

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**  
**BỘ MÔN KIỂM THỬ PHẦN MỀM**

-----o0o-----



**TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH**  
HO CHI MINH CITY OPEN UNIVERSITY

**BÀI TẬP LỚN**  
**BÁO CÁO UNIT TEST**

Giảng viên hướng dẫn: Phan Trần Minh Khuê, Nguyễn Trung Hậu

Sinh viên thực hiện:

Nguyễn Văn Bin                      MSSV: 2151012001

Đặng Trần Thiên Phúc              MSSV: 2151013071

Trương Lâm Quốc Thái              MSSV: 2151010338

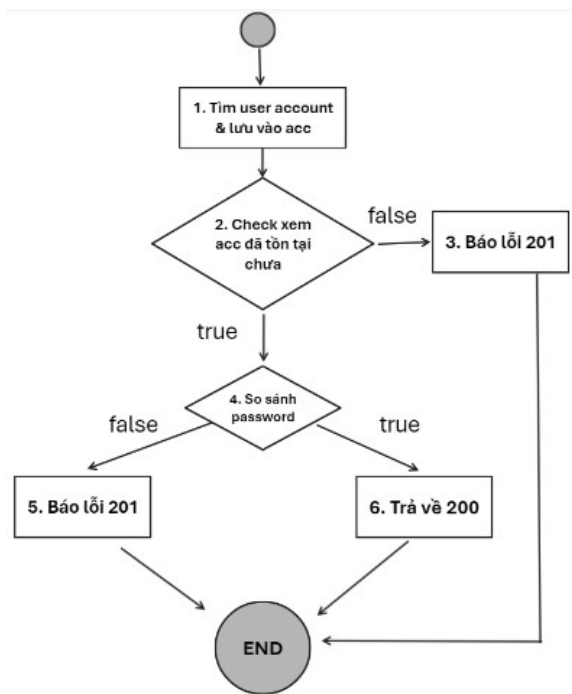
**TP. HỒ CHÍ MINH, THÁNG 05 NĂM 2024**

# MỤC LỤC

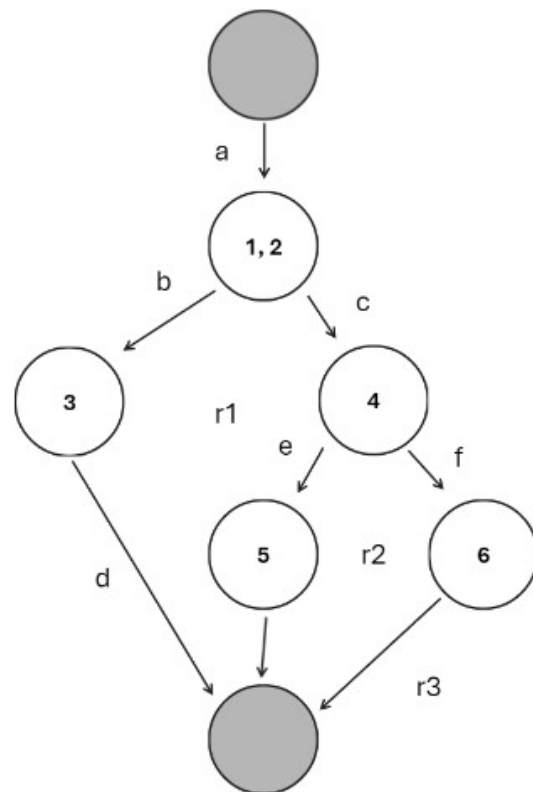
Test Case ID: b2 - Kiểm tra tài khoản và mật khẩu khách nhập trong phần đăng nhập có hợp lệ không? .....	3
Test Case ID: b3 - Kiểm tra mật khẩu có được mã hóa trong cơ sở dữ liệu không? .....	4
Test Case ID: c1 - 1: Hàm có trả về đầy đủ thông tin của cuốn sách không? .....	6
Test Case ID: c1 - 2: Hàm trả về thông tin 12 cuốn sách mới nhất .....	7
Test Case ID: c2 - 1: Hàm filter có trả đúng danh sách sách theo checkbox hay không? .....	8
Test Case ID: c2 - 2: Giỏ hàng có chứa đủ lượng sách đã được thêm không? .....	9
Test Case ID: c2 - 3: Hàm có trả về đúng số lượng trong giỏ hàng không? .....	10
Test Case ID: c2 - 4: Hàm có trả về được đúng theo ký tự đã nhập trong thanh tìm kiếm không? .....	11
Test Case ID: c2 - 5: Hàm có trả về đủ và đúng lịch sử đơn đã đặt theo mỗi tài khoản không? .....	12

## Test Case ID: b2 - Kiểm tra tài khoản và mật khẩu khách nhập trong phần đăng nhập có hợp lệ không?

### Sơ đồ luồng



### Đồ thị luồng



```

const checkAccount = require('../function/checkAccount');

describe('kiểm tra tài khoản và mật khẩu', () => {
  let Phuduongdan1, Phuduongdan2, Phuduongdan3, Phuduongdan4, Phuduongdan5, Phuduongdan6, Phuduongdan7;
  let Phunhanh1, Phunhanh2
  beforeEach(async () => {
    // Gọi hàm checkAccount và lưu kết quả vào biến result
    Phunhanh1 = await checkAccount('thai123', '343');
    Phunhanh2 = await checkAccount('123', '123');

    Phuduongdan1 = await checkAccount('123', '123');
    Phuduongdan2 = await checkAccount('123', '456');
    Phuduongdan3 = await checkAccount('thai123', '343');
    Phuduongdan4 = await checkAccount('', '343');
    Phuduongdan5 = await checkAccount('', '');
    Phuduongdan6 = await checkAccount('', '343');
    Phuduongdan7 = await checkAccount('thai123', '');
  });

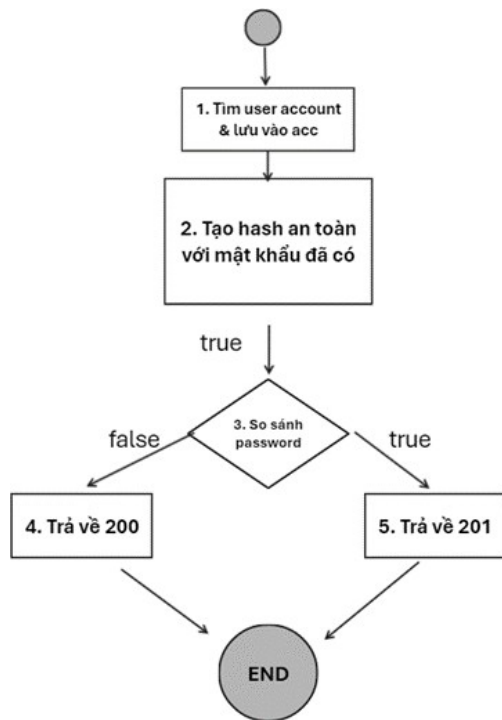
  test('', () => {
    // Kiểm tra kết quả
    expect(Phunhanh1).toBe(201);
    expect(Phunhanh2).toBe(201);

    expect(Phuduongdan1).toBe(201);
    expect(Phuduongdan2).toBe(201);
    expect(Phuduongdan3).toBe(201);
    expect(Phuduongdan4).toBe(201);
    expect(Phuduongdan5).toBe(201);
    expect(Phuduongdan6).toBe(201);
    expect(Phuduongdan7).toBe(201);
  });
});

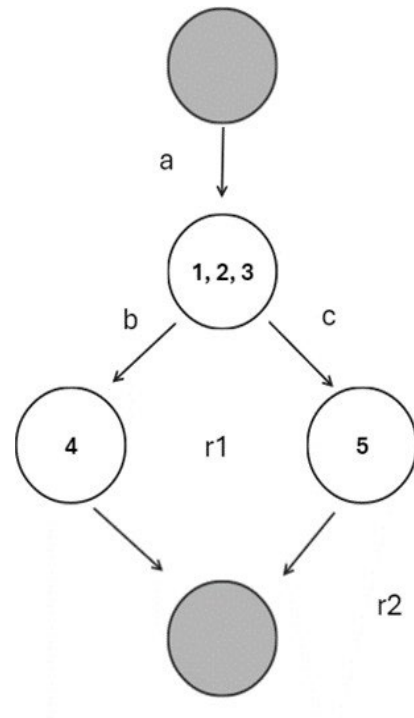
```

**Test Case ID: b3 - Kiểm tra mật khẩu có được mã hóa trong cơ sở dữ liệu không?**

## Sơ đồ luồng



## Đồ thị luồng



```
const checkEncryptedPass = require('../function/checkEncryptedPass');

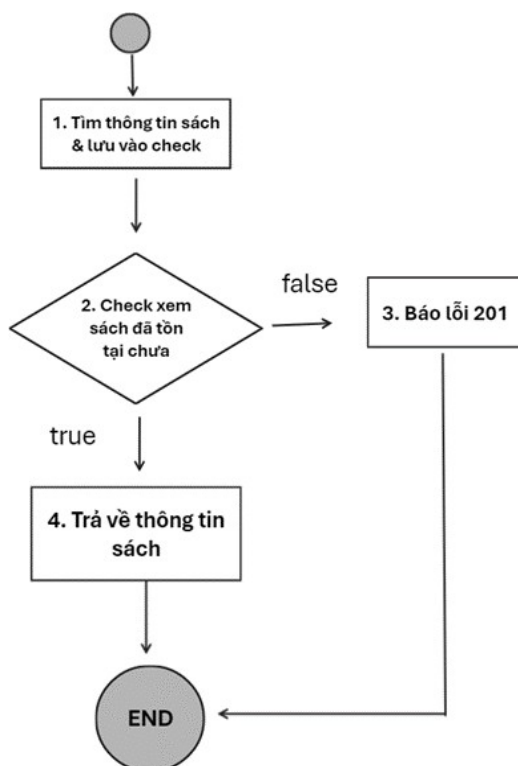
describe('kiểm tra tài mật khẩu được mã hóa', () => {
  let result;

  beforeEach(async () => {
    // Gọi hàm checkAccount và lưu kết quả vào biến result
    ExpectedResult = await checkEncryptedPass('123', '123');
  });

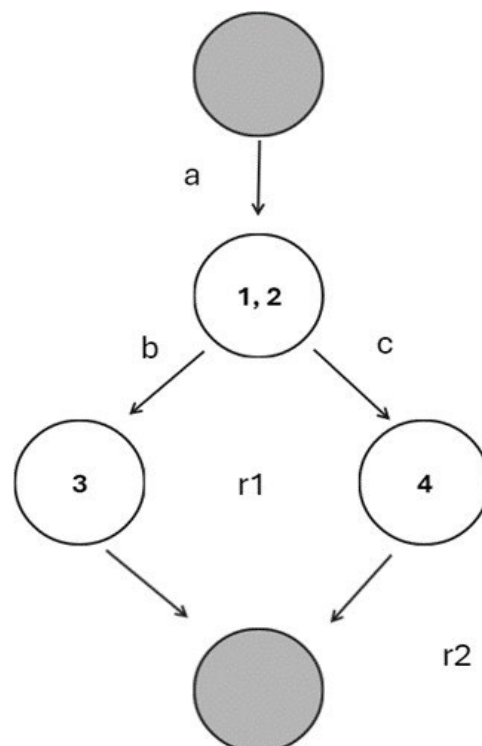
  test('', () => {
    expect(ExpectedResult).toBe(200);
  });
});
```

## Test Case ID: c1 - 1: Hàm có trả về đầy đủ thông tin của cuốn sách không?

### Sơ đồ luồng



### Đồ thị luồn



```
const takeFullBookInfor = require('../function/takeFullBookInfor');

describe('kiểm tra hàm lấy toàn bộ thông tin cuốn sách', () => {
  test('', async () => {
    // Gọi hàm takeFullBookInfor và lưu kết quả vào biến takeFullBookInforResult
    const notExist = await takeFullBookInfor('YOUR NAME');
    const takeFullBookInforResult = await takeFullBookInfor('BOCCHI THE ROCK - TẬP 1');

    // Kiểm tra xem kết quả có tồn tại không
    expect(takeFullBookInforResult).toBeTruthy();

    // Kiểm tra các thông tin của cuốn sách
    expect(takeFullBookInforResult.name).toBe('BOCCHI THE ROCK - TẬP 1');
    expect(takeFullBookInforResult.author).toBe('Aki Hamazi');
    expect(takeFullBookInforResult.description).toBe('Goto Hitori là một nữ sinh hướng nội. Giao tiếp kém, học');
    expect(takeFullBookInforResult.TrangThai).toBe('Đang phát hành');
    expect(takeFullBookInforResult.NamXuatBan).toBe('2024');
    expect(takeFullBookInforResult.KichThuoc).toBe('14,5 x 20,5 cm');
    expect(takeFullBookInforResult.DoiTuong).toBe('15+');

    expect(notExist).toBe(201);
  });
});
```

## Test Case ID: c1 - 2: Hàm trả về thông tin 12 cuốn sách mới nhất

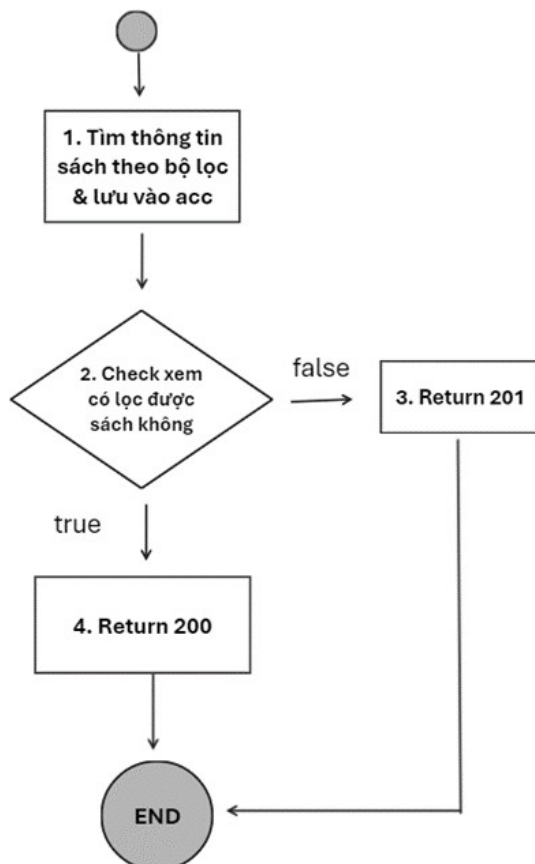
```
const newestProduct = require('../function/check_newest_product');

describe('kiểm tra 12 cuốn sách mới nhất mới nhất ', () => {
  beforeEach(async () => {
    New12Product = await newestProduct();
  });

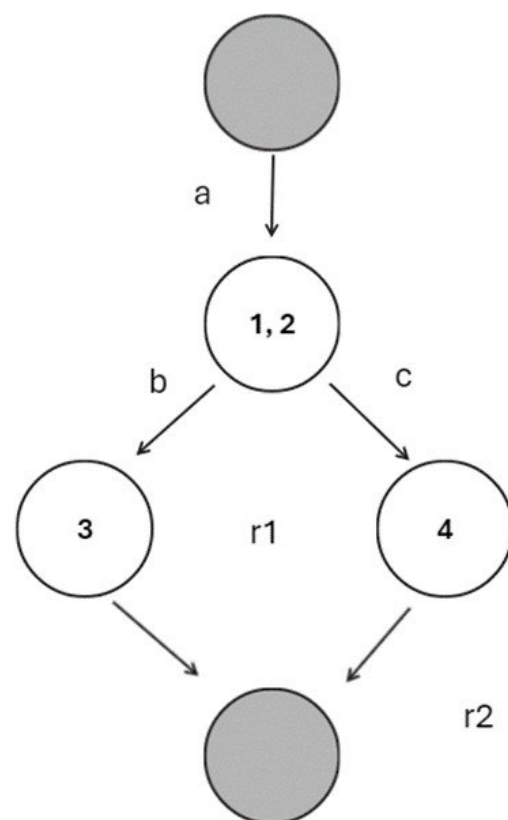
  test('', () => {
    expect(New12Product[0].name).toBe('NHÂN VẬT HẠ CẤP TOMOZAKI - TẬP 6.5');
    expect(New12Product[1].name).toBe('KHẼ HÁT LỜI YÊU - TẬP 3');
    expect(New12Product[2].name).toBe('THIÊN SỬ NHÀ BÊN (MANGA) - TẬP 1');
    expect(New12Product[3].name).toBe('KHÁT VỌNG TỎA SÁNG CỦA NANAMI MINAMI (MANGA) - TẬP 1');
    expect(New12Product[4].name).toBe('SKIP AND LOAFER - NHỊP BƯỚC TUỔI XANH - TẬP 4');
    expect(New12Product[5].name).toBe('NGÔN NGỮ YÊU THƯƠNG - TẬP 3');
    expect(New12Product[6].name).toBe('CHÀNG BĂNG GIÁ VÀ NÀNG LẠNH LÒNG - TẬP 6');
    expect(New12Product[7].name).toBe('CÓ CHỖ CỐ MÈO, NGÀY NÀO CŨNG VUI - TẬP 5');
    expect(New12Product[8].name).toBe('NEON GENESIS EVANGELION - COLLECTORS EDITION - TẬP 4');
    expect(New12Product[9].name).toBe('OSHI NO KO - DƯỚI ÁNH HÀO QUANG - TẬP 7');
    expect(New12Product[10].name).toBe('FRIEREN - PHÁP SƯ TIỀN TÁNG - TẬP 3');
    expect(New12Product[11].name).toBe('NHÂN VẬT HẠ CẤP TOMOZAKI - TẬP 6');
  });
});
```

## Test Case ID: c2 - 1: Hàm filter có trả đúng danh sách sách theo checkbox hay không?

### Sơ đồ luồng



### Đồ thị luồng



```
const checkFilter = require('../function/checkFilter');

describe('test Filter --- task 4.2-1', () => {
  let result1, result2;

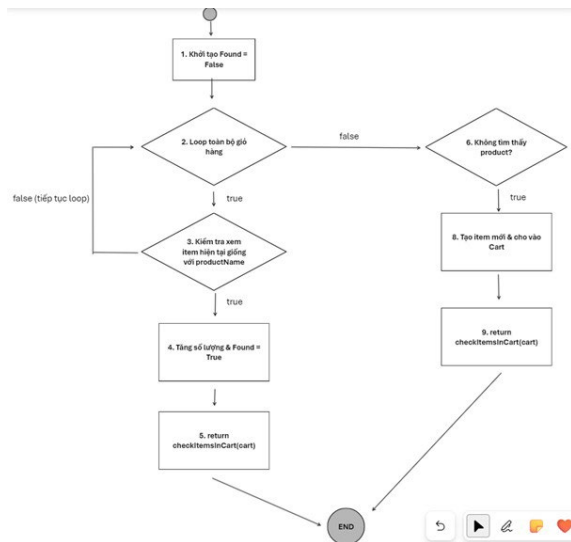
  beforeEach(async () => {
    result1 = await checkFilter('Âm nhạc', 'BOCCHI THE ROCK - TẬP 1');
    result2 = await checkFilter('Âm nhạc', 'KHẼ HÁT LỜI YÊU - TẬP 1');
  });

  test('Check filter work', () => {
    // Kiểm tra kết quả
    expect(result2).toBe(201);
    expect(result1).toBe(200);
  });
});
```

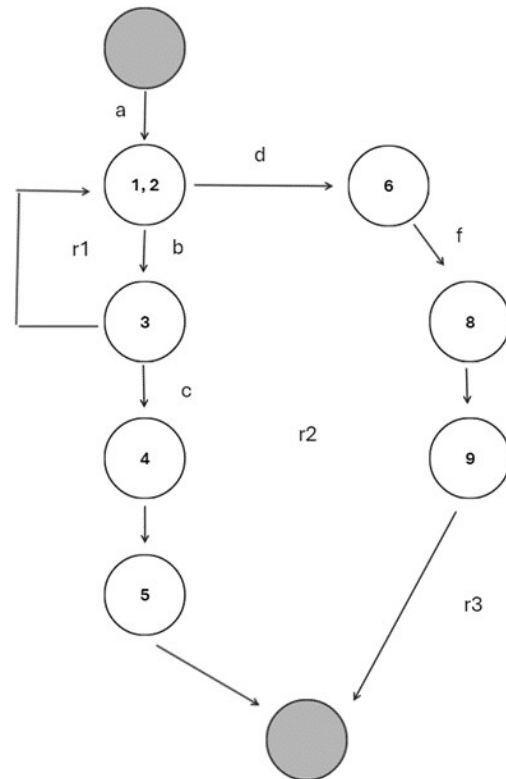


## Test Case ID: c2 - 2: Giỏ hàng có chứa đủ lượng sách đã được thêm không?

### Sơ đồ luồng



### Đồ thị luồng

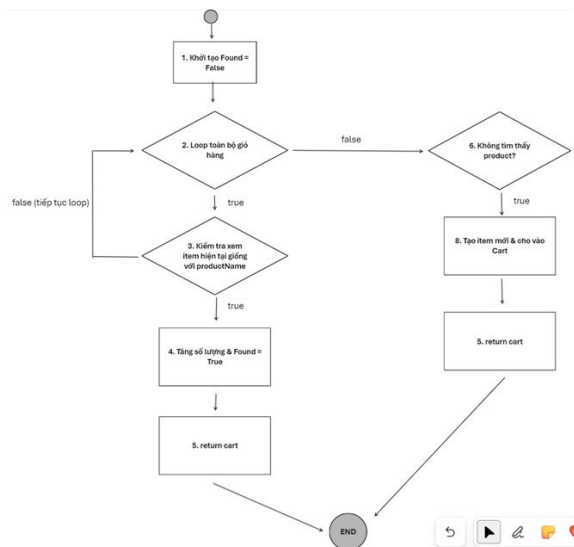


```
const {additem} = require('../function/checkItemsInCart');
```

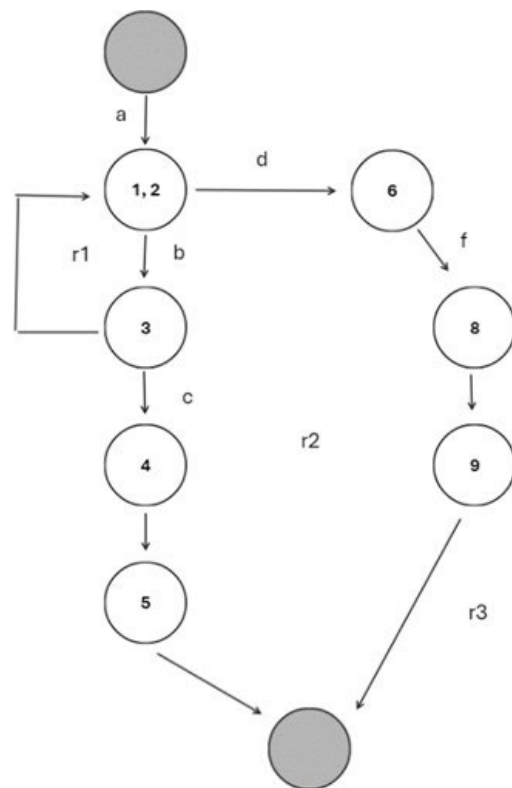
```
describe('kiểm tra tổng item hiện có trong giỏ hàng', () => {
  test('', async () => {
    let cart = [];
    const result1 = additem('BOCCHI THE ROCK - TẬP 1', cart)
    const result2 = additem('KHẾ HÁT LỜI YÊU - TẬP 1', cart)
    expect(result1).toBe(1)
    expect(result2).toBe(2)
    //hàm trả về số lượng item hiện có trong giỏ hàng
  });
});
```

## Test Case ID: c2 - 3: Hàm có trả về đúng số lượng trong giỏ hàng không?

### Sơ đồ luồng



### Đồ thị luồng



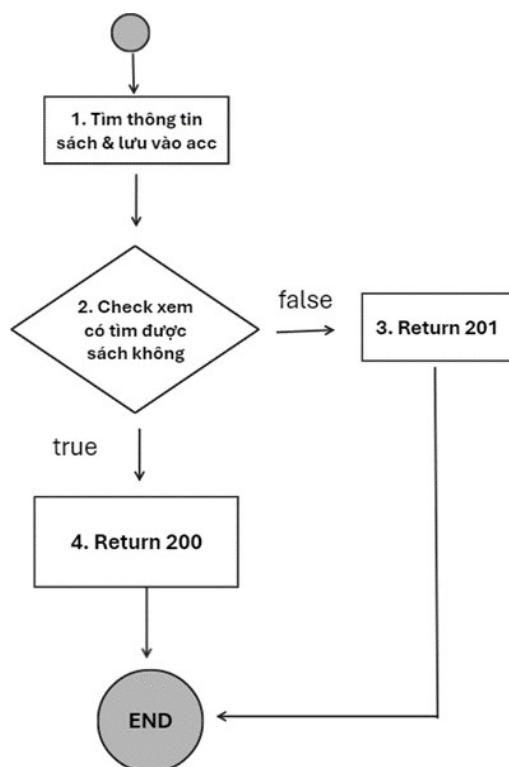
```
const itemsInCart = require('../function/itemsInCart');

describe('kiểm tra số lượng riêng của mỗi sản phẩm trong giỏ hàng', () => {
  test('Test itemsInCart function', async () => {
    let cart = [];
    const result1 = itemsInCart('BOCCHI THE ROCK - TẬP 1', cart)
    const result1_1 = itemsInCart('BOCCHI THE ROCK - TẬP 1', cart)

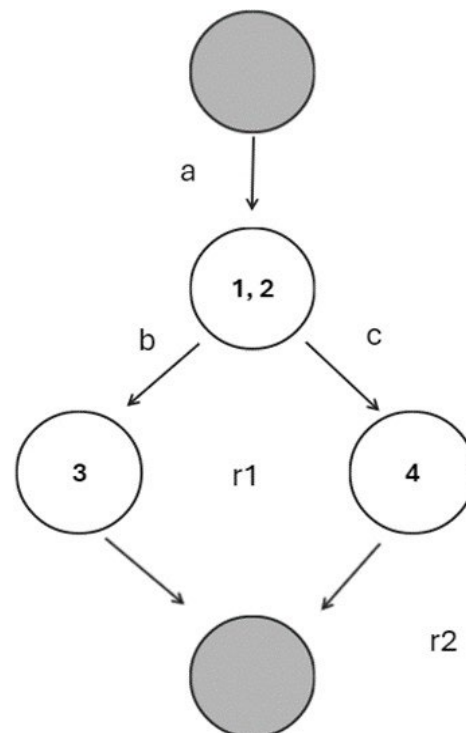
    const result2 = itemsInCart('KHẾ HÁT LỜI YÊU - TẬP 1', cart)
    expect((result1_1.find(cart => cart.name === 'BOCCHI THE ROCK - TẬP 1')).quantity).toBe(2)
    expect((result2.find(cart => cart.name === 'KHẾ HÁT LỜI YÊU - TẬP 1')).quantity).toBe(1)
  });
  // kiểm tra số lượng mỗi item
});
```

## Test Case ID: c2 - 4: Hàm có trả về đúng theo ký tự đã nhập trong thanh tìm kiếm không?

### Sơ đồ luồng



### Đồ thị luồng



```
const testSearchbar = require('../function/testSearchbar');

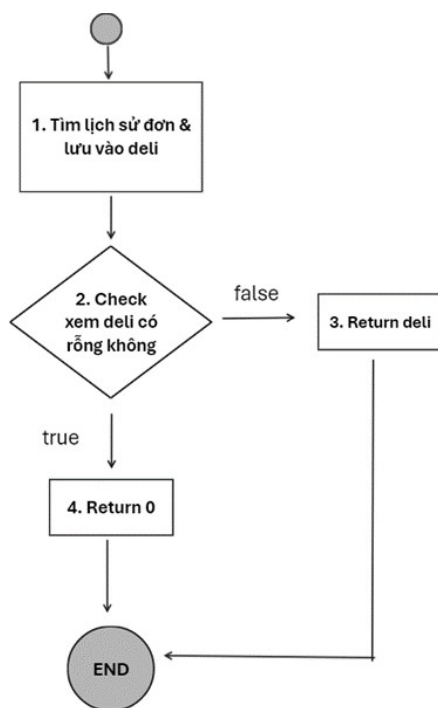
describe('kiểm tra hàm tìm kiếm trong search bar', () => {
  let result;

  beforeEach(async () => {
    result = await testSearchbar('BOCCHI THE ROCK - TẬP 1');
    result2 = await testSearchbar('YOUR NAME');
  });

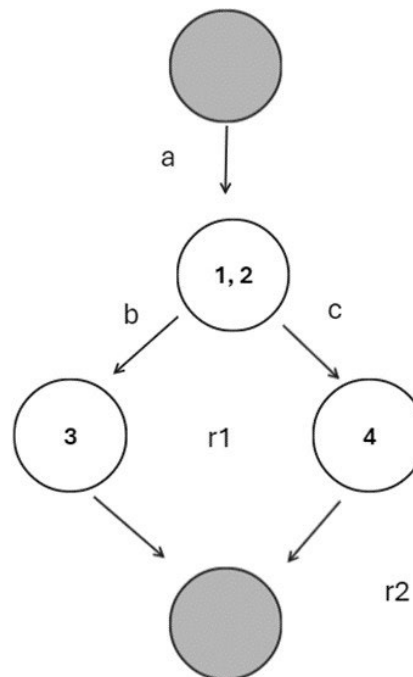
  test('', () => {
    // Kiểm tra kết quả
    expect(result).toBe(200);
    expect(result2).toBe(201);
  });
});
```

## Test Case ID: c2 - 5: Hàm có trả về đủ và đúng lịch sử đơn đã đặt theo mỗi tài khoản không?

### Sơ đồ luồng



### Đồ thị luồng



```
const checkHistoryOrder = require('../function/checkHistoryOrder');
```

```
describe('kiểm tra lịch sử đơn hàng', () => {
  test('', async () => {
    const history_bin = await checkHistoryOrder('bin')
    const history_phuc = await checkHistoryOrder('phuc')
    const history_nullclient = await checkHistoryOrder('')
    const history_binprovip = await checkHistoryOrder('bin#provip123')

    expect(history_bin).not.toBe(0);
    expect(history_phuc).not.toBe(0);
    expect(history_nullclient).toBe(0);
    expect(history_binprovip).toBe(0);
  });
});
```

