# Exercise: While Loops

## 1. Clock

Write a program that prints the hours of the day from a given hour and minutes to 23:59 (inclusive), each on a separate line.

### Input

The input is consists of 2 lines:

- First line - hours - integer in range [0…23]
- Second line - minutes - integer in range [0…59]

### Output

- Hours must be printed on the console in "{hour} : {minutes}" format.

### Example

| Input | Output |
|-------|--------|
| 8 | 08 : 05 |
| 5 | 08 : 06 |
| | ... |
| | 23 : 57 |
| | 23 : 58 |
| | 23 : 59 |

## 2. Clock - Part 2

Write a program to print the hours of the day from a given hour, minutes, seconds to 23:59:59 (inclusive), each on a separate line.

### Input

The input is entered from the console as 3 lines:

- First line - hours - integer in range [0…23]
- Second line - minutes - integer in range [0…59]
- Third line - seconds - integer in range [0…59]

## Output

- The output should be printed in the following format "{hour} : {minutes} : {seconds}".

## Example

| Input | Output |
|-------|--------|
| 21 | 21 : 59 : 58 |
| 59 | 21 : 59 : 59 |
| 58 | 22 : 00 : 00 |
|  | 22 : 00 : 01 |
|  | ... |
|  | 23 : 59 : 58 |
|  | 23 : 59 : 59 |

# 3. Combinations

Write a program that calculates how many solutions in natural numbers (including zero) has the equation:

**x1** \+ **x2** \+ **x3** \+ **x4** \+ **x5** = **n**

## Input

- The input consists a single number - n - integer

## Output

- Print on the console the total number of solutions

## Example

| Input | Output | Comments |
|-------|--------|----------|
| 25 | 23751 | Generate all combinations of 5 digits, the first is: |
|  |  | 0+0+0+0+0=0, but because it is not equal to 25, |
|  |  | continuing 0+0+0+0+1=1 - again, it's not 25, etc. |

| | | We come to the first valid combination: |
| --- | --- | --- |
| | | 0+0+0+0+25=25, expanding the number of valid |
| | | combinations of 1, the second valid combination is: |
| | | 0+0+0+1+24=25 |
| | | The third: 0+0+0+2+23=25 etc. |
| | | After generating all possible combinations, |
| | | the number of valid combinations is 23751. |

## 4. Word Wars

Write a program that calculates the ASCII value of a few words, as the word with the highest value is the winner.

The value of a word is calculated when you sum the ASCII values of all the letters of which it consists of.

From the console read words until the command "STOP", then print:

- "Winner is {winner word} – {value of the word}!"

## Example

| Input | Output |
| --- | --- |
| House | Winner is Destination - 1154! |
| Dog | |
| Destination | |
| STOP | |

### Comments

- The first letter is 'H' and it's ASCII value is 72, 'o' has value 111, 'u' has value 117, 's' has value 115, 'e' has value 101. Their sum is 516.
- Keep doing the same calculations for the other words and get that Destination has the highest value - 1154.

## 5. Coding

Write a program that reads an integer from the console. The console should print as many lines as digits as numbers. As the first line of the corresponding units, of the second - tens, third order - hundreds of numbers, etc., as there are no more digits of the number.

Each line you must print symbol, which satisfies the following conditions:

© Kingsland – https://kingslanduniversity.com. Unauthorized copy, reproduction or use is not permitted.

KINGSLAND
UNIVERSITY
Follow us:

Page PAGE \*
MERGEFORMAT 1
of NUMPAGES \*
MERGEFORMAT 1

- The symbol, which must be printed on a line, is located on the [ASCII](http://www.asciitable.com/) table. Its decimal ASCII code is calculated as adding 33 to the digit of the input number that corresponds to a given row
- The symbol shall be printed as many times as the digit corresponding to this line
- If for a given row matches a digit 0, on this line is printed "ZERO" once

## Example

| Input | Output |
|-------|--------|
| 2049 | ******** |
|  | %%% |
|  | ZERO |
|  | ## |

## Comments

The number 2049 has four digits so we will print four rows.

The first line corresponds to the digit 9. We add 33 to 9 and get 42.

This is the decimal ASCII code of the symbols it should print on the first row.

From the ASCII table we find out that of the 42 corresponds symbol '*'.

Because the first line corresponds with digit 9 we print 9 times '*'.

For the second line the digit is 4. 4+33=37. Using ASCII table, we find that the symbol to print is '%'. We print 4 times '%'.

On the third line matches a digit 0. In this case we do NOT searching anything in ASCII table and on this line we print ZERO once.

The last digit of the number is 2. 2+33=35.

From the ASCII table we found the symbol to print - '#' and we print it twice.

## Hints

- Read the number as a String, save its length in a variable using the length() method. Find more information about it on the Internet
- To take the last digit of the number, divide it modularly by 10 (num % 10) and save it in a variable. Then remove the last digit of the number, dividing it by 10 (num / 10) for the next time again to grab the last digit.

# 6. Train the Trainers

The course "Train the trainers" is ending and the final evaluation approaches.

Help the jury by writing a program to calculate the average score of each presentation by a student, and the average of all of them.

# Input

- From the console of the first row read the number of people on the jury - n - an integer in the range [1...20]

- Then on a separate line read the name of the presentation - String

- For each presentation of the new line is read n - the number of ratings - real number in the interval [2.00 ... 6.00]

# Output

- After calculating the average score for a particular presentation, print to the console:
  "{name of the presentation} - {average score}."

- After receiving the command "Finish" on the console, print:
  "Student's final assessment is {average presentations of all presentations}." and the program ends.

All scores must be formatted to the second decimal point.

# Example

| Input | Output |
|---|---|
| 2 | While-Loop - 5.75. |
| While-Loop | For-Loop - 5.75. |
| 6.00 | Student's final assessment is 5.75. |
| 5.50 | |
| For-Loop | |
| 5.84 | |
| 5.66 | |
| Finish | |

# Comments

- 2 – the number of people on the jury - therefore gain a 2 evaluations of the presentation:

  - (6.00 + 5.50) / 2 = 5.75

  - (5.84 + 5.66) / 2 = 5.75

- (6.00 + 5.50 + 5.84 + 5.66) / 4 = 5.75