# Lab: JS Basic Syntax, Conditional Statements, and Loops

Problems for exercise and homework for the ["JS Fundamentals" Course @ SoftUni.](https://softuni.bg/trainings/4382/programming-fundamentals-with-javascript-january-2024)   
Submit your solutions in the SoftUni judge system at: <https://judge.softuni.org/Contests/1189>

## Multiply Number by 2

Write a function that receives a number and **prints** as result that **number** **multiplied** by **two**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | 4 |
| 5 | 10 |
| 20 | 40 |

### Hints

Create a function called solve (or some other name). As parameters, it will receive a number num.  
 

Print the result inside the function.



If you want to test your code locally, you need to call the function.



## Student Information

You will be given **3 parameters** – student name (string), age (number), and average grade (number). Your task is to print all the info about the student in the following format:

**`**Name: {student name}, Age: {student age}, Grade: {student grade}**`**

**Note**: The grade should be formatted to the **second decimal** point.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'John', 15, 5.54678 | Name: John, Age: 15, Grade: 5.55 |
| 'Steve', 16, 2.1426 | Name: Steve, Age: 16, Grade: 2.14 |
| 'Marry', 12, 6.00 | Name: Marry, Age: 12, Grade: 6.00 |

### Hint

Use toFixed() method to format the grade.

1. First, receive the input:

A picture containing logo

Description automatically generated

1. Print the output:



## Excellent Grade

Write a function that receives a single number and checks if the grade is **excellent** or **not**.   
If it is, print "**Excellent**", otherwise print "**Not excellent**".

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5.50 | Excellent |
| 4.35 | Not excellent |

### Hints

Check if the number given is greater or equal to 5.50 and print the corresponding result.



## 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**.

### Output

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

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| USA | English |  | Germany | unknown |

### Hint

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

## Month Printer

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

### Input

You will receive a **single number**.

### Output

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

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 2 | February |  | 13 | Error! |

## 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 i**n** the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Day / Age** | **0 <= age <= 18** | **18 < age <= 64** | **64 < age <= 122** |
| **Weekday** | 12$ | 18$ | 12$ |
| **Weekend** | 15$ | 20$ | 15$ |
| **Holiday** | 5$ | 12$ | 10$ |

### Input

The input comes in **two parameters**. The **first** one will be the **type of day (string)**. The **second** – the **age** of the person (number).

### 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** |  | **Input** | **Output** |  | **Input** | **Output** |  |
| 'Weekday',  42 | 18$ |  | 'Holiday', -12 | Error! | 'Holiday', 15 | 5$ |  |

## Numbers from 1 to 5

Write a function that **prints** all the **numbers** from **1** **to** **5** (inclusive) each on a separate line.

### Hints

Create a for loop starting from 1 and continuing until 5 and print the number.



## 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.

## Numbers from N to 1

Write a function that receives a number **N** and prints all the numbers from **N** **to 1**. Try using the while loop.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 5  4  3  2  1 |
| 3 | 3  2  1 |

### Hints

Create a while loop with condition **N >= 1**. Print **N** and decrease it with each step.



## Numbers from M to N

Write a function that receives a number **M** and a number **N** (M will always be bigger than N). Print all numbers from **M to N**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6, 2 | 6 5  4  3  2 |
| 4, 1 | 4  3  2  1 |

### Hints

Use for or while loop and print the numbers.



## 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

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 **newlines**.

On the last line, print the **sum** of these numbers in the following format: **`Sum: {total sum}`**

### Constraints

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

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 5 | 1  3  5  7  9  Sum: 25 |  | 3 | 1  3  5  Sum: 9 |