Exercise: Conditional Statements

1. Sum Seconds

Write a program, which reads three numbers, which represent seconds **(between 1 and 50)** and sums them. Print the result in the format "**`minutes:seconds`**". The seconds should be printed **with a leading zero** (2 ⍰ "02", 7 ⍰ "07", 35 ⍰ "35").

Examples

input	output
35 45 44	2:04
22 7 34	1:03
50 50 49	2:29
14 12 10	0:36

Hints:

- Sum the three numbers to get the total seconds.
- If the result is between 0 and 59, print 0 minutes + the seconds.
- If the result is between 60 and 119, print 1 minute + the seconds minus 60.
- If the result is between 120 and 179, print 2 minutes + the seconds minus 120.
- If the seconds are less than 10, print them with a leading zero.

2. Bonus Points

You are given an **integer**, which represents a **number of points**. Bonus points are given by **the rules below**. Write a program, which calculates **the bonus points for this number** and **the total number of points** with the bonus.

- If the number is **less than or equal to 100**, the bonus points are **5**.
- If the number is **bigger than 100 and less than or equal to 1000**, the bonus points are **20%** of the number.
- If the number is **bigger than 1000**, the bonus points are **10%** of the number.
- Additional bonus points (added after the previous ones):
    - For an **even** number □ + 1 p.
    - For a number **ending in 5** □ + 2 p.

Examples

input	output
20	6 26
175	37 212

2703	270.3 2973.3
15875	1589.5 17464.5

Hints:

- Calculate the main bonus points with an **`if`**-**`else`**-**`if`**-**`else`**-**`if`** statement (you will have 3 cases).
- Calculate the additional bonus points with another **`if`**-**`else`**-**`if`** statement (you will have 2 more cases).

3. Speed Info

Write a program, which reads a floating point number, which represents speed and prints information about it. If the speed is **less than or equal to 10** print "**slow**". If the speed is above 10 and up to 50 print "**average**". If the speed is above 50 and up to 150 print "**fast**". If the speed is above 150 and up to 1000 print "**ultra fast**". If it's above 1000 print "**extremely fast**".

Examples

input	output
8	slow
49.5	average
126	fast
160	ultra fast
3500	extremely fast

Hint: use a series of **`if`**-**`else`**-**`if`**-**`else`**-... statements, to cover all 5 cases.

4. Metric Converter

Write a program, which converts distance between the following 8 measuring units: **`m`**, **`mm`**, **`cm`**, **`mi`**, **`in`**, **`km`**, **`ft`**, **`yd`**. Use the table below:

Input unit	output unit
1 meter (**m**)	**1000** millimeters (**mm**)
1 meter (**m**)	**100** centimeters (**cm**)
1 meter (**m**)	**0.000621371192** miles (**mi**)
1 meter (**m**)	**39.3700787** inches (**in**)
1 meter (**m**)	**0.001** kilometers (**km**)
1 meter (**m**)	**3.2808399** feet (**ft**)
1 meter (**m**)	**1.0936133** yards (**yd**)

The input consists of three lines:

- First line: number to convert – **real number**
- Second line: input unit - **text**
- Third line: output unit - **text**

Print the **result of converting the measurement units** formatted **to 10 decimal places**.

Examples

input	output
12 km ft	39370.0788000000
150 mi in	9503999.9939359918
450 yd km	0.4114799994

5. Time + 15 Minutes

Write a program, which reads **hours and minutes** from the 24-hour day and calculates **what the time will be after 15 minutes**. The result should be printed in the format **hh:mm**. The hours will always **be between 0 and 23**, the minutes will always **be between 0 and 59.** The hours will be written with **one or two digits**, whereas the minutes will be written **always with two digits (with a leading zero if necessary).**

Examples

input	output
1 46	2:01
0 01	0:16
23 59	0:14
11 08	11:23
12 49	13:04

Hint: add 15 minutes and make several checks. If the minutes become more than 59, increment the hours by 1 and subtract 60 from the minutes. Similarly cover the case when the hours become more than 23. When printing the minutes check for a leading zero.

6. Choreography

A group of dancers is rehearsing for a final contest. They must learn new **choreography**. The dance consists of **N steps**, which are being **distributed between the dancers**. The whole choreography must be learned **for a certain number of days**. All dancers can memorize **no more than 13% of the total steps per day**. Write a program, which calculates whether the dancers are **going to manage to memorize the new dance** and **what percent of the steps should each dancer memorize**.

When calculating the percent of steps per day, round the number to the bigger integer.

Input

The input consists of **3 lines**:

1. **Number of steps - an integer in the range [1 ... 100 000]**
2. **Number of dancers - an integer in the range [1 ... 50]**
3. **Number of days to rehearse – an integer in the range [1 ... 31]**

Output

What you print on the console depends on the result:

- If the total percentage of the steps is **less than or equal to 13%**, print:
  - **"Yes, they will succeed in that goal! {the percent of steps which every dancer should memorize per day}%."**
- If the total percentage of steps **is more than 13%**, print:
  - **"No, they will not succeed in that goal! Required {the percent of steps which every dancer should memorize per day}% steps to be learned per day."**

Both outputs should be **formatted to two decimal places.**

Examples

Input	Output	Explanation
10464 20 20	`Yes, they will succeed in that goal! 0.25%.`	Steps per day: (10464 / 20) / 10464 = 5% Percent of steps per dancer: 5 / 20 = 0.25%
55555 30 7	`No, they will not succeed in that goal! Required 0.50% steps to be learned per day.`	Steps per day: (55555 / 7) / 55555 = 14.28..% = 15% Percent of steps per dancer: 15 / 30 = 0.50%

7. World Swimming Record

Ivan has decided to beat the world swimming record. **You must read the record in seconds, which Ivan has to beat, the distance in meters, which he has to swim and the time in seconds per meter.** Write a program, which calculates **whether he will succeed**, having in mind that **the water resistance will slow him down with 12.5 seconds for every 15 meters** he swims. Calculate the **time in seconds**, for which Ivan will swim the distance and the difference from the world record.

When calculating Ivan's slowdown as a result of the water resistance, round the result to the smaller integer.

Input

The input consists of **3 lines**:

1. **The record in seconds - a real number in the range [0.00 ... 100000.00]**
2. **The distance in meters - a real number in the range [0.00 ... 100000.00]**
3. **The time in seconds per meter - a real number in the range [0.00 ... 1000.00]**

Output

What you print on the console depends on the result:

- If Ivan **managed to beat the World Swimming Record**, print:
  - `" Yes, he succeeded! The new world record is {Ivan's time} seconds."`
- If Ivan **did NOT manage to beat the World Swimming Record**, print:
  - `"No, he failed! He was {the difference between his time and the record} seconds slower."`

The result should be formatted to two decimal places.

Examples

Input	**Output**	**Hints**
10464 1500 20	No, he failed! He was 20786.00 seconds slower.	Ivan’s Time(seconds): 1500 * 20 = 30000 Slowdown(seconds): parseInt(1500 / 15) * 12.5 = 1250 Ivan’s Time w/water resistance: 30000 + 1250 = 31250 Ivan’s Total Time > 10464 31250 - 10464 = 20786.00
Input	**Output**	
55555.67 3017 5.03	Yes, he succeeded! The new world record is 17688.01 seconds.	Ivan’s Time(seconds): 3017 * 5.03 = 15175.51 Slowdown(seconds): parseInt(3017 / 15) * 12.5 = 2512.5 Ivan’s Time w/water resistance: 15175.51 + 2512.5 = 17688.01 Ivan’s Total Time < 55555.67 17688.01