CS 315 Programming Languages
Homework 2

Yasin Balcancı / 21501109 / Section: 1

Parameter Correspondence and Default Parameters:

-C has positional parameters only, it also does not have default parameters:

```
void foo(int a, int b, int c){
   printf("%d\n", a * b + c);
}
```

We are allowed to call foo in the form of 3 proper parameters only, such as foo(9, 8, 7).

-C++ language has positional parameters only but it has default parameter values:

```
void foo(int a = 5, int b = 4, int c = 3){
   printf("%d\n", a * b + c);
}
```

Which allows us to call the function by foo(), foo(9), foo(10, 9, 8) etc., values that have not been entered will be used from the default values.

-PHP has positional parameters only, it has default parameter values:

```
function foo($a = 5, $b = 4, $c = 3){
        echo $a * $b + $c;
        echo "\n";
}
```

We are able to call this function by foo(), foo(10), foo(10, 9, 8) etc., values that have not been entered will be used from the default values.

-JavaScript has positional parameters only but it has default parameter values:

```
function foo(a = 5, b = 4, c = 3){
   alert("foo: " + (a * b + c));
}
```

This function can be called by foo(), foo(10), foo(10, 9) etc., unentered values will be used from the default ones.

-Python has both positional and keyword parameters, it also has default parameters:

```
def foo(a = 5, b = 4, c = 3):
   print(a * b + c)
```

Function above can be called by foo(c = 1, a = 2, b = 3), foo(), foo(10) etc.

Variable Number of Actual Parameters
JavaScript, Python and PHP allows this with these ways respectively:

```
function bar(){
   var sum = 0;
   for(var i = 0; i < arguments.length; i++)
         sum += arguments[i];
   alert("bar: " + sum);
}
Call: bar(), bar(10), bar(10, 20, 30, 40)
```

```
function bar(){
         echo array_sum(func_get_args());
         echo "\n";
}
Call: bar(), bar(10, 20, 30)
```

```
def bar(*nums):
   print(sum(nums))
Call: bar(), bar(20, 10, 50, 60, 70)
```

Parameter Passing Methods
-C has both pass-by-value and pass-by-reference:

```
void passBy(int a, int* b){
   a = 3;
   *b = 3;
   printf("Inside passBy a = %d, b = %d\n", a, *b);
}
```

This function changes b outside but does not change a

-C++ has both pass-by-value and pass-by-reference:

```
void passBy(int a, int* b){
   a = 3;
   *b = 3;
   printf("Inside passBy a = %d, b = %d\n", a, *b);
}
```

This function changes b outside but does not change a

-JavaScript does not make assignments but changes values inside objects:

```
function passBy(x, y){
    y.t = 5;
    x = 5;
}
```

Changes t in object y, but x remains the same outside.

-PHP has both pass-by-value and pass-by-reference:

```
function passBy($x, &$y){
            $x = 5;
            $y = 5;
}
```

y changes outside while x remains the same.

-Python has pass-by-assignment which does not change outside values when an assignment is made inside but it changes(appends items etc.) mutable objects.

```
def passBy(x, y):
    x = 5
    y.append(5)
    print("Inside passBy x = ",x,", y = ",y)
```

REFERENCES:

https://stackoverflow.com/questions/986006/how-do-i-pass-a-variable-by-reference
https://codeburst.io/javascript-pass-by-value-and-pass-by-reference-in-javascript-fcf10305aa9c
https://stackoverflow.com/questions/879/are-php-variables-passed-by-value-or-by-reference