

Student Name: Wiktor Kubasiak

Student Number: R00162970

Student Class: SD2-A

1. Spy

```
<?php
```

```
    // Get first name entered by a user in the HTML form and store it in a
    variable.
```

```
    $first_name = $_GET['fname'];
```

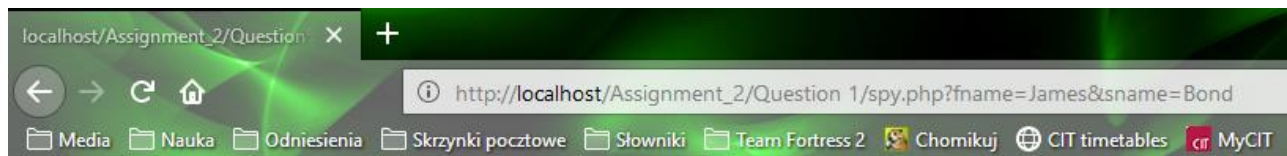
```
    // Get surname entered by a user in the HTML form and store it in a
    variable.
```

```
    $surname = $_GET['sname'];
```

```
    // Display an appropriate sentence within the paragraph tags with the
    variables used.
```

```
    echo '<p>Your name is '.$surname.', '.$first_name.' '.$surname.'.</p>';
```

```
?>
```



2. Weight and Height Conversion

Weight

<?php

// Get user's weight entered by the user in the HTML form and store it in a variable.

```
$user_weight = $_GET['weight_kg'];
```

// Calculate the total user's weight in pounds and round it down to the nearest integer.

```
$user_weight_in_pounds = floor($user_weight * 2.2);
```

// Calculate the total user's weight in stones and round it down to the nearest integer.

```
$user_weight_in_stones = floor($user_weight_in_pounds / 14);
```

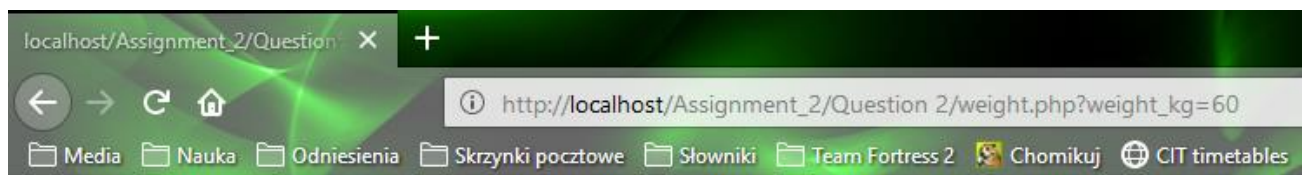
// Calculate the user's weight left over (remainder) after estimating the weight in stones.

```
$user_weight_in_pounds_left_over = $user_weight_in_pounds % 14;
```

// Display an appropriate sentence within the paragraph tags with the variables used.

```
echo '<p>You weigh '.$user_weight.' kg, which is  
'.$user_weight_in_stones.' stones and '.$user_weight_in_pounds_left_over.'  
pounds.</p>';
```

?>



Height

<?php

// Link to the stylesheet "table.css" to make the output look nicer.

```
echo '<link rel="stylesheet" type="text/css" href="table.css">';
```

// Get user's height in feet entered by the user in the HTML form and store it in a variable.

```
$user_height_in_feet = $_GET['feet'];
```

// Get user's height left over (remainder) entered by the user in the HTML form and store it in a variable.

```
$user_height_in_inches_left_over = $_GET['inches'];

// Declare units for conversion.
// 1 inch = 2.54 cm
$inch_in_centimetres = 2.54;
// 1 foot = 12 inches
$foot_in_inches = 12;
// 1 m = 100 cm
$metre_in_centimetres = 100;

// Calculate the total user's height in inches.
$user_height_in_inches = ($user_height_in_feet * $foot_in_inches) +
$user_height_in_inches_left_over;

// Calculate the total user's height in centimetres rounding up to the
nearest centimetre.
$user_height_in_centimetres = ceil($user_height_in_inches *
$inch_in_centimetres);

// Calculate the total user's height in metres rounding down to the
nearest metre.
$user_height_in_metres = floor($user_height_in_centimetres /
$metre_in_centimetres);

// Calculate the user's height left over (remainder) after estimating the
height in metres.
$user_height_in_centimetres_left_over = $user_height_in_centimetres
% $metre_in_centimetres;

// Display the table with the variables used. (2 rows and 5 columns)
echo'<table>
```

```

<tr>

    <th>Imperial:</th>

    <td>'.$user_height_in_feet.'</td>

    <td>foot/feet</td>

    <td>'.$user_height_in_inches_left_over.'</td>

    <td>inch(es)</td>

</tr>

<tr>

    <th>Metric:</th>

    <td>'.$user_height_in_metres.'</td>

    <td>metre(s)</td>

    <td>'.$user_height_in_centimetres_left_over.'</td>

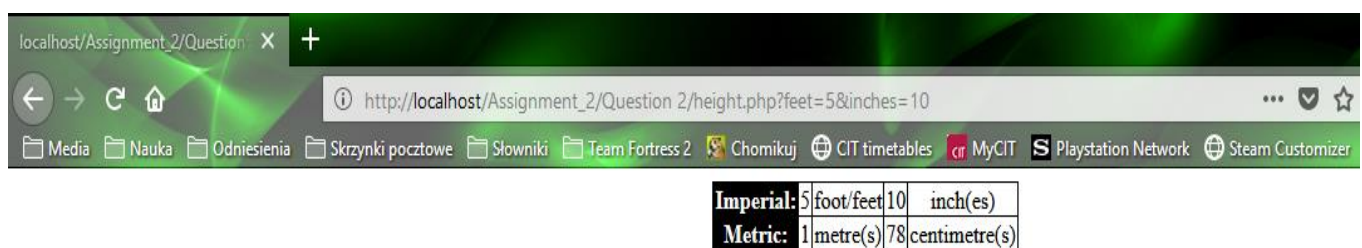
    <td>centimetre(s)</td>

</tr>

</table>';

```

?>



3. Exam Results System

<?php

// Get student's result from written examination entered by the student in the HTML form and store it in a variable.

```
$written_exam_result = $_GET['exam'];
```

// Get student's result from continuous assessment entered by the student in the HTML form and store it in a variable.

```
$continuous_assessment_result = $_GET['cont'];
```

// Get student's result from project entered by the student in the HTML form and store it in a variable.

```
$project_result = $_GET['proj'];
```

// If the result from written examination is greater than or equal to 40 and the result from continuous assessment is greater than or equal to 40 and the result from project is greater than or equal to 40, do the following:

```
if ($written_exam_result >= 40 && $continuous_assessment_result >= 40 && $project_result >= 40) {
```

// Display an appropriate message within the paragraph tags to the user.

```
echo '<p>Congratulations! You passed in each type of assessment.</p>';
```

// In the ordered list display the results from each type of assessment using the variables defined above.

```
echo '<ol>
```

```
<li>Written exam: '.$written_exam_result.'%</li>
```

```
<li>Continuous assessment: '.$continuous_assessment_result.'%</li>
```

```
<li>Project: '.$project_result.'%</li>
```

```

        </ol>';
    }

    // If the result from written examination is greater than or equal to 40
    and the result from continuos assessment is greater than or equal to 40 and
    the result from project is lesser than 40, do the following:

    elseif ($written_exam_result >= 40 && $continuos_assessment_result >=
    40 && $project_result < 40) {
        echo '<p>You will need to re-do your project.</p>';
        echo '<ol>
        <li>Written exam: '.$written_exam_result.'%</li>
        <li>Continuos assessment: '.$continuos_assessment_result.'%</li>
        <li>Project: '.$project_result.'%</li>
        </ol>';
    }

    // If the result from written examination is greater than or equal to 40
    and the result from continuos assessment is lesser than 40 and the result from
    project is greater than or equal to 40, do the following:

    elseif ($written_exam_result >= 40 && $continuos_assessment_result <
    40 && $project_result >= 40) {
        echo '<p>You will need to resubmit continuos assessments.</p>';
        echo '<ol>
        <li>Written exam: '.$written_exam_result.'%</li>
        <li>Continuos assessment: '.$continuos_assessment_result.'%</li>
        <li>Project: '.$project_result.'%</li>
        </ol>';
    }

    // If the result from written examination is greater than or equal to 40
    and the result from continuos assessment is lesser than 40 and the result from
    project is also lesser than 40, do the following:

```

```

elseif ($written_exam_result >= 40 && $continuous_assessment_result <
40 && $project_result < 40) {

    echo '<p>You will need to resubmit continuos assessments and re-
do your project.</p>';

    echo '<ol>

<li>Written exam: '.$written_exam_result.'%</li>

<li>Continuos assessment: '.$continuous_assessment_result.'%</li>

<li>Project: '.$project_result.'%</li>

</ol>';

}

// In every other case a student will fail.
else {

    echo '<p>Unfortunately, you failed.</p>';

    echo '<ol>

<li>Written exam: '.$written_exam_result.'%</li>

<li>Continuos assessment: '.$continuous_assessment_result.'%</li>

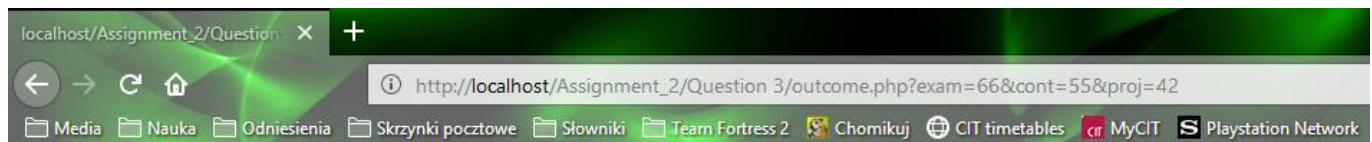
<li>Project: '.$project_result.'%</li>

</ol>';

}

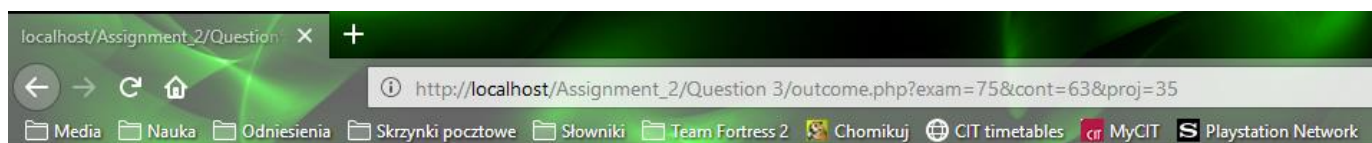
?>

```

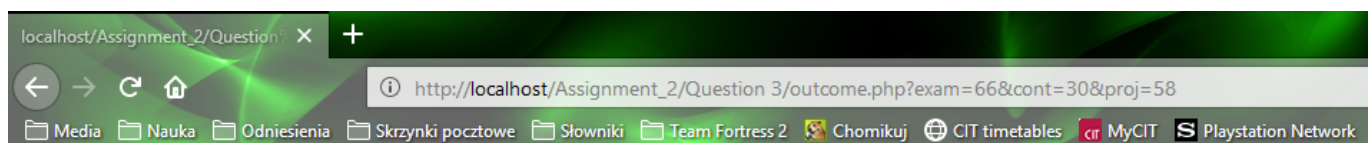
Congratulations! You passed in each type of assessment.

1. Written exam: 66%
2. Continuous assessment: 55%
3. Project: 42%



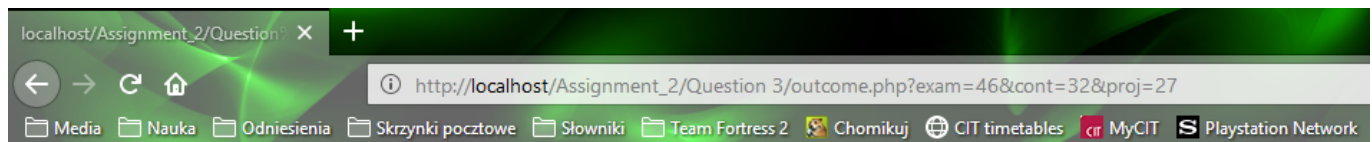
You will need to re-do your project.

1. Written exam: 75%
2. Continuous assessment: 63%
3. Project: 35%

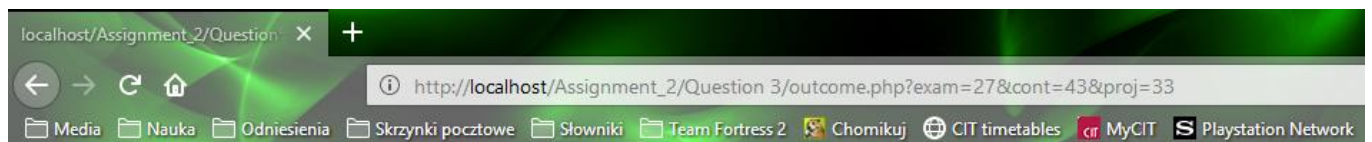


You will need to resubmit continuous assessments.

1. Written exam: 66%
2. Continuous assessment: 30%
3. Project: 58%



1. Written exam: 46%
2. Continuous assessment: 32%
3. Project: 27%



1. Written exam: 27%
2. Continuous assessment: 43%
3. Project: 33%

4. Yes, we have no bananas.

<?php

// Create the following associative array with the prices for each fruit.

```
$fruit_prices = array('Apples' => 1.59, 'Pears' => 2.34, 'Kumquats' => 4.05,  
'Jujubes' => 2.34);
```

// Get fruit which was chosen by a customer with the radio buttons in the HTML form and store it in a variable.

```
$fruit_chosen = $_GET['fruit'];
```

// Get weight which was entered by a customer into the textfield in the HTML form and store it in a variable.

```
$weight_entered = $_GET['weight'];
```

// If a customer chose apples, do the following:

```
if ($fruit_chosen == 'Apples') {
```

// Calculate the price for apples with a specific weight and refer to the array while doing it as well as round the total price to 2 decimal places.

```
    $price_for_apples = round($weight_entered *  
    $fruit_prices['Apples'], 2);
```

// Display an appropriate message to the user stating the weight and the total price for it.

```
    echo '<p>'.$weight_entered.'kg of apples costs  
    '.$price_for_apples.'$';
```

```
}
```

// If a customer chose pears, do the following:

```
elseif ($fruit_chosen == 'Pears') {
```

// Calculate the price for pears with a specific weight and refer to the array while doing it as well as round the total price to 2 decimal places.

```
    $price_for_pears = round($weight_entered *  
    $fruit_prices['Pears'], 2);
```

```
    echo '<p>'.$weight_entered.'kg of pears costs  
    '.$price_for_pears.'$';
```

```
}
```

// If a customer chose kumquats, do the following:

```
elseif ($fruit_chosen == 'Kumquats') {
```

// Calculate the price for kumquats with a specific weight and refer to the array while doing it as well as round the total price to 2 decimal places.

```
$price_for_kumquats = round($weight_entered *  
$fruit_prices['Kumquats'], 2);
```

```
echo '<p>'.$weight_entered.'kg of kumquats costs  
'.$price_for_kumquats.'&#364;</p>';
```

```
}
```

// If a customer chose jujubes, do the following:

```
elseif ($fruit_chosen == 'Jujubes') {
```

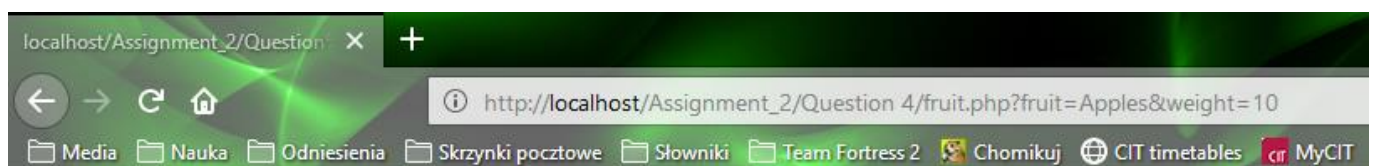
// Calculate the price for jujubes with a specific weight and refer to the array while doing it as well as round the total price to 2 decimal places.

```
$price_for_jujubes = round($weight_entered *  
$fruit_prices['Jujubes'], 2);
```

```
echo '<p>'.$weight_entered.'kg of jujubes costs  
'.$price_for_jujubes.'&#364;</p>';
```

```
}
```

?>



10kg of apples costs 15.9€

5. The Pants Pizza Parlour

```
<?php
```

```
    // Get the size of a pizza chosen by a customer in the HTML form and
    store it in a variable.
```

```
    $pizza_size = $_GET['size'];
```

```
    // Get the type of crust in a pizza chosen by a customer in the HTML form
    and store it in a variable.
```

```
    $type_of_crust = $_GET['crust'];
```

```
    // Create an empty array which will hold all the pizza toppings and assign
    it to a variable.
```

```
    $pizza_toppings = array();
```

```
    // Get the type of delivery for pizza defined by a customer in the HTML
    form and store it in a variable.
```

```
    $pizza_delivery = $_GET['delivery'];
```

```
    // Initialise the variable "total_price" to be further used in the program.
```

```
    $total_price = 0;
```

```
    // Declare variables with prices.
```

```
    // A small pizza with a thin crust costs 8€
```

```
    $price_for_small_thin_crust_pizza = 8;
```

```
    // A medium pizza with a thin crust costs 12€
```

```
    $price_for_medium_thin_crust_pizza = 12;
```

```
    // A large pizza with a thin crust costs 16€
```

```
    $price_for_large_thin_crust_pizza = 16;
```

```
    // Mushrooms on a pizza cost 0.5€
```

```

$price_for_mushrooms = 0.5;
// Olives on a pizza cost 0.5€ too.
$price_for_olives = 0.5;
// Finger nails in a pizza cost 1€
$price_for_finger_nail = 1;
// Spicy beef on a pizza costs 1.5€
$price_for_spicy_beef = 1.5;

// Declare variables with extra charges.

// A pizza with a deep pan crust costs additional 2€ to be included in the
entire price.
$deep_pan_crust_surcharge = 2;
// A home delivery costs additional 3€ to be included in the entire price.
$home_delivery_charge = 3;

// If the pizza is small-sized, consider the following:
if ($pizza_size == 'small') {
    // Add the price for a small pizza with the thin crust (8€) to the
total price to be paid.
    $total_price += $price_for_small_thin_crust_pizza;
    // If a customer wants a small pizza with the deep pan crust,
she/he has to pay extra 2€
    if ($type_of_crust == 'deep') {
        // Add the extra amount to the total price.
        $total_price += $deep_pan_crust_surcharge;
    }
    // If a customer wants a home delivery, she/he has to pay extra 3€

```

```
        if ($pizza_delivery == 'home') {
            $total_price += $home_delivery_charge;
        }
    }

    // If the pizza is medium-sized, consider the following:
    else if ($pizza_size == 'medium') {

        // Add the price for a medium pizza with the thin crust (12€) to the
        total price to be paid.

        $total_price += $price_for_medium_thin_crust_pizza;

        // If a customer wants a medium pizza with the deep pan crust,
        she/he has to pay extra 2€

        if ($type_of_crust == 'deep') {
            $total_price += $deep_pan_crust_surcharge;
        }

        if ($pizza_delivery == 'home') {
            $total_price += $home_delivery_charge;
        }
    }

    // If the pizza is large-sized, consider the following:
    else if ($pizza_size == 'large') {

        // Add the price for a large pizza with the thin crust (16€) to the
        total price to be paid.

        $total_price += $price_for_large_thin_crust_pizza;

        // If a customer wants a large pizza with the deep pan crust,
        she/he has to pay extra 2€

        if ($type_of_crust == 'deep') {
            $total_price += $deep_pan_crust_surcharge;
        }
    }
}
```

```
        if ($pizza_delivery == 'home') {
            $total_price += $home_delivery_charge;
        }
    }

    // Use the for loop to go through the list of toppings available.
    foreach ($_GET['toppings'] as $toppings) {

        // If a customer checks mushrooms as a topping on a pizza, then
do the following operation:

        if ($toppings == 'mushrooms') {

            // Append the price for mushrooms (0.5€) to the array of
toppings.

            $pizza_toppings[] = $price_for_mushrooms;
        }

        // If a customer checks olives as a topping on a pizza, then do the
following operation:

        if ($toppings == 'olives') {

            // Append the price for olives (0.5€) to the array of toppings.

            $pizza_toppings[] = $price_for_olives;
        }

        // If a customer checks finger nails as a topping in a pizza, then do
the following operation:

        if ($toppings == 'nail') {

            // Append the price for finger nails (1€) to the array of
toppings.

            $pizza_toppings[] = $price_for_finger_nail;
        }

        // If a customer checks spicy beef as a topping on a pizza, then do
the following operation:
```



```

        if ($toppings == 'beef') {

            // Append the price for spicy beef (1.5€) to the array of
toppings.

            $pizza_toppings[] = $price_for_spicy_beef;

        }

        // Calculate the total price for all the toppings chosen by a
customer. The array now stores the total price for them.

        $toppings_total_price = array_sum($pizza_toppings);

    }

    // Add the price for all the toppings to the total price to be paid by a
customer.

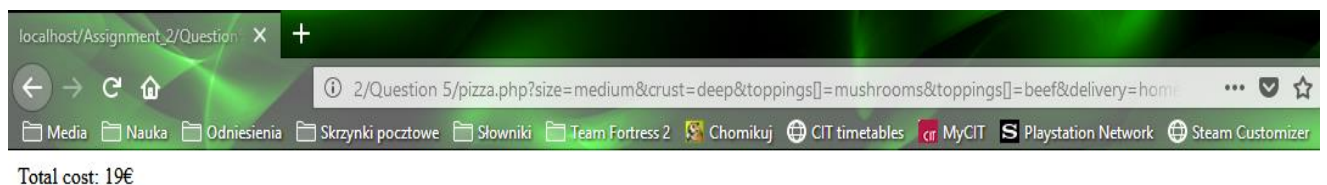
    $total_price += $toppings_total_price;

    // Display a sentence within the paragraph tags with the total price to be
paid by a customer.

    echo '<p>Total cost: '.$total_price.'&#8364;</p>';

?>

```



6. for practice

```
<?php
```

```
// I display a statement relating to numbers in the output from for loops.
```

```
echo "Ten numbers: ";
```

```
// I start from 0 and I go as far as 9 as it is lesser than 10 and I increment  
by 1 with each iteration.
```

```
for ($index = 0; $index < 10; $index++) {
```

```
    // I display a number in turn with a space between each one of  
    them.
```

```
    echo $index.' ';
```

```
}
```

```
// A line break HTML tag – now I move to the next line.
```

```
echo '<br>';
```

```
echo "Eleven numbers: ";
```

```
// I start from 0 and I go as far as 10 as it is lesser than 11 and I  
increment by 1 with each iteration.
```

```
for ($index = 0; $index < 11; $index++) {
```

```
    echo $index.' ';
```

```
}
```

```
echo '<br>';
```

```
echo "Teenage years: ";
```

// I start from 13 and I go as far as 19 as it is lesser than 20 and I increment by 1 with each iteration.

```
for ($index = 13; $index < 20; $index++) {  
    echo $index.' '  
}
```

```
echo '<br>';
```

```
echo "Evens: ";
```

// I start from 2 and I go as far as 20 as it is lesser than 22 and I increment by 2 with each iteration.

```
for ($index = 2; $index < 22; $index += 2) {  
    echo $index.' '  
}
```

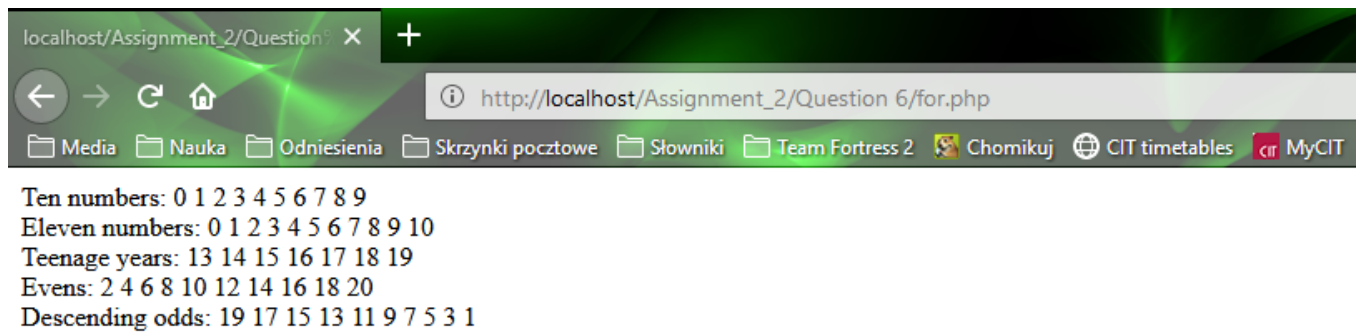
```
echo '<br>';
```

```
echo "Descending odds: ";
```

// I start from 19 and I go down to 1 as it is greater than 0 and I decrement by 2 with each iteration.

```
for ($index = 19; $index > 0; $index -= 2) {  
    echo $index.' '  
}
```

?>



7. Slip into something more comfortable?

```
<?php
```

```
// Create my validation function.
```

```
function my_validation() {
```

```
    // Get dress size entered by a user in the HTML form and store it in  
a variable.
```

```
    $dress_size = $_GET['size'];
```

```
    // Assign an error message to a variable.
```

```
    $error_message = 'Error: an incorrect value has been entered, it  
must be between 4 and 24 inclusive.';
```

```

        // Test the following block of statements within the try clause.
        try {
            // If the dress size is lesser than 4 or the dress size is greater
than 24, do the following:
            if ($dress_size < 4 || $dress_size > 24) {
                // Throw an exception with the error message defined
above.

                throw new Exception($error_message);
            }
            // Otherwise, do the following:
            else {
                // Display the size entered by a user within the
paragraph tags.

                echo '<p>Your size: '.$dress_size.'</p>';
            }

            // Return the value "dress_size" from my_validation
function.

            return $dress_size;

        }

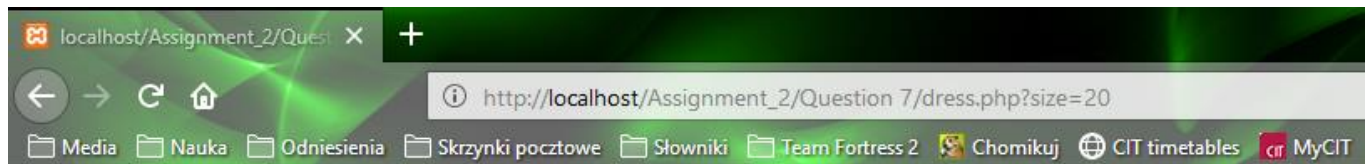
        // Include the exception in the catch clause.
        catch (Exception $dress_size) {
            // Display the error message declared above.
            echo $error_message;
        }
    }
}

```

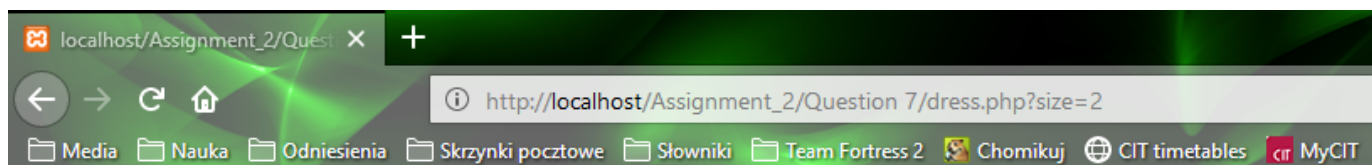
```
// Call the function.
```

```
my_validation();
```

```
?>
```



Your size: 20



Error: an incorrect value has been entered, it must be between 4 and 24 inclusive.

8. Funny farm

<?php

/*

I did not really know how to apply nested loops and standard for-loops rather than for-each loops in this question.

I was not sure of the array of user selections.

It would be valuable if the solution to this task was posted on BlackBoard.

I would like to learn how to do it properly in the future.

*/

// Create an empty array of user selections.

\$user_selections = array();

// Create an array of animals.

\$animals = array('duck', 'cow', 'dog', 'pig', 'badger');

// Create an associative array of animals and their noises.

\$animal_noises = array('duck' => 'quack', 'cow' => 'moo', 'dog' => 'woof', 'pig' => 'oink', 'badger' => 'chirr');

// Create a string variable which will hold the starting and ending line in each verse.

\$initial_and_final_line_in_a_verse = 'Old Macdonald had a farm ee-eye, ee-eye-oh.';

// Create a string variable in order to change 'a' to 'an' if a user selects the pig.

\$a_to_an = 'an';

// For each of the animals from those selected by a user, consider the following:

```
foreach($_GET['animals'] as $animal) {
```

```
    // If the animal checked is a duck, do the following:
```

```
    if ($animal == 'duck') {
```

```
        // Append the duck to the "user_selections" array.
```

```
        $user_selections[] = $animal;
```

```
        // Assign an element at index 0 (duck) of the array "animals"
to a variable.
```

```
        $animal = $animals[0];
```

```
        // Assign the association of duck (quack) from the array
"animal_noises" to a variable.
```

```
        $animal_noise = $animal_noises['duck'];
```

```
        // Display the starting line in the verse and write out the
whole verse within the paragraph tags with line breaks and variables declared
above. At the end of the verse display the ending line.
```

```
        echo'<p>'.$initial_and_final_line_in_a_verse.' <br>
```

```
        And on that farm he had a '.$animal.', ee-eye, ee-eye-oh.
```

```
<br>
```

```
        With a '.$animal_noise.', '.$animal_noise.' here and a
'.$animal_noise.', '.$animal_noise.' there. <br>
```

```
        Here a '.$animal_noise.', there a '.$animal_noise.',
everywhere a '.$animal_noise.', '.$animal_noise.'. <br>
```

```
        '.$initial_and_final_line_in_a_verse.'</p>;
```

```
    }
```

```
    // If the animal checked is a cow, do the following:
```

```
    if ($animal == 'cow') {
```

```
        // Append the cow to the "user_selections" array.
```



```
$user_selections[] = $animal;  
// Assign the element at index 1 (cow) of the array "animals"  
to a variable.
```

```
$animal = $animals[1];  
// Assign the association of cow (moo) from the array  
"animal_noises" to a variable.
```

```
$animal_noise = $animal_noises['cow'];  
echo'<p>'.$initial_and_final_line_in_a_verse.' <br>  
And on that farm he had a '.$animal.', ee-eye, ee-eye-oh.  
<br>
```

```
With a '.$animal_noise.', '.$animal_noise.' here and a  
'.$animal_noise.', '.$animal_noise.' there. <br>
```

```
Here a '.$animal_noise.', there a '.$animal_noise.',  
everywhere a '.$animal_noise.', '.$animal_noise.'. <br>
```

```
'.$initial_and_final_line_in_a_verse.'</p>;  
}
```

```
// If the animal checked is a dog, do the following:  
if ($animal == 'dog') {  
    // Append the dog to the "user_selections" array.  
    $user_selections[] = $animal;  
    // Assign the element at index 2 (dog) of the array "animals"  
to a variable.
```

```
$animal = $animals[2];  
// Assign the association of dog (woof) from the array  
"animal_noises" to a variable.
```

```
$animal_noise = $animal_noises['dog'];  
echo'<p>'.$initial_and_final_line_in_a_verse.' <br>  
And on that farm he had a '.$animal.', ee-eye, ee-eye-oh.  
<br>
```

With a '.\$animal_noise.', '.\$animal_noise.' here and a
'.\$animal_noise.', '.\$animal_noise.' there.

Here a '.\$animal_noise.', there a '.\$animal_noise.',
everywhere a '.\$animal_noise.', '.\$animal_noise.'.

'.\$initial_and_final_line_in_a_verse.'</p>;

}

// If the animal checked is a pig, do the following:

if (\$animal == 'pig') {

 // Append the pig to the "user_selections" array.

 \$user_selections[] = \$animal;

 // Assign the element at index 3 (pig) of the array "animals"
to a variable.

 \$animal = \$animals[3];

 // Assign the association of pig (oink) from the array
"animal_noises" to a variable.

 \$animal_noise = \$animal_noises['pig'];

 echo'<p>'.\$initial_and_final_line_in_a_verse.'

And on that farm he had a '.\$animal.', ee-eye, ee-eye-oh.

With '.\$a_to_an.' '.\$animal_noise.', '.\$animal_noise.' here
and '.\$a_to_an.' '.\$animal_noise.', '.\$animal_noise.' there.

Here '.\$a_to_an.' '.\$animal_noise.', there '.\$a_to_an.'
'.\$animal_noise.', everywhere an '.\$animal_noise.', '.\$animal_noise.'.

'.\$initial_and_final_line_in_a_verse.'</p>;

}

// If the animal checked is a badger, do the following:

if (\$animal == 'badger') {

 // Append the badger to the "user_selections" array.

```

        $user_selections[] = $animal;

        // Assign the element at index 4 (badger) of the array
"animals" to a variable.

        $animal = $animals[4];

        // Assign the association of badger (chirr) from the array
"animal_noises" to a variable.

        $animal_noise = $animal_noises['badger'];

        echo'<p>'.$initial_and_final_line_in_a_verse.' <br>
        And on that farm he had a '.$animal.', ee-eye, ee-eye-oh.
<br>

        With a '.$animal_noise.', '.$animal_noise.' here and a
'.$animal_noise.', '.$animal_noise.' there. <br>

        Here a '.$animal_noise.', there a '.$animal_noise.',
everywhere a '.$animal_noise.', '.$animal_noise.'. <br>

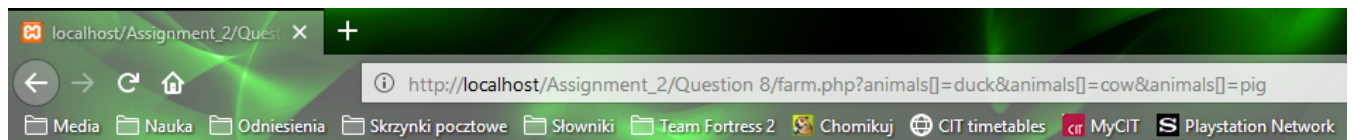
        '.$initial_and_final_line_in_a_verse.'</p>';

    }

}

?>

```



Old Macdonald had a farm ee-eye, ee-eye-oh.
And on that farm he had a duck, ee-eye, ee-eye-oh.
With a quack, quack here and a quack, quack there.
Here a quack, there a quack, everywhere a quack, quack.
Old Macdonald had a farm ee-eye, ee-eye-oh.

Old Macdonald had a farm ee-eye, ee-eye-oh.
And on that farm he had a cow, ee-eye, ee-eye-oh.
With a moo, moo here and a moo, moo there.
Here a moo, there a moo, everywhere a moo, moo.
Old Macdonald had a farm ee-eye, ee-eye-oh.

Old Macdonald had a farm ee-eye, ee-eye-oh.
And on that farm he had a pig, ee-eye, ee-eye-oh.
With an oink, oink here and an oink, oink there.
Here an oink, there an oink, everywhere an oink, oink.
Old Macdonald had a farm ee-eye, ee-eye-oh.

9. Don't foul your own nest

<?php

/*

I could not figure out how to display Triangle 2 pattern in this question.

It would be valuable if the solution to this part of the task was posted on BlackBoard.

*/

// The square number pattern for the numbers from 1 to 5.

echo '<p>Square</p>';

// The outer loop controls how many rows will be displayed. It starts from 1 and goes as far as 5 as it is lesser than 6. The "outer_index" increments by 1 with each iteration.

```

for ($outer_index = 1; $outer_index < 6; $outer_index++) {
    // The inner loop will display numbers from 1 to 5 in each of the
rows. It starts from 1 and goes as far as 5 as it is lesser than 6. The
"inner_index" also increments by 1 each time.

    for ($inner_index = 1; $inner_index < 6; $inner_index++) {
        // Display a number at a specific index with a space
separating every one of them. (numbers from 1 to 5)

        echo $inner_index.' ';
    }

    // In order to distinguish rows, there is a need to insert a line break
tag after each iteration of the outer loop.

    echo '<br>';
}

```

```

// The increasing number pattern forming a triangle for the numbers
from 1 up to 5.

echo '<p>Triangle 1</p>';

// The outer loop in this case also controls how many rows will be
displayed, progressing from 1 up to 5 as it is lesser than 6 and incrementing by
1 with each iteration.

for ($outer_index = 1; $outer_index < 6; $outer_index++) {
    // The inner loop will decide how many numbers to display in each
row, given the condition such that the current "inner_index" is lesser than or
equal to the "outer_index" of the outer loop.

    for ($inner_index = 1; $inner_index <= $outer_index;
$inner_index++) {
        echo $inner_index.' ';
    }

    echo '<br>';
}

```

```
}
```

```
// The decreasing number pattern forming a triangle for the numbers  
from 5 down to 1.
```

```
echo '<p>Triangle 2</p>';
```

```
// The outer loop controls how many rows will be displayed like in the  
other 2 cases, coming from 5 down to 1 as it is greater than 0 and  
decrementing by 1 with each iteration.
```

```
for ($outer_index = 5; $outer_index > 0; $outer_index--) {
```

```
// The inner loop will display values accordingly to the outer loop  
when it is writing out the numbers in an increasing order, but one number less  
with each row.
```

```
    for ($inner_index = 1; $inner_index <= $outer_index;  
    $inner_index++) {
```

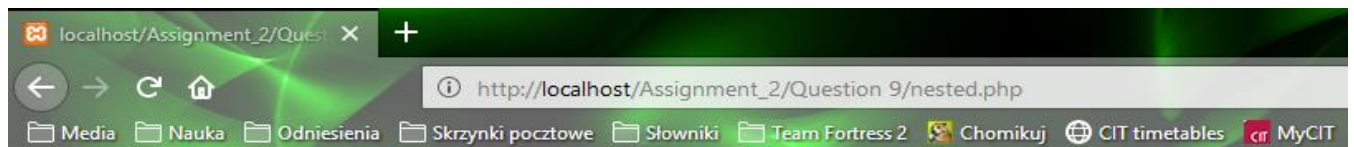
```
        echo $inner_index.' ';
```

```
    }
```

```
    echo '<br>';
```

```
}
```

```
?>
```



Square

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

Triangle 1

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Triangle 2

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

10. A library of functions

Library

```
<?php
```

```
/*
```

I did not know how to complete this task using only the built-in functions stated in the question's specifications.

It would be valuable if the solution to this problem was posted on BlackBoard.

```
*/
```

```
// Define a function "get_distinct()"
function get_distinct() {
    // Declare an array "a" with integers only.
    $a = array(2, 3, 1, 2, 1, 4);
    // Display the label indicating that it is the original array declared
above.
```

```
    echo 'The original array: ';
    // Display the array "a" with all its elements.
    print_r($a);
    // Distinguish the values which are unique for the array.
    $a_without_duplicates = array_unique($a);
    // Insert a line break HTML tag.
    echo '<br>';
    // Display the label indicating that it is the changed array.
    echo 'The same array without duplicates: ';
    // Display the array without any duplicates.
    print_r($a_without_duplicates);
    echo '<br>';
}
```

```
// Call the function "get_distinct()"
get_distinct();
```

```
echo '<br>';
```

```
// Define a function "get_flip()"
function get_flip() {
```



```

// Declare an associative array "a" with the following letters.
$a = array('a' => 'x', 'c' => 'y', 'b' => 'z', 'd' => 'y');
echo 'The original array: ';
print_r($a);

// Flip the values of the array "a" and assign it to a variable.
$a_with_flipped_values = array_flip($a);
echo '<br>';

echo 'The same array with flipped associative values: ';

// Display the array with all the associative values flipped.
print_r($a_with_flipped_values);
echo '<br>';
}

```

```

// Call the function "get_flip()"
get_flip();

```

```

echo '<br>';

```

```

// Define a function "get_frequencies()"
function get_frequencies() {
    $a = array('a' => 'x', 'c' => 'y', 'b' => 'z', 'd' => 'y');
    echo 'The original array: ';
    print_r($a);

    // Count the associations from the array "a" for each of the letters
    and assign it to a variable.

    $a_with_counted_values = array_count_values($a);
    echo '<br>';
}

```

```
        echo 'The same array with counted associative values: ';
```

// Display the array with all the associative values and their counters.

```
        print_r($a_with_counted_values);
        echo '<br>';
    }
```

```
// Call the function "get_frequencies()"
get_frequencies();
```

```
?>
```

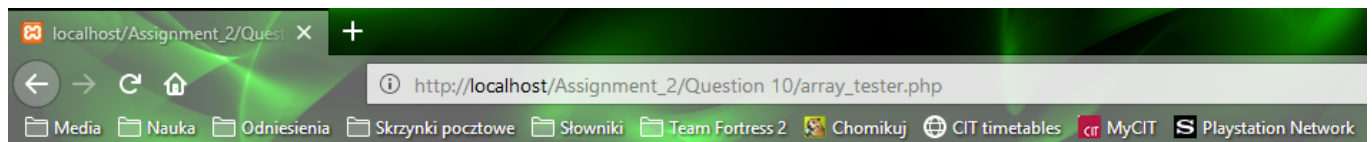
Tester

```
<?php
```

```
    // Load in the file "array_library.php" and call all the functions which are
    contained there.
```

```
    require_once('array_library.php');
```

```
?>
```



The original array: Array ([0] => 2 [1] => 3 [2] => 1 [3] => 2 [4] => 1 [5] => 4)

The same array without duplicates: Array ([0] => 2 [1] => 3 [2] => 1 [5] => 4)

The original array: Array ([a] => x [c] => y [b] => z [d] => y)

The same array with flipped associative values: Array ([x] => a [y] => d [z] => b)

The original array: Array ([a] => x [c] => y [b] => z [d] => y)

The same array with counted associative values: Array ([x] => 1 [y] => 2 [z] => 1)