Task 1:

Problem Statement
Create an image gallery using a CSS grid.

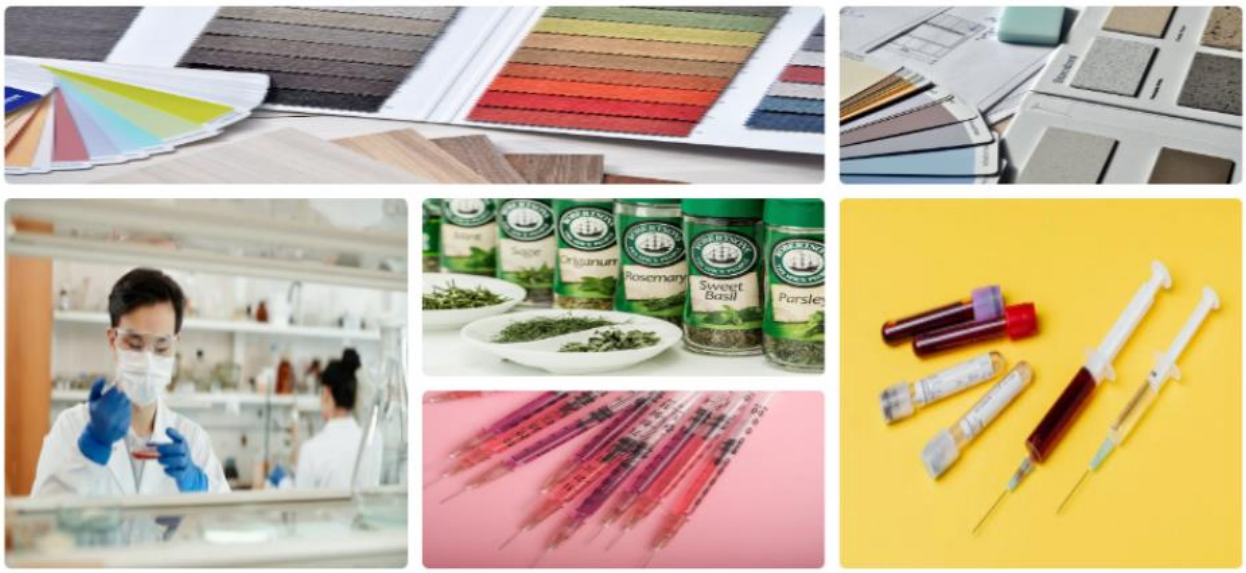
Expected Behaviour

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Grid Image Gallery</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            text-align: center;
        }

        .gallery {
            display: grid;
            grid-template-columns: repeat(3, 1fr);
            grid-template-rows: auto;
            gap: 10px;
            max-width: 800px;
            margin: 0 auto;
```

```
        }

        .gallery img {
            width: 100%;
            height: 100%;
            object-fit: cover;
            border-radius: 5px;
        }

        .item1 {
            grid-column: span 2;
        }

        .item2 {
            grid-column: span 1;
        }

        .item3 {
            grid-row: span 2;
        }

        .item4 {
            grid-column: 2;
        }

        .item5 {
            grid-column: 2;
        }

        .item6 {
            grid-column: 3;
            grid-row: span 2;
        }
    </style>
</head>

<body>

    <h2>Image Gallery using CSS Grid</h2>
    <div class="gallery">
        <img src="image1.png" alt="Image 1" class="item1">
        <img src="image2.png" alt="Image 2" class="item2">
        <img src="image3.png" alt="Image 3" class="item3">
        <img src="image4.png" alt="Image 4" class="item4">
        <img src="image5.png" alt="Image 5" class="item5">
```

```
        <img src="image6.png" alt="Image 6" class="item6">
    </div>

</body>

</html>
```

Output:

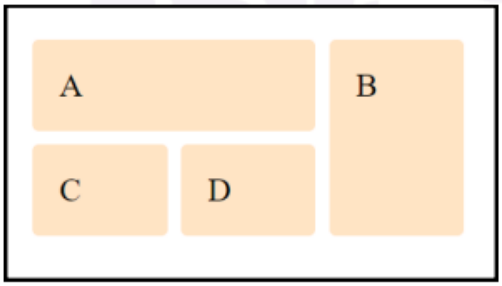


Task 2:

Problem Statement

Write code to arrange containers with texts A, B, C, and D as shown in the below image.

Expected Output



Ans:

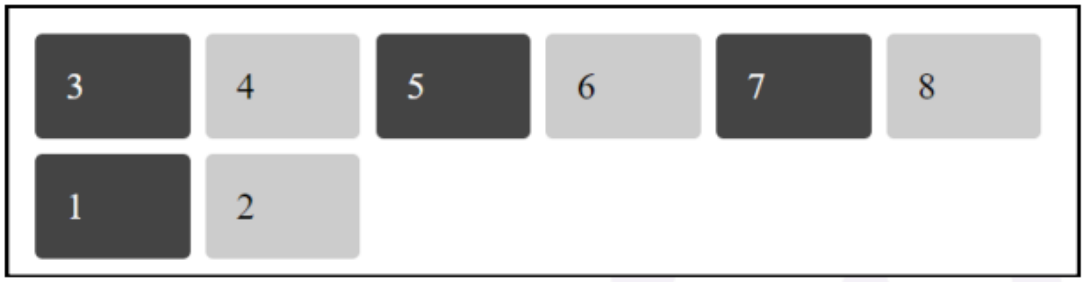
Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Grid Layout</title>
    <link rel="stylesheet" href="style.css">

</head>
<body>

    <div class="container">
        <div class="box a">A</div>
        <div class="box b">B</div>
        <div class="box c">C</div>
        <div class="box d">D</div>
    </div>

</body>
</html>
```

4.

```
<div class="container">
    <div class="box box1">1</div>
    <div class="box box2">2</div>
    <div class="box box3">3</div>
    <div class="box box4">4</div>
    <div class="box box5">5</div>
    <div class="box box6">6</div>
    <div class="box box7">7</div>
    <div class="box box8">8</div>
</div>
```

Expected Output



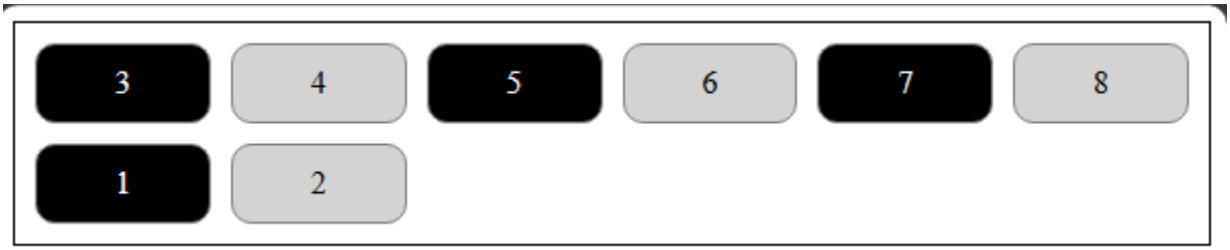
Code:

```html
<!DOCTYPE html>
<html>

<head>
    <title>Grid Layout</title>
    <style>
        .container {
            padding: 10px;
            display: grid;
            grid-template-columns: repeat(6, 1fr);
            grid-template-rows: auto auto;
            /* Rows adjust to content */
            gap: 10px;
            border: 1px solid black;
        }

        .box {
            display: flex;
            justify-content: center;
            align-items: center;
            background-color: lightgray;
```

```
        border: 1px solid gray;
        padding: 10px;
        border-radius: 10px;
        text-align: center;
        min-width: 50px;
    }

    .box1 {
        grid-area: 2 / 1 / 3 / 2;
        background-color: darkslategray;
        color: white;
    }

    .box2 {
        grid-area: 2 / 2 / 3 / 3;
    }

    .box3 {
        grid-area: 1 / 1 / 2 / 2;
    }

    .box4 {
        grid-area: 1 / 2 / 2 / 3;
    }

    .box5 {
        grid-area: 1 / 3 / 2 / 4;
    }

    .box6 {
        grid-area: 1 / 4 / 2 / 5;
    }

    .box7 {
        grid-area: 1 / 5 / 2 / 6;
    }

    .box8 {
        grid-area: 1 / 6 / 2 / 7;
    }

    .box1,
    .box3,
    .box5,
    .box7 {
```

```
            background-color: black;
            color: white;
        }
    </style>
</head>

<body>
    <div class="container">
        <div class="box box3">3</div>
        <div class="box box4">4</div>
        <div class="box box5">5</div>
        <div class="box box6">6</div>
        <div class="box box7">7</div>
        <div class="box box8">8</div>
        <div class="box box1">1</div>
        <div class="box box2">2</div>
    </div>

</body>

</html>
```

Output:



Task 5:

Problem Statement

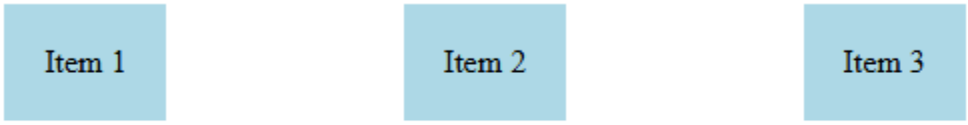
Explain the difference between justify-items and justify-self using code examples.

Justify-items:

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

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        .grid-container {
            display: grid;
            grid-template-columns: repeat(3, 1fr);
            justify-items: center;
            /* Aligns all grid items horizontally at the center */
        }

        .grid-item {
            background-color: lightblue;
            padding: 20px;
        }
    </style>
</head>

<body>
    <div class="grid-container">
        <div class="grid-item">Item 1</div>
        <div class="grid-item">Item 2</div>
        <div class="grid-item">Item 3</div>
    </div>

</body>

</html>
```

Output:



justify-self:

code:

```
<!DOCTYPE html>
```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        .grid-container {
            display: grid;
            grid-template-columns: repeat(3, 1fr);
            justify-items: start;
        }

        .grid-item {
            background-color: lightgreen;
            padding: 20px;
        }

        .item-2 {
            justify-self: center;
        }
    </style>
</head>

<body>
    <div class="grid-container">
        <div class="grid-item">Item 1</div>
        <div class="grid-item item-2">Item 2</div>
        <div class="grid-item">Item 3</div>
    </div>

</body>

</html>
```

Output: