# 1 Exchange numbers

## Description

Write a program that reads two double values from the console A and B, stores them in variables and exchanges their values if the first one is greater than the second one. Use an if-statement. As a result print the values of the variables A and B, separated by a space.

## Input

- On the first line, you will receive the value of A
- On the second line, you will receive the value of B

## Output

- On the only output line, print the values of the two variables, separated by a whitespace

## Constraints

- A and B will always be valid real numbers between -100 and 100
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| 5<br>3 | 3 5 |
| 2<br>4 | 2 4 |

| | |
|---|---|
| 3.3<br>3.3 | 3.3 3.3 |

# 2 Bonus Score

## Description

Write a program that applies bonus score to given score in the range [1…9] by the following rules:

- If the score is between 1 and 3, the program multiplies it by 10.
- If the score is between 4 and 6, the program multiplies it by 100.
- If the score is between 7 and 9, the program multiplies it by 1000.
- If the score is less than 0 or more than 9, the program prints "invalid score".

## Input

- The only input line will contain one integer number - the score

## Output

- Output the score with the applied bonus

## Constraints

- The score will always be a valid integer number
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
|---|---|

| | |
|---|---|
| 2 | 20 |
| 4 | 400 |
| 9 | 9000 |
| 10 | invalid score |

# 3 Play card

## Description

Classical play cards use the following signs to designate the card face: `2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K and A`. Write a program that enters a string and prints "yes CONTENT_OF_STRING" if it is a valid card sign or "no CONTENT_OF_STRING" otherwise.

## Input

- On the only line you will receive a single string.

## Output

- Output "yes CONTENT_OF_STRING" if the string is a card sign, otherwise print "no CONTENT_OF_STRING".

## Constraints

- String length will always be between 1 and 5
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| 5 | yes 5 |
| 1 | no 1 |
| Q | yes Q |
| q | no q |
| P | no P |
| 10 | yes 10 |
| 500 | no 500 |

# 4 Multiplication sign

## Description

Write a program that shows the sign (+, - or 0) of the product of three real numbers, without calculating it.

- Use a sequence of `if` operators.

## Input

- On the first three lines, you will receive the three numbers, each on a separate line

## Output

- Output a single line - the sign of the product of the three numbers

## Constraints

- The input will always consist of valid floating-point numbers
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| 2<br>5<br>2 | + |
| 2<br>5<br>-2 | - |
| -0.5<br>0<br>-2 | 0 |

# 5Biggest of 3

## Description

Write a program that finds the biggest of three numbers that are read from the console.

## Input

- On the first three lines you will receive the three numbers, each on a separate line.

## Output

- On the only output line, write the biggest of the three numbers.

## Constraints

- The three numbers will always be valid floating-point numbers in the range `[-200, 200]`.
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| 4<br>4<br>4 | 4 |
| -0.5<br>-10<br>0 | 0 |

# 6 Biggest of 5

## Description

Write a program that finds the biggest of 5 numbers that are read from the console, using only 5 if statements.

## Input

- On the first 5 lines you will receive the 5 numbers, each on a separate line

## Output

- On the only output line, write the biggest of the 5 numbers

## Constraints

- The 5 numbers will always be valid floating-point numbers in the range `[-200, 200]`
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| 4<br>4<br>4<br>4<br>4 | 4 |
| -0.5<br>-10<br>0<br>-1<br>-3 | 0 |

# 7 Sort 3 Numbers

## Description

Write a program that enters 3 real numbers and prints them sorted in descending order.

- Use nested `if` statements.
- *Don't use arrays and the built-in sorting functionality.*

## Input

- On the first three lines, you will receive the three numbers, each on a separate line.

## Output

- Output a single line on the console - the sorted numbers, separated by a whitespace

## Constraints

- The three numbers will always be valid integer numbers in the range `[-1000, 1000]`
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| 3<br>2<br>1 | 3 2 1 |
| -5<br>3<br>-5 | 3 -5 -5 |
| 1<br>2<br>20 | 20 2 1 |

# 8 Digit as Word

## Description

Write a program that read a digit [0-9] from the console, and depending on the input, shows the digit as a word (in English).

- Print "not a digit" in case of invalid input.
- Use a switch statement.

## Input

- The input consists of one line only, which contains the digit.

## Output

- Output a single line - should the input be a valid digits, print the english word for the digits. Otherwise, print "not a digit".

## Constraints

- The input will never be an empty line.
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| -0.1 | not a digit |
| 1 | one |
| 9 | nine |
| -9 | not a digit |

| | |
|---|---|
| kek | not a digit |

# 9 Int, Double, String

## Description

Write a program that, depending on the first line of the input, reads an int, double or string variable.

- If the variable is int or double, the program increases it by one.
- If the variable is a string, the program appends * at the end.
- Print the result at the console. Use switch statements.

## Input

- On the first line you will receive the type of input as string in the following form:
  - integer for int
  - real for double
  - text for string
- On the second line you will be given the value of the variable.

## Output

- Write a single line on the console - the value transformed according to the rules from the description.
  - Print all double variables with exactly 2-digits precision after the floating point. *Example:* 0.99

## Constraints

- The input will always be valid and in the described format.
- All input numbers will be between -1000 and 1000.
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

| Input | Output |
| --- | --- |
| integer<br>2 | 3 |
| real<br>-2.5 | -1.50 |
| text<br>gosho | gosho* |