**Exercise: For Loop**

Problems for exercise and homework for the "**Programming Basics**" course [@ SoftUni Global](https://softuni.org).

**Submit** your solutions in the **SoftUni Judge** system at: <https://judge.softuni.org/Contests/Compete/Index/3547>

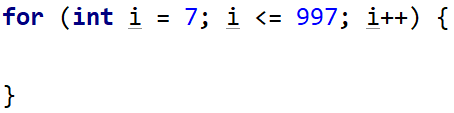
## Numbers Ending in 7

Write a program that prints numbers **in the range 1 to 1000** that end in **7**.

|  |  |
| --- | --- |
| **Input** | **Output** |
| (no input) | 7  17  27  …  997 |

### Hints and Guidelines

1. Make a **for loop** to **print a number** if it **ends in 7**. The initial step from which you can **start is 7 and the final one is 997**. The first number in your series is **7** and the last is **997**.



1. To print numbers **ending in 7**, you must check that the **current number ends in 7**. To do this, use a **modular division of 10** and print the number if **the result of this division is 7**.

**\* Note: With a modular division of 10 you can find the last digit of any integer, regardless of its length.**

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#0>

## Half Sum Element

Write a program that reads **n-number integers** entered by the user and checks if there is a number among them which is equal to the sum of all the others.

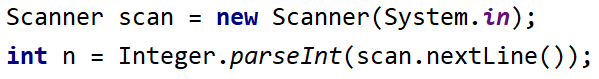
* If there is such an element, print "Yes" and on a new line "Sum = " + **its value**
* If there is no such element, print "No" and on a new line "Diff = " + **the difference between the largest number and the sum of the other** (in absolute value)

### Sample Input and Output

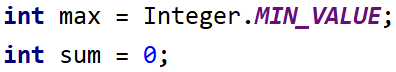
|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 7  3  4  1  1  2  **12**  1 | Yes  Sum = 12 | 3 + 4 + 1 + 2 + 1 + 1 = 12 |
| 4  **6**  1  2  3 | Yes  Sum = 6 | 1 + 2 + 3 = 6 |
| 3  1  1  **10** | No  Diff = 8 | |10 - (1 + 1)| = 8 |
| 3  5  5  1 | No  Diff = 1 | |5 - (5 + 1)| = 1 |
| 3  1  1  1 | No  Diff = 1 |  |

### Hints and Guidelines

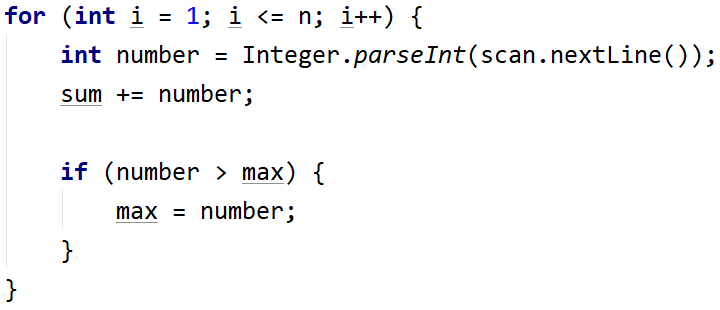
1. Read the input **n** (the number of the input numbers):



1. Create two **new variables**. In one **calculate the** **maximum number**, in the other the **sum of all numbers**. Set **initial values** for both variables. Set the maximum variable to **Integer.MIN\_VALUE**, and the sum variable to **0**.



1. Make a **for loop** **to read n numbers**. Each time you read a number, **add it to the amount**. Check if **the number entered is greater than the current maximum**. If it is **greater**, the **maximum** becomes **equal to this number**.



1. Once you have the **sum of all the numbers** and the **maximum number among them**, find the sum of the numbers **without the maximum number**. Create a new variable in which to calculate **the amount without the maximum number**.



1. 5. Check if the sum without the maximum number is equal to the **maximum number**, print two lines of output - (**Yes…) and the maximum numbe**r, or - **(No…) and the absolute value of the difference** between the maximum number and the sum of all numbers without the maximum (use **Math.abs** to find the **absolute value**).

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#1>

## Histogram

**N** integers are given in the interval [1… 1000]. Some of these, percentage **p1,** are below 200, another percentages **p2** are from 200 to 399, another percentages **p3** are from 400 to 599, other percentages **p4** are from 600 to 799 and the remaining **p5** percent are above 800. Write a program that calculates and prints the percentages **p1**, **p2**, **p3**, **p4**, and **p5**.

**Example:** we have n = **20** numbers: 53, 7, 56, 180, 450, 920, 12, 7, 150, 250, 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. We get the following distribution and visualization:

|  |  |  |  |
| --- | --- | --- | --- |
| **Range** | **Numbers in the range** | **Numbers** | **Percentage** |
| < 200 | 53, 7, 56, 180, 12, 7, 150, 2, 199, 46, 128, 65 | 12 | p1 = 12 / 20 \* 100 = **60.00**% |
| 200 … 399 | 250, 200 | 2 | p2 = 2 / 20 \* 100 = **10.00**% |
| 400 … 599 | 450 | 1 | p3 = 1 / 20 \* 100 = **5.00**% |
| 600 … 799 | 680, 600, 799 | 3 | p4 = 3 / 20 \* 100 = **15.00**% |
| ≥ 800 | 920, 800 | 2 | p5 = 2 / 20 \* 100 = **10.00**% |

### Input Data

The first line of the input is the integer n (1 ≤ **n** ≤ 1000) - count of numbers. The next **n lines** contain an **integer in the interval** [**1… 1000**] - the numbers with which to calculate the histogram.

### Output Data

Print the **histogram** on the console - **5 lines**, each of which contain a number between 0% and 100%, formatted two digits after the decimal point, for example 25.00%, 66.67%, 57.14%.

### Sample Input and Output

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| **3**  1  2  999 | 66.67%  0.00%  0.00%  0.00%  33.33% | **4**  53  7  56  999 | 75.00%  0.00%  0.00%  0.00%  25.00% | **7**  800  801  250  199  399  599  799 | 14.29%  28.57%  14.29%  14.29%  28.57% | **9**  367  99  200  799  999  333  555  111  9 | 33.33%  33.33%  11.11%  11.11%  11.11% | **14**  53  7  56  180  450  920  12  7  150  250  680  2  600  200 | 57.14%  14.29%  7.14%  14.29%  7.14% |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#2>

## Clever Lily

Lily is already N years old. She receives a gift for each of her **birthdays**.

* For the **odd** birthdays (**1, 3, 5...n**), she receives toys**.**
* For the **even** birthdays (**2, 4, 6...n**), she receives money.

For the **second birthday** she receives **10.00 USD**, as the amount increases by **10.00 USD**, for each subsequent even birthday (**2** -> **10**, **4** -> **20**, **6** -> **30** ... etc.). Over the years, Lily has secretly saved money. Lily's brother, in the years she receives money, **takes 1.00 USD from them**. Lily sold the toys received over the years, each for **p** USD, and **added the amount to the saved money**. With the money, she wanted to buy a washing machine for **x** USD. Write a program to calculate how much money **she has raised** and whether she **has enough to buy a washing machine**.

### Input Data

3 lines are read from the console:

* **Lily’s age – an integer in the range** **[1...77]**
* **Price of washing machine – a floating-point number in the range** **[1.00...10 000.00]**
* **Single price of a toy – an integer in the range [0...40]**

### Output Data

On the console print:

* If Lily **has enough** money:
  + **"**Yes! {N}**"** - where **N** is the remaining money after the purchase
* If the money is **not enough**:
  + **"No! {М}"** - where **M** is the amount that is not enough

**The numbers N and M must be formatted to the second decimal place.**

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 10  **170.00**  **6** | Yes! 5.00 | On her first birthday, she receives a **toy**  On her second birthday, she receives money (10 USD)  She receives a **toy** on her third birthday  On her fourth birthday, she receives **money** (20 USD)  She gets a **toy** on her fifth birthday  On her sixth birthday receives **money** (30 USD)  She receives a **toy** on her seventh birthday  On her eighth birthday, she receives **money** (40 USD)  On her ninth birthday, she receives a **toy**  On her tenth birthday, she receives **money** (50 USD)  **She saved**: 10 + 20 + 30 + 40 + 50 = **150** USD  She sold 5 toys and got **\* 6** USD = **30** USD  Her brother took 5 times **\* 1** USD = **5** USD  Total money saved: (**150** + **30**) – **5** = **175** USD  **175 >= 170** (the price of the washing machine)  => she managed to buy it  **Remaining:** **175** - **170** = **5** USD |
| 21  **1570.98**  **3** | No! 997.98 | She saved **550** USD.  She sold **11** toys \* **3** USD = **33** USD  Her brother took **10** years \* **1** USD = **10** USD  Total money saved**:** (**550** + **33**) – **10** = **573** USD  **573 < 1570.98** – failed to buy a washing machine  Needed money**:** **1570.98** – **573** = **997.98** USD |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#3>

## Salary

**A company boss notices that more and more employees are spending time on sites that distract them.**

**To prevent this, he introduces surprise checks on the open tabs of his employees' browsers.**

**According to the open site, the following fines are imposed in the tab:**

**• "Facebook" -> 150 USD**

**• "Instagram" -> 100 USD**

**• "Reddit" -> 50 USD**

**Two lines are read from the console:**

**• Number of open tabs in the browser n - integer in the range [1 ... 10]**

**• Salary - a number in the interval [500 ... 1500]**

**Then n-times - the name of the website - text (string)**

### Output Data

* **If during the inspection, the salary becomes less than or equal to USD0, it is printed on the console  
  "You have lost your salary."** and the program stops.
* **Otherwise, after checking the console, the rest of the salary is displayed (to be written as an integer).**

### Sample Input and Output

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | |
| **10**  **750**  Facebook  Dev.bg  Instagram  Facebook  Reddit  Facebook  Facebook | You have lost your salary. | There are **10** open tabs in the browser.  The initial salary is: **750**  For the first tab -> Facebook -> fine **150** USD (salary: **750** – **150** = **600**)  For the second tab -> Dev.bg -> no fine  For the third tab -> Instagram -> fine **100** USD (salary: **600** – **100** = **500**)  For the fourth tab -> Facebook -> fine **150** USD (salary: **500** – **150** = **350**)  For the fifth tab -> Reddit -> fine **50** USD (salary: **350** – **50** = **300**)  For the sixth tab -> Facebook -> fine **150** USD (salary: **300** – **150** = **150**)  For the seventh tab -> Facebook -> fine **150** USD (salary: **150** – **150** = **0**)  The salary is equal to **0** => the corresponding output is printed and the program ends | |
| **Input** | **Output** | **Input** | **Output** |
| 3  500  Github.com  Stackoverflow.com  softuni.bg | 500 | 3  500  Facebook  Stackoverflow.com  softuni.bg | 350 |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#4>

## Oscars

You are invited by the academy to write software to calculate the points for an actor / actress. The academy will give you initial **points for the actor**. Then each **evaluator** will give **his evaluation**. The points that the **actor receives** are formed with the length of the evaluator's name multiplied by the points he gives divided by two.

**If the result at some point exceeds 1250.5 the program must be interrupted, and it must be printed that the actor has received a nomination.**

### Input Data

* Actor's name- **string**
* Points from the academy – **a floating-point number in the range [2.0... 450.5]**
* Number of evaluators **n** – **integer in the range [1… 20]**

On the next **n number of rows**:

* Name of evaluator - **string**
* Evaluator points – **a floating-point number in the range [1.0... 50.0]**

### Output Data

On the console print:

* If the points are above **1250.5**:

**"Congratulations, {actor’s name} got a nominee for leading role with {points}!"**

* If the points are not enough:

**"Sorry, {actor’s name} you need {needed points} more!"**

**The result should be formatted to the first digit after the decimal point!**

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| Mark Zuckerberg  **205**  **4**  Johnny Depp  **45**  Will Smith  **29**  Jet Lee  **10**  Matthew Mcconaughey  39 | Sorry, Mark Zuckerberg you need 247.5 more! | Mark Zuckerberg starts with **205** points, and **4** people will rate him.  First is Johnny Depp  => **205** + ((11 \* **45**) / 2) = **452.5**  Second is Will Smith  => **452.5** + ((10 \* **29**) / 2) = **597.5**  Third is Jet Lee  => **597.5** + ((7 \***10**) / 2) = **632.5**  Fourth is Matthew Mcconaughey  => **632.5** + ((19 \* **39**) / 2) = **1003.0**  **1003.0** < **1250.5**  => The points are not enough.  Neded points: **1250.5** - **1003.0** = 247.5 |
| Sandra Bullock  340  5  Robert De Niro  50  Julia Roberts  40.5  Daniel Day-Lewis  39.4  Nicolas Cage  29.9  Will Smith  33 | Congratulations, Sandra Bullock got a nominee for leading role with 1268.5! |  |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#5>

## 7. Trekking Mania

Climbers from all over the world gather in groups and mark the next peaks to climb. Depending on the **size of the group**, the climbers will climb **different peaks**.

* Group of up to **5 people** – climb **Mount Funi**
* Group of **6 to 12 people** – climb **Mont Blanc**
* Group of **13 to 25** **people** – climb **Kilimanjaro**
* Group of **26 to 40** **people** – climb **K2**
* Group of **41 or more people** – climb **Everest**

Write a program that calculates the percentage of climbers climbing each peak**.**

### Input Data

From the console read:

* **On the first row** – **number of groups of climbers** – **an integer in the range [1...1000]**
* **For each group print the number of the climbers** – **an integer in the range [1...1000]**

### Output Data

5 rows are printed on the console, each containing a **percentage between 0.00% and 100.00%, formatted to the second digit after the decimal point**.

* **First row** – the percentage of people climbing **Mount Funi**
* **Second row** – the percentage of people climbing **Mont Blanc**
* **Third row** – the percentage of people climbing **Kilimanjaro**
* **Fourth row** – the percentage of people climbing **K2**
* **Fifth row** – the percentage of people climbing **Everest**

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 10  10  5  1  100  12  26  17  37  40  78 | 1.84%  6.75%  5.21%  31.60%  54.60% | All climbers: 10 + 5 + 1 + 100 + 12 + 26 + 17 + 37 + 40 + 78 = 326  Climbing Mount Fuji: 6 / 326 \* 100 = 1.84%  Climbing Mont Blanc: 22/326\*100 = 6.75%  Climbing Kilimanjaro: 17/326\*100 = 5.21%  Climbing K2: 103/326\*100 = 31.60%  Climbing Everest: 178/326\*100 = 54.60% |
| **Input** | **Output** | **Comments** |
| 5  25  41  31  250  6 | 0.00%  1.70%  7.08%  8.78%  82.44% | All climbers: 25 + 41 + 31 + 250 + 6 = 353  Climbing Mount Fuji: 0 / 353 \* 100 = 0.00%  Climbing Mont Blanc: 6 / 353 \* 100 = 1.78%  Climbing Kilimanjaro: 25 / 353 \* 100 = 7.08%  Climbing K2: 31 / 353 \* 100 = 8.78%  Climbing Everest: 291 / 353 \* 100 = 82.44% |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#6>

## 8. Tennis Ranklist

Peter Parker is a tennis player whose next goal is to rise in the world rankings in men's tennis.

During the year Peter **participates in several tournaments**, receiving points for each tournament, which depend on the position in which he finished in the tournament. There are **three options** for completing a tournament:

* **W - if he is a winner, he receives 2000 points**
* **F - if he is a finalist, he receives 1200 points**
* **SF - if he is a semifinalist, he gets 720 points**

Write a program that calculates **how many points will Peter have after playing all the tournaments**, knowing how many **points the season starts with**. Also, calculate **how many points he won on average** in all tournaments played and **what percentage** of tournaments he has **won**.

### Input Data

2 lines are read from the console:

* **Number of tournaments in which he participated - an integer in the range [1…20]**
* **The initial number of points in the rankings - an integer in the range [1...4000]**

For each tournament a row is read from the console**:**

* **Reached tournament stage – string –** "**W**"**,** "**F**", **or** "**SF**"

### Output Data

3 lines are printed on the console:

* "**Final points: {number of points after playing the tournament}**"
* "**Average points: {average points earned per tournament}**"
* "**{percentage of won tournaments}%**"

**The average points should be rounded down to the nearest whole number, and the percentage should be formatted to the second digit after the decimal point.**

### Sample Input and Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | | |
| **5**  **1400**  **F**  **SF**  **W**  **W**  **SF** | Final points: 8040  Average points: 1328 40.00% | **5 tournaments and starting number of points: 1400**  **1st tournament -> final (F) -> points = 1400 + 1200 = 2600**  **2nd tournament -> semifinal (SF) -> points = 2600 + 720 = 3320**  **3rd tournament -> winner (W) -> points = 3320 + 2000 = 5320**  **4th tournament -> winner (W) -> points = 5320 + 2000 = 7320**  **5th tournament -> semifinal (SF) -> points = 7320 + 720 = 8040**  **Points after playing tournaments: 8040**  **Average tournament points earned:**  **(1200 + 720 + 2000 + 2000 + 720) / 5 = 6640 / 5 = 1328**  **Number of won tournaments: 2**  **Percentage of tournaments won: (2 / 5) \* 100 = 40 %** | | |
| **Input** | **Output** | **Input** | **Output** |
| 4  750  SF  W  SF  W | Final points: 6190  Average points: 1360  50.00% | 7  1200  SF  F  W  F  W  SF  W | Final points: 11040  Average points: 1405  42.86% |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3547#7>