -> **first Pesho attacks with 30 damage and Gosho is now 40 health**. After that **both players receive 10 health**. 4th round **-> After healing Gosho is 50 health and Pesho is 70**. It is **Gosho's turn and he does 40 damage to Pesho** -> Pesho is now 30 health. 5th round -> **Pesho attacks and reduces Gosho from 50 to 20 health**. 6th round -> **Gosho attacks with 40 damage** and **kills Pesho**. They will **not receive healing**, because **one of the player is dead** and we should **print the final result**.

Input	Output
20 10	Pesho used Roundhouse kick and reduced Gosho to 80 health. Gosho used Thunderous fist and reduced Pesho to 90 health. Pesho used Roundhouse kick and reduced Gosho to 60 health. Gosho used Thunderous fist and reduced Pesho to 90 health. Pesho used Roundhouse kick and reduced Gosho to 50 health. Gosho used Thunderous fist and reduced Pesho to 80 health. Pesho used Roundhouse kick and reduced Gosho to 40 health. Gosho used Thunderous fist and reduced Pesho to 80 health. Pesho used Roundhouse kick and reduced Gosho to 20 health. Gosho used Thunderous fist and reduced Pesho to 80 health. Pesho used Roundhouse kick and reduced Gosho to 10 health. Gosho used Thunderous fist and reduced Pesho to 70 health. Pesho won in 13th round.
100 100	Pesho won in 1th round.