```
0.----------------------------------------------------------------------------------
---------------------------------------------
# Having the following application created with `create-react-app` add a `Company`
component and modify `CompanyList` so that:
- the app renders correctly (0.5 pts);
- `CompanyList` is rendered as a child of `App` (0.5 pts);
- `CompanyList` is rendered as a list of `Company` (0.5 pts);
- `Company` has a property called item containing the company it's supposed to
render (0.5 pts);
- `Company` can be deleted via a button with the label `delete` (0.5 pts);
-------APP-------
import React, { Component } from 'react'
import CompanyList from './CompanyList'

class App extends Component {
  render() {
    return (
      <div>
            A list of companies
            <CompanyList />
      </div>
    )
  }
}

export default App

-------COMPANY---------
import React, { Component } from 'react'

class Company extends Component{

    render(){
        let {item} = this.props

        return(
            <div>{item.name}
            <input type="button" value="delete" onClick={() =>
this.props.onDelete(item.id)} />
            </div>

        )
    }
}



export default Company


-------COMPANY LIST-----
import React, { Component } from 'react'
import CompanyStore from '../stores/CompanyStore'
import Company from './Company'

class CompanyList extends Component {
      constructor(){
            super()
            this.state = {
```

```
                        companies : []
                }
                this.deleteCompany = (id) => {
                        this.store.deleteOne(id)
                }
        }
        componentDidMount(){
                this.store = new CompanyStore()
                this.setState({
                        companies : this.store.getAll()
                })
                this.store.emitter.addListener('UPDATE', () => {
                        this.setState({
                                companies : this.store.getAll()
                        })
                })
        }
  render() {
    return (
      <div>
                        {
                        this.state.companies.map((e, i) =>
                          <Company item={e} key={i} onDelete={this.deleteCompany}/>
                        )
                }
      </div>
    )
  }
}

export default CompanyList
```

1.--

```
# Having the following application created with `create-react-app` modify `Company` component and modify `CompanyList` so that:
- the app renders correctly (0.5 pts);
- `CompanyList` is rendered as a list of `Company` and each `Company` has a button labeled `edit`(0.5 pts);
- If the edit button is clicked on a `Company` it goes into edit mode (0.5 pts);
- If a `Company` is in edit mode and the button labeled `cancel` is clicked, it goes into view mode (0.5 pts);
- A company can be saved and the changes are reflected in the company list (0.5 pts);
--------------APP-------------
import React, { Component } from 'react'
import CompanyList from './CompanyList'

class App extends Component {
  render() {
    return (
      <div>
            A list of companies
            <CompanyList />
      </div>
    )
  }
}

export default App
--------------COMPANY---------------
import React, { Component } from 'react'

class Company extends Component {
      constructor(props){
            super(props)
            let {item} = this.props
            this.state = {
                  name : item.name,
                  employees : item.employees,
      revenue : item.revenue,

      isEditing: false
            }
            this.handleChange = (evt) => {
                  this.setState({
                        [evt.target.name] : evt.target.value
                  })
            }
      }
  render() {
    let {item} = this.props
    if (this.state.isEditing){
      return (
        <div>
          <input type="text" id="name" name="name" onChange={this.handleChange}
value={this.state.name} />
           <input type="text" id="employees" name="employees"
onChange={this.handleChange} value={this.state.employees} />
            <input type="text" id="revenue" name="revenue"
onChange={this.handleChange} value={this.state.revenue} />
            <input type="button" value="save" onClick={() => {
                                          this.props.onSave(item.id, {
                                                name : this.state.name,
                                                employees : this.state.employees,
                                                revenue : this.state.revenue
                                          })
                                          this.setState({isEditing : false})
                                          }
                                    } />
          <input type="button" value="cancel" onClick={() =>
this.setState({isEditing : false})} />
        </div>
      )
    }
    else{
      return (
        <div>
          Name {item.name} with {item.employees} employees {item.revenue} revenue
          <input type="button" value="edit" onClick={() => this.setState({isEditing
: true})} />
        </div>
      )
    }
  }
}
```

```
export default Company


--------------------COMPANYLIST----------------------
import React, { Component } from 'react'
import CompanyStore from '../stores/CompanyStore'
import Company from './Company'

class CompanyList extends Component {
      constructor(){
            super()
            this.state = {
                  companies : []
            }
            this.saveCompany = (id, company) => {
                  this.store.saveOne(id, company)
            }
      }
      componentDidMount(){
            this.store = new CompanyStore()
            this.setState({
                  companies : this.store.getAll()
            })
            this.store.emitter.addListener('UPDATE', () => {
                  this.setState({
                        companies : this.store.getAll()
                  })
            })
      }
  render() {
    return (
      <div>
                  {
                  this.state.companies.map((e, i) =>
                        <Company item={e} key={i} onSave={this.saveCompany} />
                  )
            }
      </div>
    )
  }
}

export default CompanyList



----------COMPANY STORES--------------
import {EventEmitter} from 'fbemitter'

class CompanyStore{
      constructor(){
            this.companies = [{
                  id : 1,
                  name : 'acme inc',
                  employees : 100,
                  revenue : 1000
            },{
                  id : 2,
                  name : 'apex llc',
                  employees : 20,
                  revenue : 100
            }]
            this.emitter = new EventEmitter()
      }
      addOne(company){
            this.companies.push(company)
            this.emitter.emit('UPDATE')
      }
      getAll(){
            return this.companies
      }
      deleteOne(id){
            let index = this.companies.findIndex((e) => e.id === id)
            if (index !== -1){
                  this.companies.splice(index, 1)
            }
            this.emitter.emit('UPDATE')
      }
      saveOne(id, company){
            let index = this.companies.findIndex((e) => e.id === id)
            if (index !== -1){
                  Object.assign(this.companies[index], company)
            }
            this.emitter.emit('UPDATE')
      }

}

export default CompanyStore
```

2.---

# Given the server `app.js` and the file `index.html` in the `public` directory:

# Complete the following tasks:
- the file `index.html`, which contains the text `A simple app` should be delivered from the server as static content (0.5 pts);
- a button with the id `load` exists in the page and can be clicked (0.5 pts);
- when the button with the id `load` is clicked, a list of cars should be fetched from the server; cars with the color `red` loaded in the table with the id `main` with a `tr` for each car (0.5 pts);
- the table contains a `tr` for each car loaded from the server (0.5 pts);
- only `red` cars are shown (0.5 pts);

```
---------COMPANY-------------
import React, { Component } from 'react'

class Company extends Component {
  render() {
      let {item} = this.props
    return (
      <div>
            Name {item.name} with {item.employees} employees {item.revenue} revenue
        <input type="button" value="select" onClick={() =>
this.props.onSelect(item.id)} />
```

```
      </div>
    )
  }
}

export default Company


-------COMPANY DETAILS-----------
import React, { Component } from 'react'

class CompanyDetails extends Component{
    render() {
        let {item} = this.props
      return (
        <div>
            Details for {item.name}: {item.employees} employees and {item.revenue}
revenue
            <input type="button" value="cancel" onClick={() =>
this.props.onCancel()} />
        </div>
      )
    }
}

export default CompanyDetails



------COMPANYLIST------------
import React, { Component } from 'react'
import CompanyStore from '../stores/CompanyStore'
import Company from './Company'
import CompanyDetails from './CompanyDetails'

class CompanyList extends Component {
      constructor(){
            super()
            this.state = {
                  companies : [],
                  selected : null
            }

            this.selectCompany = (id) => {
                  this.store.selectCompany(id)
            }

            this.cancelSelection = () => {
                  this.setState({
                        selected: null
                  })
            }
      }
      componentDidMount(){
            this.store = new CompanyStore()
            this.setState({
                  companies : this.store.getAll()
            })
            this.store.emitter.addListener('UPDATE', () => {
                  this.setState({
                        companies : this.store.getAll(),
                        selected: this.store.getSelected()
                  })
            })
      }
  render() {
      if (this.state.selected){
            return (
                  <CompanyDetails item={this.state.selected}
onCancel={this.cancelSelection} />
            )
      }
      else{
            return (
              <div>
                  {
                        this.state.companies.map((e, i) =>
                              <Company item={e} key={i} onSelect =
{this.selectCompany}/>
                        )
                  }
              </div>
            )
      }
  }
}

export default CompanyList



-----company store---------
import {EventEmitter} from 'fbemitter'

class CompanyStore{
      constructor(){
            this.companies = [{
                  id : 1,
                  name : 'acme inc',
                  employees : 100,
                  revenue : 1000
            },{
                  id : 2,
                  name : 'apex llc',
                  employees : 20,
                  revenue : 100
            }]
            this.emitter = new EventEmitter()

            //ADAUGAT DE MN
            this.selected = null
      }

      //ADAUGAT DE MN
      getSelected(){
            return this.selected
      }
      //ADAUGAT DE MN
      selectCompany(id){
            let index = this.companies.findIndex((e) => e.id === id)
            if (index !== -1){
                  this.selected = this.companies[index]
            }
            this.emitter.emit('UPDATE')
      }

      addOne(company){
            this.companies.push(company)
            this.emitter.emit('UPDATE')
      }
      getAll(){
            return this.companies
      }
      deleteOne(id){
            let index = this.companies.findIndex((e) => e.id === id)
            if (index !== -1){
                  this.companies.splice(index, 1)
            }
            this.emitter.emit('UPDATE')
      }
}

export default CompanyStore
```

3.---
# Subject 4
# Topic: REACT

# Having the following application created with `create-react-app` complete the following tasks:
- `AddDevice` component should be rendered inside `DeviceList` component;
- `AddDevice` component should contain 2 inputs with `id`: `name` and `price`;
- `AddDevice` component should contain a `button` with the value `Submit`, used to trigger `addItem` method;
- `AddDevice` component inside `DeviceList` should contain a `props` called `onAdd`;
- When pressing `Submit` button a new item should be displayed and added to the state of `DeviceList` component;

### USEFUL INFORMATION: Objects that are added in the array of the `DeviceList` component state have the following structure: { name: String, price: Number }.

----ADD DEVICE-----------

```
import React from 'react';

class AddDevice extends React.Component {
    constructor(props) {
        super(props);
        this.state={

            name:"",
            price:""
        }

        this.handleChange = this.handleChange.bind(this);

      }

      handleChange(event) {
        let obj = {};
        obj[event.target.name] = event.target.value;
        this.setState(obj);
        console.log(obj);
      }

    render() {
        return (
                <div>
            <label>Name</label>
            <input type='text' id='name' defaultValue={this.state.name}
onChange={this.handleChange}/> <br/>
            <label>Price</label>
            <input type='text' id='price' defaultValue={this.state.price}
onChange={this.handleChange}/> <br/>
            <button type="button" value="Submit" onClick={() =>
                  this.props.onAdd({
                    name: this.state.name,
                    price: this.state.price
                  })
                }
                />
            </div>
        )
    }
}

export default AddDevice;


-------DEVICE LIST--------
import React from 'react';
import AddDevice from './AddDevice';

class DeviceList extends React.Component {
    constructor(){
        super();
        this.state = {
            devices: []
        };
        this.addDevice = device => {
            console.log(device)
                  this.state.devices.push(device)
            };
    }



    render(){
        return (
            <div>
               <AddDevice onAdd={this.addDevice}/>
            </div>
        )
    }
}

export default DeviceList;



4.-----------------------------------------------------------------------------------
-------------------
# Having the following automatic vending machine created with `create-react-app`
modify it so that:
- the app renders correctly (0.5 pts);
- the list of products is loaded from the ProducStore when `VendingMachine` is
rendered (0.5 pts)
- add the component `Product` to display the name, price and a button with the
label buy that calls the onBuy method recieved by props (0.5 pts)
- implement addTokens that increments the number of tokens by 1 at each press of
the add token button (0.5 pts)
- implement buyProduct thet substracts the tokens with the price of the product; if
there are not enough tokens nothing happens (0.5 pts)

---APP-----
import React, { Component } from 'react'
import VendingMachine from './VendingMachine'

class App extends Component {
  render() {
    return (
      <div>
            Vending Machine
            <VendingMachine />
      </div>
    )
  }
}

export default App



----------PRODUCT-----------
import React, { Component } from 'react'

class Product extends Component {
    render(){
        return(
            <div>
                <h1>{this.props.name}</h1>
                <h1>{this.props.price}</h1>
                <input type="button" value="buy" onClick={() =>
this.props.onBuy(this.props.price) } ></input>


            </div>

        )
    }
```

```
}

export default Product


-----Vending MACHINE-----------
import React, { Component } from 'react'
import Product from './Product'
import ProductStore from '../stores/ProductStore'

class VendingMachine extends Component {
    constructor() {
        super()
        this.state = {
            products: [],
            tokens: 0
        }

        this.addToken = () => {

            this.setState({tokens: this.state.tokens + 1})

        }

        this.buyProduct = (price) => {
            if(this.state.tokens >= price  ){
                this.setState({tokens:this.state.tokens -price})
             }else{
                alert("Nu sunt suficienti tokens")
             }
        }
    }
    componentDidMount(){
        let productStore = new ProductStore()

        this.setState({products: productStore.getAll()})

}


    render() {
        return (
            <div>
                {this.state.products.map((el, index) => <Product key={index}
name={el.name} price={el.price} onBuy={this.buyProduct}  />)}
                <div>Tokens: {this.state.tokens}</div>
                <input type="button" value="add token"  onClick={this.addToken}/>
            </div>
        )
    }
}

export default VendingMachine




-----PRODUCT STORE----------
import {EventEmitter} from 'fbemitter'
```

```
class ProductStore{
      constructor(){
            this.products = [{
                  id : 1,
                  name : 'coffe',
                  price : 3
            },{
                  id : 2,
                  name : 'cappuccino',
                  price: 3
            },{
                  id : 3,
                  name : 'latte',
                  price: 3
            }]
            this.emitter = new EventEmitter()
      }

      getAll(){
            return this.products
      }
}

export default ProductStore


--------------V1_2020--------------------
# Having the following application created with `create-react-app` modify `Company`
component and modify `CompanyList` so that:
- the app renders correctly (0.5 pts);
- `CompanyList` is rendered as a list of `Company` and each `Company` has a button
labeled `edit`(0.5 pts);
- If the edit button is clicked on a `Company` it goes into edit mode (0.5 pts);
- If a `Company` is in edit mode and the button labeled `cancel` is clicked, it
goes into view mode (0.5 pts);
- A company can be saved and the changes are reflected in the company list (0.5
pts);

----------COMPANY:
import React, { Component } from 'react'

class Company extends Component {
      constructor(props){
            super(props)
            let {item} = this.props
            this.state = {
                  name : item.name,
                  employees : item.employees,
      revenue : item.revenue,
      isEditing:false
            }
            this.handleChange = (evt) => {
                  this.setState({
                        [evt.target.name] : evt.target.value
                  })
            }
  }
  startEdit=()=>{this.setState({isEditing:true})}

  succes=()=>{
    let company ={
      name:this.state.name,
      employees:this.state.employees,
      revenue:this.state.revenue
    }
    this.props.onSave(this.props.item.id,company)
  }

  render() {
    let {item} = this.props
    if (this.state.isEditing){
      return (
        <div>
          <input id="name" type="text" name="name" onChange={this.handleChange}/>
          <input id="employees" type="text" name="employees"
onChange={this.handleChange}/>
          <input id="revenue" type="text" name="revenue"
onChange={this.handleChange}/>
          <input type="button" value="save" onClick={this.succes} />
          <input type="button" value="cancel"
onClick={()=>{this.setState({isEditing:false})}} />
        </div>
      )
    }
    else{
      return (
        <div>
          Name {item.name} with {item.employees} employees {item.revenue} revenue
          <input type="button" value="edit" onClick={this.startEdit}/>
        </div>
      )
    }
  }
}

export default Company

--------COMPANY LIST:
import React, { Component } from 'react'

class Company extends Component {
      constructor(props){
            super(props)
            let {item} = this.props
            this.state = {
                  name : item.name,
                  employees : item.employees,
      revenue : item.revenue,
      isEditing:false
            }
            this.handleChange = (evt) => {
                  this.setState({
                        [evt.target.name] : evt.target.value
                  })
            }
  }


  startEdit=()=>{this.setState({isEditing:true})}

  succes=()=>{
    let company ={
      name:this.state.name,
      employees:this.state.employees,
      revenue:this.state.revenue
    }
    this.props.onSave(this.props.item.id,company)
  }

  render() {
    let {item} = this.props
    if (this.state.isEditing){
      return (
        <div>
          <input id="name" type="text" name="name" onChange={this.handleChange}/>
          <input id="employees" type="text" name="employees"
onChange={this.handleChange}/>
          <input id="revenue" type="text" name="revenue"
onChange={this.handleChange}/>
          <input type="button" value="save" onClick={this.succes} />
          <input type="button" value="cancel"
onClick={()=>{this.setState({isEditing:false})}} />
        </div>
      )
    }
    else{
      return (
        <div>
          Name {item.name} with {item.employees} employees {item.revenue} revenue
          <input type="button" value="edit" onClick={this.startEdit}/>
        </div>
      )
    }
  }
}

export default Company
```

--------------------V2_2020------

# Having the following application created with `create-react-app` modify `Company` component and add `CompanyDetails` so that:
- the app renders correctly (0.5 pts);
- `CompanyList` is rendered as a list of `Company` and each `Company` has a button labeled `select`(0.5 pts);
- `CompanyDetails` has a property called item containing the company whose details it's supposed to render (0.5 pts);
- If the select button is clicked on a `Company` the details component is shown (0.5 pts);
- If the `CompanyDetails` component is shownand the button labeled `cancel` is

```
clicked, the company list is displayed (0.5 pts);

---------------company:
import React, { Component } from 'react'

class Company extends Component {



  select=()=>{
    this.props.onSelect(this.props.item.id)
  }

  render() {
      let {item} = this.props
    return (
      <div>
            Name {item.name} with {item.employees} employees {item.revenue} revenue
        <button value="select" id="select" onClick={this.select}>Select</button>
      </div>
    )
  }
}

export default Company


-----------COMPANY DETAILS:
import React, { Component } from 'react'

class CompanyDetails extends Component {

    constructor(props) {
        super(props)
    }

    cancel=()=>{
        this.props.onCancel();
    }

    render(){
        return (
            <div>
                Details for the company: {this.props.item}
                <button value="cancel" onClick={this.cancel}>Cancel </button>
            </div>
        )
    }
}

export default CompanyDetails


COMPANY LIST:
import React, { Component } from 'react'
import CompanyStore from '../stores/CompanyStore'
import Company from './Company'
import CompanyDetails from './CompanyDetails'
```

```
-----------------class CompanyList extends Component {
      constructor(){
            super()
            this.state = {
                  companies : [],
                  selected:0,
            }

      }

      select=(id)=>{
            this.setState({
                  selected:id
            })
      }
      componentDidMount(){
            this.store = new CompanyStore()
            this.setState({
                  companies : this.store.getAll()
            })
            this.store.emitter.addListener('UPDATE', () => {
                  this.setState({
                        companies : this.store.getAll()
                  })
            })
      }

      cancel=()=>{
            this.setState({selected:0})
      }

  render() {
      if (this.state.selected){
            return <CompanyDetails onCancel={this.cancel}
item={this.state.selected}/>
      }
      else{
            return (
              <div>
                  {
                        this.state.companies.map((e, i) =>
                              <Company item={e} key={i} onSelect={this.select}/>
                        )

                  }
              </div>
            )
      }
  }
}

export default CompanyList



-----------------COMPANY STORE:

import {EventEmitter} from 'fbemitter'
```

```
class CompanyStore{
      constructor(){
            this.companies = [{
                  id : 1,
                  name : 'acme inc',
                  employees : 100,
                  revenue : 1000
            },{
                  id : 2,
                  name : 'apex llc',
                  employees : 20,
                  revenue : 100
            }]
            this.emitter = new EventEmitter()
      }
      addOne(company){
            this.companies.push(company)
            this.emitter.emit('UPDATE')
      }
      getAll(){
            return this.companies
      }
      deleteOne(id){
            let index = this.companies.findIndex((e) => e.id === id)
            if (index !== -1){
                  this.companies.splice(index, 1)
            }
            this.emitter.emit('UPDATE')
      }
}

export default CompanyStore




-------------V3 2020----------------
# Having the following application created with `create-react-app` complete the
following tasks:
- `AddDevice` component should be rendered inside `DeviceList` component;
- `AddDevice` component should contain 2 inputs with `id`: `name` and `price`;
- `AddDevice` component should contain a `button` with the value `Submit`, used to
trigger `addItem` method;
- `AddDevice` component inside `DeviceList` should contain a `props` called
`onAdd`;
- When pressing `Submit` button a new item should be displayed and added to the
state of `DeviceList` component;


------ADD DEVICE:
import React from "react";

class AddDevice extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      device: {
        name: "",
        price: 0,
      },
    };
  }
  onHandleChange = (evt) => {
    let newdevice = this.state.device;
    newdevice[evt.target.name] = evt.target.value;
    newdevice.price = parseFloat(newdevice.price);
    this.setState({
      device: newdevice,
    });
    console.log(this.state.device);
  };
  render() {
    return (
      <div>
        <input
          name="name"
          type="text"
          id="name"
          onChange={this.onHandleChange}
        />
        <input
          name="price"
          type="number"
          id="price"
          onChange={this.onHandleChange}
        />
        <button
          onClick={() => {
            this.props.onAdd(this.state.device);
          }}
        >
          Submit
        </button>
      </div>
    );
  }
}

export default AddDevice;




------DEVICE LIST:
import React from 'react';
import AddDevice from './AddDevice'

class DeviceList extends React.Component {
    constructor(){
        super();
        this.state = {
            devices: []
        };
    }

    addItem =(newdevice)=>{

        let oldDevices=this.state.devices
        oldDevices.push(newdevice);
        this.setState({devices: oldDevices});
        console.log(this.state.devices)
    }

    render(){
        return (
            <div>
                <AddDevice onAdd={this.addItem}></AddDevice>
            </div>
        )
    }
}

export default DeviceList;
```

-------------V4 2020----------------
# Having the following automatic vending machine created with `create-react-app` modify it so that:
- the app renders correctly (0.5 pts);
- the list of products is loaded from the ProducStore when `VendingMachine` is rendered (0.5 pts)
- add the component `Product` to display the name, price and a button with the label buy that calls the onBuy method recieved by props (0.5 pts)
- implement addTokens that increments the number of tokens by 1 at each press of the add token button (0.5 pts)
- implement buyProduct thet substracts the tokens with the price of the product; if there are not enough tokens nothing happens (0.5 pts)

----------------APP:
```
import React, { Component } from 'react'
import VendingMachine from './VendingMachine'

class App extends Component {
  render() {
    return (
      <div>
            Vending Machine
            <VendingMachine />
      </div>
    )
  }
}

export default App
```

--------------PRODUCT:
```
import React, { Component } from 'react'

class Product extends Component {

    render(){
        let {item} = this.props
        return (
            <div>
                  {item.name}
                  <button value="buy" onClick={this.props.onBuy} />
            </div>
        )
    }
}

export default Product


---------------VENDING MACHINE:
import React, { Component } from 'react'
import Product from './Product'
import ProductStore from '../stores/ProductStore'

class VendingMachine extends Component {
    constructor() {
        super()
        this.state = {
            products: [],
            tokens: 0
        }
        this.addToken = () => {
            let oldTokens=this.state.tokens;
            oldTokens=oldTokens+1;
            this.setState({
                tokens: oldTokens
            })
        }

        this.buyProduct = (price) => {
           if(this.state.tokens>=price){
               let oldTokens=this.state.tokens;
               oldTokens=oldTokens-price;
               this.setState({
                   tokens: oldTokens
               })
           }
        }
    }
    componentDidMount(){
            this.store = new ProductStore()
            this.setState({
                  products:  this.store.getAll()
            })
      }
    render() {
        return (
            <div>
                {this.state.products.map((el, index) =>
                <Product key={index} item={el}
onBuy={()=>{this.buyProduct(el.price)}}  />)}
                <div>Tokens: {this.state.tokens}</div>
                <input type="button" value="add token" onClick={this.addToken} />
            </div>
        )
    }
}

export default VendingMachine



--------------PRODUCT STORE:

import {EventEmitter} from 'fbemitter'

class ProductStore{
      constructor(){
            this.products = [{
                  id : 1,
                  name : 'coffe',
                  price : 3
            },{
                  id : 2,
                  name : 'cappuccino',
                  price: 3
            },{
                  id : 3,
                  name : 'latte',
                  price: 3
            }]
            this.emitter = new EventEmitter()
      }

      getAll(){
            return this.products
      }
}

export default ProductStore






------------------------------------------------------------------------------------
------------------------------

# Having the following automatic vending machine created with `create-react-app`
modify it so that:
- the app renders correctly (0.5 pts);
- the list of products is loaded from the ProductStore when `VendingMachine` is
rendered (0.5 pts)
- add the component `Product` to display the name, price and a button with the
label buy that calls the onBuy method recieved by props (0.5 pts)
- implement addTokens that increments the number of tokens by 1 at each press of
the add token button (0.5 pts)
- implement buyProduct thet substracts the tokens with the price of the product; if
there are not enough tokens nothing happens (0.5 pts)


-----------VENDING MACHINE:
import React, { Component } from 'react'
import Product from './Product'
import ProductStore from '../stores/ProductStore'

class VendingMachine extends Component {
    constructor() {
        super()
        this.state = {
            products: [],
            tokens: 0
        }

        this.addToken = () => {

        }

        this.buyProduct = (price) => {

        }
    }


    render() {
        return (
            <div>
                {this.state.products.map((el, index) => <Product key={index}
name={el.name} price={el.price} onBuy={this.buyProduct}  />)}
                <div>Tokens: {this.state.tokens}</div>
                <input type="button" value="add token" />
            </div>
        )
    }
}

export default VendingMachine
```