Here you'll build a cash register app that will return change to the customer based on the price of the item, the amount of cash provided by the customer, and the amount of cash in the cash drawer. You'll also need to show different messages to the user in different scenarios, such as when the customer provides too little cash or when the cash drawer doesn't have enough to issue the correct change.

In the script.js file, you have been provided with the price and cid variables. The price variable is the price of the item, and the cid variable is the cash-in-drawer, which is a 2D array listing the available currency in the cash drawer.

The other variable you will need to add is the cash variable, which is the amount of cash provided by the customer for the item, which is provided via an input element on the page.

If you'd like to test your application with different values for price and cid, make sure to declare them with the let keyword so they can be reassigned by our tests.

Your application should show different messages depending on the price of the item, the amount of cash provided by the customer, and the amount of cash in the drawer:

* "Status: INSUFFICIENT\_FUNDS": if cash-in-drawer is less than the change due, or if you cannot return the exact change.
* "Status: CLOSED": if cash-in-drawer is equal to the change due.
* "Status: OPEN": if cash-in-drawer is greater than the change due and you can return change, with the change due in coins and bills sorted in highest to lowest order.

| **Currency Unit** | **Amount** |
| --- | --- |
| Penny | $0.01 (PENNY) |
| Nickel | $0.05 (NICKEL) |
| Dime | $0.1 (DIME) |
| Quarter | $0.25 (QUARTER) |
| Dollar | $1 (ONE) |
| Five Dollars | $5 (FIVE) |
| Ten Dollars | $10 (TEN) |
| Twenty Dollars | $20 (TWENTY) |
| One Hundred Dollars | $100 (ONE HUNDRED) |

**Objective:** Build an app that is functionally similar to [https://cash-register.freecodecamp.rocks](https://cash-register.freecodecamp.rocks/)

**User Stories:**

1. ~~You should have an input element with an id of "cash"~~
2. ~~You should have a div, span or p element with an id of "change-due"~~
3. ~~You should have a button element with an id of "purchase-btn"~~
4. ~~When the value in the #cash element is less than price, an alert should appear with the text "Customer does not have enough money to purchase the item"~~
5. ~~When the value in the #cash element is equal to price, the value in the #change-due element should be "No change due - customer paid with exact cash"~~
6. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 1.01], ["NICKEL", 2.05], ["DIME", 3.1], ["QUARTER", 4.25], ["ONE", 90], ["FIVE", 55], ["TEN", 20], ["TWENTY", 60], ["ONE HUNDRED", 100]] , and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: OPEN QUARTER: $0.5"
7. When price is 3.26, the value in the #cash element is 100, cid is [["PENNY", 1.01], ["NICKEL", 2.05], ["DIME", 3.1], ["QUARTER", 4.25], ["ONE", 90], ["FIVE", 55], ["TEN", 20], ["TWENTY", 60], ["ONE HUNDRED", 100]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: OPEN TWENTY: $60 TEN: $20 FIVE: $15 ONE: $1 QUARTER: $0.5 DIME: $0.2 PENNY: $0.04"
8. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 0.01], ["NICKEL", 0], ["DIME", 0], ["QUARTER", 0], ["ONE", 0], ["FIVE", 0], ["TEN", 0], ["TWENTY", 0], ["ONE HUNDRED", 0]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: INSUFFICIENT\_FUNDS"
9. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 0.01], ["NICKEL", 0], ["DIME", 0], ["QUARTER", 0], ["ONE", 1], ["FIVE", 0], ["TEN", 0], ["TWENTY", 0], ["ONE HUNDRED", 0]] , and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: INSUFFICIENT\_FUNDS"
10. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 0.5], ["NICKEL", 0], ["DIME", 0], ["QUARTER", 0], ["ONE", 0], ["FIVE", 0], ["TEN", 0], ["TWENTY", 0], ["ONE HUNDRED", 0]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: CLOSED PENNY: $0.5"

Fulfill the user stories and pass all the tests below to complete this project. Give it your own personal style. Happy Coding!

**Tests**

* ~~Waiting:1. You should have the HTML file link to the JavaScript file.~~
* ~~Waiting:2. You should have a global variable called price.~~
* ~~Waiting:3. You should have a global variable called cid.~~
* ~~Waiting:4. You should have an input element with an id of "cash".~~
* ~~Waiting:5. You should have a div, span, or p element with an id of "change-due".~~
* ~~Waiting:6. You should have a button element with an id of "purchase-btn".~~
* ~~Waiting:7. When price is 20 and the value in the #cash element is 10, an alert should appear with the text "Customer does not have enough money to purchase the item".~~
* ~~Waiting:8. When the value in the #cash element is less than price, an alert should appear with the text "Customer does not have enough money to purchase the item".~~
* ~~Waiting:9. When price is 11.95 and the value in the #cash element is 11.95, the value in the #change-due element should be "No change due - customer paid with exact cash".~~
* ~~Waiting:10. When the value in the #cash element is equal to price, the value in the #change-due element should be "No change due - customer paid with exact cash".~~
* Waiting:11. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 1.01], ["NICKEL", 2.05], ["DIME", 3.1], ["QUARTER", 4.25], ["ONE", 90], ["FIVE", 55], ["TEN", 20], ["TWENTY", 60], ["ONE HUNDRED", 100]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: OPEN QUARTER: $0.5".
* Waiting:12. When price is 3.26, the value in the #cash element is 100, cid is [["PENNY", 1.01], ["NICKEL", 2.05], ["DIME", 3.1], ["QUARTER", 4.25], ["ONE", 90], ["FIVE", 55], ["TEN", 20], ["TWENTY", 60], ["ONE HUNDRED", 100]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: OPEN TWENTY: $60 TEN: $20 FIVE: $15 ONE: $1 QUARTER: $0.5 DIME: $0.2 PENNY: $0.04".
* Waiting:13. When price is less than the value in the #cash element, total cash in drawer cid is greater than the change due, individual denomination amounts allows for returning change due, and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: OPEN" with required change due in coins and bills sorted in highest to lowest order.
* Waiting:14. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 0.01], ["NICKEL", 0], ["DIME", 0], ["QUARTER", 0], ["ONE", 0], ["FIVE", 0], ["TEN", 0], ["TWENTY", 0], ["ONE HUNDRED", 0]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: INSUFFICIENT\_FUNDS"
* Waiting:15. When the price is less than the value in the #cash element and the total cash in the drawer (cid) is insufficient to cover the change due, the purchase should not proceed. When the #purchase-btn is clicked under these conditions, the #change-due element should display "Status: INSUFFICIENT\_FUNDS".
* Waiting:16. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 0.01], ["NICKEL", 0], ["DIME", 0], ["QUARTER", 0], ["ONE", 1], ["FIVE", 0], ["TEN", 0], ["TWENTY", 0], ["ONE HUNDRED", 0]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: INSUFFICIENT\_FUNDS".
* Waiting:17. When price is less than the value in the #cash element, total cash in drawer cid is greater than change due, but the individual denomination amounts make it impossible to return needed change, when the #purchase-btn element is clicked, the value in the #change-due element should be "Status: INSUFFICIENT\_FUNDS"
* Waiting:18. When price is 19.5, the value in the #cash element is 20, cid is [["PENNY", 0.5], ["NICKEL", 0], ["DIME", 0], ["QUARTER", 0], ["ONE", 0], ["FIVE", 0], ["TEN", 0], ["TWENTY", 0], ["ONE HUNDRED", 0]], and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: CLOSED PENNY: $0.5".
* Waiting:19. When price is less than the value in the #cash element, total cash in drawer cid is equal to change due, and the #purchase-btn element is clicked, the value in the #change-due element should be "Status: CLOSED" with change due in coins and bills sorted in highest to lowest order.