****

**SCHOOL OF ELECTRICAL, ELECTRONICS AND COMPUTER ENGINEERING**

**FLOWER SHOP APPLICATION**

**dOCUMENTATION**

**Mark Angelo B. Aquino**

**Cyrille Lan Chua**

**Angelo Hubalde**

**Krisha Rain T. Martinez**

October 2018

**CONTENTS**

**…………………………………………………………………………………………………………………………..**

1. **Introduction 3**
2. Purpose 3
3. Product Backlog 3
4. User Manual 4

**…..………………………………………………………………………………………………………………………**

1. **Requirements 6**
2. User Requirements 6
3. Functional Requirements 6
4. Non-functional Requirements 7

**……………………………………………………………………………………………………………..……………**

1. **High-Level Design 7**
2. User Interface Sketches 7
3. Use Case Diagram 8
4. Conceptual Class Diagram 9

**…………………………………………………………………………………………………..………………………**

1. **Low-Level Design 9**
2. Database Diagram (ERD) 9
3. Detailed Class Diagram 10

**………………………………………………………………………………………………………………..…………**

1. **Development 10**
2. Source Codes 10
3. Data Access 10
4. Business Logic 12
5. User Interface 14

**………………………………………………………………………………………………………..…………………**

1. **Testing 25**
2. Unit testing codes 25
3. Data Access 25
4. Business Logic 26
5. User Interface 26

**……………………………………………………………………………………………………………….**

**INTRODUCTION**

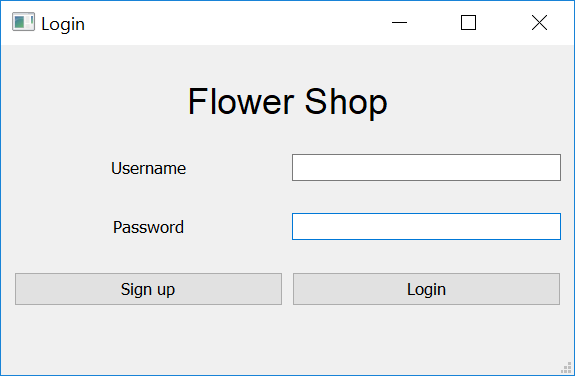
1. **PURPOSE**

Flower shops are one of the places to buy a gift for our loved ones, dead or alive. Many flower shops still rely on manual ordering system and manual computation for the prices. The flower shop application erases those hassles of manual labors it offers a feature wherein the users have their own account created by them, it lets the user know when he/she entered a wrong username or password. They can purchase their desired flowers by exploring the menu for the flowers showing their ordered type of flower, the quantity of each flower to be ordered, the price of each flower and the current total price.

1. **PRODUCT BACKLOG**

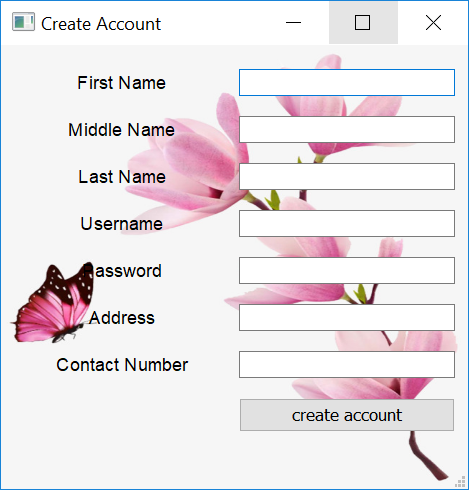
|  |  |
| --- | --- |
| User Login Function  As a user I want to be able to login by typing my created username and password  Status: Done | Product Search Function  As a user I want to be able to search for a specific type of flower I want to order  Status: Done |
| User account Creation Function  As a user I want to be able to create my own account instead of having them create for me  Status: Done | User’s Manual  As a user I want to have a user’s manual that can guide me in my orders  Status: Done |
| Product list Reset  As a user I want to have a feature wherein you can clear all your current orders and start from zero  Status: Done | Continuous Price Update  As a user I want to see the current price of my order every time I add a flower to order  Status: Done |
| Final Order Checking  As a user I want to have a feature that asks me first whether my order is final or not  Status: Done | User Login Page Error  As a user I want to know if I have entered my username and password wrong  Status: Done |

1. **USER MANUAL**



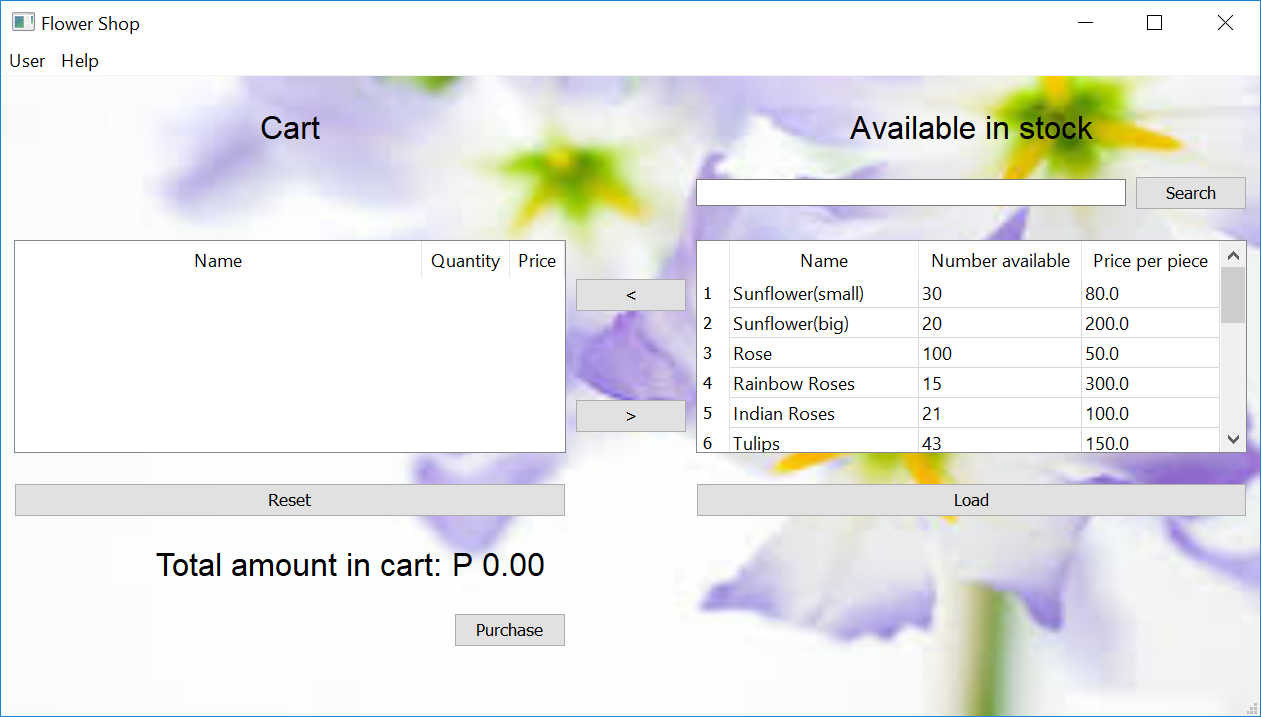
Here is the window at the start of the program.

The user can either Sign up, to create a new account, or Login to his/her existing account.



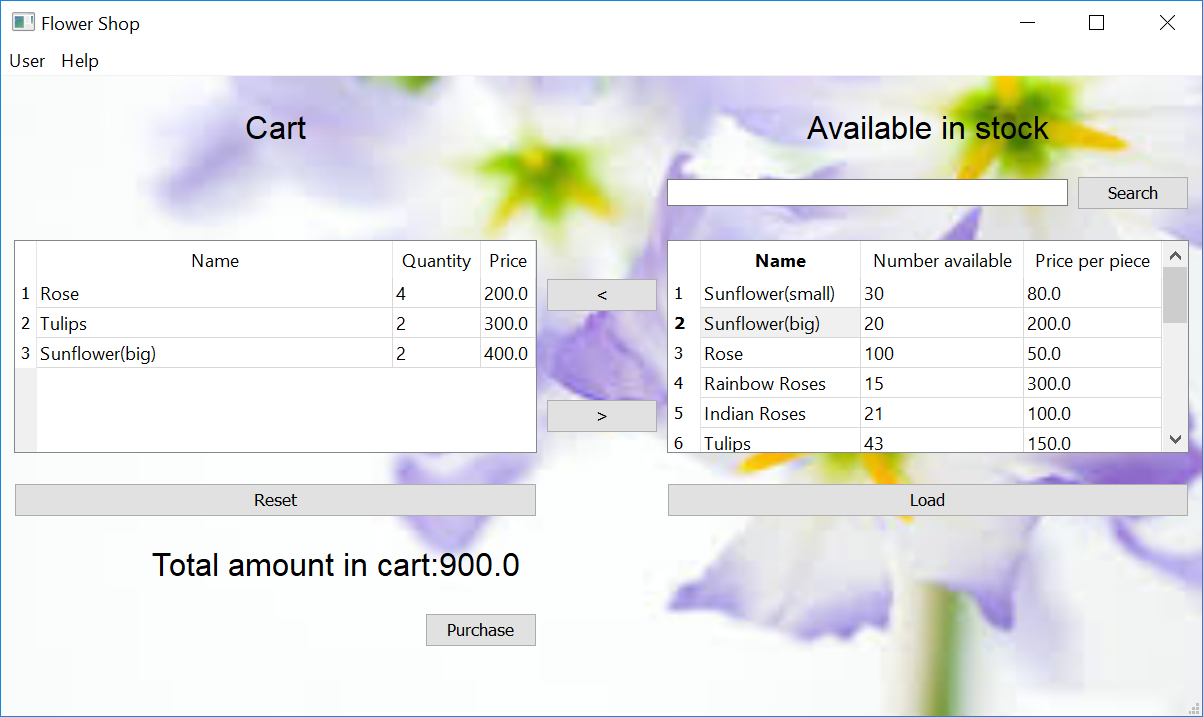
If the user chooses to Sign up, this window will pop up.

The user is required to input his First Name, Middle Name, Last Name, Username, Password, Address and Contact Number. Pressing on the Create Account button will save the user inputs into the database.



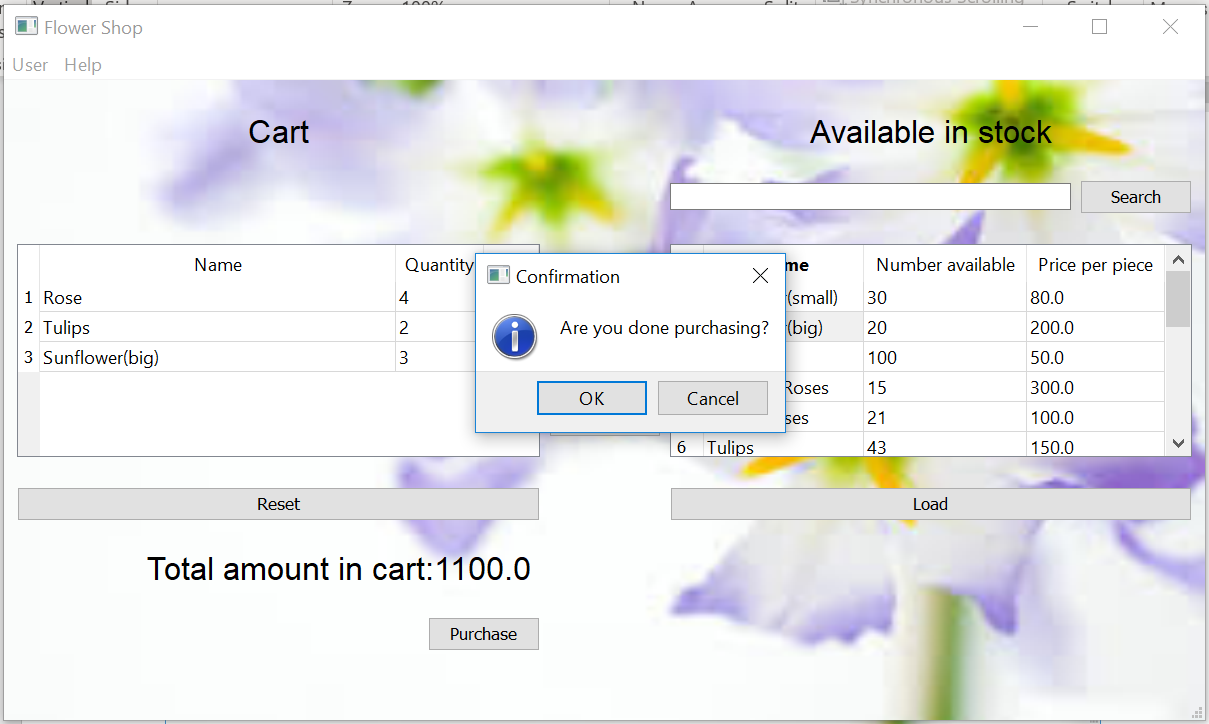
If the user chooses the Login button and the input credentials are correct, this window will appear. This window shows the items in the user’s cart as well as the available items in the shop.

Pressing the Load button will load the available items in the stock to the table. The user will be able to see the Name, Number of items available, and the Price per piece of each item.



If the user wants to put an item into his/her cart, the user must first select the item in the table and then push the < button. The Name, Quantity, and Price of each item will appear on the table. The Total price of the items will also appear as seen from the picture.

The user can also choose to return items to the available stock table, or choose to empty his/her cart and return all the items.



After the user if done selecting his/her desired items, pressing the purchase button will require a confirmation and then the transaction will be done.

**REQUIREMENTS**

1. **USER REQUIREMENTS**

* Must be able to log in with an existing account
* Must be able to create a new account
* Must be able to select items the user wants to buy
* Must be able to add/remove items from the cart
* Should be able to search for specific items
* Could clear the cart if necessary

1. **FUNCTIONAL REQUIREMENTS**

* The program should be able to load the items being sold
* The program will show the user the total value of the items added to the cart
* The program will check the validity of the user inputs in the login page
* The program will display the selected item to the cart
* The program should be able to let the user search for the item name

1. **NON-FUNCTIONAL REQUIREMENTS**

* The password of the user must be masked on login
* The program will deduct the sold items from the stock
* The program will record the new user account on the database
* The program will create records for purchases

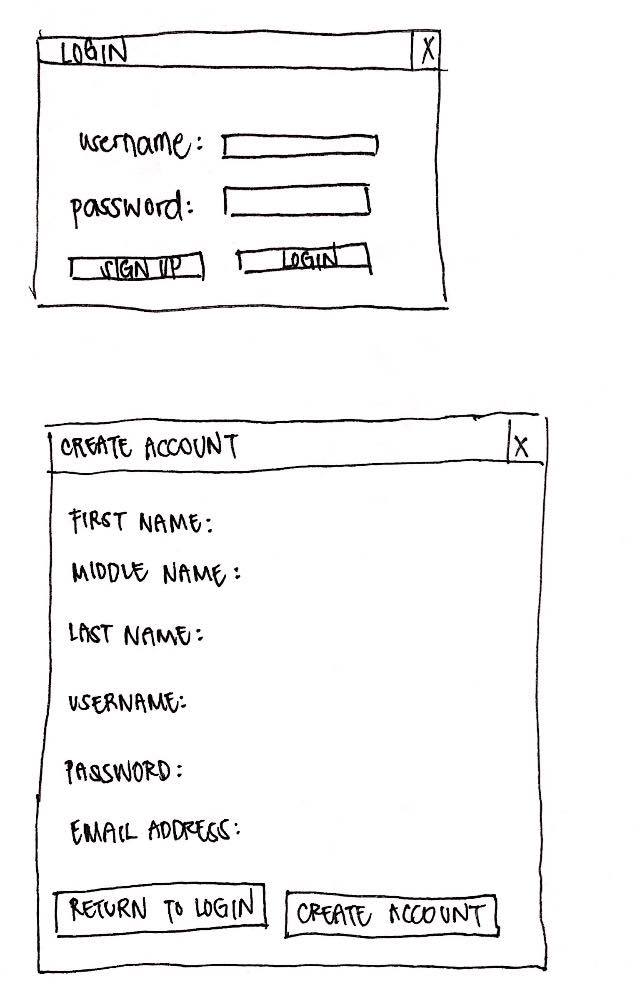
**HIGH-LEVEL DESIGN**

1. **USER INTERFACE SKETCHES**
2. **LOGIN**

A close up of text on a white background

Description generated with very high confidence

1. **CREATE ACCOUNT**

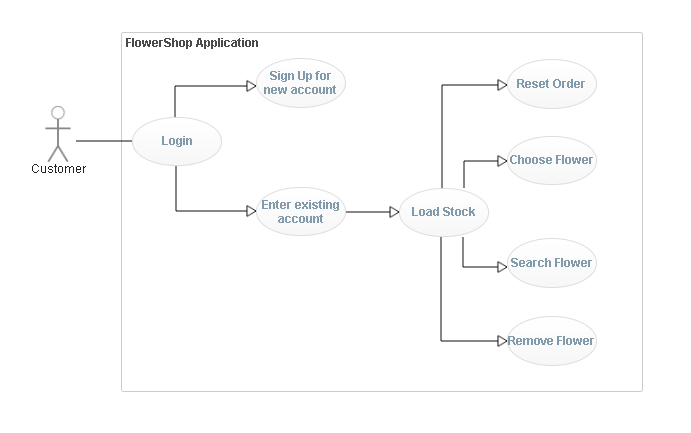


1. **MAIN WINDOW**

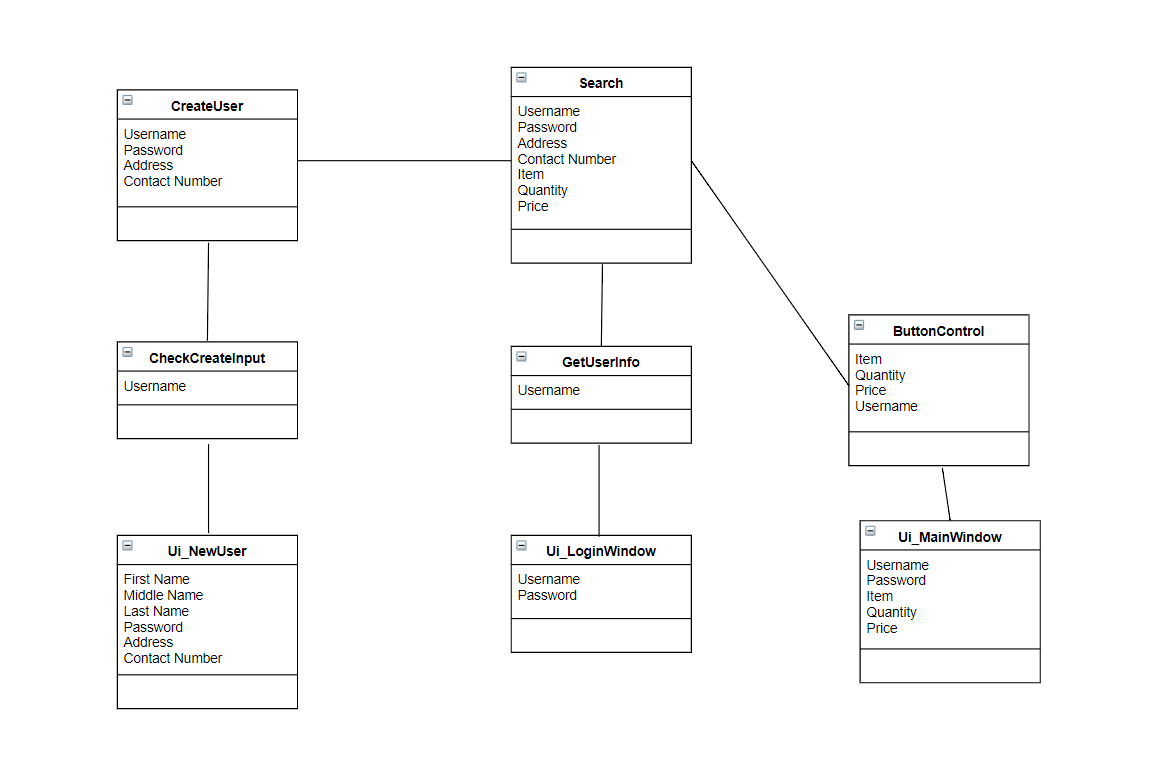
A close up of text on a white background

Description generated with high confidence

1. **USE CASE DIAGRAM**



1. **CONCEPTUAL CLASS DIAGRAM**



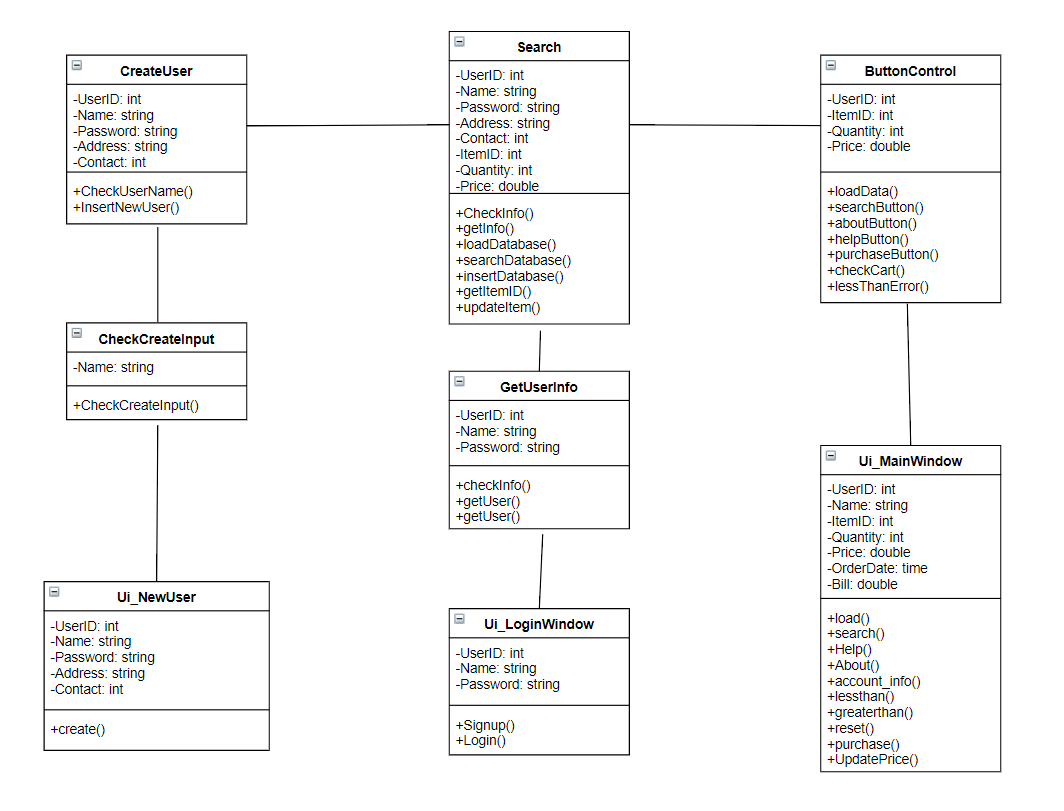
**LOW-LEVEL DESIGN**

1. **DATABASE DIAGRAM (ERD)**

![A close up of a device

Description generated with high confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4RDuRXhpZgAATU0AKgAAAAgABAE7AAIAAAAMAAAISodpAAQAAAABAAAIVpydAAEAAAAYAAAQzuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAE1hcmsgQXF1aW5vAAAFkAMAAgAAABQAABCkkAQAAgAAABQAABC4kpEAAgAAAAM0NwAAkpIAAgAAAAM0NwAA6hwABwAACAwAAAiYAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAxODoxMDowMyAxNjozMTo0MAAyMDE4OjEwOjAzIDE2OjMxOjQwAAAATQBhAHIAawAgAEEAcQB1AGkAbgBvAAAA/+ELHmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTgtMTAtMDNUMTY6MzE6NDAuNDY3PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPk1hcmsgQXF1aW5vPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAi4CuwMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAorgfH76hceL/DOlWWs3+lwXkd48zWMiozlBEV5ZSO57d6oSeHLmJtsvj/xIjejXkAP/AKKoA9NorzOLw1dzgmHx74lkA6lLyE/+0qk/4RPUP+h48Uf+BUP/AMaoA9Iorzf/AIRPUP8AoePFH/gVD/8AGqP+ET1D/oePFH/gVD/8aoA9Iorzf/hE9Q/6HjxR/wCBUP8A8ao/4RPUP+h48Uf+BUP/AMaoA9Iorzf/AIRPUP8AoePFH/gVD/8AGqP+ET1D/oePFH/gVD/8aoA9Iorzf/hE9Q/6HjxR/wCBUP8A8ao/4RPUP+h48Uf+BUP/AMaoA9Iorzf/AIRPUP8AoePFH/gVD/8AGqP+ET1D/oePFH/gVD/8aoA9Iorzf/hE9Q/6HjxR/wCBUP8A8ao/4RPUP+h48Uf+BUP/AMaoA9Iorzf/AIRPUP8AoePFH/gVD/8AGqP+ET1D/oePFH/gVD/8aoA9Iorzf/hE9Q/6HjxR/wCBUP8A8ao/4RPUP+h48Uf+BUP/AMaoA9Iorzf/AIRPUP8AoePFH/gVD/8AGqP+ET1D/oePFH/gVD/8aoA9Iorzf/hE9Q/6HjxR/wCBUP8A8arc+F9/ean8ONMu9Supbu5czB5pTlnxM6jJ+gFAHWUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHB+Mv+SmeDv+uGof+gw025vL+1uLw6ZZ6XdyfaAHGoXDRbV8pMbSsb55z6U7xl/yUzwd/wBcNQ/9Bhq/PptndSGSe3V3PUmmmJq5T0q5nur6V7y3tbefyIxJFaSF4gdz9GKqTxjsK1qgtrO3swwtoljDdcd6noerBKyCiiikMKKKKACsb+15/wDhOP7G2R/Z/wCzvtW7B37/ADNuM5xjHtWzXL6sX0jxrb65Na3dxZSWDWcj2lu87ROJA6kxoC5B5GQDgjnGaX2l8/yf62E78rt5fmr/AIEz+JhaeKtXstReCDT9PsLe684g7t0jyqQeefuLgAZye+RVm28WaNc295MbqS1WxQSXK3tvJavEhzhysqq204OGxgkEdq5yWwi1ubxFrWoaVqMum3dtbWsNusLQ3LrC7sZVRirqQz5AIDHZkA5XOPe2Ov6zomoadpt9rWpadbraTQy6nYi2ujJFMHeNPMjQSEoucuhG7ALHJw+th+f9dP6/HY7m28XaNc2l5cC4mhWyhNxOlzaSwSJEATv8t1DFeDyAQSCOtRjxtobW0U8ct3KszEQrFp9xI8wABLoioWePBH7xQV5HPIrj7zT7rV7bWL23k8Q6m0ehXdtHJqdklqxeQA+UkQt43kJ2A56DgDJJxueIbK0WPRLm4XXbG6tYSkN/o9u0zQ5C7onjVHyrbQfmjKgp1Bxk/r8/8vxHpZfP8l/mdVp+oWuq6fDe2Eomt5l3I+CPqCDyCDwQcEEEHmrNYnhGfVbjw1BJr29ros4EksPkySRhyI3eP+BmXBK8YJ6DoNum9xBRRRSAKKKKACiiigArO+EX/JLNJ/3rj/0fJWjWd8Iv+SWaT/vXH/o+SgDtKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDzn4h6pYaR8QPB91q19bWNuIb9TNcyrGgJWHAyxAp3/Cf+Dv8AobND/wDBjD/8VW1r6K/xE8KB1DDZe8EZ/wCWaV032eH/AJ5R/wDfIoA8/wD+E/8AB3/Q2aH/AODGH/4qj/hP/B3/AENmh/8Agxh/+Kr0D7PD/wA8o/8AvkUfZ4f+eUf/AHyKAPP/APhP/B3/AENmh/8Agxh/+Ko/4T/wd/0Nmh/+DGH/AOKrT+JsEQ8EEiJAf7T03oo/5/oK637PD/zyj/75FAHn/wDwn/g7/obND/8ABjD/APFUf8J/4O/6GzQ//BjD/wDFV6B9nh/55R/98ij7PD/zyj/75FAHn/8Awn/g7/obND/8GMP/AMVR/wAJ/wCDv+hs0P8A8GMP/wAVXoH2eH/nlH/3yKPs8P8Azyj/AO+RQB5//wAJ/wCDv+hs0P8A8GMP/wAVR/wn/g7/AKGzQ/8AwYw//FV6B9nh/wCeUf8A3yKPs8P/ADyj/wC+RQB5/wD8J/4O/wChs0P/AMGMP/xVH/Cf+Dv+hs0P/wAGMP8A8VXoH2eH/nlH/wB8ij7PD/zyj/75FAHn/wDwn/g7/obND/8ABjD/APFUf8J/4O/6GzQ//BjD/wDFV6B9nh/55R/98ij7PD/zyj/75FAHn/8Awn/g7/obND/8GMP/AMVR/wAJ/wCDv+hs0P8A8GMP/wAVXoH2eH/nlH/3yKPs8P8Azyj/AO+RQB5//wAJ/wCDv+hs0P8A8GMP/wAVR/wn/g7/AKGzQ/8AwYw//FV6B9nh/wCeUf8A3yKPs8P/ADyj/wC+RQB5/wD8J/4O/wChs0P/AMGMP/xVH/Cf+Dv+hs0P/wAGMP8A8VXoH2eH/nlH/wB8ij7PD/zyj/75FAHn/wDwn/g7/obND/8ABjD/APFVf+D7rJ8KdIeNgysZyrKcgjz5Oa7H7PD/AM8o/wDvkVzXw0AHw50kAY+R/wD0Y1AHU0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRVDWdZtND083d8zEM4jiijG6SeRuFjRf4mJ6D8TgAmgC/TJZY4U3zSLGv8AedgBXLro3iHxBmXxDqcukWzfc03SZdrY/wCmlxjcW9o9gHqetSD4beDCwe48NadeOBjfewi5Y/VpNxJ96AN7+0rH/n8t/wDv6v8AjWU/iyzTxPJpG6Aqlh9s8/7QME7yuzH4Zzn8Kh/4Vx4I/wChN8P/APgrg/8AiayJPhT4ZbxVJfL4Y8PjT20/yFt/7Pi4m3k79uzH3SBnrQBHa6/D4l1/wDq0YSJ76wnumt1lDmHzII22k4GcZxnA6dK2fHmqPpHh+CddX/saOS+t4Zr7MQ8mNnAY5lVkHHcg1y2heDNP8H+IPAdrDpunW+pR6dNBfXVnAqG4kSCMMxfaGbLZOW55zXfazpX9rRWied5P2a8hus7N27y2DbeoxnHWjqvVfnqJ35Xbs/yPP7jxfNZWurPoHjFfElnbaY9zNft9lkFlKHUIu+FFT5lMh2sCRszwDXQ6B4iuPGfiC4utHvlg0HTZPKwiqZL+QqG3HcCUh2sCpGC/3gQuN294i0dfEHh290mSQRpeRGJmZN4APquRn6VVuvDjf8JTaa5pV0tjOsf2e9j8nel5AMlVIyNrKSSrc4ywIIPD7f1/XYWuv9f1/XqUPiSgfwUQf+gjp549r2E1p+Lr640vwTrd/YyeVc2unzzQybQ211jYqcHIPIHWs34kMF8FsW4H9o6ePzvYRW5rUEd3od5az2kt5FcQtDJBCVDurjaQNxA6E9TUSTcWluawaUk3scFqXi3W7f4X+CNXhvdt9qt1pkd5L5SHzVm2+YMYwM5PQDHbFS6B8VbnxB8QLzw/beHo0s7S4mgku21e3+0J5Z2s7WpIkCbuM88EHviud0v4e+IoLnSLXW9U8Qar4f0W4Sew0s2dlEymM/uhJMs2ZAo7EDPtVm98Da1q/wAQrTxDrcmq31pY3IuLa1/szT4p025Kxm6SUOYwWPy45HXnmtW05NrZtv0Wlv10WnmZa8tuqS+e/wDwNS14c+Odj4g8XQ6WNMhgsb15I7G7GqQyzSugJAktl+eIEK2C3+z61Utvin4k8RWHhvUrXQX0HTdR1q2tBcSXUc5u0YyiRAuzKgFB83BPaq3hn4Xy+GvFC38On3E+mwM72tm+i6Z58bNnG668wyNtycdD05wOdax8HXll4P8AC+hfYtVf/hH9SS+8/wAi2Hn7Wc7Nv2j5fv8AXJ6dKmFvdb8vz1v/AMDQqX2kuzt9zt+hk2l38Sdbh8T67pPjeGCDR9Uu4YNJn0mF45UhOQjSjDgEcZ6+9ep+FNc/4SXwhpWt+UITqFpHcGIHIQsoJGe+DXlL+AfGSXGsW1jr+tWOiaxezXN1YQaXY79sp+dVma4LKdvG4DjrivXdCtIbDQbKztLOSxgtoVhit5SpaNVG0A7SR0HrSj8Ovl99tfvCXxu3d/dfT8C/RRRQAUUUUAFFFFABRRRQAUUUUAFct8Nf+SdaT/uP/wCjGrqa5b4a/wDJOtJ/3H/9GNQB1NFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBX1DULTStOuL/AFG4S2tLaMyTTSHCooGSTXLeGLC71/Vv+Ev16N4i6FNIsJBj7HAf42H/AD1kHJ9FwvrUHzfEHxF/C/hbSLj6jUrtD+sUZH0Zx6Lz3FABRRRQAUUUUActrv8AyUXwp/uXv/otK6muI8aaNZ67428K2Wo/aPJIvGP2e6kgbIjT+ONlb8M1Z/4Vj4a9NY/8H19/8eoA66iuR/4Vj4a9NY/8H19/8eo/4Vj4a9NY/wDB9ff/AB6gBfid/wAiOf8AsJ6b/wCl0FdbXlfxC+HugWHhAz241Tf/AGhYJ+81q8kGGvIVPDSkZwxweoOCMEA103/CsfDXprH/AIPr7/49QB11Fcj/AMKx8Nemsf8Ag+vv/j1H/CsfDXprH/g+vv8A49QB11Fct/wrnw7/AHdU/wDB1ef/AB2j/hXPh3+7qn/g6vP/AI7QB1NFct/wrnw7/d1T/wAHV5/8do/4Vz4d/u6p/wCDq8/+O0AdTRXLf8K58O/3dU/8HV5/8do/4Vz4d/u6p/4Orz/47QB1NFct/wAK58O/3dU/8HV5/wDHaRvhv4cdSCuq49tbvR/KWgDqqK5H/hWPhr01j/wfX3/x6j/hWPhr01j/AMH19/8AHqAOuorkf+FY+GvTWP8AwfX3/wAeo/4Vj4a9NY/8H19/8eoA66iuR/4Vj4a9NY/8H19/8eo/4Vj4a9NY/wDB9ff/AB6gDrq5b4a/8k60n/cf/wBGNUX/AArHw16ax/4Pr7/49T/hiixfDXR40ztWJlGSScB27nk0AdXRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVx/ie+u9e1b/AIQ/Q5HiLoH1a+jbBs7duiKf+esnIHouW9K0PFniKXRra3stKgF3repOYbC27bsfNI/pGg+Zj9AOSKseGPDsPhrSfsySvdXUzme8vJfv3Mzfedv5AdAAAOlAGhp+n2mladb2GnW6W1rbRiOGGMYVFAwAKsUUUAFFFFABRRRQBy2u/wDJRfCn+5e/+i0rqa5bXf8AkovhT/cvf/RaV1NABRRRQBynxJQP4KIP/QR088e17Ca6DVr7+y9Fvb/y/N+y27zeXu27tqk4zzjOK5/4kuE8FEn/AKCOnjj3vYRXQatY/wBp6Le2G/Z9qt5Id393cpGf1qJ83K+XcqNuZX2M/WfEX9k6JY6h9l877XdWlv5fmbdnnypHuzg52784747Utp4v0W91U6fb3MvnZcLI9rKkMpT74SZlEbkYOQrEja3ocYFxb67r9jpWi3vh+fTvsd3a3FzfSXMLwMIJFfEW1zI25kAG9E4JJwRg5c/hXW7vU7yw0i21XQLG9F1Hf+bqEVzYypIsoV4ELNJG5eRWwqxKMNnOBnWVrO3d/krfjfzJituby/W51kHjvw9cRXMqXsqxW1s92ZZbSaNJYUGWkiZkAmQDB3R7hgr6jMcnxA8PRCPM16zyo8kUcemXLvKikBpEQRlmT5l+cArg5ziuP/4RG+u/Dd7anw3q8OpR6Vc28c1/4he6haZ4TGPIRp3GGyfmdY8Aj3A7C30m9Txfpt60OLeDRntZH3r8speIhcZz0VuenFS97L+t/wDJfeL7Kf8AX2f839w+98eeHbG3hna8luYprYXatY2c11thPIkfykbYpwcFsA4PoaL3x74b09ytxqJIWKKd5IbeWVIopM7JXdFKohwfnYhRjkiuIg8NeK4vDemaLeWOqSW0WkRwx21hqcVrFHdAOGNxIrCUrgpgRl1POVJAq5a+E9c/4Q3XbOTTzHc3nhu1sYYmmjJadIZFZMhiBgsBnOOetO17/wBd/wDgFNWmo/e/uOvg8baBcQXsy3kiJZKjy+dayxllckI0YZQZQ5BClNwY8DJIq9pGu2GuRSvp7zAwvsliubaS3ljOMjdHIqsMg5BIwe1cj4y8KalrGoPcWdvK8cFvaOkcN2bd52inZ2iV1YMjbTw2QM45HJGt4P0hLK4vr3+xtT02S4WOInVdWe9uJFTcRnMsqooLnGHycnIGBlR1Tb/rz+ZPb+vl8v8AgHUUUUUDCiiigAooooAKKKKACuW+Gv8AyTrSf9x//RjV1Nct8Nf+SdaT/uP/AOjGoA6miiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACs/XdcsvDmjT6nqchSCEDhV3PIxOFRV/iZiQAO5NXpJEijaSVlREBZmY4CgdSTXGaMknjfXovEl1/yArJj/Y0BX/j4fobtge3URjsMt3GAC74T0W9FzceI/EQ/wCJzqKBfJzlbGDOVgX+bHu2ewFdRRRQAUUUUAFFFFABRRRQBy2u/wDJRfCn+5e/+i0rqa5bXf8AkovhT/cvf/RaV1NABRRRQByXxN/5Edj6alpxP4X0B/yK62uS+J3/ACI5/wCwnpv/AKXQV1tABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFct8Nf+SdaT/uP/6Mauprlvhr/wAk60n/AHH/APRjUAdTRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFcz4r1q9W4t/Dvh3nWdRUnzsZWxgBw1w306KO7Y7A0AUdZeTxtr0vhu1/wCQHZMP7ZnDf69+otFI9eDIewwv8Rx2UcaRRrHEioiAKqqMBQOgAqjoWiWXh3RoNM0yMpBCDyx3PIxOWdm/iZiSSe5NaFABRRRQAUUUUAFFFFABRRRQBy2u/wDJRfCn+5e/+i0rqa5bXf8AkovhT/cvf/RaV1NABRRRQBy3xG/5Exv+whYf+lkNaviaaS38JavNBI0UsdlMyOjEMpCEggjoax/iYxXwSSvB/tLTh+d9AK6TULKPUtMurGdmWK6heFyhAYBgQcZ781FRNwaRUGlJNnnekPA95ozeFz4sa8aWNrttUOo/ZjBtzJuN38nPG3Z82cY+XdTrL4g6tL4htrWCfStbiu0uAkemWlwsccqRmRY1vWJimPylThVIJzgYIr0WO2jjs1teWjWMR8nkjGK5a0+HttbS6aZNc1q4j0tWjsonnjVYY2jMez5EUnCkYdiXBUYbls6S1bt5/wBf15/OFolfy/r+vIx9O8daxdaLqbwz6fqWqWsUbNp8WnXNrdWhZlDmS1kZpJFVW3AqV342gZINWtH8cz3IspZ9S0nUbSS+ks7mezglgkgYRM6iWGRi0LBkZSrEk5U/LnFasfghf3st34g1m8vTb/Z7e+kkiSa1TcrEIY41ByyITvDbtoByMgsuvAFjqOi6hYatqOpX02otG1xfPJHHORGQUVTGiqoHPRf4jnrSfl/Wv+X/AA7GvP8ArT/P+kY0fxG1FPDEOoXenQreItxPdWqlgFjUDyl3c7WYzW+ScjBfAOBUus+LPEnhaG5h1ZNLv7trBr22ktoZIIwUkjR4nVncn/WqQ4PrlRgZ6F/B+kS3msTzxSSrrECQXULSEJtVSuVxgqSMAkHPyr6VRfwBaXVtcpq2ranqk81uLVLq6eLzIItwYomyNV5ZVJYgscDJOBh6X+X+f62/rQWtv6/ruv60ztS1PxfOur6ZYXmk/adLshLcT/ZJY/tDyByqR4lJhKqo+cmTJbO0Ywep8MyyXHhPSJp5GllksYWd3YlmJjBJJPU1S1vwhb6zdz3C6lqGnG7t/st4tlIii6jG7AYsjFSNzYZCrc9eBjY0+yj03TLWxgLNFawpChc5YhVAGcd+KUdnfy/X/NCd+ZP1/T/JliiiigoKKKKACiiigAooooAK5b4a/wDJOtJ/3H/9GNXU1y3w1/5J1pP+4/8A6MagDqaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5r4i6te6F8Ota1PSp/s95a2xeGXYrbWyOcMCD+IrCOmeKF+94/1IfWxsv/AIzWj8W/+SR+I/8Arzb+YpmpPsuYCLWK8YJIy28zBVkYAYBJBA+uKa3E9igNO8Tk4HxA1En2sbL/AOM07+yvFX/Q+6n/AOAFl/8AGaZFfz3FxAtz4fstKYTpsktrhJWkHOQdqLjtXQ0NAmYP9leKv+h91P8A8ALL/wCM0f2V4q/6H3U//ACy/wDjNb1FIZg/2V4q/wCh91P/AMALL/4zR/ZXir/ofdT/APACy/8AjNb1FAGD/ZXir/ofdT/8ALL/AOM0f2V4q/6H3U//AAAsv/jNZ9l4l8R6rDeXGlaBpstvbXU9svnas8ckhicoTtFuQMlePm/GtzRdesdd020u7OQKbq3W5WB2HmKh9VBPfIz0yOtC1VxtWZT/ALK8Vf8AQ+6n/wCAFl/8Zo/srxV/0Pup/wDgBZf/ABmrlx4i0SztVurvWLCC3fcVmlukVG2sFbBJxwxAPoSBTpNe0eHVItMm1Wxjv5gGitGuUEsgPQhM5PQ9B2oEUf7K8Vf9D7qf/gBZf/GaP7K8Vf8AQ+6n/wCAFl/8ZpumeMtKvbl7S7vLOxvjeTW0NpLdJ5s3lytGGVTgnJU8AH0yav6h4h0XSLmK31XV7CxnmGYorm5SNn5xwGIJ544oWqT7g9G0+hS/srxV/wBD7qf/AIAWX/xmj+yvFX/Q+6n/AOAFl/8AGavX+vaPpV1DbapqtjZT3H+piuLlI2k5x8oYgnnjil1HXNJ0eSFNW1SysXuCRCtzcJGZDxwu4jPUdPWgCh/ZXir/AKH3U/8AwAsv/jNH9leKv+h91P8A8ALL/wCM1vUUAYP9leKv+h91P/wAsv8A4zR/ZXir/ofdT/8AACy/+M1vUUAYP9leKv8AofdT/wDACy/+M0f2V4q/6H3U/wDwAsv/AIzW9RQBg/2V4q/6H3U//ACy/wDjNW/BV7rP/CW69pOsa1Pq0Vpa2c0LzwQxshkM4b/VouR+7Xrnp71p1leFP+Sm+Kf+wfp3/oV1QB3FFFFABRRRQAUUUUAFFFFABRRVe/v7XS9PnvtRnS3tbeMyTTSHCooGSTQBQ8S+IYfDek/aXie5uZnEFpaRffuZm+6i/wAyegAJPAqv4V8PS6Pb3F7qkwuta1JxNfXA6bsfLGnpGg4UfUnkmqHhqyutd1b/AIS7WkeLehTSbGRcG0gbq7DtLIME/wB1cL6566gAooooAKKKKACiiigAooooAKKKKAOW13/kovhT/cvf/RaV1Nctrv8AyUXwp/uXv/otK6mgAooooA5L4m/8iOc/9BLTsfX7dBXW1yXxO/5Ec/8AYT03/wBLoK6qWaO3heaeRYoo1LO7sAqgckknoKAH0VWbUbJLOG7a8t1tpygimMqhJC5AQK2cHcSAMdcjFWaACiiigAooooAKKKKACiiigAooooAKKKKACiiigArlvhr/AMk60n/cf/0Y1dTXLfDX/knWk/7j/wDoxqAOpooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDjvi3/ySPxH/ANebfzFT3VnBeKFuE3BemGKkfiKg+Lf/ACSPxH/15t/MVF/wkuhf9BrTv/ApP8aAJYNHsbeZZYoSHU5BaRmx+Zq9WZ/wkuhf9BrTv/ApP8aP+El0L/oNad/4FJ/jTu2Bp0Vmf8JLoX/Qa07/AMCk/wAaP+El0L/oNad/4Fp/jSA06KzP+El0L/oNad/4FJ/jR/wkuhf9BrTv/ApP8aAOQ8JaFd6jpmqNH4j1Wxgk1a+U21qtuFA89wcM0LOM+u7jtirbT6P4N8ZWlvcSxaZp39jLbWjTvhGMcn+rDMfmfDA4ySeetdJ/wkuhf9BrTv8AwKT/ABo/4SXQv+g1p3/gUn+NJK1vL/K36jk+a/n/AJ3OA8HRxXN94NmlhO5IdWkjEqFWjYzoOh5BwxHrya09B1PQtNttS0nxPJapqk+rTzPZXCB5rrdMTC6R43SDYI8FQcbMcbTjrP8AhJdC/wCg1p3/AIFJ/jR/wkuhf9BrTv8AwKT/ABprT+vO4nflt/XX/M4G8toI/hp4mmjhRZW12eVnCjcXW8AVs+oCjB7YFV/Fmop9q8W2kmrW+ku6kGwitxLfamPs4ClfNLAxkkjCR8bWO4HdXo3/AAkuhf8AQa07/wACk/xo/wCEl0L/AKDWnf8AgUn+NS43jy+VvwS/Qrm99z87/i3+p5vcX1np2lx6jaeJNLhnvtEgFzpOtW3mRahiIbPLOVYkjcpC+YMt93PWe4vrqx8Uapc6xqGj+H01GxtxBBq9k02+IR/PCjCaMEq7NlACTuB7ivQf+El0L/oNad/4FJ/jR/wkuhf9BrTv/ApP8auXvNvv/wAEzguVJdv+B/kQ+ELWSx8G6VbTmcvFaov+kReXIBjgMu5tpAwMbjjHWtmsz/hJdC/6DWnf+BSf40f8JLoX/Qa07/wKT/Ghu7uNaI06KzP+El0L/oNad/4FJ/jR/wAJLoX/AEGtO/8AApP8aQzTorM/4SXQv+g1p3/gUn+NH/CS6F/0GtO/8Ck/xoA06yvCn/JTfFP/AGD9O/8AQrqnf8JLoX/Qa07/AMCk/wAar+Cry2vviP4omsriK4i+wacN8Lh1zuuuMigDvqKKKACiiigAooooAKKKKACuLOfHviDAKv4Y0qfnuNRukPT3ijI+jOPRebHiW+u9a1QeE9DkaJpED6rextg2dueiKf8AnrJggegy3pnpNPsLXStOt7DTrdLa1toxHDDGMKigYAFAFiiiigAooooAKKKKACiiigAooooAKKKKAOW13/kovhT/AHL3/wBFpXU1y2u/8lF8Kf7l7/6LSupoAKKKKAOW+I3/ACJjf9hCw/8ASyGtLxZ/yJmtf9eE/wD6Lasv4lPs8FE4z/xMtOH53sArqJYo7iF4Z41likUq6OoKsDwQQeoqJx5oOPcqL5ZJnkGsBvCmm6J4ecN/ZWoarps+kvyRC4uYmltiewHLp/s7lHCCp7nUbz/hJ9H1XT59TitrrXfsn2i/1sj7WNzo8cdkmYvLAHDHZINu4gnLH1O4sLO6hiiurSCaOF0kiSSMMI3U5VlB6EHoR0qkPC+gLfzXy6Hpou55FlluBaR+ZI6sGVmbGSQwBBPQjNbc2q9b/LTT8LfnczcXy2Xa356/j/wx59pNld6hb+F7mfXdZ36peXdtegajKolhUTMqqAwEZBjT50w+ARu5qpaT+INW1dfD0Mkt3a2f2wwG48QXFhLN5d5JEMyxI8kpjQRjkgfOC244I9Yi02xhWBYbK3jW3ZnhCRKBEzZyV44J3Nkjrk+tVb/wzoOq2ottT0TTry3WVpxFcWkciCRiSz4II3Ekknqcms0rJL+v66/1cpfDZ76f18xnhZ72TwvYnU7u2vbry8SXFrN5scmCQCH2ruOMZO0ZOeB0rWpscaRRrHEioiAKqqMBQOgAp1U3d3EtFYKKKKQwooooAKKKKACiiigAooooAK5b4a/8k60n/cf/ANGNXU1y3w1/5J1pP+4//oxqAOpooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDlfieqv8LPEYdQw/s+XgjPatf/AIRrQv8AoC6d/wCAif4Vk/E3/klviP8A7B8v/oNdSTgGk3ZXAzP+Ea0L/oC6d/4CJ/hR/wAI1oX/AEBdO/8AARP8K5Gy8S+Kk8I6Z4n1G+0Oa1uxbs1hHYSwyHzWVQiSmdwWBYYGz5iMcZyLA8d3Vyf7J02G2uPENze3cNvByI4IIZ2j+0TDOdoCjgEF24XHJWrO9upTi1e/9b/5HTf8I1oX/QF07/wET/CuT+GPhjR1+H9mbjTLG4kae6cyPaoThrmQgcjsCB+Fd5AJVt4xcukkwUCR40KKzY5IUkkDPbJ+prmfht/yIFh/vz/+j3pEmt/wjWhf9AXTv/ARP8KP+Ea0L/oC6d/4CJ/hXFaj4r8da94l1bT/AIdWGhLZ6LKLa5utaeX9/PtDFI1j6BQQCW6k8dKih+NdjBouly6toGrf2rfT3Vm+m6dCLl47i3xvQcgtnIwQCOeSMZpJ3V/6/r/gDtrb+v6/yZ3X/CNaF/0BdO/8BE/wo/4RrQv+gLp3/gIn+FefeLPjZa6Xa6jBoWl6hNe2LwxSXU9kz2cMzOgaCSRG4kCueOmRjJ6Hdu/itodl4qOiyWWrPFHcpaXGrR2ZNjbztjETy54bLKOhALDnrikr7f1/VxbHSf8ACNaF/wBAXTv/AAET/Cj/AIRrQv8AoC6d/wCAif4VwevfGSKHT/E0fh7RdSlvdB8+OW6urT/Q1kiIGGdXGd2eAPm4yQBUV14t+I2seO9c0XwZB4YFvpEVq7nVFuBI5mi34BRsHkN1A7dala7epXK9b9P+B/meg/8ACNaF/wBAXTv/AAET/Cj/AIRrQv8AoC6d/wCAif4VR8DeKT4w8KQapLZtY3PmSQXNqzh/JljYo65HUZHB9K6GqIWpmf8ACNaF/wBAXTv/AAET/Cj/AIRrQv8AoC6d/wCAif4Vp0UhmZ/wjWhf9AXTv/ARP8KP+Ea0L/oC6d/4CJ/hWnRQBmf8I1oX/QF07/wET/Cj/hGtC/6Aunf+Aif4Vp0UAZn/AAjWhf8AQF07/wABE/wo/wCEa0L/AKAunf8AgIn+FadFAGZ/wjWhf9AXTv8AwET/AArC8PWVrYfEzxNFY20NtGbDTyUhjCDO655wK7CuW0n/AJKl4m/7B+n/APoVzQB1NFFFABRRRQAUUUUAFYXinxBLo9tBa6ZALvWNQcw2Nt2LY+aR/SNB8zH6AckVe1vWbLw/o8+palIUggGSFXczsThUVerMSQAB1JFZPhbRrwXVx4i8QA/2vqCBRBnK2MGcrAvv3Y929gKAL3hrw9D4c0s26yvc3U8hnvLuT79zM33nP8gOgAAHSteiigAooooAKKKKACiiigAooooAKKKKACiiigDltd/5KL4U/wBy9/8ARaV1Nctrv/JRfCn+5e/+i0rqaACiiigDkviaM+BzjtqenE/+B0FdbXJfE7/kRz/2E9N/9LoK62gAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuW+Gv/JOtJ/3H/wDRjV1Nct8Nf+SdaT/uP/6MagDqaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5b4m/wDJLfEf/YPl/wDQa6k8g1y3xN/5Jb4j/wCwfL/6DXU0mrqwHI+EPh7oXh3SdLaXQdHGs2cCrJfwWaeY0m3DMJNobnnnqc1ST4d/Z9Nv7jT2trLxA+rXGqW1/ECcu0jFFkOAWQxsEZemCcc4Nd3RVN3dxt33/rf/ADZDaNctZwtfxxRXJQGVIZC6K2OQrEAkZ7kD6Vznw2/5ECw/35//AEe9dTXJ/DGaOf4e2LxNuUTXKE4xytxIpH5g0hHGajb+NNB8S6tqHw6W3az1qUXNza61ptz+4n2hS8bRryGABIboRx1qHR/CF1o2ueEr7fqF++l3V/eapcTafKj3E1ymMogUgAHsSOB+Fey0UR93b+l/X6A9Xf8ArXQ8G1/wn4umTX9G0CWJfD2sagNSK3mm3P2lJDIjvGGVSoTK5BwT2461T1H4Vi5+JVxrA0S1vdMvb77bNNeWuordwEkFkRYnWMjIyC3Pzc5AAr6Fooj7trdP+B/kgl7yafX+v1PHH8N6i3gfx1owt7j7R4i1O5vLRjaT7ESTZtDnZkH5TnAP1qvc23xA0fx5rmt+DLXTGttXitUddVsrwuhhi2cCNMDkt3PbpXtdFH+SXyRTk3fzv+Lv+hy3w60RfD/g6KwMlzPcCaSW6uLm3MDTTSMXdgp6DLYH0rqaKKZIUUUUgCiiigAooooAKKKKACuW0n/kqXib/sH6f/6Fc11NctpP/JUvE3/YP0//ANCuaAOpooooAKKKKACkd1jjZ5GCooyzMcAD1pa47WHk8Y63J4ctCP7FtGH9szqf9c3BFopHqMFz2GF/iOABNIV/GmuReIbg50KzY/2RARxcPyDdsPTqI/bLdxjsqbHGkUaxxKqIgCqqjAUDoAKdQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUActrv/ACUXwp/uXv8A6LSuprltd/5KL4U/3L3/ANFpXU0AFFFFAHKfElA/gog/9BHTzx7XsJre1nVbfQ9DvdVvdwt7KB55No52qCTj34rC+JDBfBbFuB/aOnj872EVu6zpVvrmh3ulXoY297A8Em087WBBx780nsONrq5iwXHjNGtb25h0q4tp5EE2m28bpNbIx5YTtJtlKdSPLTIzjkAFyeOLF9QWM2N8mnyXf2KPVWRPs0k+7bsHz7xlwUDFApbgMcjMcEHjaQW1jczaTawQyL52pW8jyTXEanoIHjCxlwME+Y+3JwCcEUIfCetpBb6C0liuh2uorfJdrK5uXVZvPWExFNo+fAL7zlR90E8UrcyXT9NP+D5/gLW2u/6/18vxatWnxCtb2S3EOi6ssN40sVncSJCqXM0asWiXMm4MfLfBcKh2/e5GY9E8b6hf+EtI1Gbwxqt1e31qLiWCySEKg4+be8oTBzwu8vjqo5pLHwff22m+GbeSa2L6TqM11OVZsMjrOAF+Xk/vV646H8chfAmttpmjWGo2ej6paWFoLQ2l3fT/AGYFThbhoRHtmYrz5bgbSo2vzuEa8vn/AMP+tl/VytOW/wDX9df6sdBH4/sr21tJtF0vU9X+0WaX0kdpHGHtoXztLrI6nJ2thV3Mdh46ZxtT8eeIEh14aToQM9hqdpaWYvCEW5WXysg7XLKxDkglQAGXIyGAXw54S8S+D7C0XSU0m9nbTobK6S4uZIY42hL7JE2xMWBDnKHbjaMN1NX5vC2sS6lfs8li8N1qVjqHnB3Rt0IiWRfL2nAIiyp3n72DjGTorc3l/wAH/L+uhN7Rfp/X9f8ADux/wnFrNLot1byRR6XfW11PcyTgq9v5Kgsrc/KVO4MCDgqRVvSPF0Oq6hFaTaXqOmPcwmeza+jRRdRjGSu12KkBlOxwjYP3eGxg618OJNT8V3Fzb3iwaPqFndRXlr/Ek8sYjMsfGPmUDcDxlQepatGy0rxPLqNpeawmlBtKtZI7VLW4kxeSsoXfIWj/AHK4X7o8w/N1O35pXVv+t/8AgW/qxbSy/ra363/q/W0Uilig3gBscgHIB+tLQAUUUUAFFFFABRRRQAVy3w1/5J1pP+4//oxq6muW+Gv/ACTrSf8Acf8A9GNQB1NFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAct8Thn4WeJB66fL0/3ah/4Vvp//Qd8U/8AhQ3f/wAcqb4m/wDJLfEf/YPl/wDQa6knAo2A5D/hW+n/APQd8U/+FDd//HKP+Fb6f/0HfFP/AIUN3/8AHKsaN4yudchs7m18J61HY3iq8d5LJZhAjchyouC+Mc4259qkPjfTm0nR7u3gurifWkV7KwiVPtEikBmJBYKoUHLMWAHTOSAXZjs1cp/8K30//oO+Kf8Awobv/wCOVy3w4+HlrP4EtJbjWfEETvPdNstdbuYowPtEmPlVwAcYyccnJ5zmvWK5b4bf8iBYf78//o96QiH/AIVvp/8A0HfFP/hQ3f8A8cp8fw8sYs7dc8TnP97Xrpv5vW/YapBqNxfwwLIrWFx9nlLgAFtivkYPTDjrjnNZ1p400C4s9OnuNTtdPfU/+PS3vbiOKWY7tuFUt8xzgcZ6ijsBU/4QGz/6DfiT/wAHdx/8XR/wgNn/ANBvxJ/4O7j/AOLrZutf0ex1WDTL3VrG31C5AMFpLcoksoJIG1CctyCOB2qpq3jDRNH82Oe/gluogpazhmRp9pcJu2FgcAsMmgCj/wAIDZ/9BvxJ/wCDu4/+Lo/4QGz/AOg34k/8Hdx/8XUtz4vmTV7+w0/wzrGqf2e6xzz2rWqpuaNZMASTox+Vx/DWzpep2usaZDf2EhkgmXKllKsOcEMp5VgQQQeQQQaOlwejszB/4QGz/wCg34k/8Hdx/wDF0f8ACA2f/Qb8Sf8Ag7uP/i66migDlv8AhAbP/oN+JP8Awd3H/wAXR/wgNn/0G/En/g7uP/i66migDlW8AWTKQdb8S4IxxrlyP/Z6i/4Vvp//AEHfFP8A4UN3/wDHK6+igDkP+Fb6f/0HfFP/AIUN3/8AHKP+Fb6f/wBB3xT/AOFDd/8AxyuvooA5D/hW+n/9B3xT/wCFDd//AByj/hW+n/8AQd8U/wDhQ3f/AMcrr6KAOQ/4Vvp//Qd8U/8AhQ3f/wAcqr4O0aLQ/iH4ntbe6vrpTZae++/vJLmTJNzxvck446dK7muEutZj0Lxl401BzGz2ui2c6RO4XzCgum2/jjH40Ad3RXN23jG2n1/TtOJt1S80x75pfPHyMrRLs/HzCc/7Nbf9pWP/AD+W/wD39X/GgCzRWbd+I9EsFLX2s6fbKBkma6RAB+JrGk8Zvq5Fv4Jsn1SV+t9NG8VlEP7xkI/ef7sec+o60AWPFGs3iT2+geH8HWNQUkSkZWyh6NOw9uij+JsDoDjU0TRbLw/pEOm6ahWGIE7nO55GJyzsf4mYkknuTUWh6FFo0MztK13f3b+ZeXsgw879v91QOFUcKPxJ1KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5bXf8AkovhT/cvf/RaV1Nctrv/ACUXwp/uXv8A6LSupoAKKKKAOS+J3/Ijn/sJ6b/6XQV1tcl8TT/xQ7e+pacD+N9AP8mutoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArlvhr/AMk60n/cf/0Y1dTXLfDX/knWk/7j/wDoxqAOpooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDlvib/yS3xH/ANg+X/0GuoP3T9K5f4m/8kt8R/8AYPl/9BrqaTV1YDifh/4Sg0zwtol3OdXhv47NPMt7jU7sxoxTBUwNJsGM/d24HYDFc54U8P6t4L0bQ/Erpf6nNJZR2mrWkkBae3g48sQRqMjyyfmQDLgsxywGfWaKvm1v/XX/ADKbvdPqIDuUEZwRnkYrl/ht/wAiBYf78/8A6Peuprj/AIV3P2r4c2Um3ZtuLuPGc/dupVz+OM1JIQ6n/wAIxreuR6lYanIt/eC6tZLLT5rpZFMMaEExKwQhkI+fbwQQcZxxsujahoei2scVrqUOtXGlxQ3Fi2kjUtPv/mlIt5Sg+QqZCCxeNfmB+YA49ioo7fL8P663BaX/AK8zya90a9k1PXrDWrvxNA2rXKTRwaPp0M0EymNFUC4aBvLZCpXLyJjaCMZBrTvdIvP+Ee8XpFYXBmutbiljAhO6ZB9n+YcfMBtPPsfSvRqKNlbyt+X+Qmvzv+f+ZxNtrkPh/wAWeJl1Cw1hvtN5FNA9ro91cJIotol4eONl+8pHXtWz4PsLrT/D+NQh+z3FzdXF20BYMYfNmeQISCRkBgDgkZzgmt2infSxUm2FFFFIQUUUUAFFFFABRRRQAUUUUAFeb614R0zxb418X2t/pun3V3/Y1pFaT3lusht3f7SNykgleQpyPQelekVy2k/8lS8Tf9g/T/8A0K5oAyrf4S+Gote0+7l8OaC9rb6a9tPD/Z8Z82YtGRJjbg4COMnn5vc1tf8ACuPBH/Qm+H//AAVwf/E10lFAGPY+EPDWlsG0zw9pVmwOQbeyjjP6KK2KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOW13/AJKL4U/3L3/0WldTXLa7/wAlF8Kf7l7/AOi0rqaACiiigDk/iYpbwSQoyf7S04/lfQGusrlviN/yJjf9hCw/9LIa1vEbamvhy9GgxebqLxFLYblAV24DncQMLncfYcZPFJ7aDWrM7QPGCa7r2o6d9ie2S3Be1nZ8i7jWR4nYDA24eMjHPDIc/NgQL8RNHudf07TNPZ5xevKPtLxSRRbI0ZjJG7IFmXK4LIxA3A55GcZ/AGpaIdDl0fW9S1JNOH2FrSdLRAtrKoRyrJHGcqQj8sSdncmqjeG/EGp6RoXhufR3sLfS7Sayn1MTxbGzayQJJCiuX2kkHDBSMrweSH9l26fj/Wnq722F19fw/wA+vob3/CzNDfV4baBpWtGsbm+lvJIJYlWKLYd6BkHmowYkOhI+XjOa0JPHOhx20U3mX0gmLeVHFplzJJIq4zIsaxlmj+Zf3gGz5hzyK5HWdK8T69YpbHw0bM22gX9iXe6hYSTyRxqgjw+fLYqcFtp4GVWr2u+Gbwa7Y6o2naxqUP8AZqWctvpGrtZSwujFgxxNErqdzA5YkFVwDkkLrb+uv+S+8rTkT6/8N/m/uOhu/HXh2z8ky37SRywJcedb28s0UUTZ2ySOilYkOD8zlR8rHPBxNJ4u0aPWhpbXMv2gyiEyC1lMCyEZEZnC+WrnIwpYHkDGSK4zUvDGpWUVt/wjeg6rpuoraqsF/putJIqPukOy6Wc4lQGQtu2St8z4wQN1u80XXo/FiXGi2WpafdSXUL3d9DfxPp13HiPzTJbuxZXKoyjy4wc7cvgnFKzlb+v67/gRrb+v6/roehUUUUhhRRRQAUUUUAFFFFABRRRQAVy3w1/5J1pP+4//AKMauprlvhr/AMk60n/cf/0Y1AHU0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBy3xN/wCSW+I/+wfL/wCg11Nct8Tf+SW+I/8AsHy/+g11J4U0m7K4BRXmngfWbzXLXSLi88T+Kpbu4iWaaCTQ0is2bbuZfO+xgbOoBEnPGCSab4X8bX/jW30jSdI1RY7lLKO61fU1jQtu4zFCpUoz7iA5wQgOMbiMXyu9v6/rQbVr36Hptcn8MYY4Ph7YpEu1TNcuRnPLXEjE/mTXWVy3w2/5ECw/35//AEe9SI6mivPL3xN4jS91SPTjb3At/E1vYIs8giCQPDAxQERtnLO3J5GeDwBUn/C2NLOvLaLPpDWrXv2HA1ZPtvmeZ5e77Ltzs399+7b823FO17ef+Sf6oTdm12/za/Q7+iuFi8e6xcGwe28NRPDqV1PY2jNqOG86MycyL5Z2xkRMdwLMOmylg8W6nfa9pcDxrYF7fUkvLVHEy+dbvGqlZCoJHzMRwud3I4qHJJXLjFydv6/rQ7mivPbSfX9N8C2PiyfxPf6kVs4ry6sru3tVikRlDSBTHCjqQCSp3HkDINehda0lFxdmZxkpK6CiiipKCiiigAooooAKKKKACiiigArltJ/5Kl4m/wCwfp//AKFc11NeV+LdevNE8TeNjYaVql3JJoNuRdWBjAtCFusOxaRWHXIKgn5T7ZAPVKK8qsvHWsHxbo6t4W8TSRnQ5S9tutz57+ZBibBnwccjJIb5+nJx1H/Cb6j/ANCD4o/75s//AJIoA62iuQPj6aI/6X4M8U249RZRzf8AoqR61tI8W6Jrl09rp9+pvIxmSznRobiMf7UTgOPxFAGzRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHLa7/AMlF8Kf7l7/6LSuprk/FYv7XxR4f1Wx0i81WKzFys0dmY9670UKf3jqOoPen/wDCY6j/ANCN4k/8k/8A5IoA6miuW/4THUf+hG8Sf+Sf/wAkUf8ACY6j/wBCN4k/8k//AJIoAZ8TGK+CSV4P9pacPzvoBXWV5z4x1bWvEHhz7BZeCNfSX7ZaT5lNoF2xXMcrDic87UOPfHTrW7/wmOo/9CN4k/8AJP8A+SKAOporlv8AhMdR/wChG8Sf+Sf/AMkUf8JjqP8A0I3iT/yT/wDkigDqaK5b/hMdR/6EbxJ/5J//ACRR/wAJjqP/AEI3iT/yT/8AkigDqaK5b/hMdR/6EbxJ/wCSf/yRR/wmOo/9CN4k/wDJP/5IoA6miuW/4THUf+hG8Sf+Sf8A8kUf8JjqP/QjeJP/ACT/APkigDqaK5b/AITHUf8AoRvEn/kn/wDJFH/CY6j/ANCN4k/8k/8A5IoA6miuW/4THUf+hG8Sf+Sf/wAkUf8ACY6j/wBCN4k/8k//AJIoA6miuW/4THUf+hG8Sf8Akn/8kUf8JjqP/QjeJP8AyT/+SKAOporlv+Ex1H/oRvEn/kn/APJFH/CY6j/0I3iT/wAk/wD5IoA6muW+Gv8AyTrSf9x//RjUf8JjqP8A0I3iT/yT/wDkip/AVjd6b4F0y01G3e2uY0bzIXILIS7HBwSO/Y0AdFRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHLfE3/klviP/sHy/wDoNdSRkEVy3xN/5Jb4j/7B8v8A6DXU0PVWAzfD2kf2D4Z0/SPP8/7FbpB5uzbv2jGcZOPpmsJfASW3hjQ7HTb5LTU9CVfseorb5GcYkDRhhuSQZ3Lu7g5yAa6+indt3HdiDO0biCcckDFcv8Nv+RAsP9+f/wBHvXU1xfwjdpPhpZM7s7fa70ZY5xi7mGPoOg9hSEWbnwZOw1WSz1KOK5vdXh1SF5bUyJC0aRJsZQ6lwfKzkFfve3Mmn+GtX0q8aDT9dhh0Vrtrr7L9h3XC723vGJmcrsLlusZYKxAYEBh09FNNr+vJL8khNJu/9df82czY+D/scGjR/bt/9l6hcXufJx5vmiYbfvcY87rznb0GeGw+DPJ1uLUPt+fLN+fL8nr9pdG67v4dmPfPauooqWk9/T7ylJq1un/B/wA2cbY+D9c/sWy0PWdesbnSLaKKJ4rTTHt5Z1jAwrSNO4wdo3YUEjIBFdlRRVNtu7JUUtgooopDCiiigAooooAKKKKACiiigArjoLKDUviB4vsbtS9vdaVYwyqCRlW+1AjI6cGuxrltJ/5Kl4m/7B+n/wDoVzQBtRaLYw6la30cbC4tbRrOJt54iYoSMd+Y159qv0UUAFZWt+GtJ8RRKuq2aSyR8w3CkpNAf70ci4ZDz1BFatFAHI2Wp6l4V1CDSfE9wb3T7iQRWGsuAG3nhYbgDgOeiuOHPBwxG7rqr39haarp89hqNvHc2twhjlhkXKup6giua0vULvwtqcHh/wAQXElzZ3DeXpOqzNlpD2tpm/56gfdc/wCsA5+cHcAdbRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAct8Tf+SW+I/8AsHy/+g11Nct8Tf8AklviP/sHy/8AoNdTQAUUUUAFct8Nv+RAsP8Afn/9HvXU1y3w2/5ECw/35/8A0e9AF698aeFtO1F7DUPEuj2l7GwV7ae/iSRSQCAVLZBII/OtGTU7CGdoZr22jlVkVo2lUMC5wgxnqxBA9ccVwb2PiG8l8apper6RZ2b3jq6XumvM2TaxZJcTqoGPVDj36VjRajLfLYDbNb2obw/JDZySFhBudsgA9+Bk9TgZpxV7f9u/+Tf1/wAMRKVk3/i/A9at7mC8hE1pPHPESVDxOGXIJBGR6EEH3FS1zHw8/wCRNi/6+7v/ANKZK6ekUndBRRRQMKKKKACiiigAooooAKKKKACiiigArltJ/wCSpeJv+wfp/wD6Fc11NctpP/JUvE3/AGD9P/8AQrmgDqaKKKACiiigAqpqml2WtaXPp2qW63FpcLtkjbPPcEEcgggEEYIIBBBFW6KAOV0bU73QtVi8N+JLhrgy5Gl6pJgG8UDPlSY4E6gc9nA3DkMq9VVHWdHste0qXT9SjLwyYIKsVeNgcq6MOVZSAQw5BANY2haze2Opr4b8USB9QClrG+2hU1KJepwOFmUffQdfvr8pIQA6eiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDlvib/yS3xH/wBg+X/0Gupqpq2l2mt6RdaZqUZltLuMxTIHKllPBGRgj6isH/hXukf8/wB4h/8ACivv/j1AHU0Vy3/CvdI/5/vEP/hRX3/x6j/hXukf8/3iH/wor7/49QB1Ncn8MZo5/h7YvE25RNcoTjHK3EikfmDT/wDhXukf8/3iH/wor7/49UNp8MPD2n2wt7CXW7WBWZhFDr16igsxZjgS4ySST6kk0Aal74L8LajqL3+oeGtHu72Rgz3M9hE8jEAAEsVySAB+VaMmmWE07TTWVtJKzIzSNEpYlDlDnHVSSR6Z4rA/4V7pH/P94h/8KK+/+PUf8K90j/n+8Q/+FFff/HqAOkt7aCzhENpBHBECWCRIFXJJJOB6kkn3NS1y3/CvdI/5/vEP/hRX3/x6j/hXukf8/wB4h/8ACivv/j1AHU0Vy3/CvdI/5/vEP/hRX3/x6j/hXukf8/3iH/wor7/49QB1NFct/wAK90j/AJ/vEP8A4UV9/wDHqP8AhXukf8/3iH/wor7/AOPUAdTRXLf8K90j/n+8Q/8AhRX3/wAeo/4V7pH/AD/eIf8Awor7/wCPUAdTRXLf8K90j/n+8Q/+FFff/HqP+Fe6R/z/AHiH/wAKK+/+PUAdTRXLf8K90j/n+8Q/+FFff/HqP+Fe6R/z/eIf/Civv/j1AHU0Vy3/AAr3SP8An+8Q/wDhRX3/AMeo/wCFe6R/z/eIf/Civv8A49QB1NctpP8AyVLxN/2D9P8A/Qrmj/hXukf8/wB4h/8ACivv/j1aGh+FtM8PXF1caf8AbHnu1RZpby+muXYJu2jdKzEAb24HrQBs0UUUAFFFFABRRRQAVna5odn4h0trK/DqAwkhmhbbLbyLyskbfwsp5B/A5BIrRooA5vQNcvI9Rbw74nKLq8SGSC5RdseowjA81B2YZAeP+EkEZVlJ6SsvxBoNv4g09YJnkt54XE1rdwkCW1lH3ZEPryQQeCCQQQSKp+HdfuLm6m0TX0jt9ds03yKgIju4s4FxDnqhPBXkoxweCrMAdBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVkeIvD0WvWsRSd7LULR/Nsb+IAvbSYxnB+8pHDIeGBIPqNeigDB8O+IZdRln0vWYEstdsVBurZSSkiHIWeInlomwcHqCCp5BrerF8R+H/AO2ooLmyuPsOr2LF7G9C7jExxuRh/FG2AGTvgEYZVIPDniD+2Y57a9t/sOr2JCX1kW3eWxztdT/FG2CVfvgg4ZWAANqiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArB8ReHpdRlg1TRp0stdsVItblgSkinBaCUDlomwMjqCAwwQK3qKAMjw74hi161mDwPZajZv5V9YSkF7aTGcZH3lI5VxwwII9Br1z/iLQLi6uodb0B47bXbNNkbuSI7uLOTbzY6oTyG5KN8w4LK1zw/r1v4g09p4Y5Le4hcw3dpMAJbWUdY3A78ggjggggkEGgDUooooAKKKKACiiigAooooAKK5Lxh4o1fRtb0bStBsLK7uNTW4cte3DwogiCHqqMTnf6dqof2/8QP8AoC+Gv/Bpcf8AxigDvKK4P+3/AIgf9AXw1/4NLj/4xR/b/wAQP+gL4a/8Glx/8YoA7yiuD/t/4gf9AXw1/wCDS4/+MUf2/wDED/oC+Gv/AAaXH/xigDvKK4P+3/iB/wBAXw1/4NLj/wCMUf2/8QP+gL4a/wDBpcf/ABigDvKK4P8At/4gf9AXw1/4NLj/AOMUf2/8QP8AoC+Gv/Bpcf8AxigDvKK4P+3/AIgf9AXw1/4NLj/4xR/b/wAQP+gL4a/8Glx/8YoA7yiuD/t/4gf9AXw1/wCDS4/+MUf2/wDED/oC+Gv/AAaXH/xigDvKK4P+3/iB/wBAXw1/4NLj/wCMUf2/8QP+gL4a/wDBpcf/ABigDvKK4P8At/4gf9AXw1/4NLj/AOMUf2/8QP8AoC+Gv/Bpcf8AxigDvKK4P+3/AIgf9AXw1/4NLj/4xR/b/wAQP+gL4a/8Glx/8YoA7yiuD/t/4gf9AXw1/wCDS4/+MUf2/wDED/oC+Gv/AAaXH/xigDvKK4P+3/iB/wBAXw1/4NLj/wCMVveCfEM3irwdY6zdW8dtNc+ZvijcsqlZGTgkDP3fSgDeooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArm9f0O8j1FfEXhgIusQoI5rd22x6jCMnynPZhklJP4SSDlWYHpKKAM7Qtcs/EOlre2BdQGMc0Ey7ZbeVeGjkX+FlPUfiMgg1o1zGu6Ne2Opt4k8LxhtRChb2x3BU1KJegyeFmUfcc/7rfKQV2dG1my17SotQ02QvDJkEMpV42BwyOp5VlIIKnkEEUAXqKKKACiiigAooooA4Pxl/yUzwd/1w1D/0GGkvdUsbCed9Wl1QL5wjiWwtZp+BGrHIjRsct1OP0pfGX/JTPB3/AFw1D/0GGrE+mSyXEslvePAJSGdQoOSABn8gKaExmlX1vezO1hJdvatCjp9sjeOQEs4OVcBh90dR7961KpWOnm0kllknaeSQAFmGOBn/ABNXaHuC2CiiikMKKKKACszWNbj0g28Qtbm+u7pisFpaqpeTAyxy7KqgDklmA6DqQDp1zfiWx0vU9X0m1udTu9L1YNJLp9xaNsc7QDIgLKyMCMEowOQM44JCAkXxjZRWWpzapa3emTaXB9pubW5VGkWLBIceWzKwO1hwx5GDit6NxJErrnDAEZryvxMNZj8J+K/D3iq+tPEK22jNfRXv2NYZI3JkCq6AlcjaCrAA/KfrWnaaF4f1LxJqWn69Y2Utrp+n2/2G0njURW8DKTJJGnRTuGC4AI2gZp9L/wBdf6/ENnb+un+f6HodUG1eBfEKaMUk+0PatdBsDZsDBcZznOT6V574Zii8S3mjWfitE1O0/sY3FlHfL5i3J84r5rK3DuIvKO4jI8wn+I1B4njuYdegi8FyLLZ2+jTLOLa5Z5vIS4XzI4G+bEnBUZ+7jAwcYV9V53/BP9UD6/L8bf5nrFFZvh9dLXw7Y/8ACPpEmmNCrWwhHy7CMg/XnJ75zmtKqas7CTurhRRRSGFFFFABRRRQAVnfCL/klmk/71x/6PkrRrO+EX/JLNJ/3rj/ANHyUAdpRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVyus6Ze6FqsviTw3btcGXB1TS48D7YoGPNjzwJ1A46BwNp6Ky9VRQBU0vVLLWtLg1HS7hbi0uF3RyLkZ7EEHkEEEEHBBBBAIq3XJapp934X1OfxB4et5LmzuG8zVtKhXLSetxCv/PUAfMg/1gHHzgbul0/ULTVdOgv9OuI7m1uEEkU0bZV1PQg0AWKKKKACiiigDg/GX/JTPB3/AFw1D/0GGtmsnxrottrnjvwnBdS3UKhL3D2ly8D/AHIz95CD2FW/+Fb6T/0EvEH/AIPLr/4ugC3RVT/hW+k/9BLxB/4PLr/4uj/hW+k/9BLxB/4PLr/4ugC3RXJ+OPBlpovhc3mn6tr8c/26yh3HWrlvlkuoo3GC/dXYV0P/AArfSf8AoJeIP/B5df8AxdAFuiqn/Ct9J/6CXiD/AMHl1/8AF0f8K30n/oJeIP8AweXX/wAXQBbqrqGm2OrWbWmq2Vve2zEFobmJZEJHIyrAik/4VvpP/QS8Qf8Ag8uv/i6P+Fb6T/0EvEH/AIPLr/4ugBtno2l6dp72Gn6baWtm+d1vBAqRtng5UDBz3qvN4X8P3Fra21xoWmy29nn7NE9nGyQZ5OwEYXp2q1/wrfSf+gl4g/8AB5df/F0f8K30n/oJeIP/AAeXX/xdACalo+m6zbLb6xp1pfwK29YrqBZVDdMgMCM8nmnwadY2rRNbWdvC0MIgiMcSr5cYx8i4HC8DgccU3/hW+k/9BLxB/wCDy6/+Lo/4VvpP/QS8Qf8Ag8uv/i6AJLSytdPthb2FtDbQKSRFDGEUEkknA45JJ+pqeqn/AArfSf8AoJeIP/B5df8AxdH/AArfSf8AoJeIP/B5df8AxdAFuiqn/Ct9J/6CXiD/AMHl1/8AF0f8K30n/oJeIP8AweXX/wAXQBboqp/wrfSf+gl4g/8AB5df/F0f8K30n/oJeIP/AAeXX/xdAFuiqn/Ct9J/6CXiD/weXX/xdH/Ct9J/6CXiD/weXX/xdAFus74Rf8ks0n/euP8A0fJU3/Ct9J/6CXiD/wAHl1/8XR8LLdLX4Z6RDFuKqkhyzZJzK5PP40AddRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFcje6bqPhXUZ9W8M27XunXMhm1DR0wG3nlp7fPAc9WjPDnkYYnd11FAGXoviTSfEMJk0m9jnZB+9hJ2ywn+68Z+ZD7ECtSsTWfBug69dpeajpyG9jGEvIHaG4Qe0qEOPzqjH4KuLY/wCg+MPEluOwa4huMf8Af6J6ANrV9c0vQLRbnWr+3sYWbYrTyBd7egHc+wqbT9SstWsI73S7uG7tZRlJoHDq34iuLu9H1fw94tsdenl1XxbbR2s1sVdLYT2bOyHfGiJGGDBSrdWHGOM1z2s/DnxJ4uh1PULTVbvw5Df34uhobmPyp0EaITMVDAM5QsQQ6jjKk5oA6K78R6TrPxU8PWmk3iXj2iXgneAF40Yovy+YPlLccqDkd8V0viTWLrRtOhlsLOG8ubi6itoop7gwpmRguS4RyAM/3TXnfh+O+sfiJ4a03U/7Yt5baC7VbS8gtRbIuxeYJLeGNWHscEcZUZrvPGOjS69pdpZxLMyjULeWUwXDQOsayAswdWVgQP7pz6UdV6r8xPSLfk/yM288Z6pov2mLxBotrDcpZveW62WoNPHKqMiurM0SFWBkTHykEE88Vqr4ja88Uto+kWn2tLT/AJCN20myO1JGVjHynfIcgleAq8kglQcjXvAlr/wietW+ipezajf2ht1nudRluJguc7VkmdigzzgEDODT9M8O3Pg7xBCnhy1kudC1Di8tzMC9rOB/x8gu2XDgYcZLFsMASWp6af1/X+YtdST4loX8EkD/AKCWnHn2vYDW5r+qf2H4b1LVfJ8/7DaS3Plbtu/YhbbnBxnGM4NY3xG/5Exv+whYf+lkNXfG0LXPgPXbaNo1kuNPnhj8yRY1LuhVQWYgDJIHJ71E7qLtua00nNKW1zmbD4yeH7vxfpnhy43Wt7e6cl7Iz5McLtGsixbtvJ2FmLHCgKO5wNLTPiv4I1m21W40zXoriPR4TPelYZBsjGcuAVy446rnt6jPnGoaRfvrOlLbSaa9reeDzoF7cf2rChsJCMlyuSZBnAwue/PTPOaN4O1U6PrUeqz3f2yHQ7jT7E3vi22u4Z2dNuyKPYDEhKg/M4x8o55xcrXdvP8AN2/T1v5ELaPyv9yv+vpY9e/4Xb8OjaXdyPFFsYrMoJcRSEnf93au3L++0HHfFGt/FG20x9Tawsk1G2stAXW4rhLnatwrOVCD5TjgZ3c9elcrBpEcfiy1u/M0xbWLwT/ZO4XsGFudw/d43Z6Z5+771gxeHL9fCl1ZmSw+0SeCYtKVP7St+boOxMed+OhHzfd96U9G0vP/ANu/VJ/NAtk/T843/BtfI73R/idrZ1fSrbxl4Jn8P2msuIbG9TUYrtHlZdyo4QApuHQnv26kej14paX2veNL7wxpniPSNL8Pabod5DeyTnXoLqS5eJSI0RI+mWIJyenvwfa6uVunf8P6uSr9e34/1YKKKKgoKKKKACiiigAooooAKKKKACuW+Gv/ACTrSf8Acf8A9GNXU1y3w1/5J1pP+4//AKMagDqaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDltd/5KL4U/3L3/ANFpXU1xPjKHUp/G/hVNFvbayutt4RLc2puEA8tMjYHQ/juq7/Znjr/oadD/APBBL/8AJVAHU0Vy39meOv8AoadD/wDBBL/8lUf2Z46/6GnQ/wDwQS//ACVQBH8Tv+RHP/YT03/0ugrrHRZFKyKGU9QwyDXmHxD0/wAYx+Dy174j0ieL+0LAbItFkjbcbyEKcm5bgNgkY5AIyM5HT/2Z46/6GnQ//BBL/wDJVAHR/Y7b/n2h/wC/Yo+x23/PtD/37Fc5/Znjr/oadD/8EEv/AMlUf2Z46/6GnQ//AAQS/wDyVQB0f2O2/wCfaH/v2KPsdt/z7Q/9+xXOf2Z46/6GnQ//AAQS/wDyVR/Znjr/AKGnQ/8AwQS//JVAHR/Y7b/n2h/79ipq5b+zPHX/AENOh/8Aggl/+SqP7M8df9DTof8A4IJP/kqgDqaK5b+zPHX/AENOh/8Aggk/+SqP7M8df9DTof8A4IJf/kqgDqaK5b+zPHX/AENOh/8Aggl/+SqP7M8df9DTof8A4IJf/kqgDqaK5b+zPHX/AENOh/8Aggl/+SqP7M8df9DTof8A4IJf/kqgDqaK5b+zPHX/AENOh/8Aggl/+SqP7M8df9DTof8A4IJf/kqgDqaK5b+zPHX/AENOh/8Aggl/+SqP7M8df9DTof8A4IJf/kqgDqa5b4a/8k60n/cf/wBGNR/Znjr/AKGnQ/8AwQS//JVN+GIcfDbRxIwZxE25lGATvbnHagDq6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDltd/5KL4U/wBy9/8ARaV1Nctrv/JRfCn+5e/+i0rqaACiiigDk/iWhfwSQP8AoJacefa9gNdNd3UNjZT3d0/lwQRtJI+CdqqMk4HJ4Fc58RyB4LbJx/xMNPHP/X5DWl4s/wCRM1r/AK8J/wD0W1RUk4wcl0KguaSTLVrq9je3n2W2n3zfZo7rbsYfupCwRskY5KNx145q7XjWtWsdtdX9zpcVtb6zc+GbExzrDm4kjWRhcFNhWRyIguQhDcIMj5arS2Olr4d1FtC1zQbuylm09Jbfw1p7Wtur/bIsOzLNIol25GBtbGCc4XGsklK39bkrW3nb8Un+p7dWfp+u6Zql/qNlYXazXOmSiG7iAIaJioYZyOQQeoyOvoa811vwloNhD40ns9KtoG06GK5sUjjASzm8rcZIV6RuSqksuCdozmr4srmx1LW/FWjwtNe2Gpypd28Y+a8tDHEXQDu6n509wV43movo35P80v1/rYTeqS7/AOf+R6DpmpWmsaXb6jpsvnWlzGJIpNpXcp6HBAI/GrVeAJNpOpeCdCsL+10NZf8AhHoZILjWo5Lp5twk/d2lqCuZMquXRtwygKt8uNB7Kz1rwjr2p3yLe3Vr4WsLm1upGLtDMIZWEqN/C+QDvHPvVcrd/J/5/wCRTtzqEev/AAP8z2+ivIvFekRaFcX9voEVvp1nPaWM2qMYGkjeP7Q4klmRGUyZX753Asu7cSM10nw4tLO3fU30fWdGv9PkMW2HQdPNvZwyANuKkSyIXYbdwUjG0EjLZKj7yb/rQm9rf1v+vc7miiigYUUUUAFFFFABRRRQAVy3w1/5J1pP+4//AKMauprlvhr/AMk60n/cf/0Y1AHU0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHLa7/yUXwp/uXv/otK6muW13/kovhT/cvf/RaV1NABRRRQByXxO/5Ec/8AYT03/wBLoK62uS+Jp/4odvfUtOB9wb6AEflXW0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVy3w1/5J1pP+4//oxq6muW+Gv/ACTrSf8Acf8A9GNQB1NFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBy2u/wDJRfCn+5e/+i0rqa5bXf8AkovhT/cvf/RaV1NABRRRQByvxIUN4LYMMj+0dPP5XsJro7++t9M064vr2QRW1tE0srn+FVGSfyFc38Sn2eCicZ/4mWnD872AVt69pQ1zw7qGltKYftlu8PmAZ2FlIBx3xSew1vqYU/izWdPtF1bV/DsdpohKtJKL/ddW0Z/5aSw7AiqvVtsjEDJwcGtmXxLpMFtdXEt3iKzuks528tzslcoFXpzkyJyOOevBrivF19421PQJNEh8GTzJeW7WuoXUF5bbEVgA0sG+UM/y7sI6Ickc8YJqWha60OsaTZaLLNBfava6hDe/aIljWJGtyyMpbeHHlNxtwePmzxVaf16r9L/1oT2v8/T/AIc6o+N9BGrpp32qbzXuPsqzCzm+ztNz+7E+zyt2QRt3ZyMdeKZb+PPD11eLbW93PI7tIkbCyn8uV492+NH2bWcbG+QEtweK5DUNC8UalrVvNf2Gq3c9rrMU4mOpxR2SWqz5URQI43sEIJMybuGwxwoOpaeG9Wi0nwzDJa4ksddnu7geYnyRN9p2t15/1icDJ56cGp15U/P/AC1/F/cP7bS2t/np+H4j3+KFovhWLW/sEw8zTbi+Fo8U6ysYigwB5PK5cZc4AHOCMkb9v4u0ifSbrUGmmijskR7lJbWVJI96B1HlsoYkhhgAZJ468VxL+FNevNGtbM6Y9vJbaBqWmFpZotrSyeUIyNrE7W2sQTgjHIFadzpt1eeL9EDWRtodRtY5dStrh1MkZtHDxfcLKfnkAJDHgCqjZpX6/wDB/RFSskref6f8E74HIooopEhRRRQAUUUUAFFFFABRRRQAVy3w1/5J1pP+4/8A6Mauprlvhr/yTrSf9x//AEY1AHU0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHLa7/yUXwp/uXv/AKLSuprltd/5KL4U/wBy9/8ARaV1NABRRRQByXxN/wCRIP8A2E9N/wDS6Cutrkvid/yI5/7Cem/+l0FdbQAUUUUAFFFFABUEdjaRX017FawpdTqqSzrGA8irnaGbqQMnAPTJqeigAooooAKKKKACiiigAooooAKKKKACuW+Gv/JOtJ/3H/8ARjV1Nct8Nf8AknWk/wC4/wD6MagDqaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDltd/5KL4U/3L3/0WldTXLa7/AMlF8Kf7l7/6LSupoAKKKKAOW+I3/ImN/wBhCw/9LIa1fE00lv4S1eaCRopY7KZkdGIZSEJBBHQ1jfE0keCDg4/4memjj/r+grpdQso9S0y6sZ2ZYrqF4XKEBgGBBxnvzUVE3BpFQaUk2ed6Q8D3mjN4YbxabxpY2vH1M6ibYwbcybvtfyc8bdnzZxj5d1Fv421VND8P+I9eNoba9hmumgs45ozEiWskjAnzSshJXIBXC57kBq9IghWC2jgQkrGgQE9cAYrCt/BOl2+l6Lp5aea30eN44VlZT5qtG0ZEny8/Kx6Yq5N+9y/L+v66kxSsuby/r1/4Bk3fiTxJ4f8ADb65rqaZdQTRReTb2UEyyQTSuiJGcFzMvz8sqo3y8Id2Ft+EPEWqarqF3Z6pFJMkUaSxXo0W701GJJBjKXGSSMA7lY5B5Axy9PAds2ny2GoazrGoWRgEFvb3Fwii1UEFShRFZnUqu15C7DbkHJJOro+jTaZJLLd61qWrTSAKHvXjARRnACRIidSfm27j0JwABV1fT+u39fmL7Pn/AMMalFFFSMKKKKACiiigAooooAKKKKACiiigArlvhr/yTrSf9x//AEY1dTXLfDX/AJJ1pP8AuP8A+jGoA6miiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5bXf+Si+FP9y9/9FpXU1x/i67/szxd4a1Ka0v7i1txdrK1lZS3JQsihcrGrEZwecVP/AMLC0j/nw8Rf+E7ff/GaAOporlv+FhaR/wA+HiL/AMJ2+/8AjNH/AAsLSP8Anw8Rf+E7ff8AxmgCP4mDPghsAk/2lp2Mdv8AToOa62vNvG/jC11bwwbTTNK8QzXBvbOTafD96vyJdRO5yYgOEVj+FdB/wsLSP+fDxF/4Tt9/8ZoA6miuW/4WFpH/AD4eIv8Awnb7/wCM0f8ACwtI/wCfDxF/4Tt9/wDGaAOporlv+FhaR/z4eIv/AAnb7/4zR/wsLSP+fDxF/wCE7ff/ABmgDqaK5b/hYWkf8+HiL/wnb7/4zR/wsLSP+fDxF/4Tt9/8ZoA6miuW/wCFhaR/z4eIv/Cdvv8A4zR/wsLSP+fDxF/4Tt9/8ZoA6miuW/4WFpH/AD4eIv8Awnb7/wCM0f8ACwtI/wCfDxF/4Tt9/wDGaAOporlv+FhaR/z4eIv/AAnb7/4zR/wsLSP+fDxF/wCE7ff/ABmgDqaK5b/hYWkf8+HiL/wnb7/4zR/wsLSP+fDxF/4Tt9/8ZoA6miuW/wCFhaR/z4eIv/Cdvv8A4zR/wsLSP+fDxF/4Tt9/8ZoA6muW+Gv/ACTrSf8Acf8A9GNR/wALC0j/AJ8PEX/hO33/AMZp/wAOoZrf4faVHdQTW8ojYtFPG0brl2IyrAEHB6EUAdNRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAf/Z)

1. **DETAILED CLASS DIAGRAM**



**DEVELOPMENT**

1. **SOURCE CODES**
2. **DATA ACCESS**

* searchdarabase.py

from PyQt4 import QtCore, QtGui

import sys#dal login and main user get

import sqlite3

class Search(object):

def CheckInfo(user,password):

connection = sqlite3.connect('ShopDatabase.db')

cursor = connection.cursor()

getInfo=connection.execute('SELECT Username,Password FROM Users WHERE Username=? AND Password=?',[user,password]).fetchall()

if not getInfo: #check if the list is empty, if it is empty the input pass and user is wrong

valid=False

else:

valid=True

cursor.close()

connection.close()

return valid

def getInfo(username,password):

connection = sqlite3.connect('ShopDatabase.db')

cursor=connection.cursor()

getUser=connection.execute('SELECT UserID,Name,ContactNo,Address FROM Users WHERE Username=? AND Password=?',[username,password]).fetchall()

cursor.close()

connection.close()

return getUser

def loadDatabase():

connection = sqlite3.connect('ShopDatabase.db')

query = "SELECT ItemName,Quantity,ItemPrice FROM Stock"

cursor=connection.cursor()

result = connection.execute(query)

return (result, connection, cursor)

def searchDatabase(search=None):

connection = sqlite3.connect('ShopDatabase.db')

cursor=connection.cursor()

result=connection.execute("SELECT ItemName,Quantity,ItemPrice FROM Stock WHERE ItemName LIKE ?",('%'+search+'%',))

return (result, connection, cursor)

* create\_user.py

from PyQt4 import QtCore, QtGui

import sys#dal create

import sqlite3

class CreateUser(object):

def CheckUsername(username=None):

connection = sqlite3.connect('ShopDatabase.db')

cursor = connection.cursor()

getInfo=connection.execute('SELECT Username FROM Users WHERE Username=?',[username]).fetchall()

if not getInfo:

dup=False

else:

dup=True

cursor.close()

connection.close()

return dup

def InsertNewUser(user=None,password=None,name=None,contact=None,address=None):

connection = sqlite3.connect('ShopDatabase.db')

cursor = connection.cursor()

cursor.execute("INSERT INTO Users (Username, Password, Name,ContactNo,Address)VALUES(?,?,?,?,?)",(user,password,name,contact,address))

connection.commit() #save

cursor.close()

connection.close()

1. **BUSINESS LOGIC**

* check\_input.py

from PyQt4 import QtCore, QtGui

import sys#bll create

import sqlite3

from create\_user import CreateUser

class CheckCreateInput(object):

def SaveUser(user,password,name,fName,mName,lName,contact,address):

if fName=="" or mName=="" or lName=="" or user=="" or password=="" or address=="" or contact=="":

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("Place make sure that all fields have been filled.")

msg.setWindowTitle("Error")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

else:

#DAL statement?

dup=CreateUser.CheckUsername(user)

#check if the username is already in the databse

if dup==True:

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("There is already an existing username")

msg.setWindowTitle("Duplicate")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

else:

CreateUser.InsertNewUser(user,password,name,contact,address)

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("The account has been successfully created.")

msg.setWindowTitle("Success!!!")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

* Get\_User.py

from PyQt4 import QtCore, QtGui

import sys #bll login and main userinfo

import sqlite3

from searchdatabase import Search

class GetUserInfo(object):

def \_\_init\_\_(self,username,password):

self.username=username

self.password=password

self.user=Search.getInfo(self.username,self.password)

self.UserID=self.user[0][0]

self.Name=self.user[0][1]

self.Contact=self.user[0][2]

self.Address=self.user[0][3]

def checkInfo(self):

#DAL statement?

valid=Search.CheckInfo(self.username,self.password)

#check if the username is already in the databse

if valid==True:

return True

else:

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("The supplied username or password is not correct")

msg.setWindowTitle("Wrong info")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

def getUser(self):

return self.user

def accountInfo(self):

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("Name: "+self.Name)

msg.setInformativeText("Username: "+self.username+"\n User ID: "+str(self.UserID)+"\n Contact: "+str(self.Contact)+"\n Address: "+self.Address)

msg.setWindowTitle("Account Details")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

* mainButtonControl.py

from PyQt4 import QtCore, QtGui

import sys #bll main buttons

import sqlite3

from searchdatabase import Search

class ButtonControl(object):

def loadData():

data, connection, cursor=Search.loadDatabase()

return (data, connection, cursor)

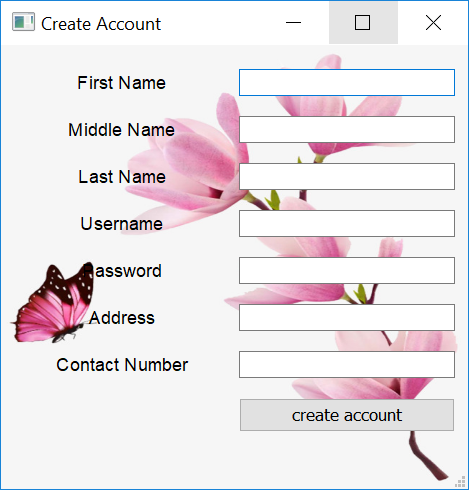
def searchButton(search=None):

result, connection, cursor =Search.searchDatabase(search)

return (result, connection, cursor)

1. **USER INTERFACE**

* create.py



# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'create.ui'

#

# Created by: PyQt4 UI code generator 4.11.4

#

# WARNING! All changes made in this file will be lost!

from PyQt4 import QtCore, QtGui

import sqlite3

from check\_input import CheckCreateInput

try:

\_fromUtf8 = QtCore.QString.fromUtf8

except AttributeError:

def \_fromUtf8(s):

return s

try:

\_encoding = QtGui.QApplication.UnicodeUTF8

def \_translate(context, text, disambig):

return QtGui.QApplication.translate(context, text, disambig, \_encoding)

except AttributeError:

def \_translate(context, text, disambig):

return QtGui.QApplication.translate(context, text, disambig)

class Ui\_NewUser(object):

## def \_\_init\_\_ (self,createWindow=None):

## self.createWindow=createWindow

def setupUi(self, NewUser):

NewUser.setObjectName(\_fromUtf8("NewUser"))

NewUser.resize(467, 444)

NewUser.setEnabled(True)

self.centralwidget = QtGui.QWidget(NewUser)

self.centralwidget.setObjectName(\_fromUtf8("centralwidget"))

self.gridLayout = QtGui.QGridLayout(self.centralwidget)

self.gridLayout.setObjectName(\_fromUtf8("gridLayout"))

self.label\_fn = QtGui.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_fn.setFont(font)

self.label\_fn.setAlignment(QtCore.Qt.AlignCenter)

self.label\_fn.setObjectName(\_fromUtf8("label\_fn"))

self.gridLayout.addWidget(self.label\_fn, 0, 0, 1, 1)

self.lineEdit\_First = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_First.setObjectName(\_fromUtf8("lineEdit\_First"))

self.gridLayout.addWidget(self.lineEdit\_First, 0, 1, 1, 1)

self.label\_Mn = QtGui.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_Mn.setFont(font)

self.label\_Mn.setAlignment(QtCore.Qt.AlignCenter)

self.label\_Mn.setObjectName(\_fromUtf8("label\_Mn"))

self.gridLayout.addWidget(self.label\_Mn, 1, 0, 1, 1)

self.lineEdit\_Middle = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Middle.setObjectName(\_fromUtf8("lineEdit\_Middle"))

self.gridLayout.addWidget(self.lineEdit\_Middle, 1, 1, 1, 1)

self.label\_LN = QtGui.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_LN.setFont(font)

self.label\_LN.setAlignment(QtCore.Qt.AlignCenter)

self.label\_LN.setObjectName(\_fromUtf8("label\_LN"))

self.gridLayout.addWidget(self.label\_LN, 2, 0, 1, 1)

self.lineEdit\_Last = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Last.setObjectName(\_fromUtf8("lineEdit\_Last"))

self.gridLayout.addWidget(self.lineEdit\_Last, 2, 1, 1, 1)

self.label\_user = QtGui.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_user.setFont(font)

self.label\_user.setAlignment(QtCore.Qt.AlignCenter)

self.label\_user.setObjectName(\_fromUtf8("label\_user"))

self.gridLayout.addWidget(self.label\_user, 3, 0, 1, 1)

self.lineEdit\_User = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_User.setObjectName(\_fromUtf8("lineEdit\_User"))

self.gridLayout.addWidget(self.lineEdit\_User, 3, 1, 1, 1)

self.label\_pass = QtGui.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_pass.setFont(font)

self.label\_pass.setAlignment(QtCore.Qt.AlignCenter)

self.label\_pass.setObjectName(\_fromUtf8("label\_pass"))

self.gridLayout.addWidget(self.label\_pass, 4, 0, 1, 1)

self.lineEdit\_Pass = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Pass.setObjectName(\_fromUtf8("lineEdit\_Pass"))

self.gridLayout.addWidget(self.lineEdit\_Pass, 4, 1, 1, 1)

self.label\_address = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_address.sizePolicy().hasHeightForWidth())

self.label\_address.setSizePolicy(sizePolicy)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_address.setFont(font)

self.label\_address.setAlignment(QtCore.Qt.AlignCenter)

self.label\_address.setObjectName(\_fromUtf8("label\_address"))

self.gridLayout.addWidget(self.label\_address, 5, 0, 1, 1)

self.lineEdit\_Address = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Address.setObjectName(\_fromUtf8("lineEdit\_Address"))

self.gridLayout.addWidget(self.lineEdit\_Address, 5, 1, 1, 1)

self.label\_contact = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_contact.sizePolicy().hasHeightForWidth())

self.label\_contact.setSizePolicy(sizePolicy)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(9)

self.label\_contact.setFont(font)

self.label\_contact.setAlignment(QtCore.Qt.AlignCenter)

self.label\_contact.setObjectName(\_fromUtf8("label\_contact"))

self.gridLayout.addWidget(self.label\_contact, 6, 0, 1, 1)

self.lineEdit\_Contact = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Contact.setObjectName(\_fromUtf8("lineEdit\_Contact"))

self.gridLayout.addWidget(self.lineEdit\_Contact, 6, 1, 1, 1)

self.btn\_create = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_create.sizePolicy().hasHeightForWidth())

self.btn\_create.setSizePolicy(sizePolicy)

self.btn\_create.setObjectName(\_fromUtf8("btn\_create"))

self.gridLayout.addWidget(self.btn\_create, 7, 1, 1, 1)

NewUser.setCentralWidget(self.centralwidget)

self.statusbar = QtGui.QStatusBar(NewUser)

self.statusbar.setObjectName(\_fromUtf8("statusbar"))

NewUser.setStatusBar(self.statusbar)

self.retranslateUi(NewUser)

QtCore.QMetaObject.connectSlotsByName(NewUser)

NewUser.setStyleSheet("#NewUser { border-image: url(flower\_pink.jpg) 0 0 0 0 stretch stretch; }")

self.btn\_create.clicked.connect(self.on\_pushButton\_clicked\_create)

#self.btn\_create.setFlat(True)

#self.btn\_create.setAutoFillBackground(True)

## ####

## self.btn\_create.setStyleSheet("QPushButton { background-image: url(Yamcha\_Missing\_Tooth\_LOL.jpg)0 0 0 0 stretch stretch ; }")

## self.btn\_create.setIcon(QtGui.QIcon("Yamcha\_Missing\_Tooth\_LOL.jpg"))

## self.btn\_create.setIconSize(QtCore.QSize(100,100))

## ####

def retranslateUi(self, NewUser):

NewUser.setWindowTitle(\_translate("NewUser", "Create Account", None))

self.label\_fn.setText(\_translate("NewUser", "First Name", None))

self.label\_Mn.setText(\_translate("NewUser", "Middle Name", None))

self.label\_LN.setText(\_translate("NewUser", "Last Name", None))

self.label\_user.setText(\_translate("NewUser", "Username", None))

self.label\_pass.setText(\_translate("NewUser", "Password", None))

self.label\_address.setText(\_translate("NewUser", "Address", None))

self.label\_contact.setText(\_translate("NewUser", "Contact Number", None))

self.btn\_create.setText(\_translate("NewUser", "create account", None))

def on\_pushButton\_clicked\_create(self):

self.fName=self.lineEdit\_First.text()

self.mName=self.lineEdit\_Middle.text()

self.lName=self.lineEdit\_Last.text()

self.user=self.lineEdit\_User.text()

self.password=self.lineEdit\_Pass.text()

self.address=self.lineEdit\_Address.text()

self.contact=self.lineEdit\_Contact.text()

self.Name=self.lName+", "+self.fName+", "+self.mName

CheckCreateInput.SaveUser(self.user,self.password,self.Name,self.fName,self.mName,self.lName,self.contact,self.address)

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtGui.QApplication(sys.argv)

NewUser = QtGui.QMainWindow()

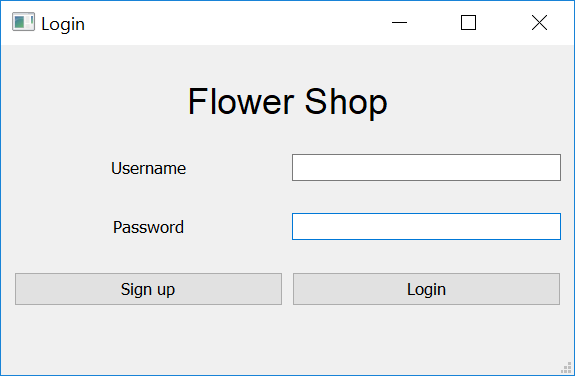
ui = Ui\_NewUser()

ui.setupUi(NewUser)

NewUser.show()

sys.exit(app.exec\_())

* login.py



# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'login.ui'

#

# Created by: PyQt4 UI code generator 4.11.4

#

# WARNING! All changes made in this file will be lost!

from PyQt4 import QtCore, QtGui

import create

import main

import sqlite3

from Get\_User import GetUserInfo

try:

\_fromUtf8 = QtCore.QString.fromUtf8

except AttributeError:

def \_fromUtf8(s):

return s

try:

\_encoding = QtGui.QApplication.UnicodeUTF8

def \_translate(context, text, disambig):

return QtGui.QApplication.translate(context, text, disambig, \_encoding)

except AttributeError:

def \_translate(context, text, disambig):

return QtGui.QApplication.translate(context, text, disambig)

class Ui\_LoginWindow(object):

def setupUi(self, LoginWindow):

LoginWindow.setObjectName(\_fromUtf8("LoginWindow"))

LoginWindow.resize(573, 330)

self.centralwidget = QtGui.QWidget(LoginWindow)

self.centralwidget.setObjectName(\_fromUtf8("centralwidget"))

self.gridLayout = QtGui.QGridLayout(self.centralwidget)

self.gridLayout.setObjectName(\_fromUtf8("gridLayout"))

self.label\_pass = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Preferred)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_pass.sizePolicy().hasHeightForWidth())

self.label\_pass.setSizePolicy(sizePolicy)

self.label\_pass.setAlignment(QtCore.Qt.AlignCenter)

self.label\_pass.setObjectName(\_fromUtf8("label\_pass"))

self.gridLayout.addWidget(self.label\_pass, 2, 0, 1, 1)

self.lineEdit\_Pass = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Pass.setText(\_fromUtf8(""))

self.lineEdit\_Pass.setObjectName(\_fromUtf8("lineEdit\_Pass"))

self.gridLayout.addWidget(self.lineEdit\_Pass, 2, 1, 1, 1)

self.btn\_Signup = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Signup.sizePolicy().hasHeightForWidth())

self.btn\_Signup.setSizePolicy(sizePolicy)

self.btn\_Signup.setObjectName(\_fromUtf8("btn\_Signup"))

self.gridLayout.addWidget(self.btn\_Signup, 3, 0, 1, 1)

self.btn\_Login = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Login.sizePolicy().hasHeightForWidth())

self.btn\_Login.setSizePolicy(sizePolicy)

self.btn\_Login.setObjectName(\_fromUtf8("btn\_Login"))

self.gridLayout.addWidget(self.btn\_Login, 3, 1, 1, 1)

self.label\_user = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Preferred)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_user.sizePolicy().hasHeightForWidth())

self.label\_user.setSizePolicy(sizePolicy)

self.label\_user.setAlignment(QtCore.Qt.AlignCenter)

self.label\_user.setObjectName(\_fromUtf8("label\_user"))

self.gridLayout.addWidget(self.label\_user, 1, 0, 1, 1)

self.lineEdit\_User = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_User.setText(\_fromUtf8(""))

self.lineEdit\_User.setObjectName(\_fromUtf8("lineEdit\_User"))

self.gridLayout.addWidget(self.lineEdit\_User, 1, 1, 1, 1)

self.label\_3 = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_3.sizePolicy().hasHeightForWidth())

self.label\_3.setSizePolicy(sizePolicy)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(18)

self.label\_3.setFont(font)

self.label\_3.setAlignment(QtCore.Qt.AlignCenter)

self.label\_3.setObjectName(\_fromUtf8("label\_3"))

self.gridLayout.addWidget(self.label\_3, 0, 0, 1, 2)

LoginWindow.setCentralWidget(self.centralwidget)

self.statusbar = QtGui.QStatusBar(LoginWindow)

self.statusbar.setObjectName(\_fromUtf8("statusbar"))

LoginWindow.setStatusBar(self.statusbar)

self.retranslateUi(LoginWindow)

QtCore.QMetaObject.connectSlotsByName(LoginWindow)

##########

self.lineEdit\_Pass.setEchoMode(QtGui.QLineEdit.Password) ##set to bullets

self.btn\_Signup.clicked.connect(self.on\_btn\_Signup\_clicked)

self.btn\_Login.clicked.connect(self.on\_btn\_Login\_clicked)

def retranslateUi(self, LoginWindow):

LoginWindow.setWindowTitle(\_translate("LoginWindow", "Login", None))

self.label\_pass.setText(\_translate("LoginWindow", "Password", None))

self.btn\_Signup.setText(\_translate("LoginWindow", "Sign up", None))

self.btn\_Login.setText(\_translate("LoginWindow", "Login", None))

self.label\_user.setText(\_translate("LoginWindow", "Username", None))

self.label\_3.setText(\_translate("LoginWindow", "Flower Shop", None))

def on\_btn\_Signup\_clicked(self):

self.NewUser = QtGui.QMainWindow()

self.ui = create.Ui\_NewUser()

self.ui.setupUi(self.NewUser)

self.dialog=self.NewUser

self.dialog.show()

#login

def on\_btn\_Login\_clicked(self):

user=self.lineEdit\_User.text()

password=self.lineEdit\_Pass.text()

obj=GetUserInfo(user,password)

valid=obj.checkInfo()

if valid ==True:

self.MainWindow = QtGui.QMainWindow()

self.ui = main.Ui\_MainWindow(user,password)

self.ui.setupUi(self.MainWindow)

self.dialog=self.MainWindow

self.dialog.show()

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtGui.QApplication(sys.argv)

LoginWindow = QtGui.QMainWindow()

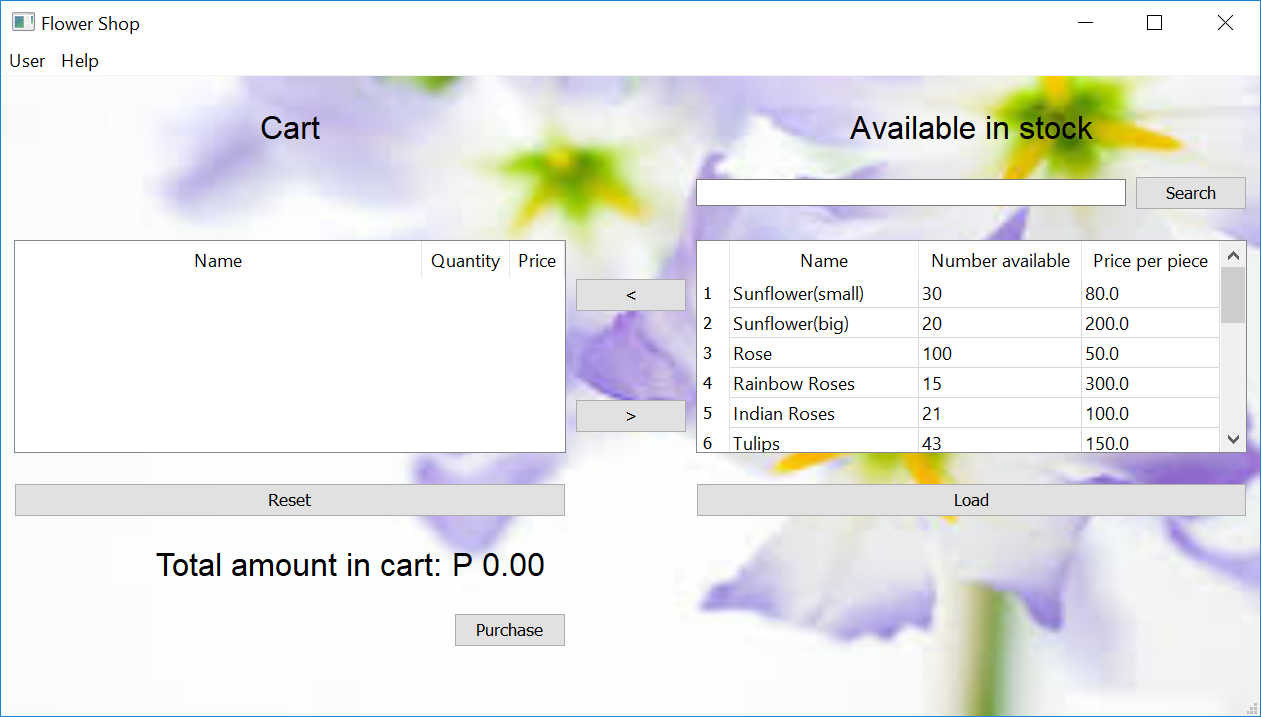
ui = Ui\_LoginWindow()

ui.setupUi(LoginWindow)

LoginWindow.show()

sys.exit(app.exec\_())

* main.py



# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'mainv3.ui'

#

# Created by: PyQt4 UI code generator 4.11.4

#

# WARNING! All changes made in this file will be lost!

from PyQt4 import QtCore, QtGui

import datetime

import sqlite3

from Get\_User import GetUserInfo

from mainButtonControl import ButtonControl

try:

\_fromUtf8 = QtCore.QString.fromUtf8

except AttributeError:

def \_fromUtf8(s):

return s

try:

\_encoding = QtGui.QApplication.UnicodeUTF8

def \_translate(context, text, disambig):

return QtGui.QApplication.translate(context, text, disambig, \_encoding)

except AttributeError:

def \_translate(context, text, disambig):

return QtGui.QApplication.translate(context, text, disambig)

class Ui\_MainWindow(object):

#the init function is used to send the user and password to this form

def \_\_init\_\_(self, gUser="cheng",gPass="1234"):#TOOOOO EDDIIT

self.obj=GetUserInfo(gUser,gPass)

def setupUi(self, MainWindow):

MainWindow.setObjectName(\_fromUtf8("MainWindow"))

MainWindow.resize(915, 669)

MainWindow.setEnabled(True)

self.centralwidget = QtGui.QWidget(MainWindow)

self.centralwidget.setObjectName(\_fromUtf8("centralwidget"))

self.gridLayout = QtGui.QGridLayout(self.centralwidget)

self.gridLayout.setObjectName(\_fromUtf8("gridLayout"))

self.label\_stock = QtGui.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(16)

self.label\_stock.setFont(font)

self.label\_stock.setAlignment(QtCore.Qt.AlignCenter)

self.label\_stock.setObjectName(\_fromUtf8("label\_stock"))

self.gridLayout.addWidget(self.label\_stock, 0, 3, 1, 2)

self.btn\_Great = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Great.sizePolicy().hasHeightForWidth())

self.btn\_Great.setSizePolicy(sizePolicy)

self.btn\_Great.setObjectName(\_fromUtf8("btn\_Great"))

self.gridLayout.addWidget(self.btn\_Great, 4, 2, 1, 1)

self.btn\_Reset = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Reset.sizePolicy().hasHeightForWidth())

self.btn\_Reset.setSizePolicy(sizePolicy)

self.btn\_Reset.setObjectName(\_fromUtf8("btn\_Reset"))

self.gridLayout.addWidget(self.btn\_Reset, 5, 0, 1, 2)

self.btn\_Load = QtGui.QPushButton(self.centralwidget)

self.btn\_Load.setObjectName(\_fromUtf8("btn\_Load"))

self.gridLayout.addWidget(self.btn\_Load, 5, 3, 1, 2)

self.tableWidget\_Cart = QtGui.QTableWidget(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Expanding)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.tableWidget\_Cart.sizePolicy().hasHeightForWidth())

self.tableWidget\_Cart.setSizePolicy(sizePolicy)

self.tableWidget\_Cart.setObjectName(\_fromUtf8("tableWidget\_Cart"))

self.tableWidget\_Cart.setColumnCount(3)

self.tableWidget\_Cart.setRowCount(0)

item = QtGui.QTableWidgetItem()

self.tableWidget\_Cart.setHorizontalHeaderItem(0, item)

item = QtGui.QTableWidgetItem()

self.tableWidget\_Cart.setHorizontalHeaderItem(1, item)

item = QtGui.QTableWidgetItem()

self.tableWidget\_Cart.setHorizontalHeaderItem(2, item)

self.tableWidget\_Cart.horizontalHeader().setMinimumSectionSize(49)

self.gridLayout.addWidget(self.tableWidget\_Cart, 3, 0, 2, 2)

self.label\_Total = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_Total.sizePolicy().hasHeightForWidth())

self.label\_Total.setSizePolicy(sizePolicy)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(16)

self.label\_Total.setFont(font)

self.label\_Total.setAlignment(QtCore.Qt.AlignCenter)

self.label\_Total.setObjectName(\_fromUtf8("label\_Total"))

self.gridLayout.addWidget(self.label\_Total, 6, 0, 1, 3)

self.lineEdit\_Search = QtGui.QLineEdit(self.centralwidget)

self.lineEdit\_Search.setObjectName(\_fromUtf8("lineEdit\_Search"))

self.gridLayout.addWidget(self.lineEdit\_Search, 2, 3, 1, 1)

self.btn\_Search = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Search.sizePolicy().hasHeightForWidth())

self.btn\_Search.setSizePolicy(sizePolicy)

self.btn\_Search.setObjectName(\_fromUtf8("btn\_Search"))

self.gridLayout.addWidget(self.btn\_Search, 2, 4, 1, 1)

self.btn\_Less = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Less.sizePolicy().hasHeightForWidth())

self.btn\_Less.setSizePolicy(sizePolicy)

self.btn\_Less.setObjectName(\_fromUtf8("btn\_Less"))

self.gridLayout.addWidget(self.btn\_Less, 3, 2, 1, 1)

self.btn\_Purchase = QtGui.QPushButton(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn\_Purchase.sizePolicy().hasHeightForWidth())

self.btn\_Purchase.setSizePolicy(sizePolicy)

self.btn\_Purchase.setObjectName(\_fromUtf8("btn\_Purchase"))

self.gridLayout.addWidget(self.btn\_Purchase, 7, 1, 1, 1)

self.tableWidget\_Stock = QtGui.QTableWidget(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Minimum)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.tableWidget\_Stock.sizePolicy().hasHeightForWidth())

self.tableWidget\_Stock.setSizePolicy(sizePolicy)

self.tableWidget\_Stock.setObjectName(\_fromUtf8("tableWidget\_Stock"))

self.tableWidget\_Stock.setColumnCount(3)

self.tableWidget\_Stock.setRowCount(0)

item = QtGui.QTableWidgetItem()

self.tableWidget\_Stock.setHorizontalHeaderItem(0, item)

item = QtGui.QTableWidgetItem()

self.tableWidget\_Stock.setHorizontalHeaderItem(1, item)

item = QtGui.QTableWidgetItem()

self.tableWidget\_Stock.setHorizontalHeaderItem(2, item)

self.tableWidget\_Stock.horizontalHeader().setMinimumSectionSize(49)

self.gridLayout.addWidget(self.tableWidget\_Stock, 3, 3, 2, 2)

self.label\_4 = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Minimum)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_4.sizePolicy().hasHeightForWidth())

self.label\_4.setSizePolicy(sizePolicy)

self.label\_4.setObjectName(\_fromUtf8("label\_4"))

self.gridLayout.addWidget(self.label\_4, 2, 0, 1, 1)

self.label\_cart = QtGui.QLabel(self.centralwidget)

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Fixed)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.label\_cart.sizePolicy().hasHeightForWidth())

self.label\_cart.setSizePolicy(sizePolicy)

font = QtGui.QFont()

font.setFamily(\_fromUtf8("Arial"))

font.setPointSize(16)

self.label\_cart.setFont(font)

self.label\_cart.setAlignment(QtCore.Qt.AlignCenter)

self.label\_cart.setObjectName(\_fromUtf8("label\_cart"))

self.gridLayout.addWidget(self.label\_cart, 0, 0, 1, 2)

self.lineEdit\_Search.raise\_()

self.btn\_Search.raise\_()

self.btn\_Great.raise\_()

self.btn\_Reset.raise\_()

self.btn\_Load.raise\_()

self.label\_stock.raise\_()

self.tableWidget\_Cart.raise\_()

self.label\_cart.raise\_()

self.tableWidget\_Stock.raise\_()

self.label\_Total.raise\_()

self.btn\_Purchase.raise\_()

self.label\_4.raise\_()

self.btn\_Less.raise\_()

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtGui.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 915, 26))

self.menubar.setObjectName(\_fromUtf8("menubar"))

self.menuUser = QtGui.QMenu(self.menubar)

self.menuUser.setObjectName(\_fromUtf8("menuUser"))

self.menuHelp = QtGui.QMenu(self.menubar)

self.menuHelp.setObjectName(\_fromUtf8("menuHelp"))

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtGui.QStatusBar(MainWindow)

self.statusbar.setObjectName(\_fromUtf8("statusbar"))

MainWindow.setStatusBar(self.statusbar)

self.actionAbout = QtGui.QAction(MainWindow)

self.actionAbout.setObjectName(\_fromUtf8("actionAbout"))

self.actionHelp = QtGui.QAction(MainWindow)

self.actionHelp.setObjectName(\_fromUtf8("actionHelp"))

self.actionAccount\_info = QtGui.QAction(MainWindow)

self.actionAccount\_info.setObjectName(\_fromUtf8("actionAccount\_info"))

self.menuUser.addAction(self.actionAccount\_info)

self.menuHelp.addAction(self.actionAbout)

self.menuHelp.addAction(self.actionHelp)

self.menubar.addAction(self.menuUser.menuAction())

self.menubar.addAction(self.menuHelp.menuAction())

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

#####

MainWindow.setStyleSheet("#MainWindow { border-image: url(Flower\_main2.jpg) 0 0 0 0 stretch stretch; }")

self.tableWidget\_Cart.setEditTriggers(QtGui.QAbstractItemView.NoEditTriggers)

self.tableWidget\_Stock.setEditTriggers(QtGui.QAbstractItemView.NoEditTriggers)

header = self.tableWidget\_Stock.horizontalHeader()

header.setResizeMode(0, QtGui.QHeaderView.Stretch)

header.setResizeMode(1, QtGui.QHeaderView.ResizeToContents)

header.setResizeMode(2, QtGui.QHeaderView.ResizeToContents)

header = self.tableWidget\_Cart.horizontalHeader()

header.setResizeMode(0, QtGui.QHeaderView.Stretch)

header.setResizeMode(1, QtGui.QHeaderView.ResizeToContents)

header.setResizeMode(2, QtGui.QHeaderView.ResizeToContents)

self.btn\_Load.clicked.connect(self.btn\_load\_clicked)

self.btn\_Less.clicked.connect(self.btn\_lessthan\_clicked)

self.btn\_Reset.clicked.connect(self.btn\_reset\_clicked)

self.btn\_Great.clicked.connect(self.btn\_greaterthan\_clicked)

self.btn\_Purchase.clicked.connect(self.btn\_purchase\_clicked)

self.actionAccount\_info.triggered.connect(self.account\_info\_clicked)

self.actionAbout.triggered.connect(self.About\_clicked)

self.actionHelp.triggered.connect(self.Help\_clicked)

self.btn\_Search.clicked.connect(self.btn\_search\_clicked)

def btn\_load\_clicked(self):# this part is the connecting of database to mainwindow

result, connection, cursor=ButtonControl.loadData()

self.tableWidget\_Stock.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidget\_Stock.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidget\_Stock.setItem(row\_number, column\_number, QtGui.QTableWidgetItem(str(data)))

cursor.close()

connection.close()

def btn\_search\_clicked(self):

result, connection, cursor=ButtonControl.searchButton(self.lineEdit\_Search.text())

self.tableWidget\_Stock.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidget\_Stock.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidget\_Stock.setItem(row\_number, column\_number, QtGui.QTableWidgetItem(str(data)))

cursor.close()

connection.close()

# the next three methods are for the menubar

def Help\_clicked(self):

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText(" ")

msg.setInformativeText( " ")

msg.setWindowTitle("Help")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

def About\_clicked(self):

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("Name")

msg.setInformativeText("Info ")

msg.setWindowTitle("About")

msg.setStandardButtons(QtGui.QMessageBox.Ok)

retval = msg.exec\_()

def account\_info\_clicked(self):

self.obj.accountInfo()

def btn\_lessthan\_clicked(self):

self.rowPosition = self.tableWidget\_Cart.rowCount() #gives rowPos the current rowcount

self.selectedRow=self.tableWidget\_Stock.currentRow() #gives selectedrow the currently clicked row's index

col1=self.tableWidget\_Stock.item(self.selectedRow,0) #selects the item at the current row, and column

col2=self.tableWidget\_Stock.item(self.selectedRow,1)

col3=self.tableWidget\_Stock.item(self.selectedRow,2)

#update the price and quantity if the item is already in the cart

dup=False

rowCtr=0

while rowCtr!=self.tableWidget\_Cart.rowCount():

if self.tableWidget\_Cart.item(rowCtr,0).text()==self.tableWidget\_Stock.item(self.selectedRow,0).text():

dup=True

quantity=int(self.tableWidget\_Cart.item(rowCtr,1).text())

self.tableWidget\_Cart.setItem(rowCtr , 1, QtGui.QTableWidgetItem(str(quantity+1)))

self.itemPrice=float(col3.text())

newPrice=self.itemPrice\*(quantity+1)

self.tableWidget\_Cart.setItem(rowCtr , 2, QtGui.QTableWidgetItem(str(newPrice)))

break

rowCtr+=1

#insert a new row if the item is not yet in the cart

if dup==False:

self.tableWidget\_Cart.insertRow(self.rowPosition) #insert a new row

self.tableWidget\_Cart.setItem(self.rowPosition , 0, QtGui.QTableWidgetItem(col1.text())) #insert item at the current row and col

self.tableWidget\_Cart.setItem(self.rowPosition , 1, QtGui.QTableWidgetItem("1"))

self.tableWidget\_Cart.setItem(self.rowPosition , 2, QtGui.QTableWidgetItem(col3.text()))

self.UpdatePrice()

def btn\_greaterthan\_clicked(self):

if self.tableWidget\_Cart.item(self.tableWidget\_Cart.currentRow(),1)==None:

return

else:

if int(self.tableWidget\_Cart.item(self.tableWidget\_Cart.currentRow(),1).text())>1:

quantity=int(self.tableWidget\_Cart.item(self.tableWidget\_Cart.currentRow(),1).text())-1

self.tableWidget\_Cart.setItem(self.tableWidget\_Cart.currentRow() , 1, QtGui.QTableWidgetItem(str(quantity)))

newPrice=(float(self.tableWidget\_Cart.item(self.tableWidget\_Cart.currentRow(),2).text()))/(quantity+1)

newPrice=newPrice\*quantity

self.tableWidget\_Cart.setItem(self.tableWidget\_Cart.currentRow() , 2, QtGui.QTableWidgetItem(str(newPrice)))

else:

self.tableWidget\_Cart.removeRow(self.tableWidget\_Cart.currentRow())

self.UpdatePrice()

def btn\_reset\_clicked(self):

self.tableWidget\_Cart.setRowCount(0)

self.label\_Total.setText("Total amount in cart:0.0")

def btn\_purchase\_clicked(self):

msg = QtGui.QMessageBox()

msg.setIcon(QtGui.QMessageBox.Information)

msg.setText("Are you done purchasing?")

msg.setWindowTitle("Confirmation")

msg.setStandardButtons(QtGui.QMessageBox.Ok | QtGui.QMessageBox.Cancel)

retval = msg.exec\_()

print ("value of pressed message box button:", retval)

if retval==1024:

connection = sqlite3.connect('ShopDatabase.db')

c=connection.cursor()

now = datetime.datetime.now()

time=now.strftime("%Y-%m-%d %H:%M")

c.execute("INSERT INTO Purchase (UserID,OrderDate,Bill) VALUES (?,?,?)",(self.UserID,time,self.price))

connection.commit()

c.close()

connection.close()

def UpdatePrice(self):

ctr=self.tableWidget\_Cart.rowCount()

self.price=0.00

while ctr!=0:

self.price=self.price+float(self.tableWidget\_Cart.item(ctr-1,2).text())

ctr=ctr-1

self.label\_Total.setText("Total amount in cart:"+str(self.price))

def retranslateUi(self, MainWindow):

MainWindow.setWindowTitle(\_translate("MainWindow", "Flower Shop", None))

self.label\_stock.setText(\_translate("MainWindow", "Available in stock", None))

self.btn\_Great.setText(\_translate("MainWindow", ">", None))

self.btn\_Reset.setText(\_translate("MainWindow", "Reset", None))

self.btn\_Load.setText(\_translate("MainWindow", "Load", None))

item = self.tableWidget\_Cart.horizontalHeaderItem(0)

item.setText(\_translate("MainWindow", "Name", None))

item = self.tableWidget\_Cart.horizontalHeaderItem(1)

item.setText(\_translate("MainWindow", "Quantity", None))

item = self.tableWidget\_Cart.horizontalHeaderItem(2)

item.setText(\_translate("MainWindow", "Price", None))

self.label\_Total.setText(\_translate("MainWindow", "Total amount in cart: P 0.00", None))

self.btn\_Search.setText(\_translate("MainWindow", "Search", None))

self.btn\_Less.setText(\_translate("MainWindow", "<", None))

self.btn\_Purchase.setText(\_translate("MainWindow", "Purchase", None))

item = self.tableWidget\_Stock.horizontalHeaderItem(0)

item.setText(\_translate("MainWindow", "Name", None))

item = self.tableWidget\_Stock.horizontalHeaderItem(1)

item.setText(\_translate("MainWindow", "Number available", None))

item = self.tableWidget\_Stock.horizontalHeaderItem(2)

item.setText(\_translate("MainWindow", "Price per piece", None))

self.label\_4.setText(\_translate("MainWindow", " ", None))

self.label\_cart.setText(\_translate("MainWindow", "Cart", None))

self.menuUser.setTitle(\_translate("MainWindow", "User", None))

self.menuHelp.setTitle(\_translate("MainWindow", "Help", None))

self.actionAbout.setText(\_translate("MainWindow", "About", None))

self.actionHelp.setText(\_translate("MainWindow", "Help", None))

self.actionAccount\_info.setText(\_translate("MainWindow", "Account info", None))

import sys

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtGui.QApplication(sys.argv)

MainWindow = QtGui.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

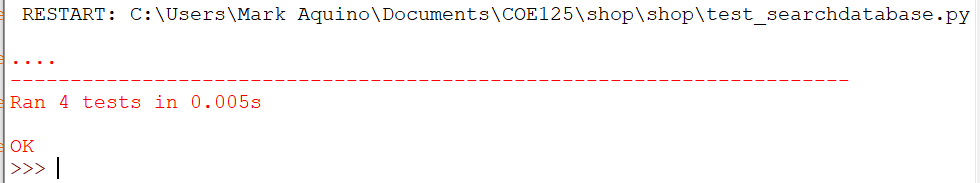
MainWindow.show()

sys.exit(app.exec\_())

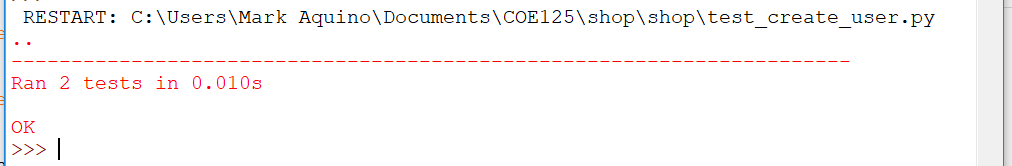
**TESTING**

1. **UNIT TESTING**
2. **DATA ACCESS**

* test\_searchdarabase.py

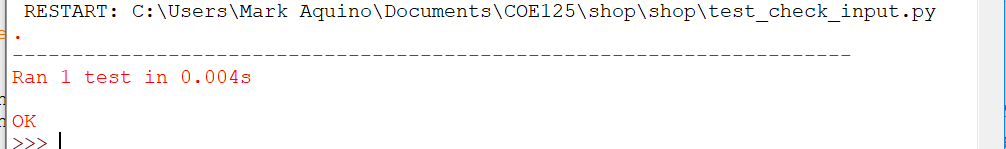


* test\_create\_user.py

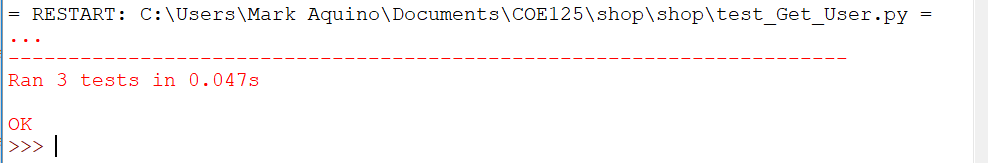


1. **BUSINESS LOGIC**

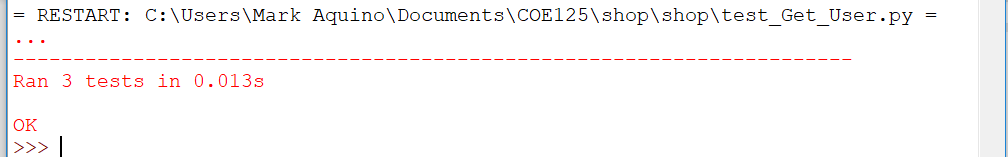
* test\_check\_input.py



* test\_Get\_User.py

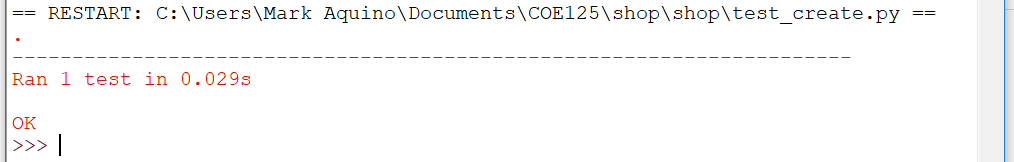


* test\_mainButtonControl.py

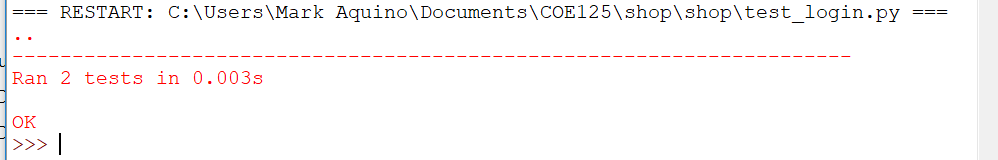


1. **USER INTERFACE**

* test\_create.py



* test\_login.py



* test\_main.py

