MEC ACADEMY

PHP Development Course

MEC Academy

Agenda

- What is PHP
- PHP syntax
- Comments
- Variables and Constants
- Datatypes
- Common functions
- Operators
- conditional statements (if, switch)

What is PHP?

- ❑ PHP stands for PHP: Hypertext Preprocessor
- ❑ PHP is a widely-used, open-source scripting language
- ❑ PHP scripts are executed on the server
- ❑ PHP is free to download and use

What is PHP Files?

- ❑ PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- ❑ PHP code is executed on the server, and the result is returned to the browser as **plain HTML.**
- ❑ PHP files have extension "**.php**"

What PHP can do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data

MEC ACADEMY

Why PHP?

- PHP runs on different platforms (Windows, Linux,
- Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP has support for a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

MEC ACADEMY

PHP Syntax

- ❑ A PHP script can be placed anywhere in the document.
- ❑ A PHP script starts with **<?php** and ends with **?>**

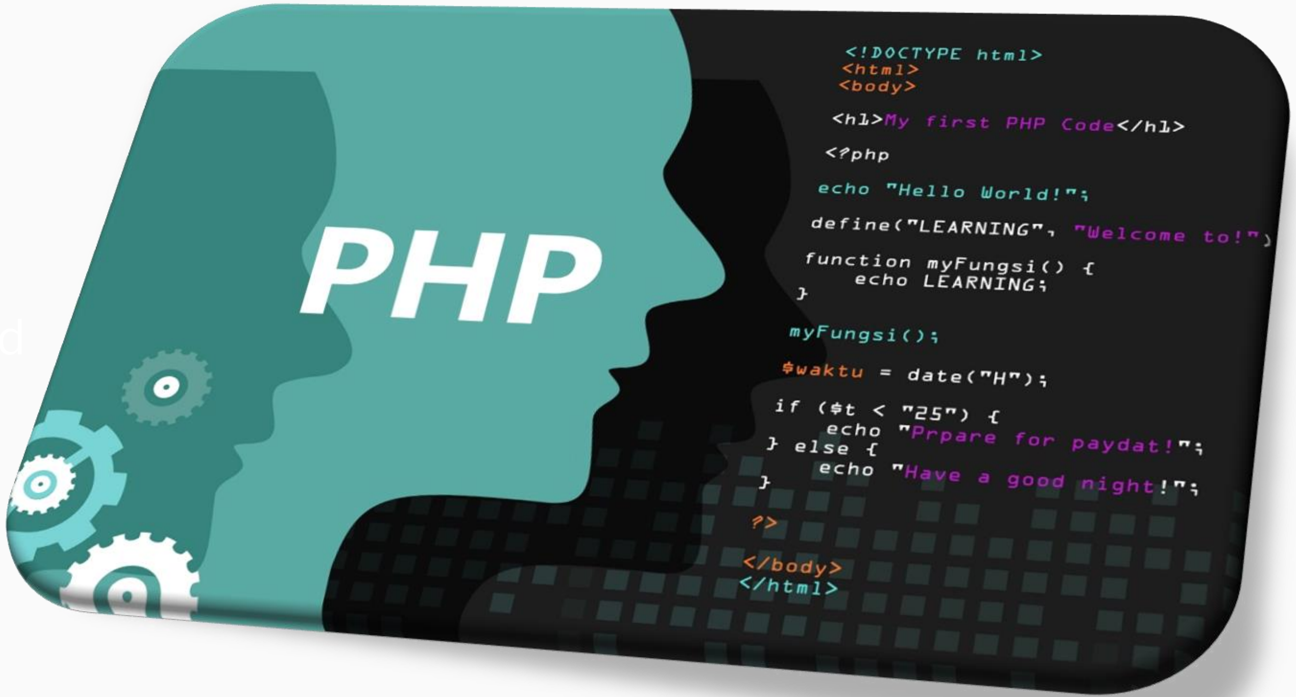
```
<?php
// PHP code goes here
?>
```

- ❑ PHP statements end with a semicolon (;)

What does Syntax mean?

Syntax is the set of rules that define what the various combinations of symbols mean.

This tells the computer how to read the code.



```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP Code</h1>
<?php
echo "Hello World!";
define("LEARNING", "Welcome to!")
function myFungsi() {
    echo LEARNING;
}
myFungsi();
$waktu = date("H");
if ($t < "25") {
    echo "Prpare for paydat!";
} else {
    echo "Have a good night!";
}
?>
</body>
</html>
```

PHP Comments

A comment in PHP code is a line that is not executed as a part of the program.

Its only purpose is to be read by someone who is looking at the code.

- **Single Line Comments** start with **//** or **#.**

```
// This is a single-line comment

# This is also a single-line comment
```

- **Multi-line Comments** start with **/*** and end with ***/.**

```
/* This is

 a multi-line

comment */
```

Printing output

There are 4 ways to print output in php by using:

- ❑ Echo
- ❑ Print
- ❑ Print_r()
- ❑ Var_dump()

The differences between print and echo are:-

- echo and print are both used to output data to the screen as a string.
- **echo** has no return value just output data to the browser while **print** has a return value of 1 so it can be used in expressions or assigned to variables.
- **echo** can take multiple parameters while **print** can take one argument.
- **echo** is marginally faster than **print**.

- **Echo** displays the outputs one or more strings separated by **commas**.
- **print_r()** outputs the detailed information about the parameter in a format with its type (of an array or an object), which can be easily understandable by humans.
- **var_dump()** is a function used to output information (i.e. **value** and **data types**) about a variable.
- **gettype():** function return only the datatype of variable.

Variables

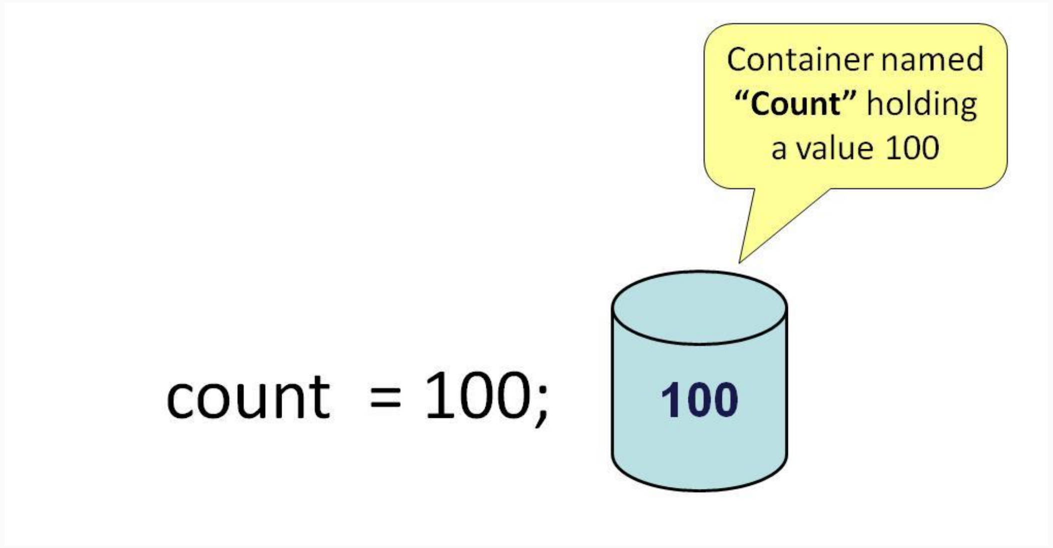
Arithmetic

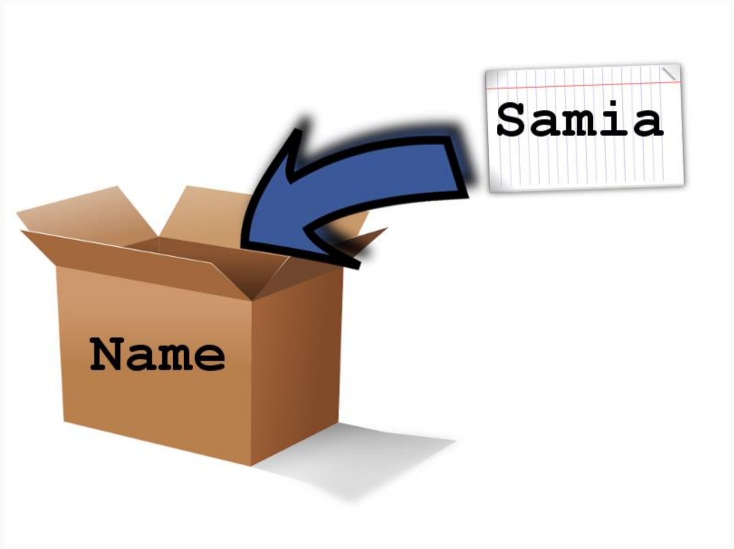
Variables

- Variables are containers for storing data values.
- Variable is a name of memory location.
- Variable is an identifier which holds data or another one.
- Variable is an identifier whose value can be changed at the execution time of program.

MEC ACADEMY

Variables


Container named **"Count"** holding a value 100
count = 100;
100


Samia
Name

Variables

In PHP, a variable starts with the $ sign, followed by the name of the variable

Rules:

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must **start** with **a letter** or **the underscore character**
- A variable name **cannot start with a number**
- A variable name **can only contain** alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are **case-sensitive** ($age and $AGE are two different variables)

Variables

- PHP has no command for declaring a variable.
- A variable is created the moment you first assign a value to it:
  - $txt="Hello world!";
  - $x=5;
- After the execution of the statements above, the variable txt will hold the value Hello world!, and the variable x will hold the value 5.
- **Note:** When you assign a text value to a variable, put quotes around the value.

PHP is a Loosely Typed Language

In the previous example, notice that we did not have to tell PHP which data type the variable is.

PHP automatically converts the variable to the correct data type, depending on its value.

In a strongly typed programming language, we will have to declare **(define) the type and name of the variable before using it**.

MEC ACADEMY

Constant

- Constants are like variables except that once they are defined, they cannot be changed or undefined.

- To create a constant, use the define() function

Syntax:

define(**name, value, case-insensitive**)

```
<?php

define(“NAME", “Ahmed", true);


echo NAME; //Ahmed

?>
```

- **One important difference between constants and variables** is that when you refer to a constant, it does not have a $ in front of it. If you want to use the value of a constant, **use its name only.**

Keep In my mind that Constants are automatically global and can be used across the entire script.!!!!!!!!

Variables Data Types

A variable’s type refers to the kind of data stored in it.

PHP supports the following basic data types:

- Integer—Used for whole numbers
- Float (also called double)—Used for real numbers
- String—Used for strings of characters
- Boolean—Used for true or false values
- Array—Used to store multiple data items
- Object—Used for storing instances of classes
- Null —

Variable Casting

You can pretend that a variable or value is of a different type by using a type cast. You simply put the temporary type in parentheses in front of the variable you want to cast.

- For example, you could have declared the two variables from the preceding section using a cast:
- $var1 = 5.3;
- $var2 = (int)$var1;

Variable functions

- To use gettype()
    - you pass it a variable. It determines the type and returns a string containing the type name: bool, int, double (for floats), string, array, object, resource, or NULL.
    - It returns unknown type if it is not one of the standard types.
- Settype()
    - you pass it a variable for which you want to change the type and a string containing the new type for that variable from the previous list.

Common variables functions

- is_array()—Checks whether the variable is an array.
- is_double(), is_float(), is_real() (All the same function)—Checks whether the variable is a float.
- is_long(), is_int(), is_integer() (All the same function) Checks whether the variable is an integer.
- is_string()—Checks whether the variable is a string.
- is_bool()—Checks whether the variable is a boolean.

Common variables functions

- is_object()—Checks whether the variable is an object.
- is_resource()—Checks whether the variable is a resource.
- Is_non() checks whether the variable is not a number.
- is_null()—Checks whether the variable is null.
- is_numeric()—Checks whether the variable is any kind of number or a numeric string.

MEC ACADEMY

Operators

Arithmetic

Operators

Operators are used to perform operations on variables and values.

Arithmetic

Comparison

Logical

Assignment

Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power

Arithmetic

Comparison Operators

Expression	Meaning	Example	Illustrate
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. Introduced in PHP 7

MEC ACADEMY

Concatenation Operator

You can use the string concatenation operator to add two strings and to generate and store a result much as you would use the addition operator to add two numbers:

```
$a = “Hello, “;
$b = “World!”;
$result = $a.$b;
```

The $result variable now contains the string “Hello, World!”

Assignment Operators

Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus
$a.=$b	$a=$a.$b	Concatination

Pre/Post-increment

- The pre-and post-increment (++) and decrement (--) operators are similar to the +=and -= operators, but with a couple of twists.
- Example

```
$a=4;
echo ++$a;//echo 5 , value of $a = 5
$a=4;
echo $a++;  //echo 4 , value of $a = 5
```

Logical Operators

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

The logical operators combine the results of logical conditions. $a, is between 0 and 100. using the AND operator, as follows:

- $a >= 0 && $a <=100

MEC ACADEMY

Operations

Example:

Given a Number *N* corresponding to a person's age (in days). Print his age in years, months and days, followed by its respective message "years", "months", "days".

Note: consider the whole year has 365 days and 30 days per month.

Input

Only one line containing a number *N* ($0 \leq N \leq 10^6$).

Output

Print the output, like the following examples.

Example:
Input

```
400
```

output

```
1 years
1 months
5 days
```

Operations

Example:

Given a number R calculate the area of a circle using the following formula:

Area = $\pi * R^2$.

Note: consider $\pi = 3.141592653$.

Input

Only one line containing the number R ($1 \leq R \leq 100$).

Output

Print the calculated area, with 9 digits after the decimal point.

Control Statements

- Until now, the code is executed sequentially(line after line in order).
- **The control statements:** special statements are used to control the execution flow of the program.
- **Conditional Control Statement: special** statements allow the program to select between the alternatives during the program execution.
- **If-else Statement ,Ternary Operator**; are examples for Conditional Control Statement

Conditional Statements

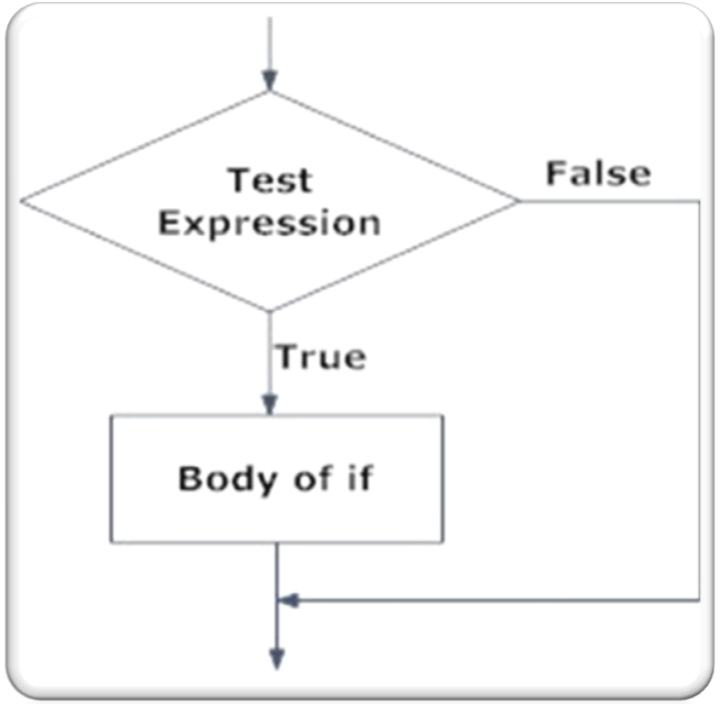
Arithmetic

MEC ACADEMY

If statement

```
if(condition)
{
 statement 1…
 statement 2…
}
```

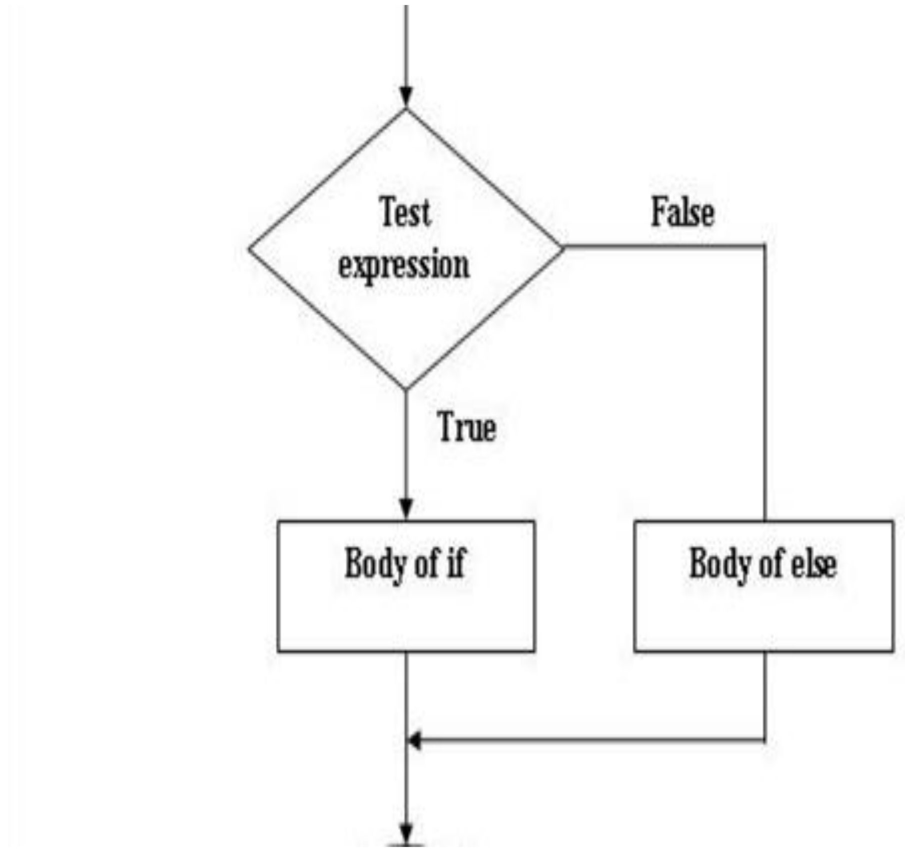
It will go inside the block only if the condition is true otherwise, it will not execute the block.



MEC ACADEMY

If-else Statement

```
if(condition){
 statements (if Block)
}
else{
 statements (Else block)
}
```



If the condition is true then, it will execute the If block. Otherwise, it will execute the Else block.

If statement

Example:

Given two numbers A and B. Print "Yes" if A is greater than or equal to B. Otherwise print "No".

Input

Only one line containing two numbers A and B ($0 \le A, B \le 100$).

Output

Print "Yes" or "No" according to the statement.

What is ternary operator ?????

Try it !

MEC ACADEMY

If statement

Example:

Given two numbers *A* and *B*. Print "Multiples" if *A* is multiple of *B* or vice versa. Otherwise print "No Multiples".

Input

Only one line containing two numbers A, B $(1 \leq A, B \leq 10^6)$

Output

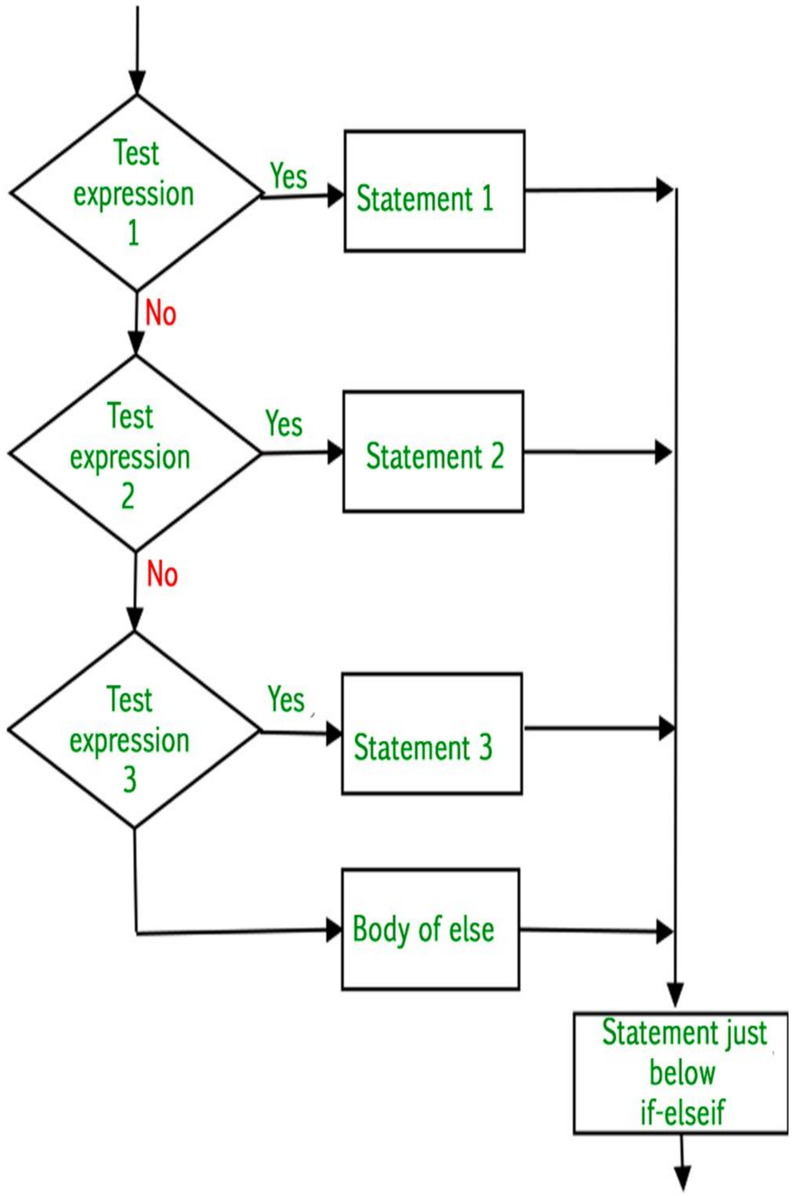
Print the "Multiples" or "No Multiples" corresponding to the read numbers.

Try Ternary operator!

MEC ACADEMY

If else-If statement

```
if(condition)
{
statements (if Block)
}
else if(condition)
 {
 statements (else if block)
}
else{
statements(else Block)
}
```



MEC ACADEMY

If else-If statement

Example:

given 3 numbers *A*, *B* and *C*, Print the minimum and the maximum numbers.

Input

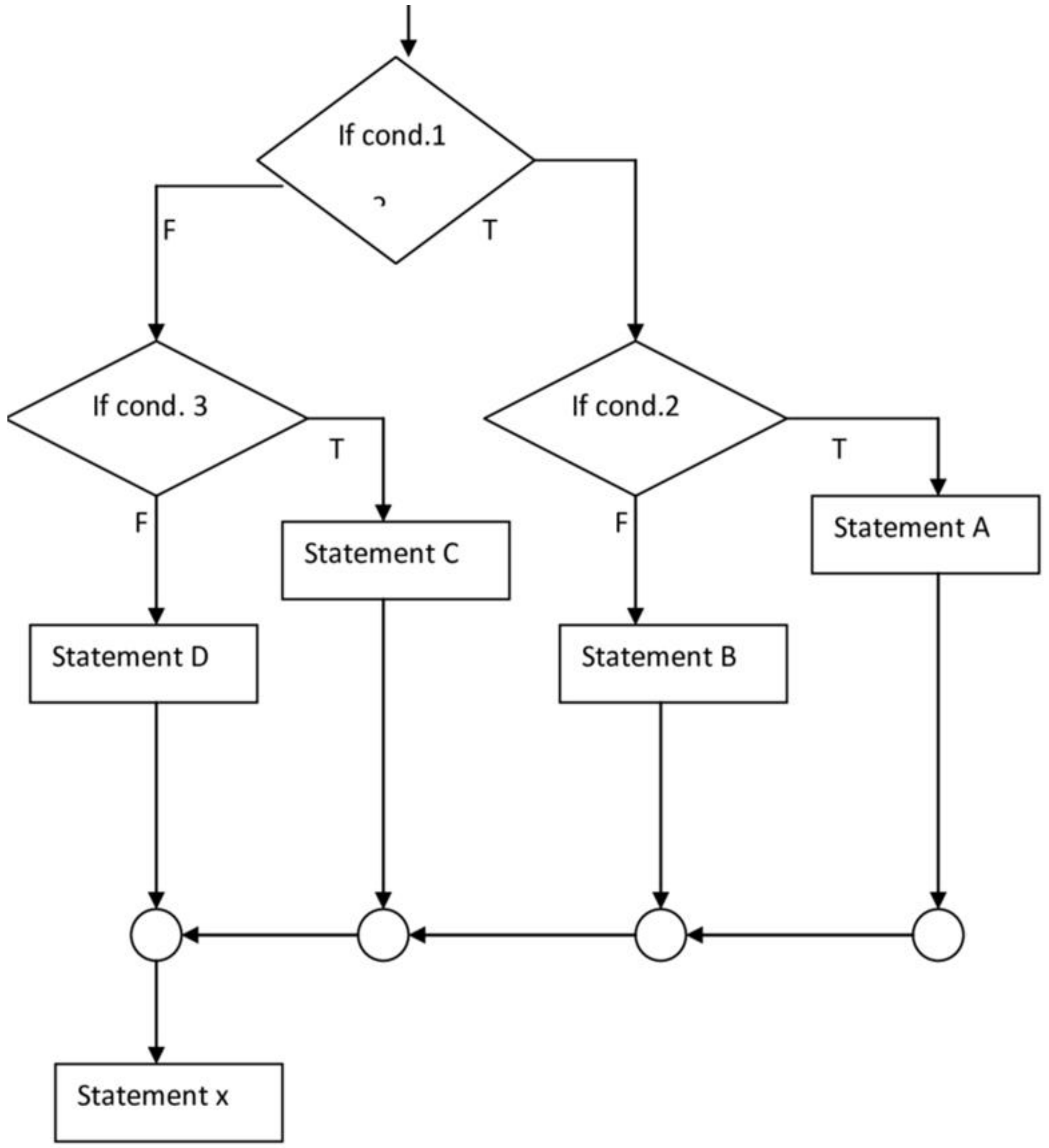
Only one line containing 3 numbers *A*, *B* and *C* ($-10^5 \leq A, B, C \leq 10^5$)

Output

Print the minimum number followed by a single space then print the maximum number.

Nested if

```
if(condition)
{
 code to be executed
    if(condition){
    code to be executed
    }
    else
}
```



MEC ACADEMY

Nested if

Example: ascii table

Given a letter *X*. Determine whether *X* is Digit or Alphabet and if it is Alphabet determine if it is Capital Case or Small Case.

Note:

Digits in ASCII '0' = 48,'1' = 49 ....etc

Capital letters in ASCII 'A' = 65, 'B' = 66 ....etc

Small letters in ASCII 'a' = 97,'b' = 98 ....etc

Input

Only one line containing a character *X* which will be a capital or small letter or digit.

Output

Print a single line contains "IS DIGIT" if *X* is digit otherwise, print "ALPHA" in the first line followed by a new line that contains "IS CAPITAL" if *X* is a capital letter and "IS SMALL" if *X* is a small letter.

Nested if

Example:

Given three numbers *A*, *B*, *C*. Print these numbers in ascending order followed by a blank line and then the values in the sequence as they were read.

Input

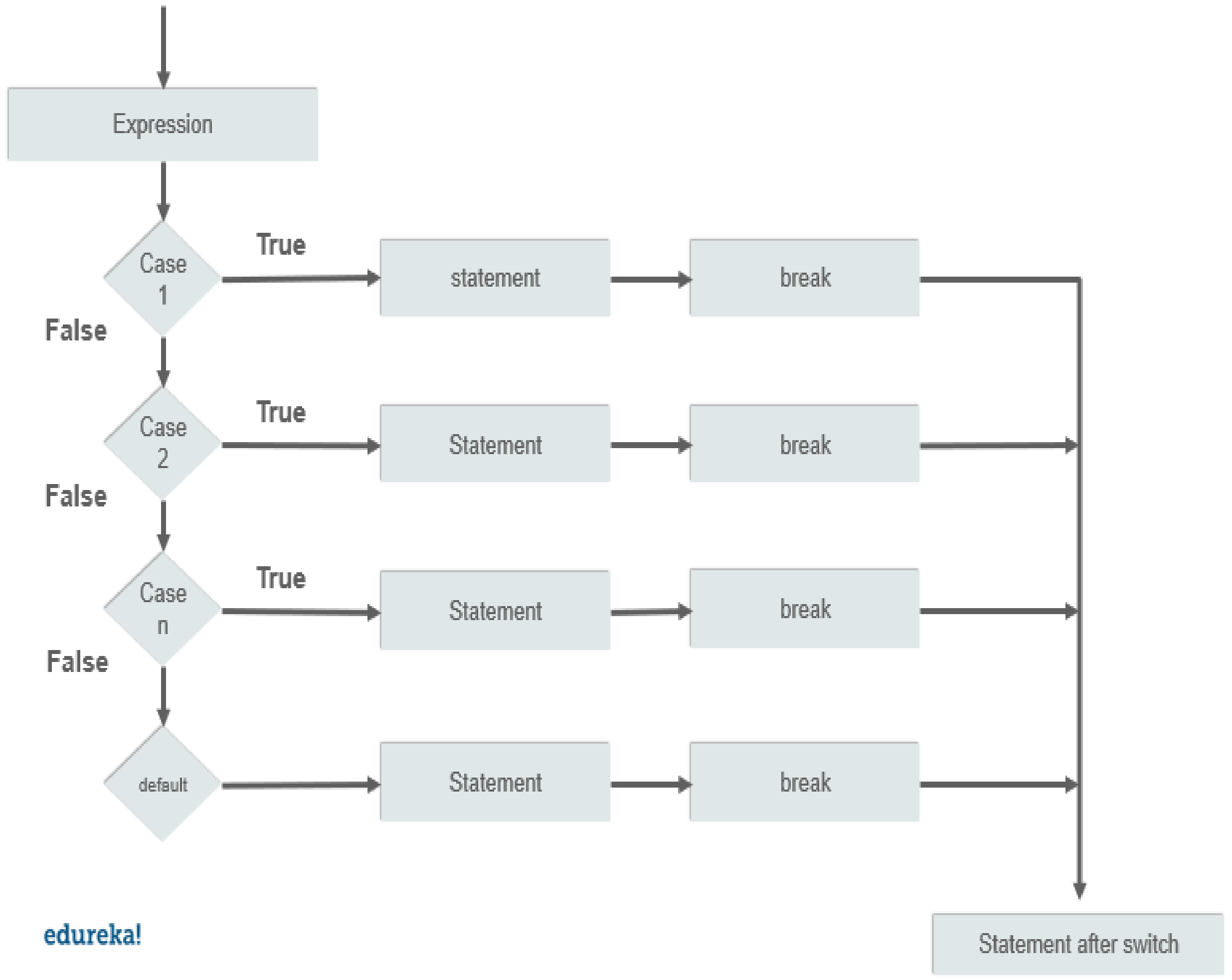
Only one line containing three numbers *A*, *B*, *C* ($-10^6 \le A, B, C \le 10^6$)

Output

Print the values in ascending order followed by a blank line and then the values in the sequence as they were read.

Switch

```
switch(expression) {
  case 1:
    // code block
    break;
  case 2:
    // code block
    break;
  default:
    // code block
}
```

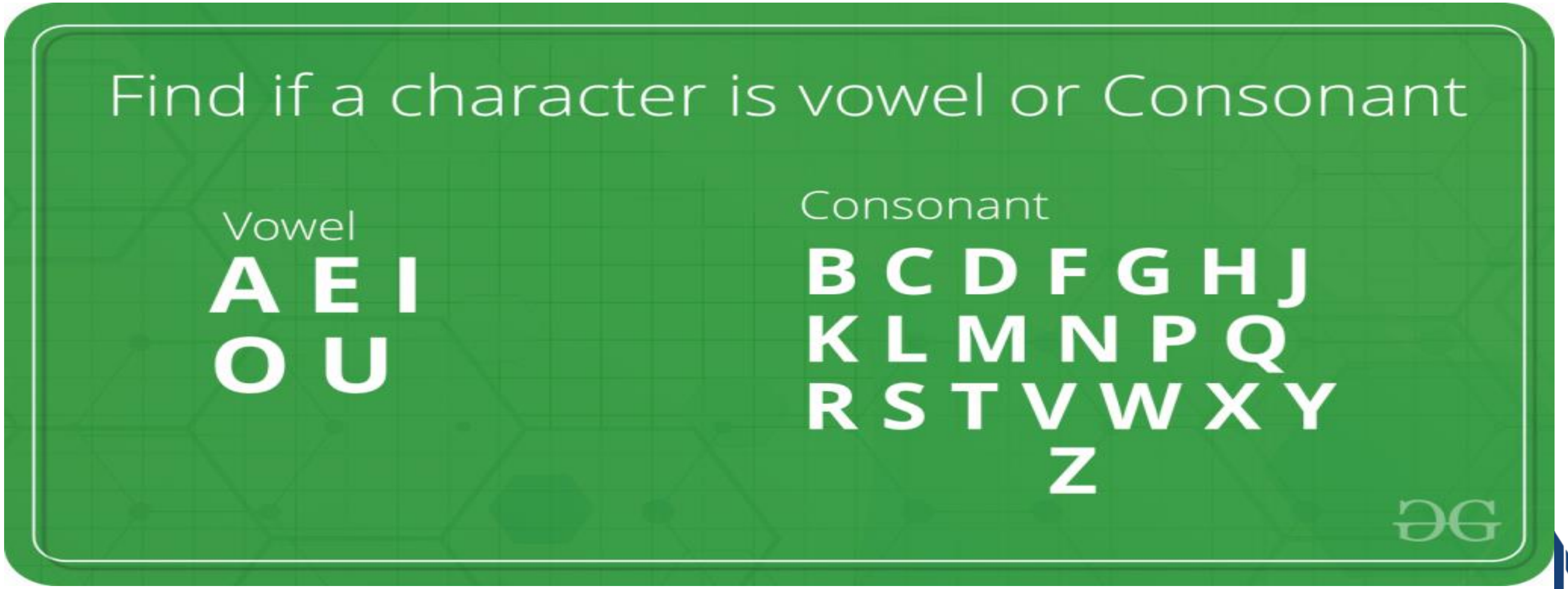


Switch

Example:

Given a character, check if it is vowel or consonant. Vowels are ‘a’, ‘e’, ‘i’, ‘o’ and ‘u’.

All other characters (‘b’, ‘c’, ‘d’, ‘f’ ….) are consonants.



Switch

Notes

- The break and default keywords are optional.
- The break keyword, it breaks out of the switch block, this will stop the execution of more code and case testing inside the block.
- The default keyword specifies some code to run if there is no case match.

MEC ACADEMY

task

- ❑ What is isset() and empty() functions and the difference between them?
- ❑ What is the meaning of variable of variable?
- ❑ What is variable scope and its types?
- ❑ What is the difference between **?** And **??**

Any

Question

MEC
ACADEMY

Next Session

- ❑ Loops
- ❑ Arrays
- ❑ Arrays Operators
- ❑ functions