Question - 1
Lifting Weights

Ollie is new to the gym and is figuring out the maximum weights she can lift. The maximum capacity of the barbell is given as *maxCapacity*. Each barbell plate has a weight, given by *weights[i]*. Now Ollie has to select as many plates as she can but the total weight of the selected plates should not exceed *maxCapacity*. What is the maximum weight of plates Ollie can add to the barbell?

For example, given barbell plates of weights of *1, 3* and *5 lbs* and a barbell of maximum capacity *7 lbs* - the right plates to insert would be *1* and *5 lbs (1+5 = 6),* thus making the right answer *6.*

Function Description

Complete the *weightCapacity* function in the editor below. The function must return an integer denoting the maximum capacity of items that he can purchase.

weightCapacity has two parameters:

weights: An array of *n* integers, where the value of each element *weights[i]* is the weight of each plate *i* (where $0 \le i < n$).

maxCapacity: An integer, the capacity of the barbell.

Constraints

- $1 \le n \le 42$
- $1 \le maxCapacity \le 10^9$
- $1 \le weights[i] \le 10^9$

▼ Input Format For Custom Testing

Locked stub code in the editor reads the following input from stdin and passes it to the function:

The first line contains an integer, *n*, denoting the number of elements in *weights*.

Each line *i* of the *n* subsequent lines contains an integer describing *weights[i]*.

The last line contains an integer, *maxCapacity*, denoting the maximum capacity of the barbell.

▼ Sample Case 0

Sample Input 0

```
STDIN      Function
-----      --------
3        → weights[] size n = 3
1        → weights[] = [ 1, 3, 5 ]
3
5
7        → maxCapacity = 7
```

Sample Output 0

```
6
```

Explanation 0

All the possible combination of items that Ollie can insert are: *{}, {1}, {3}, {5}, {1, 3}, {1, 5}, {3, 5},* and *{1, 3, 5}*.

Out of these combinations, the capacity that can be accommodated is *{1, 5}* making the total weight *6*.

▼ Sample Case 1

Sample Input 1

```
STDIN      Function
-----      --------
4        → weights[] size n = 4
4        → weights[] = [ 4, 8, 5, 9 ]
8
5
9
20       → maxCapacity = 20
```

Sample Output 1

```
18
```

Explanation

All the possible combination of items that Ollie can insert are: *{}, {4}, {8}, {5}, {9}, {4, 8}, {4, 5}, {4, 9}, {8, 5}, {8, 9}, {5, 9}, {4, 8, 5}, {4, 8, 9}, {4, 5, 9}, {8, 5, 9}, {4, 8, 5, 9}*.

Out of these combinations, the capacity that can be accommodated is *{4, 5, 9}* making the total weight *18*.

Question - 2

React: Catalog Viewer

Catalog viewer

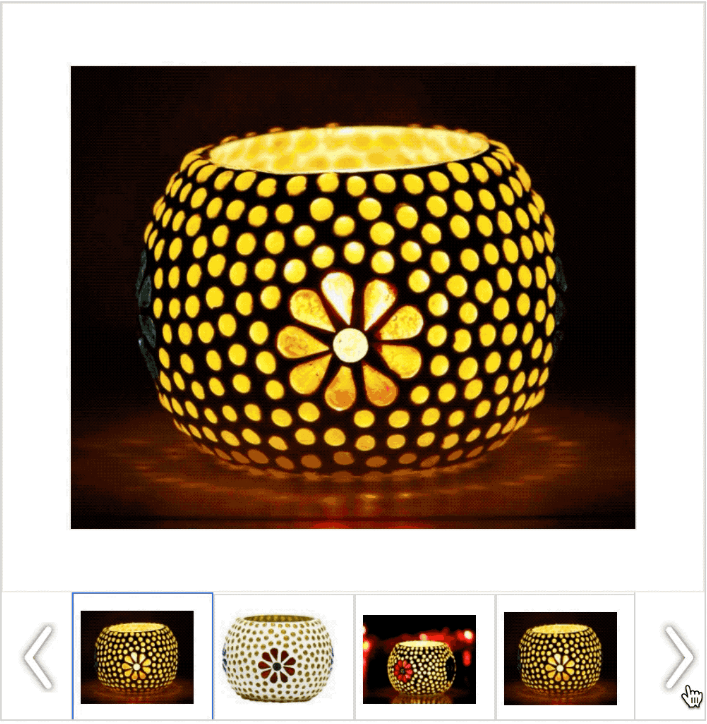
Specification Document

Goal:

Implement a simple catalog viewer.

Design:

Catalog Viewer



Implement a catalog viewer for a collection of images.

- The catalog displays the first image when opened.
- Clicking on the *previous* or *next* button displays the previous or next image respectively
- The image list is circular
  - Clicking the *next* button when the last image is showing should display the first image (cycling).
  - Clicking the *previous* button when the first image is showing should display the last image (cycling).

Note: It is not necessary to implement all of the catalog features, only those listed.

Conditions tested:

- Initially, the carousel shows the first image.
- Clicking on any carousel indicator loads the appropriate image in the main view.
- The currently selected thumbnail image should be highlighted.
- There are *previous* and *next* buttons that change the image to be displayed in the carousel.
- Cycling should be allowed.
  - Clicking *next* while showing the last image loads the first image.
  - Clicking *previous* while showing the first image loads the last image.

Question - 3

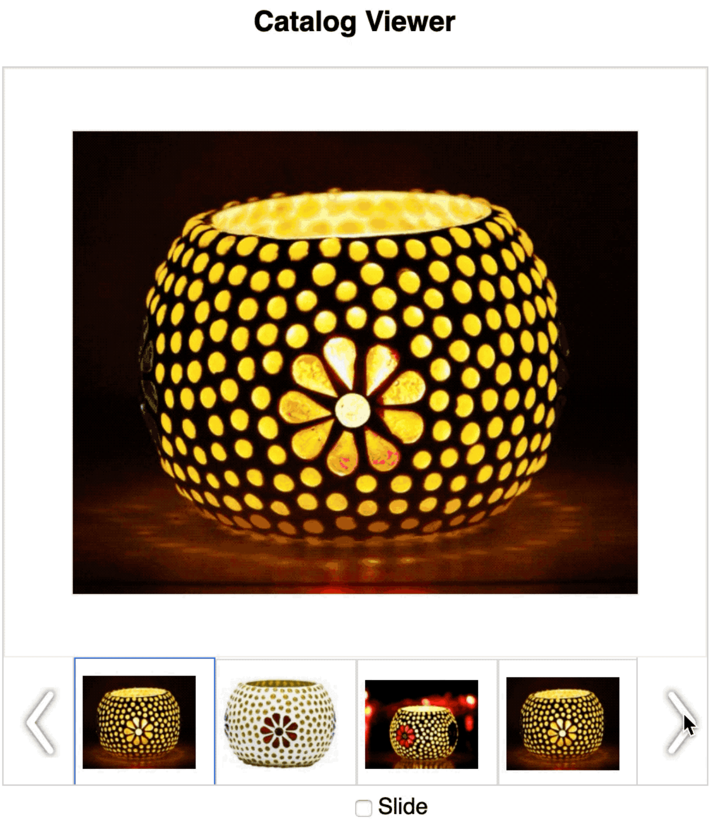
Angular: Catalog Viewer

Catalog viewer

Specification Document

Goal:
Implement a simple catalog viewer.

Design:



Implement a catalog viewer for a collection of images

- The catalog shows the first image when opened.
- Clicking on the *previous* or *next* button displays the previous or next image respectively.
- The image list is circular.
  - Clicking the *next* button when the last image is showing should display the first image (cycling).
  - Clicking the *previous* button when the first image is showing should display the last image (cycling).
- The checkbox with label *Slide* should alternately start and stop the automatic display of images in the carousel. Begin with the currently displayed image and use a 3 second interval.

Note: It is not necessary to implement all of the catalog features, only those listed.

Conditions tested:

- Initially, the carousel shows the first image.
- Clicking on any carousel indicator loads the appropriate image in the main view.
- The currently selected thumbnail image should be highlighted.
- The *previous* and *next* buttons display the appropriate image in the carousel.
- Cycling should be allowed.
    - Clicking *next* while showing the last image loads the first image.
    - Clicking *previous* while showing the first image loads the last image.
- The checkbox labeled “Slide”, when selected, causes cycling through the images with a fixed time interval of 3 seconds.
    - The slide show begins with the current image.
    - Clicking on any carousel indicator or the *previous* or *next* button when the slideshow is active loads the appropriate image and the slideshow continues from that image.