Lab: Basic Syntax, Conditional Statements and Loops

Problems for exercises and homework for the "Technology Fundamentals" course @ SoftUni.

You can check your solutions in Judge.

1. Student Information

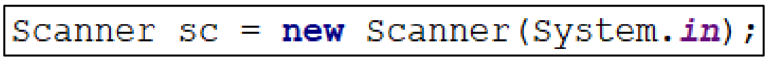
You will be given 3 lines of input – student name, age and average grade. Your task is to print all the info about the student in the following format: "**Name: {student name}, Age: {student age}, Grade: {student grade}**".

Examples

Input	Output
John 15 5.40	Name: John, Age: 15, Grade: 5.40
Steve 16 2.50	Name: Steve, Age: 16, Grade: 2.50
Marry 12 6.00	Name: Marry, Age: 12, Grade: 6.00

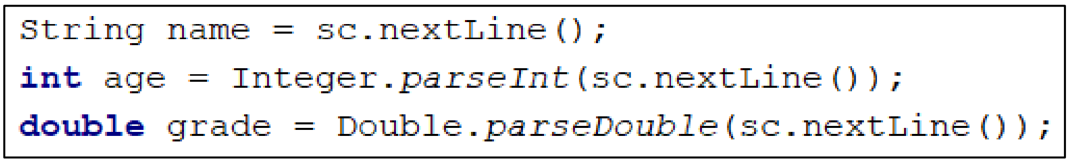
Solution

First, we need a scanner, which we can use to read data from the console.



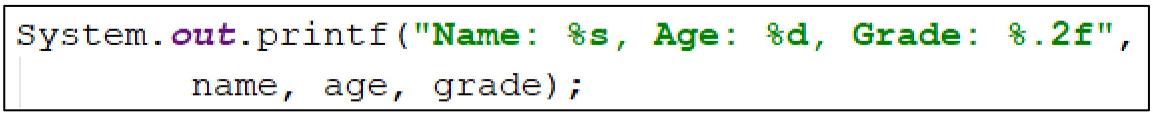
```
Scanner sc = new Scanner(System.in);
```

Read all the information – student name, age and grade



```
String name = sc.nextLine();
int age = Integer.parseInt(sc.nextLine());
double grade = Double.parseDouble(sc.nextLine());
```

Finally, we need to print the information in the specified format



```
System.out.printf("Name: %s, Age: %d, Grade: %.2f",
        name, age, grade);
```

2. Passed

Write a program, which takes as an input a **grade** and prints "**Passed!**" if the grade is **equal or more than 3.00**.

Input

The **input** comes as a single floating-point number.

Output

The **output** is either "**Passed!**" if the grade is **equal or more than 3.00**, otherwise you should print nothing.

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Examples

Input	Output
5.32	Passed!

Input	Output
2.34	*(no output)*

3. Passed or Failed

Modify the above program, so it will print "**Failed**!" if the grade is **lower than 3.00**.

Input

The **input** comes as a single double number.

Output

The **output** is either "**Passed**!" if the grade is **more than 2.99**, otherwise you should print "**Failed**!".

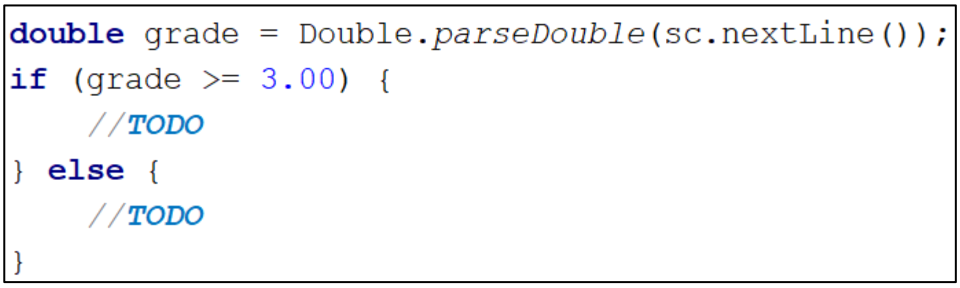
Examples

Input	Output
5.32	Passed!

Input	Output
2.36	Failed!

Hint

We need to take **floating-point** number from the console. After that print in the **`else`** statement the appropriate message.



```
double grade = Double.parseDouble(sc.nextLine());
if (grade >= 3.00) {
    //TODO
} else {
    //TODO
}
```

4. Back in 30 Minutes

Every time Stamat tries to pay his bills he sees on the cash desk the sign: **"I will be back in 30 minutes"**. One day Stamat was sick of waiting and decided he needs a program, which **prints the time** after **30 minutes**. That way he won't have to wait on the desk and come at the appropriate time. He gave the assignment to you, so you have to do it.

Input

The **input** will be on two lines. On the **first line**, you will receive the **hours** and on the **second** you will receive the **minutes**.

Output

Print on the console the time after **30** minutes. The result should be in format "**hh:mm**". The **hours** have **one or two numbers** and the **minutes** have always **two numbers (with leading zero)**.

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Constraints

- The **hours** will be between **0 and 23**.
- The **minutes** will be between **0 and 59**.

Examples

Input	Output
1 46	2:16

Input	Output
0 01	0:31

Input	Output
23 59	0:29

Input	Output
11 08	11:38

Input	Output
11 32	12:02

Hints

- Add 30 minutes to the initial minutes, which you receive from the console. If the minutes are more than 59 – increase the hours with 1 and decrease the minutes with 60. The same way check if the hours are more than 23. When you print check for leading zero.

5. Month Printer

Write a program, which takes an **integer** from the console and prints the corresponding **month**. If the number **is more than 12** or **less than 1** print "**Error!**".

Input

You will receive a **single integer** on a **single line**.

Output

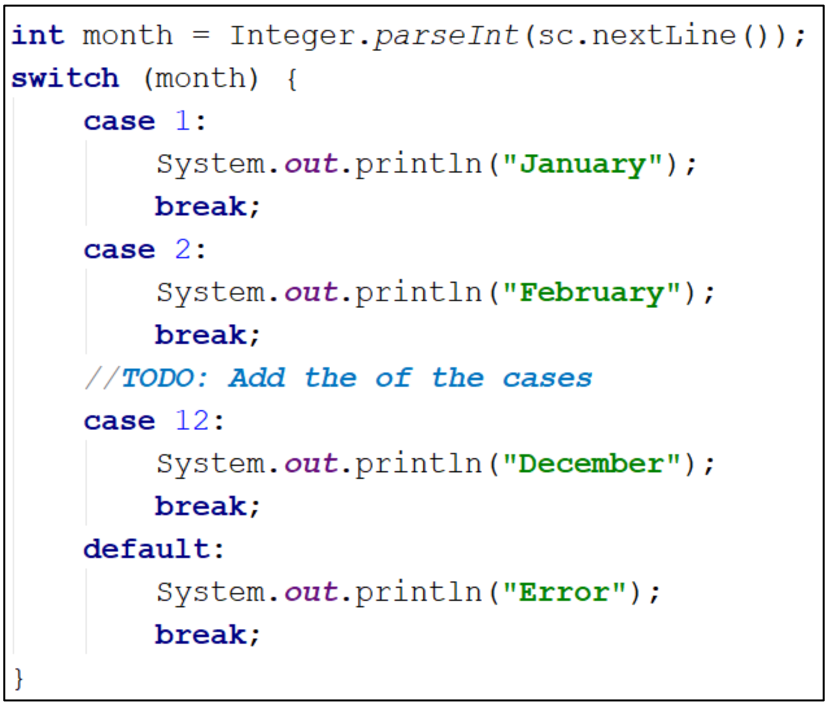
If the number is within the boundaries print the corresponding month, otherwise print "**Error!**".

Examples

Input	Output
2	February

Input	Output
13	Error!

Hints



```
int month = Integer.parseInt(sc.nextLine());
switch (month) {
    case 1:
        System.out.println("January");
        break;
    case 2:
        System.out.println("February");
        break;
    //TODO: Add the of the cases
    case 12:
        System.out.println("December");
        break;
    default:
        System.out.println("Error");
        break;
}
```

6. Foreign Languages

Write a program, which prints the language, that a given country speaks. You can receive only the following combinations: English **is spoken** in England and USA; Spanish **is spoken** in Spain, Argentina and Mexico; for the others**,** we should print "unknown".

Input

You will receive a **single country name** on a **single line**.

Output

Print the **language**, which the country **speaks**, or if it is **unknown** for your program, print **"unknown"**.

Examples

Input	Output
USA	English

Input	Output
Germany	unknown

Hint

Think how you can **merge** multiple cases, in order to **avoid** writing more code than you need to.

7. Theatre Promotions

A theatre **is doing a ticket sale**, but they need a program **to** calculate the price of a single ticket. If the given age does not fit one of the categories**,** you should print "Error!". You can see the prices **in** the table below:

Day / Age	0 <= age <= 18	18 < age <= 64	64 < age <= 122
Weekday	12$	18$	12$
Weekend	15$	20$	15$

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Holiday	5$	12$	10$

Input

The input comes in **two lines**. On the **first** line, you will receive the **type of day**. On the **second** – the **age** of the person.

Output

Print the price of the ticket according to the table, or "**Error!**" if the age is not in the table.

Constraints

- The age will be in the interval **[-1000...1000]**.
- The type of day will **always be valid**.

Examples

Input	Output

`Weekday`
`42`
	`18$`

Input	Output

`Holiday`
`-12`
	`Error!`

Input	Output

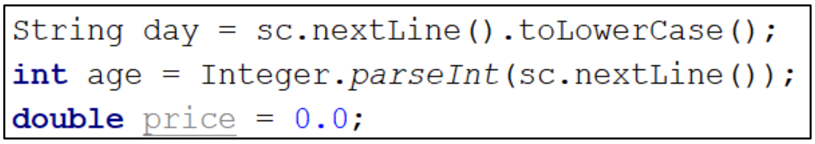
`Holiday`
`15`
	`5$`

Input	Output

`Weekend`
`122`
	`15$`

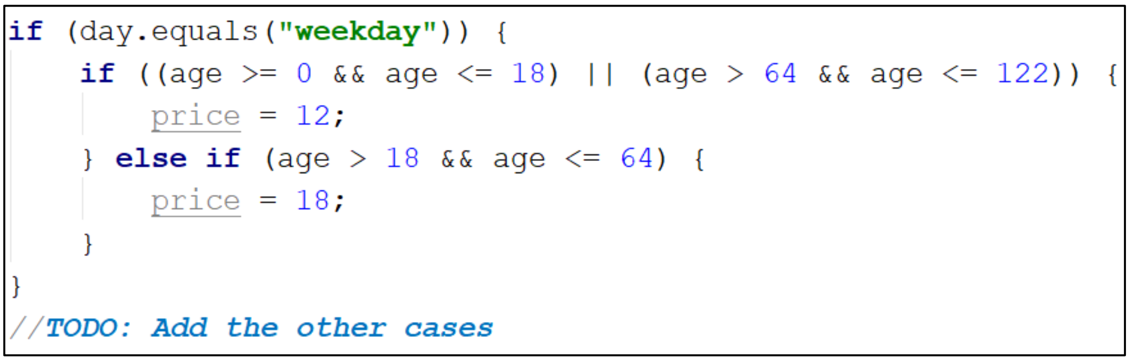
Hints

We need to read **two** lines. **First** one will be the **type of day**. We will convert it to **lower case** letters with the method "**`toLowerCase()`**". After that, we will read the **age** of the person and declare a **variable** – **`price`**, which we will use to set the price of the ticket.



```
String day = sc.nextLine().toLowerCase();
int age = Integer.parseInt(sc.nextLine());
double price = 0.0;
```

For every **type of day**, we will need to add **different cases** to check the **age** of the person and **set the price**. Some of the **age groups** have **equal prices** for the **same type** of day. This means we can use **logical operators** to **merge some of the conditions**.

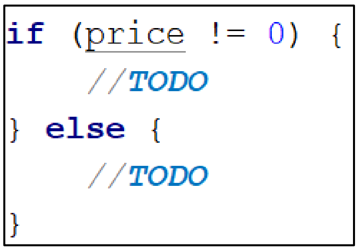


```
if (day.equals("weekday")) {
    if ((age >= 0 && age <= 18) || (age > 64 && age <= 122)) {
        price = 12;
    } else if (age > 18 && age <= 64) {
        price = 18;
    }
}
//TODO: Add the other cases
```

Think **where** and **how** you can use **logical operators** for the **other cases**.

We can check if the **`price` has a value** different, than the **initial** one. It it does, that means we got a **valid combination of day and age** and the price of the ticket is saved in the **`price`** variable. If the **`price`** has a **value of 0**, then none of the cases got hit, therefore we have to **print the error message**.

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

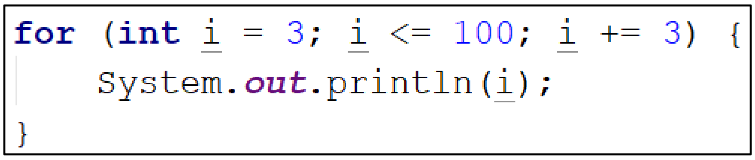


```
if (price != 0) {
    //TODO
} else {
    //TODO
}
```

8. Divisible by 3

Write a program, which prints all the numbers from **1 to 100**, which are **divisible by 3**. You have to use a single **for** loop. The program should not receive input.

Solution



```
for (int i = 3; i <= 100; i += 3) {
    System.out.println(i);
}
```

9. Sum of Odd Numbers

Write a program that prints the next **n odd numbers** (starting from 1) and on the **last row** prints the **sum of them**.

Input

On the first line, you will receive a number – **n**. This number shows how many **odd numbers** you should print.

Output

Print the next **n** odd numbers, starting from **1**, separated by **new lines**. On the last line, print the **sum** of these numbers.

Constraints

- **n** will be in the interval **[1…100]**

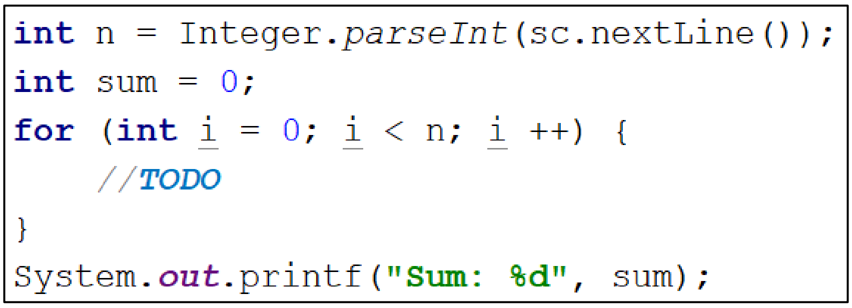
Examples

Input	Output
5	1 3 5 7 9 Sum: 25

Input	Output
3	1 3 5 Sum: 9

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Hints



```
int n = Integer.parseInt(sc.nextLine());
int sum = 0;
for (int i = 0; i < n; i ++) {
    //TODO
}
System.out.printf("Sum: %d", sum);
```

10. Multiplication Table

You will receive an **integer** as an input from the console. Print the **10 times table** for this integer. See the examples below for more information.

Output

Print every row of the table in the following format:

`{theInteger} X {times} = {product}`

Constraints

- The integer will be in the interval **[1…100]**

Examples

Input	Output
5	5 X 1 = 5 5 X 2 = 10 5 X 3 = 15 5 X 4 = 20 5 X 5 = 25 5 X 6 = 30 5 X 7 = 35 5 X 8 = 40 5 X 9 = 45 5 X 10 = 50

Input	Output
2	2 X 1 = 2 2 X 2 = 4 2 X 3 = 6 2 X 4 = 8 2 X 5 = 10 2 X 6 = 12 2 X 7 = 14 2 X 8 = 16 2 X 9 = 18 2 X 10 = 20

11. Multiplication Table 2.0

Rewrite you program so it can receive the **multiplier from the console**. Print the **table from the given multiplier to 10**. If the given multiplier is **more than 10** - print only one row with the **integer**, the given **multiplier** and the **product**. See the examples below for more information.

Output

Print every row of the table in the following format:

`{theInteger} X {times} = {product}`

Constraints

- The integer will be in the interval **[1…100]**

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Examples

Input	Output
5 1	5 X 1 = 5 5 X 2 = 10 5 X 3 = 15 5 X 4 = 20 5 X 5 = 25 5 X 6 = 30 5 X 7 = 35 5 X 8 = 40 5 X 9 = 45 5 X 10 = 50

Input	Output
2 5	2 X 5 = 10 2 X 6 = 12 2 X 7 = 14 2 X 8 = 16 2 X 9 = 18 2 X 10 = 20

Input	Output
2 14	2 X 14 = 28

12. Even Number

Take as an input an even number and **print its absolute value**. If the number is odd, print "Please write an even number." and continue reading numbers.

Examples

Input	Output
1 3 6	Please write an even number. Please write an even number. The number is: 6

Input	Output
-6	The number is: 6

13. Refactor Sum of Odd Numbers

You are assigned to **find and fix the bugs** in an existing piece of code, using the **debugger**. You should trace the program execution to find the lines of code that produce incorrect or unexpected results.

You are given a program (existing source code) that prints the next **n odd numbers** (starting from 1) and on the **last row** prints the **sum of them**.

Examples

Input	Output
5	1 3 5 7 9 Sum: 25

Input	Output
3	1 3 5 Sum: 9

SumOddNumbers.java

```
Scanner sc = new Scanner(System.in);
int n = Integer.parseInt(sc.nextLine());
int sum = 1;
for (int i = 0; i <= n; i++) {
    System.out.print(2 * i + 1);
```

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

```
    sum += 2 * i;
}
System.out.printf("Sum: %d%n", sum);
```

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Exercise: Basic Syntax, Conditional Statements and Loops

Problems for exercises and homework for the ["Programming Fundamentals" course @ SoftUni.](https://softuni.org)

You can check your solutions in [Judge.](https://judge.softuni.org)

1. Ages

Write a program that determines whether based on the given age a person is: baby, child, teenager, adult, elder. The bounders are:

- **0-2 – baby;**
- **3-13 – child;**
- **14-19 – teenager;**
- **20-65 – adult;**
- **>=66 – elder;**
- All the values are **inclusive**.

Examples

Input	Output
20	adult
1	baby
100	elder

2. Division

You will be given an integer and you have to print on the console whether that number is divisible by the following numbers: 2, 3, 6, 7, 10. You should **always take the bigger division**. If the number is divisible by both **2** and **3** it is also divisible by **6** and you should print only the division by 6. If a number is divisible by **2** it is sometimes also divisible by **10** and you should print the division by 10. If the number is not divisible by any of the given numbers print “**Not divisible”.** Otherwise print "**The number is divisible by {number}**".

Examples

Input	Output
30	The number is divisible by 10
15	The number is divisible by 3
12	The number is divisible by 6
1643	Not divisible

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

3. Vacation

You are given a group of people, type of the group, on which day of the week they are going to stay. Based on that information calculate how much they have to pay and print that price on the console. Use the table below. In each cell is the price for a **single person**. The output should look like that: "**Total price: {price}**". The price should be formatted to the second decimal point.

	Friday	Saturday	Sunday
Students	8.45	9.80	10.46
Business	10.90	15.60	16
Regular	15	20	22.50

There are also discounts based on some conditions:

- **Students –** if the group is bigger than or equal to 30 people you should reduce the **total** price by 15%
- **Business –** if the group is bigger than or equal to 100 people **10** of them can stay **for free.**
- **Regular –** if the group is bigger than or equal 10 and less than or equal to 20 reduce the **total** price by 5%

You should reduce the prices in that EXACT order

Examples

Input	Output
30 Students Sunday	Total price: 266.73
40 Regular Saturday	Total price: 800.00

4. Print and Sum

Write a program to display numbers from given start to given end and their sum. All the numbers will be integers. On the first line you will receive the start number, on the second the end number.

Examples

Input	Output
5 10	5 6 7 8 9 10 Sum: 45
0 26	0 1 2 … 26 Sum: 351
50 60	50 51 52 53 54 55 56 57 58 59 60 Sum: 605

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

5. Login

You will be given a string representing a username. The password will be that username reversed. Until you receive the correct password print on the console "**Incorrect password. Try again.**". When you receive the correct password print "**User {username} logged in.**" However on the fourth try if the password is still not correct print "**User {username} blocked!**" and end the program.

Examples

Input	Output
Acer login go let me in recA	Incorrect password. Try again. Incorrect password. Try again. Incorrect password. Try again. User Acer logged in.
momo omom	User momo logged in.
sunny rainy cloudy sunny not sunny	Incorrect password. Try again. Incorrect password. Try again. Incorrect password. Try again. User sunny blocked!

6. Strong Number

Write a program to check if a given number is a strong number or not. A number is strong if the sum of the Factorial of each digit is equal to the number. For example 145 is a strong number, because **1! + 4! + 5! = 145.** Print "**yes**" if the number is strong and "**no**" if the number is not strong.

Examples

Input	Output
2	yes
3451	no
40585	yes

7. Vending Machine

You task is to calculate the total price of a purchase from a vending machine. Until you receive "**Start**" you will be given different coins that are being inserted in the machine. You have to sum them in order to have the total money inserted. There is a problem though. Your vending machine only works with **0.1**, **0.2**, **0.5, 1, and 2** coins. If someone tries to insert some other coins you have to display "**Cannot accept {money}**", where the value is **formated to the second digit after the decimal point** and **not** add it to the total money. On the next few lines until you receive "**End**" you will be given products to purchase. Your machine has however only "**Nuts**", "**Water**", "**Crisps**", "**Soda**", "**Coke**". The prices are: **2.0**, **0.7**, **1.5**, **0.8**, **1.0** respectively. If the person tries to purchase a not existing product print "**Invalid product**". Be careful that the person may try to purchase a product for which he doesn't have money. In that case print "**Sorry, not enough money**". If the person purchases a product successfully print "**Purchased {product name}**".

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

After the “End” command print the money that are left formatted to the second decimal point in the format "**Change: {money left}**".

Examples

Input	Output
1 1 0.5 0.6 Start Coke Soda Crisps End	Cannot accept 0.60 Purchased Coke Purchased Soda Sorry, not enough money Change: 0.70

8. 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

9. *Padawan Equipment

Yoda is starting his newly created Jedi academy. So, he asked master Ivan Cho to **buy** the **needed equipment**. The number of **items** depends on **how many students will sign up**. The equipment for the Padawan contains **lightsabers, belts and robes**.

You will be given **the amount of money Ivan Cho has**, the **number of students** and the **prices of each item**. You have to help Ivan Cho **calculate** if the **money** he has is **enough to buy all of the equipment**, or how much more money he needs.
Because the lightsabres sometimes brake, Ivan Cho should **buy 10% more**, **rounded up** to the next integer. Also, every **sixth belt is free**.

Input / Constraints

The input data should be read from the console. It will consist of **exactly 5 lines**:

- The **amount of money** Ivan Cho has – **floating-point number** in **range [0.00…1,000.00]**
- The **count of students – integer in range [0…100]**
- The **price of lightsabers** for a **single sabre – floating-point number** in **range [0.00…100.00]**
- The **price of robes** for a **single robe – floating-point number** in **range [0.00…100.00]**

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

- The **price of belts** for a **single belt – floating-point number** in **range [0.00…100.00]**

The **input data will always be valid**. **There is no need to check it explicitly**.

Output

The output should be printed on the console.

- **If the calculated price of the equipment is less or equal to the money Ivan Cho has:**
  - "`The money is enough - it would cost {the cost of the equipment}lv.`"
- **If the calculated price of the equipment is more than the money Ivan Cho has:**
  - "`Ivan Cho will need {neededMoney}lv more.`"
- **All prices** must be **rounded to two digits after the decimal point.**

Examples

Input	Output	Comments
100 2 1.0 2.0 3.0	`The money is enough - it would cost 13.00lv.`	Needed equipment for 2 padawans : sabresPrice*(studentsCount + 10%) + robesPrice * (studentsCount) + beltsPrice*(studentsCount-freeBelts) 1*(3) + 2*(2) + 3*(2) = 13.00 13.00 <= 100 – the money will be enough.
Input	**Output**	**Comments**
100 42 12.0 4.0 3.0	`Ivan Cho will need 737.00lv more.`	Needed equipment for 42 padawans: 12*47 + 4*42 + 3*35 = 837.00 837 > 100 – need 737.00 lv. more.

10. *Rage Expenses

As a MOBA challenger player, Pesho has the bad habit to trash his PC when he loses a game and rage quits. His gaming setup consists of **headset, mouse, keyboard and display**. You will receive Pesho`s **lost games count**.

Every **second** lost game, Pesho trashes his **headset.**

Every **third** lost game, Pesho trashes his **mouse**.

When Pesho trashes **both his mouse and headset** in the **same** lost game, he also trashes his **keyboard**.

Every second time, when he trashes his keyboard, he also trashes his **display**.

You will receive the price of each item in his gaming setup. Calculate his rage expenses for renewing his gaming equipment.

Input / Constraints

- On the first input line - **lost games count** – integer in the range **[0, 1000]**.
- On the second line – **headset price** - floating point number in range **[0, 1000]**.
- On the third line – **mouse price** - floating point number in range **[0, 1000]**.
- On the fourth line – **keyboard price** - floating point number in range **[0, 1000]**.
- On the fifth line – **display price** - floating point number in range **[0, 1000]**.

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Output

- As output you must print Pesho`s total expenses: **"Rage expenses: {expenses} lv."**
- Allowed working **time** / **memory**: **100ms** / **16MB**.

Examples

Input	Output	Comment
7 2 3 4 5	Rage expenses: 16.00 lv.	Trashed headset -> 3 times Trashed mouse -> 2 times Trashed keyboard -> 1 time Total: 6 + 6 + 4 = 16.00 lv;
23 12.50 21.50 40 200	Rage expenses: 608.00 lv.	

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

More Exercise: Basic Syntax, Conditional Statements and Loops

Problems for exercises and homework for the "Technology Fundamentals" course @ SoftUni.

You can check your solutions in Judge.

1. Sort Numbers

Read three real numbers and sort them in descending order. Print each number on a new line.

Examples

Input	Output
2 1 3	3 2 1
-2 1 3	3 1 -2
0 0 2	2 0 0

2. English Name of the Last Digit

Write a **method** that returns the **English name** of the last digit of a given number. Write a program that reads an integer and prints the returned value from this method.

Examples

Input	Output
512	two
1	one
1643	three

3. Gaming Store

Write a program, which helps you buy the games. The **valid games** are the following games in this table:

Name	Price
OutFall 4	$39.99
CS: OG	$15.99
Zplinter Zell	$19.99
Honored 2	$59.99
RoverWatch	$29.99

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

RoverWatch Origins Edition	$39.99

On the first line, you will receive your **current balance** – a **floating-point** number in the range **[0.00…5000.00]**.

Until you receive the command "**Game Time**", you have to keep **buying games**. When a **game** is **bought**, the user's **balance** decreases by the **price** of the game.

Additionally, the program should obey the following conditions:

- If a game the user is trying to buy is **not present** in the table above, print "**Not Found**" and **read the next line**.
- If at any point, the user has **$0** left, print "**Out of money!**" and **end the program**.
- Alternatively, if the user is trying to buy a game which they **can't afford**, print "**Too Expensive**" and **read the next line**.

When you receive "**Game Time**", **print** the user's **remaining money** and **total spent on games**, **rounded** to the **2nd decimal place**.

Examples

Input	Output
120 RoverWatch Honored 2 Game Time	Bought RoverWatch Bought Honored 2 Total spent: $89.98. Remaining: $30.02
19.99 Reimen origin RoverWatch Zplinter Zell Game Time	Not Found Too Expensive Bought Zplinter Zell Out of mo-ney!
79.99 OutFall 4 RoverWatch Origins Edition Game Time	Bought OutFall 4 Bought RoverWatch Origins Edition Total spent: $79.98. Remaining: $0.01

4. Reverse String

Write a program which reverses a string and print it on the console.

Examples

Input	Output
Hello	olleH
SoftUni	inUtfoS
1234	54321

5. Messages

Write a program, which emulates **typing an SMS**, following this guide:

1	**2**	**3**

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

	abc	def
4 ghi	**5** jkl	**6** mno
7 pqrs	**8** tuv	**9** wxyz
	0 space	

Following the guide, **2** becomes “**a**”, **22** becomes “**b**” and so on.

Examples

Input	Output
5 44 33 555 555 666	hello

Input	Output
9 44 33 999 0 8 44 33 777 33	hey there

Input	Output
7 6 33 33 8 0 6 33	meet me

Hints

- A native approach would be to just put all the possible combinations of digits in a giant **`switch`** statement.
- A cleverer approach would be to come up with a **mathematical formula**, which **converts** a **number** to its **alphabet** representation:

Digit	2	3	4	5	6	7	8	9
Index	0 1 2	3 4 5	6 7 8	9 10 11	12 13 14	15 16 17 18	19 20 21	22 23 24 25
Letter	a b c	d e f	g h i	j k l	m n o	p q r s	t u v	w x y z

- Let’s take the number **222** (**c**) for example. Our algorithm would look like this:
  - Find the **number of digits** the number has “e.g. **222 ➔ 3 digits**”
  - Find the **main digit** of the number “e.g. **222 ➔ 2**”
  - Find the **offset** of the number. To do that, you can use the formula: **`(main digit - 2) * 3`**
  - If the main digit is **8 or 9**, we need to **add 1** to the **offset**, since the digits **7** and **9** have **4 letters each**
  - Finally, find the **letter index** (a ➔ 0, c ➔ 2, etc.). To do that, we can use the following formula: **`(offset + digit length - 1)`**.
  - After we’ve found the **letter index**, we can just add that to **the ASCII code** of the lowercase letter “**a**” (97)

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.