FEU INSTITUTE OF TECHNOLOGY

COLLEGE OF COMPUTER STUDEIS
INFORMATION TECHNOLOGY DEPARTMENT

IT0049

(Web System and Technologies)

TECHNICAL SUMMATIVE ASSESSMENT

1

Module 1 – Module 3

Create a controller class, "PrintSequence" that will generate a view according to the descriptions provided in each methods. Then, in your newly created controller create 3 different methods, namely :

1.) MarioFlagPole

- Accepts 1 argument, which is the Flag Pole Size.
- The segment that handles your argument should only accept numbers.

Example:

Argument : 5
Output :

```
        12
       345
      6789
  1011121314
151617181920
```

2.) BoxedFrame

- Accepts 2 arguments
- The segments that refers to your arguments should only accept numbers.

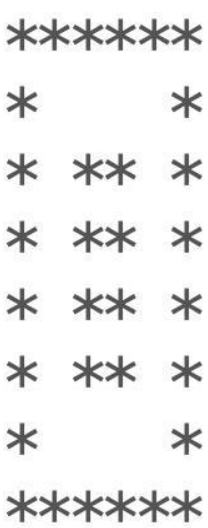
Arguments:
1 - Row Size
2 - Column Size

Examples:

Arguments:
1 = 8
2 = 5

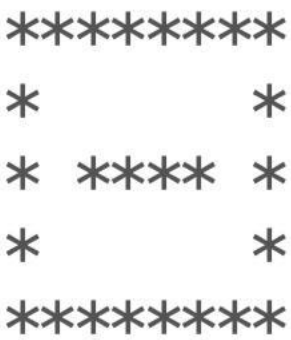
```
*****
*   *
* * *
* * *
* * *
* * *
*   *
*****
```

Arguments:
1 = 8
2 = 6



Arguments:
1 = 8
2 = 7

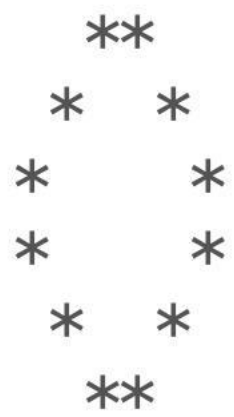
Arguments:
1 = 5
2 = 8



3.) HoneyComb
- Accepts 2 arguments
- The segments that refers to your arguments should only accept numbers.

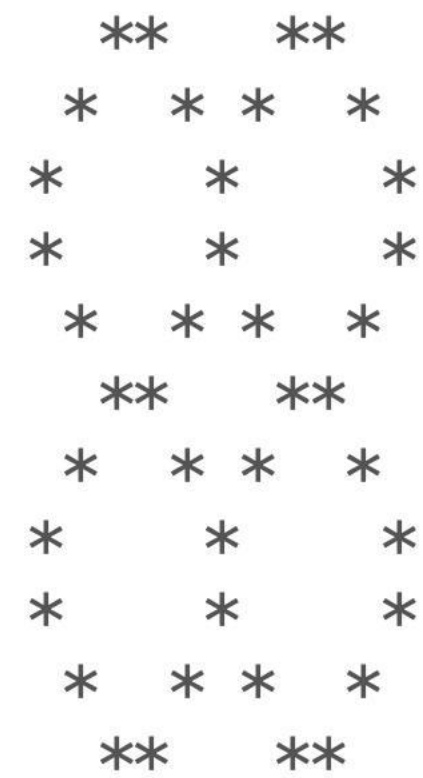
Args:
1 - Row Size
2 - Column Size

Example :
Arguments :
1 = 1
2 = 1



Arguments :
1 = 1
2 = 2

Arguments :
1 = 2
2 = 2



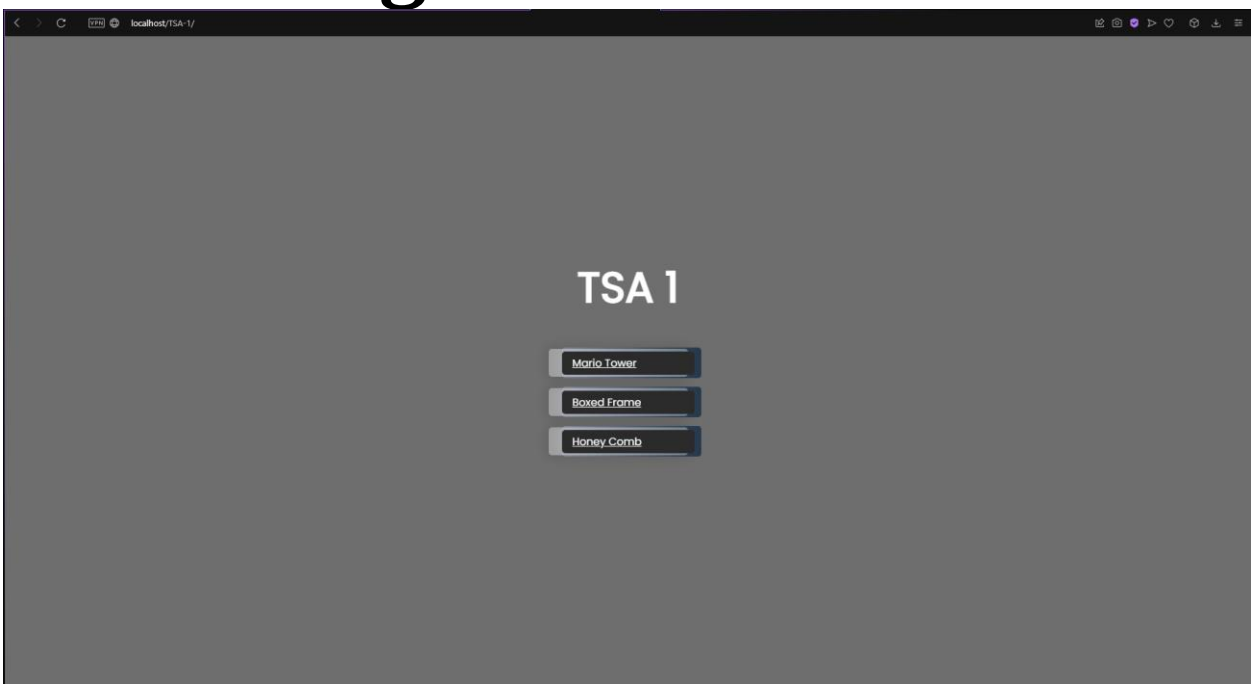
Note: The following rubrics/metrics will be used to grade students' output in the Technical Summative Assessment 1.

Program (100 pts.)	**(Excellent)**	**(Good)**	**(Fair)**	**(Poor)**
Program execution (20pts)	Program executes correctly with no syntax or runtime errors **(18-20pts)**	Program executes with less than 3 errors **(15-17pts)**	Program executes with more than 3 errors **(12-14pts)**	Program does not execute **(10-11pts)**
Correct output (20pts)	Program displays correct output with no errors **(18-20pts)**	Output has minor errors **(15-17pts)**	Output has multiple errors **(12-14pts)**	Output is incorrect **(10-11pts)**
Design of output (10pts)	Program displays more than expected **(10pts)**	Program displays minimally expected output **(8-9pts)**	Program does not display the required output (**6-7pts)**	Output is poorly designed **(5pts)**
Design of logic (20pts)	Program is logically well designed **(18-20pts)**	Program has slight logic errors that do no significantly affect the results **(15-17pts)**	Program has significant logic errors **(3-5pts)**	Program is incorrect **(10-11pts)**
Standards (20pts)	Program code is stylistically well designed **(18-20pts)**	Few inappropriate design choices (i.e. poor variable names, improper indentation) **(15-17pts)**	Several inappropriate design choices (i.e. poor variable names, improper indentation) **(12-14pts)**	Program is poorly written **(10-11pts)**
Delivery (10pts)	The program was delivered on time. **(10pts)**	The program was delivered a day after the deadline. **(8-9pts)**	The program was delivered two days after the deadline. **(6-7pts)**	The program was delivered more than two days after the deadline. **(5pts)**

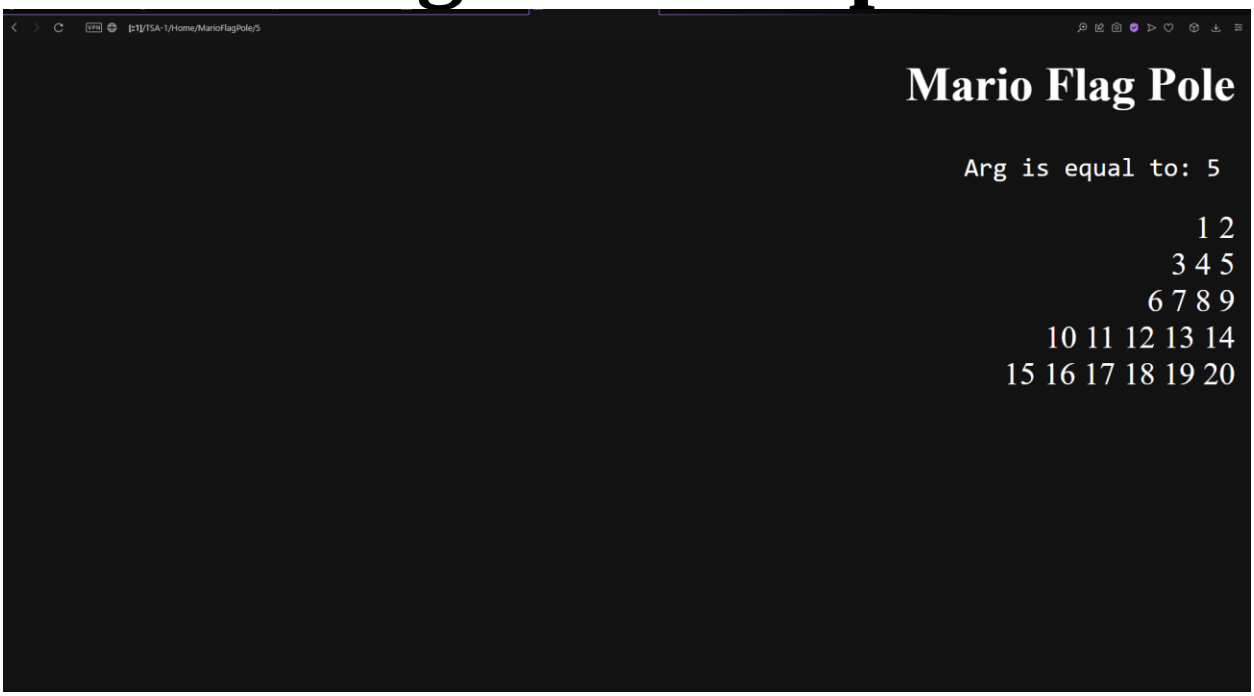
Outputs:

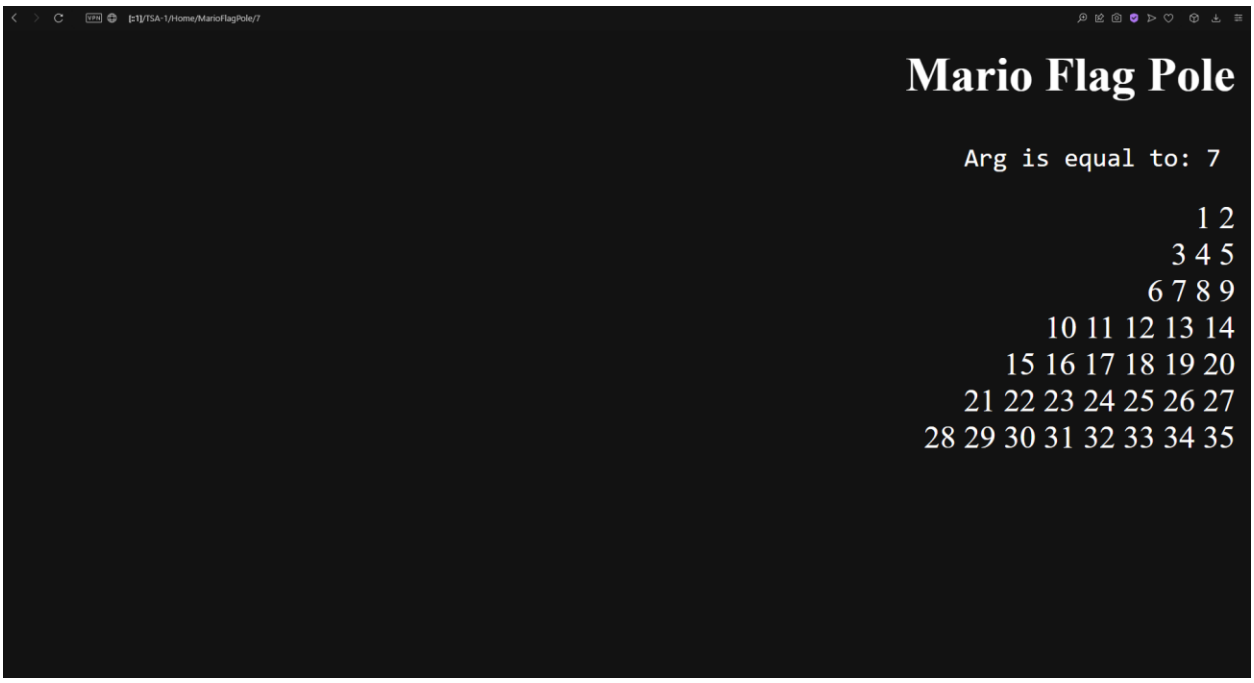
Full code can be found in my GitHub Repo

Home Page



Mario Flag Pole Outputs:



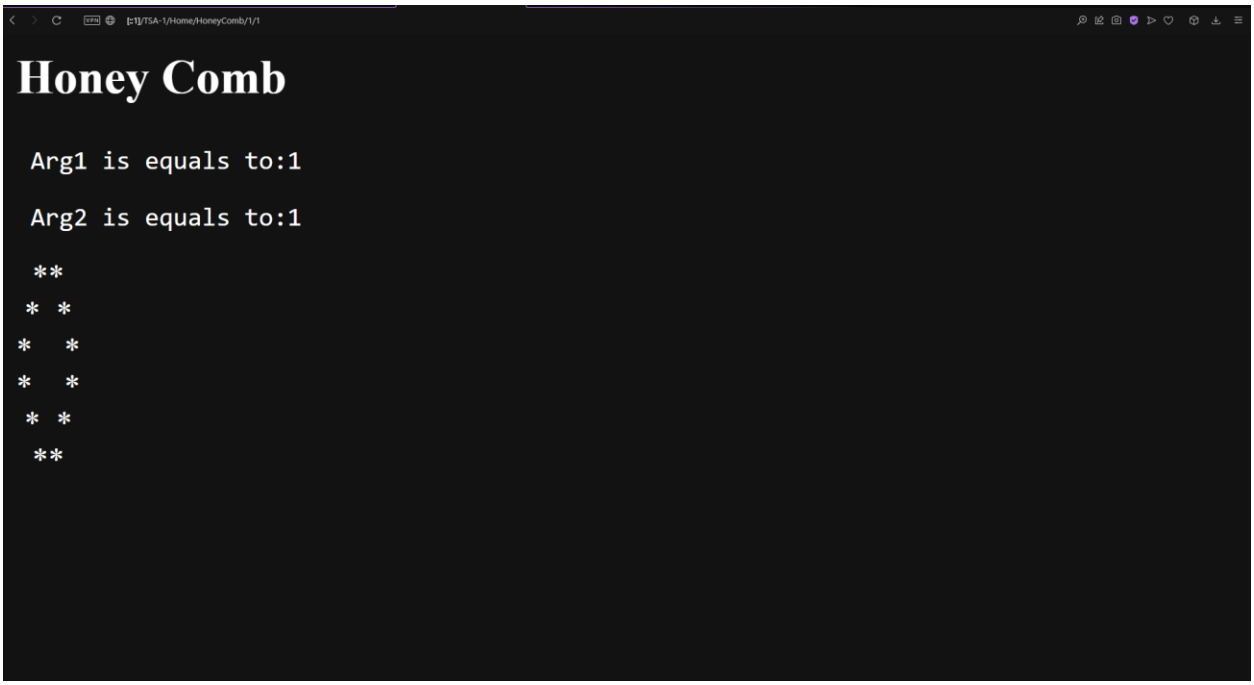


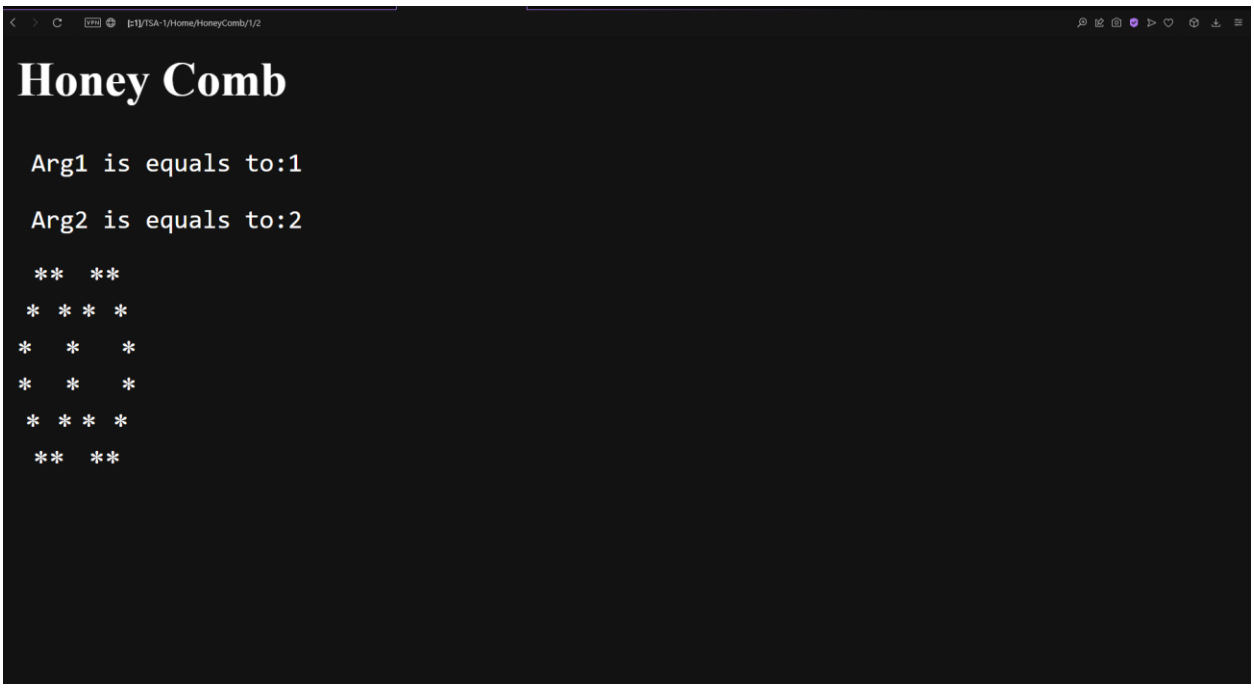
Boxed Frame Outputs:


[::1]/TSA-1/Home/BoxedFrame/8/7
Boxed Frame
Arg1 is equals to:8
Arg2 is equals to:7
****
*   *
* *** *
* *** *
* *** *
* *** *
*   *
****


[::1]/TSA-1/Home/BoxedFrame/5/8
Boxed Frame
Arg1 is equals to:5
Arg2 is equals to:8
****
*   *
* **** *
*   *
****

Honey Comb Outputs:






[::1]/TSA-1/Home/HoneyComb/2/2
Honey Comb
Arg1 is equals to:2
Arg2 is equals to:2
** **
* * * *
* * *
* * *
* * * *
** **
* * * *
* * *
* * *
* * * *
** **

More below

Codes:

Controller:

```php
<?php

class Home extends CI_Controller
{

    public function __construct()
    {
        //allways the first line
        parent::__construct();
    }
    // default lagi si index function
    public function index()
    {
        //tatawagin si home_view.php from the views folder and will load it on
the page
        $this->load->helper('url'); //for loading the css
        $this->load->view('home_view');
    }

    //Problem number 1
    public function MarioFlagPole($argu)
    {
        echo '<div style="text-align: right; ">'; // moving to the right to match
the given output
        echo "<h2>Mario Flag Pole</h2>";
        echo "<pre> Arg is equal to: " . $argu . " </pre>"; //printing the
argument in the URI

        $count = 1; // initialize to start in 1
        for ($i = 1; $i <= $argu; $i++) { // the number of lines will depend on
the input value of argument
            for ($x = 0; $x <= $i; $x++) {
```

```
                echo $count . " ";
                $count++;
            }
            echo '<br/>';
        }
        echo '</div>';
    }

    //Problem number 2
    public function BoxedFrame($argu1, $argu2)
    {
        echo "<h2>Boxed Frame</h2>";
        echo "<pre> Arg1 is equals to:" . $argu1 . "</pre>"; //printing the
argument in the URI
        echo "<pre> Arg2 is equals to:" . $argu2 . " </pre>"; //printing the
argument inthe URI

        for ($i = 0; $i < $argu1; $i++) {
            for ($j = 0; $j < $argu2; $j++) {
                if ($i == 0 || $j == 0 || $i == $argu1 - 1 || $j == $argu2 - 1) {
                    echo "*";
                } elseif ($i == 1 || $j == 1 || $i == $argu1 - 2 || $j == $argu2
- 2) {
                    echo "  ";
                } else {
                    echo "*";
                }
            }
            echo "</br>";
        }
    }


    //Problem number 3
    public function HoneyComb($argu1, $argu2)
    {

        echo "<h2>Honey Comb</h2>";
        echo "<pre> Arg1 is equals to:" . $argu1 . "</pre>"; //printing the
argument in the URI
        echo "<pre> Arg2 is equals to:" . $argu2 . " </pre>"; //printing the
argument inthe URI


        for ($a = 1; $a < 7 * $argu1; $a++) {
```

```
            for ($b = 1; $b < 7 * $argu2; $b++) {
                if ($a == 1 || $a == 6 || $a == 11) {
                    if ($b != 3 && $b != 4 && $b != 8 && $b != 9) {
                        echo " ";
                    } else {
                        echo "*";
                    }
                }
                if ($a == 2 || $a == 5 || $a == 7 || $a == 10) {
                    if ($b != 2 && $b != 5 && $b !== 7 && $b != 10) {
                        echo " ";
                    } else {
                        echo "*";
                    }
                }
                if ($a == 3 || $a == 4 || $a == 8 || $a == 9) {
                    if ($b != 1 && $b != 6 && $b != 12) {
                        echo " ";
                    } else {
                        echo "*";
                    }
                }
            }
            echo "<br>";
        }
    }
}

//Code by Justine Rome M. Guillermo
//code from: https://github.com/romeJG/TSA-1
```

View:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="<?= base_url(); ?>css/home.css">
    <title>Summative Assesment 1</title>
</head>

<body>


    <div id="container">
        <h1>TSA 1</h1>
        <ul id="navlist">
            <li><a href="<?= base_url('Home/MarioFlagPole/5') ?>">Mario
Tower</a></li>
            <li><a href="<?= base_url('Home/BoxedFrame/8/5') ?>">Boxed
Frame</a></li>
            <li><a href="<?= base_url('Home/HoneyComb/2/2') ?>">Honey
Comb</a></li>
        </ul>
    </div>
</body>

</html>
```

CSS (optional):

```css
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;

}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: rgb(110, 110, 110);
    text-align: center;
}

ul li {
    width: 200px;
    height: 35px;
    display: flex;
    align-items: center;
    margin: 1.5em;
    cursor: pointer;
    padding: 1em;
    background: rgb(43, 43, 43);
    position: relative;
    color: white;
    border-radius: 5px;
}

a {
    color: white;
}

ul li::before,
ul li::after {
    content: '';
    position: absolute;
```

```
    z-index: -1;
    border-radius: 5px;
    width: 105%;
    transition: all .4s;
}

ul li::before {
    left: 0%;
    height: 130%;
    background: linear-gradient(to right, #8b90a0, #25394d);
}

ul li::after {
    left: -10%;
    height: 120%;
    background: #ffffff56;
    backdrop-filter: blur(10px);
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.164);
}

ul li:hover::before {
    transform: translateX(-2.5%);
}

ul li:hover::after {
    transform: translateX(15%);
}

h1 {
    margin-bottom: 5vh;
    font-size: 400%;
    color: white;
}

.my_class {
    background-color: #25394d;
}
```