# Lab: Basic Syntax, Conditional Statements and Loops

## Greet by name

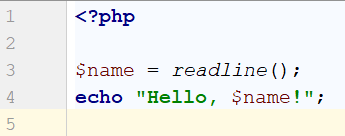
Write a program that reads a name from the console and prints "Hello, {name}!"

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| PHP | Hello, PHP! |
| SoftUni | Hello, SoftUni! |
| Sofia | Hello, Sofia! |

### Hints

Tell your program you're writing php code by writing the php tag (<?php). Read from the console and print in the given format.



## Odd or Even

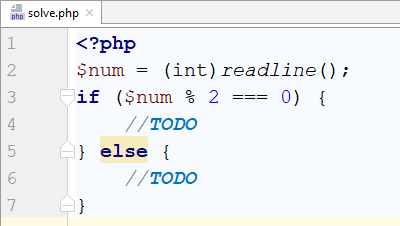
Write a program which checks if a given number is odd or even. If it's odd print "odd", if it's even -"even"

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 44 | even |
| 3 | odd |

### Hints

Check if the number given is divisible (%) by 2.



## Passed or Failed

Modify the above program, so it will print "**Failed**!" if the grade is **lower than 3.00**.

### Input

The **input** comes as a single double number.

### Output

The **output** is either "**Passed**!" if the grade is **more than 2.99**, otherwise you should print "**Failed**!".

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 5.32 | Passed! |  | 2.36 | Failed! |

### Hint

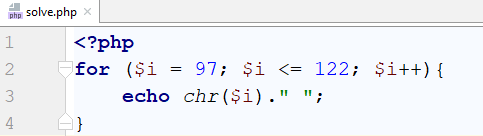
We need to take **floating-point** number from the console. After that print in the **else** statement the appropriate message.

## Alphabet

Create a program that print all small letters from the alphabet.

### Hints

Create a for loop starting from 97(the ascii code of 'a') and continuing until 122 and print the step casted to char



## Month Printer

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

### Input

You will receive a **single** **integer** on a **single line**.

### Output

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

### Examples

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

## Calculate Two Numbers

You will first start using PHP in CLI mode (console application). This means you will receive user input through the standard input and will return result into the standard output.

PHP, like most of the modern languages, supports argument passing on application start as well as waiting for user input.

### You are given two numbers from the standard input each on new line and the next line you are given the command. The commands will be sum, subtract, divide and multiply. If you have other command you should print - Wrong operation supplied.

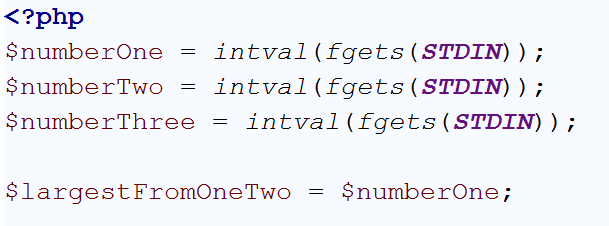
### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 2  3  sum | 6 |  | 13  7  subtract | 6 |
| 3  3  divide | 1 |  | 5  5  multiply | 25 |
| 0  5  divide | Cannot divide by zero |  | 7  4  dwada | Wrong operation supplied |

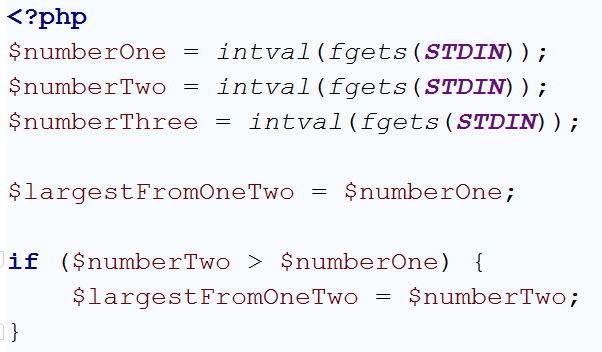
## Find Largest of Three Numbers

You are given three numbers from the standard input each on new line. Print on standard output the string “Max: “, followed by the largest number e.g. “Max: 4”.

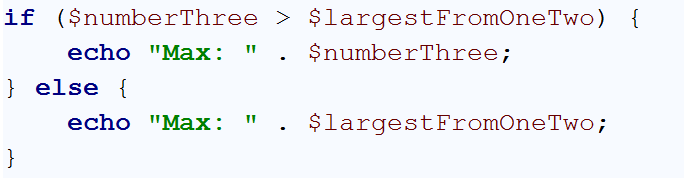
Let’s **find the largest number amongst the first and the second**. Let’s assume that the first number is the largest.



If the second is larger than the first**, assign** $largestFromOneTwo **the value of** $numberTwo**.**

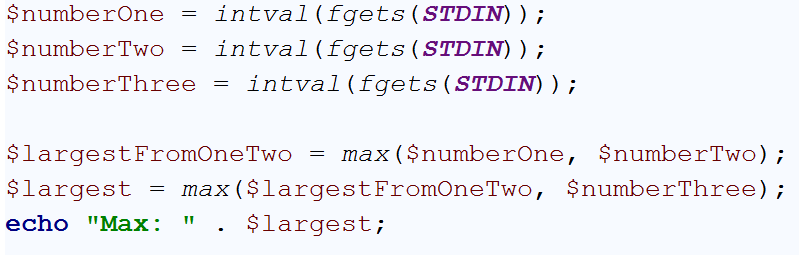


Now **the largest amongst all the numbers** is the largest **amongst the first two numbers and the third number**



Cool! It’s ready!

Of course there’s a built-in function that finds **max numbers between two numbers** and it has the **same performance as if/else**, so we will not kill the performance if we use it.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 2  3  7 | 7 |  | 1  2  3 | 3 |
| 10  20  30 | 30 |  | 57  13  99 | 99 |

## Reverse Strings

You will be given series of strings until you receive an **“end”** command. Write a program that reverses strings and printseach pair on separate line in format **"{word} = {reversed word}"**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| helLo  Softuni  bottle  end | helLo = oLleh  Softuni = inutfoS  bottle = elttob |
| Dog  caT  chAir  end | Dog = goD  caT = Tac  chAir = riAhc |

## Special Numbers

A **number** is **special** when its **sum of digits is 5, 7 or 11**.

Write a program to read an integer n and for all numbers in the range **1…n** to print the number and if it is special or not (True / False).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 15 | 1 -> False  2 -> False  3 -> False  4 -> False  5 -> True  6 -> False  7 -> True  8 -> False  9 -> False  10 -> False  11 -> False  12 -> False  13 -> False  14 -> True  15 -> False |

### Hints

To calculate the sum of digits of given number num, you might repeat the following: sum the last digit (num % 10) and remove it (sum = sum / 10) until num reaches 0.

## \*Repeat strings

### Write a program that reads an array of strings. Each string is repeated n times, where n is the length of the string. Print the concatenated string.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| hi abc add | hihiabcabcabcaddaddadd |
| work | workworkworkwork |
| ball | ballballballball |