Wasim Ahmed (2002) King’s School (26220)

7517 Computing Project:

Restaurant POS with Stock Control

# Table of Contents

[1 Table of Contents 1](#_Toc101708109)

[1 Analysis 2](#_Toc101708110)

[1.1 Background 2](#_Toc101708111)

[1.2 Investigation 3](#_Toc101708112)

[1.2.1 Primary meeting 3](#_Toc101708113)

[1.2.2 Secondary Meeting 4](#_Toc101708114)

[1.2.3 Existing Menu: Imported into program 4](#_Toc101708115)

[1.2.4 Existing method of Taking Orders 5](#_Toc101708116)

[1.3 Program Requirements 5](#_Toc101708117)

[1.4 Analysis Data Dictionary 6](#_Toc101708118)

[1.4.1 Users 6](#_Toc101708119)

[1.4.2 Ingredients 6](#_Toc101708120)

[1.4.3 Menu 6](#_Toc101708121)

[1.4.4 Suppliers 7](#_Toc101708122)

[1.4.5 Tickets / Receipt 7](#_Toc101708123)

[1.4.6 Stock Replenishment 7](#_Toc101708124)

[1.5 Entity Relationship Diagram 8](#_Toc101708125)

[1.6 Data Flow Diagram – Level 1 9](#_Toc101708126)

[1.7 Prototype Designs 10](#_Toc101708127)

[1.7.1 Example Menu Form 10](#_Toc101708128)

[1.7.2 Example Menu Form – Client Feedback 10](#_Toc101708129)

[1.7.3 Example Take Order Form 11](#_Toc101708130)

[1.7.4 Example Take Order Form – Client Feedback 12](#_Toc101708131)

[1.8 Use of Language 12](#_Toc101708132)

[2 Design 13](#_Toc101708133)

[2.1 Design Overview 13](#_Toc101708134)

[2.1.1 Top-Level Structure Chart 13](#_Toc101708135)

[2.1.2 System Flow Chart 14](#_Toc101708136)

[2.2 Key Algorithms 15](#_Toc101708137)

[2.2.1 Menu Form – SAVE (New record) 15](#_Toc101708138)

[2.2.2 Menu Form – SAVE (Update existing record) 15](#_Toc101708139)

[2.2.3 Take Order Form – SAVE AN ORDER / CREATE TICKET 17](#_Toc101708140)

[2.2.4 Suppliers Form – FIND(Search supplier contacts) 19](#_Toc101708141)

[2.2.5 Kitchen View Form – REFRESH (Using Bubble Sort) 20](#_Toc101708142)

[2.3 Forms / Functions 20](#_Toc101708143)

[2.4 Data Structures 21](#_Toc101708144)

[2.4.1 Ingredient ID Array 21](#_Toc101708145)

[2.4.2 Meal ID Array 21](#_Toc101708146)

[2.5 Data Requirements 21](#_Toc101708147)

[2.6 File organisation and processing 22](#_Toc101708148)

[2.7 Table/Record Structures 22](#_Toc101708149)

[2.7.1 Menu Table 22](#_Toc101708150)

[2.7.2 Ingredients Table 22](#_Toc101708151)

[2.7.3 Menu-Ingredients Table 23](#_Toc101708152)

[2.7.4 Suppliers Table 23](#_Toc101708153)

[2.8 Data Model 24](#_Toc101708154)

[2.9 SQL Commands 24](#_Toc101708155)

[2.10 Classes 25](#_Toc101708156)

[2.11 User Interface Design (HCI) 27](#_Toc101708157)

[3 Technical Solution 28](#_Toc101708158)

[4 Testing 28](#_Toc101708159)

[4.1 Introduction 28](#_Toc101708160)

[4.2 System Testing – Normal Data 28](#_Toc101708161)

[4.2.1 Test 1 Evidence 30](#_Toc101708162)

[4.2.2 Test 2 Evidence 31](#_Toc101708163)

[4.2.3 Test 3 Evidence 31](#_Toc101708164)

[4.2.4 Test 4 Evidence 31](#_Toc101708165)

[4.2.5 Test 5 Evidence 31](#_Toc101708166)

[4.2.6 Test 6 Evidence 32](#_Toc101708167)

[4.2.7 Test 7 Evidence 32](#_Toc101708168)

[4.2.8 Test 8 Evidence 34](#_Toc101708169)

[4.2.9 Test 9 Evidence 35](#_Toc101708170)

[4.2.10 Test 10 Evidence 35](#_Toc101708171)

[4.3 System Testing - Erroneous Data 36](#_Toc101708172)

[4.3.1 Test 1 Evidence 37](#_Toc101708173)

[4.3.2 Test 2 Evidence 37](#_Toc101708174)

[4.3.3 Test 3 Evidence 38](#_Toc101708175)

[4.3.4 Test 4 Evidence 38](#_Toc101708176)

[4.3.5 Test 5 Evidence 39](#_Toc101708177)

[4.4 System Testing – Extreme Data 39](#_Toc101708178)

[4.4.1 Test 1 Evidence 40](#_Toc101708179)

[4.4.2 Test 2 Evidence 41](#_Toc101708180)

[4.4.3 Test 3 Evidence 43](#_Toc101708181)

[5 Evaluation 44](#_Toc101708182)

[5.1 Overview 44](#_Toc101708183)

[5.2 Review of Objectives 44](#_Toc101708184)

[5.3 User Feedback 46](#_Toc101708185)

[5.4 Improvements 47](#_Toc101708186)

# Analysis

## Background

|  |  |
| --- | --- |
| Logo, company name  Description automatically generated |  |

Alaturka is a new start-up Turkish restaurant based in the heart of a busy Nottingham city centre. They can seat 115 customers serving traditional Turkish cuisine 7 days a week as well as providing a takeaway service. The restaurant is run with 10 waiters and 8 kitchen staff including 2 delivery drivers at weekends.

Currently waiters are taking orders using somewhat outdated pen and pad method. This introduces potential errors in the order due to eligible handwriting; During busy periods kitchen staff are sometimes left making guesses as to what was ordered because the handwriting on the order was not clear. It has also been identified that during busy periods some orders are misplaced when the waiter’s attention has been shifted to the immediate need of other customers. These are typical scenarios which are affecting the reputation of this new start-up restaurant.

I have been approached by the owner of the business to create a system that will streamline the ordering process from the customer’s table through to the kitchen / bar with the ability to track and report stock level records.

## Investigation

### Primary meeting

As part of my investigation, I conducted a meeting with the owner that took place at the establishment midday on Friday 1st Jan 2021. The purpose of this interview was to capture the customers’ requirements and identify any possible shortfalls in these. The customer requirements will then be used as the basis to create the detailed program requirements that will help develop the desired system.

#### Issue 1

The main issue raised in the interview was the process of taking an order from the customer’s table to the kitchen / bar. This could impose longer waiting times on the customer due to the kitchen staff misreading the order written down by the waiters and then proceeding to prepare the wrong dish which the customer did not order. This error could compel the customer in writing a bad review on leading social media sites or dedicated food review sites. Bad publicity like this could end up affecting the reputation of the business and ultimately affecting sales.

Our proposed solution to this issue was to introduce handheld devices and counter Point of Sales systems for the waiters to use in recording customers’ orders which are then submitted to the kitchen. The tickets are automatically printed off in the kitchen leaving no room for staff to misread the order. Plus, there is no error in orders being misplaced before reaching the kitchen, improving the efficiency of the order process.

#### Issue 2

Another issue raised was the current system of tracking and reporting stock levels and to know if there were enough ingredients to prepare the ordered dish. Kitchen staff are aware of the stock levels and understand how many meals they can prepare; however, the waiters are not updated regularly with this information. This will result in the waiter taking an order from a customer through to the kitchen staff who will only then inform the waiter that they have run out of the required ingredients to prepare that meal. The waiter will then have to return to the customer to ask them to choose another meal because their initial chosen dish has sold out. Again, the customer could form a negative impression of the establishment as no one likes having to choose an alternative.

To counter this problem, we would validate the order before committing it to the kitchen, this way the customer can be informed straight away if their chosen dish has sold out and an alternative can be chosen. To validate the order a real-time stock level system would be created. As orders are committed, the stock levels are automatically adjusted from the ingredients lists for each of the dishes ordered. If there aren’t sufficient ingredients the program will automatically add the ingredient to a re-ordering text file which is checked by the manager who then commits the order at the end of the week.

### Secondary Meeting

A second meeting was carried out to clarify other aspects of the program specifically the way a waiter takes an order from a customer and processes the receipt. The meeting took place on Friday 18th Jan 2021.

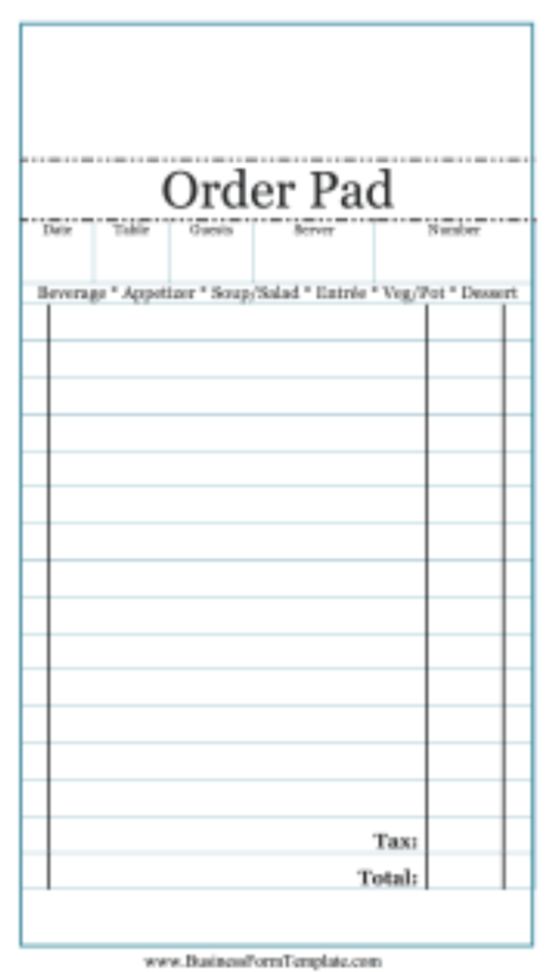
We initially discussed how the program will need to store the customer’s order so that a receipt could be printed. The information will be stored in a text file and will consist of the order number for that day, food / drink item from the menu and the waiter’s Name.

### Existing Menu: Imported into program

Text

Description automatically generated

### Existing method of Taking Orders



## Program Requirements

|  |  |
| --- | --- |
| **#** | **Requirements** |
| 1 | Each user must have a secure login which the program uses to identify the user’s role throughout the program respective of their job title. |
| 2 | The program must store details of the manager and staff, each with a unique ID, position in the business, access rights and other relevant fields depending on the user. |
| 3 | The program must apply adequate access control to different type of users of the system. For example, admin users should be able to access all areas, waiters should only be able to take orders, kitchen staff should only be able to update level of ingredients available and head chef should only have access to editing the menu. |
| 4 | Waiter must be alerted if an order cannot be fulfilled before committing it to the kitchen. The program will display a message if stock is below a threshold, in which the customer will be informed to choose another meal item. |
| 5 | The program should set a stock re-ordering threshold for each ingredient, if an item falls below this the system should automatically add the item to a re-ordering text file. |
| 6 | The program must display the stock re-ordering text file for the admin to make any changes before committing the order to replenish stock. |
| 7 | The program should display information about the individual suppliers that the business deals with and should only be accessed by the manager. |
| 8 | The program must create a ticket of each order taken which is sent through to the kitchen / bar. |
| 9 | The program must provide a receipt for each order taken and save a similar copy for the restaurant. |
| 10 | The program must track ordered meals and display this on a graph for the manager to see the most selling meal item on the menu. |
| 11 | The ‘Kitchen View’ window should have a function to click each order that has been completed and remove it from the view of the Chef, simulating the use of a written ticket. |
| 12 | In the ‘Supplier’ window the admin user should have the ability to search through each supplier by name. |

## Analysis Data Dictionary

### Users

|  |  |  |
| --- | --- | --- |
| Attribute Name | Purpose | Example |
| RoleID | A unique number to identify staff by their roles. | 01 = Manager  02 = Waiter |
| FirstName | Stores first name of worker | John |
| LastName | Stores last name of worker | Smith |
| Username | A unique string to identify each worker | JSmith007 |
| Password | A unique hash of the password | asd84bfo238sud94 = password1234 |

### Ingredients

|  |  |  |
| --- | --- | --- |
| Attribute Name | Purpose | Example |
| IngredientID | A unique number to identify each ingredient | 29 = Potato |
| Name | Stores the name of the ingredient | Potato |
| Quantity | Stores the number of units of ingredients available | 15 |
| Cost Price | Stores the cost price of the ingredient | £3.50 |
| MinOrderlvl | Stores a minimum amount level of each ingredient that must be in stock | 35 |

### Menu

|  |  |  |
| --- | --- | --- |
| Attribute Name | Purpose | Example |
| MenuID | A unique number to identify each menu option | 09 |
| Type | Stores the type of meal / beverage | S = Starter  B = Beverage |
| Name | Stores the name of the meal / beverage | Lamb Shish / Topical Juice |
| Description | Stores a description of the meal / beverage | 8pc lamb shish served with salad and Turkish rice / Mixed fruit juice |
| MealPrice | Stores the price of the meal / beverage | £4.50 / £1.25 |

The ingredients are linked to each menu option through a link table to allow for each meal having more than one ingredient assigned to it.

### Suppliers

|  |  |  |
| --- | --- | --- |
| Attribute Name | Purpose | Example |
| SupplierID | A unique number to identify each supplier | 002 |
| Name | Stores the name of the supplier | Kofte & Sons |
| Address | Stores the address of the supplier | 127 Johnsons road |
| Phone Number | Stores the phone number as a method of contact | +441234567890 |

### Tickets / Receipt

|  |  |  |
| --- | --- | --- |
| Attribute Name | Purpose | Example |
| OrderID | A unique number to identify each order | 05 |
| Quantity | Stores how many times the meal was ordered. | 3 |
| Name | Stores the name of the meal | Lamb Shish |
| Waiter’s Name | Stores name of the waiter | Jeff Ross |
| Date/Time | Stores the Date / Time of order | 04/11/2021 – 14:38 |

This data will be stored in a text file, so that a receipt can be given to both the restaurant and the customer, the order can be sent to the kitchen / bar and so that the data can be used to generate a graph which shows popular food items which the manager can only view.

### Stock Replenishment

|  |  |  |
| --- | --- | --- |
| Attribute Name | Purpose | Example |
| Ingredient Name | Name of the ingredient stored only once so that it can be collectively seen by the manager who then processes a re-order. | Tomatoes  Garlic  Vegetable Oil |

## Entity Relationship Diagram

![Diagram

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAqACoAAD/4RDaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAFAAAISodpAAQAAAABAAAIUJydAAEAAAAKAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFJhamEAAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAMwMQAAkpIAAgAAAAMwMQAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMjowMjoyNyAyMjo0MDo0NAAyMDIyOjAyOjI3IDIyOjQwOjQ0AAAAUgBhAGoAYQAAAP/hCxdodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTAyLTI3VDIyOjQwOjQ0LjAxMjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5SYWphPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIA5MEkQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APfqKKKokKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKK5/WPFsGkayulrpeqahdNbi422FuJcJuK5IyD1Hp3FVv8AhNpP+hS8Uf8Agt/+yqeeKN1h6sldI6miuW/4TaT/AKFLxR/4Lf8A7Kj/AITaT/oUvFH/AILf/sqXPEf1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqRfHcK3lpBd6Br1iLu4S3jlu7IRpvY4AyW/H8DRzxD6tV7HVUUUVZzhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN2n/JaD/2L/wD7cV29cRaf8loP/Yv/APtxXb1nHqdNb7PogoooqjAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuN+I3+p8Nf9jBa/+z12Vcb8Rv8AU+Gv+xgtf/Z6mWx0Yf8Aio36KKK0OUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5u0/5LQf8AsX//AG4rt64i0/5LQf8AsX//AG4rt6zj1Omt9n0QVwfja/kPi3SNG1HWrrQtGvYJCbu1k8lp7gMu2HzcHZ8uTxgnpmu8rlvF3ibwrphGleMgiWt1HuU3Vq0kEnJBXdggMMZwcdRim7XVzFbMoeINJl8J/DPxJJp+taxczfZJJY57y+aWSFgnGx+q+vXrXUQyyHwzHKXYyGzDF8852dc+tec6Dpra34R8ZaR4aa6k8PXEJi0Y3W8De0Z3pGX58oNtAJ4zuxV63+KOh/8ACJrYv9pXX0tPIbR/sshuPPCY2bcevfpilO/LJdbL9QjZOL6Xf6GdJd6hqXwv8ArJq2owzalfWsN1dW926TSKyPnLg5PQdfStmC1u/B3xA0PTbXXdU1Oy1lZ1mttSuDcNEY03CRHPzKOxGcHNZOveHpbL4f8AgDQb2ae2nTU7OGaS2l2SRt5b7trjoQc8iu40PwbpWg3r30Jur3UJI/Ka+1C5e4m2ddoZicDPYYrV25pPzf5IzXwxXkvzZxug+HLjxPJ4hvbrxT4is5YNau4IRbak6xRIjfKAhyuBnp0xRB4m1XVPBfgfUJ7yRbm61yK3uZYCYhcoplUkgcFW2BsdPameFPBlj4hn8Sz6lfaqYG168jksYb+SK3lXePvIpGc559a0/iPJp3h3S/Ckj+VY6dY65bE7VwkUapJ2HYCoi7KF/wC7+hctXO3979TrPE80lv4S1eaCRopY7KZkdGIZSEJBBHQ1xGuT3t74C8Cxf2pqFtJqV5ZQ3NxbXTxzSK8LFsuDk5PPPete+8beHPE/hbX7bQNVhvZotNnkdIw3yrsIzyB3Nct4t/sz/hVfgL+3/L/sz7XY/avMzt8vyGznHNJb694/mxvbTtL8kd9ofhOPQLqW4j1nW78vGU8vUNQedF5zkK3Q8da880LRbq5+FKeK5/GPiK2v47aW58x9SZ4QyM2A0b5BX5RkHrXS+Brr4cR6lcW3gOSzF5NDulS335ZFPU7vQn9a5Hwv8OrLVPhXpmtaRCP7ejDXEQuZGmt53WRvkeJyUwQMZABBwQeKHezfp8twjudJ4h8SajJ4S8J3Oo3s2g2mreWNVvoF2tbbotwUMQfLDPgbj09RXUeHPDUOiySXNtrer6nHcRqFF/ftcoB1DLnpnPUVlN8RPDdx4UsdQ11WgstQDQzrNbtLHBKuA8MuAdpzkcjBxWX4Bk04+M9QXwO07eFntA8gKutvHdb+kO4d1yWC8dKv7bt5/wBf5evzI+yr/wBa/wBfcekUUUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVxvxG/1Phr/ALGC1/8AZ67KuN+I3+p8Nf8AYwWv/s9TLY6MP/FRv0UUVocoUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBzdp/yWg/9i//AO3FdvXEWn/JaD/2L/8A7cV29Zx6nTW+z6IKKKKowCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv8A7PXZVxvxG/1Phr/sYLX/ANnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAObtP+S0H/sX/AP24qxfaNoVtePFH4c0YquMFrFM9AfSq9p/yWg/9i/8A+3FaWrf8hSb/AID/AOgiueWz9T1aWs4r+6jO/s3RP+hb0X/wAT/Cj+zdE/6FvRf/AAAT/CpaKzudnKiL+zdE/wChb0X/AMAE/wAK881D4gaZZX+pQxfDnSbiHT52hluFgXauHKgt+7IXJHGTXpFeE+LfDVwniDxNexTxvDYzxzS7sqx887gFHIOC2Dkjpn2q4Wb1OXEc0IpwPXPDVzoniHw9bap/wiei2/n7/wB19jR9u1yvXAz0z0rV/s3RP+hb0X/wAT/CsXwRZ/YPBtjbbblNnmcXUHkyDMjHlMnHX16YNb9S3qbwjeKb3Iv7N0T/AKFvRf8AwAT/AAo/s3RP+hb0X/wAT/CsW31PUtXvtQj0yWztIrGc25FxC0zyMBktgOu0c8dc9eOlasd9Gt1DY3Mi/bZITKURTtIGAxB9MkdeaNR+7/X3Ev8AZuif9C3ov/gAn+FH9m6J/wBC3ov/AIAJ/hVWfWtPtlvmmuNo08A3J2MfLBGR2549M0yHxBps93FbxztumJELtC6xykc4RyNrfgTRqHuLr+Jd/s3RP+hb0X/wAT/Cj+zdE/6FvRf/AAAT/Cs+XxJpcM08RmlkltnKTJDbyStGcA5IVSQMHr096h1DVmNzoT6dcq9tfXJVmQBhInluw57cgdKFdifIuv4mt/Zuif8AQt6L/wCACf4Uf2bon/Qt6L/4AJ/hVKzvF+1ao02pLLFbygMjRCMWo2AkFv4hzuz2zjtS2evadf3CQW8snmSIXjEsDxiVR1KFlAccjpmjUfu/0y5/Zuif9C3ov/gAn+FH9m6J/wBC3ov/AIAJ/hVCLxBYXbyxWMrzyRl1bbBIVRlyCGbGF5B6kZ7VlReIdSu9F8PzwfZYbnVX2yM8TOifIzcKHB/h9aFd/wBdxNwX4/h/w50n9m6J/wBC3ov/AIAJ/hR/Zuif9C3ov/gAn+FY8up6jpF9ax6x9luLS7lWFLm2jaIxSHOAyFmyDwNwPB7d63aBpJ6EX9m6J/0Lei/+ACf4Uf2bon/Qt6L/AOACf4VLRSuVyoi/s3RP+hb0X/wAT/Cj+zdE/wChb0X/AMAE/wAKloouHKiL+zdE/wChb0X/AMAE/wAKP7N0T/oW9F/8AE/wqWii4cqIv7N0T/oW9F/8AE/wo/s3RP8AoW9F/wDABP8ACpaKLhyoi/s3RP8AoW9F/wDABP8ACj+zdE/6FvRf/ABP8KloouHKiL+zdE/6FvRf/ABP8Kb48s7awsPC9tY28VtAniC22RQoEVc7ycAcDkk1PSfEb/U+Gv8AsYLX/wBnq1sznqK1SCXn+Rv0UUV0njBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN2n/JaD/2L/8A7cVpat/yFJv+A/8AoIrNtP8AktB/7F//ANuK19Ttp5NSlaOGRlOMFUJB4Fc8tn6nq0mlON/5TOoqb7Hc/wDPvL/3waPsdz/z7y/98Gs7M7OaPchrzzxJoep3Nx4qht7OSR9Y+x/Yyoyr+UAX3N0THbeRntmvSfsdz/z7y/8AfBo+x3P/AD7y/wDfBpq6ImoTVm/62MTw1Dd2/h62i1FLlLld+9bq5FxIPnOMyAANxj6DA7Vq1N9juf8An3l/74NH2O5/595f++DRqUpRStc4rXZNBkuLq6hvzpmuQAorRN5c8rDBVTGf9apwuBg5HApTfPZ65omp+IDHZefpzwyu52okxKNtJPC8BsZPbFdp9juf+feX/vg0fY7n/n3l/wC+DTV1/X9dyGot3v8A1dP9Dze+vIr/AEvxvcW+WiZIwjY4cCIDcPUHqD3HNaep6hY6xZ6VpulOst2LqCQwx/ftVRgWLr1TA+XnHJxXT61oNzq+i3en7JYftEZTzPJLbc98cZq6tlcqgX7PKcDH+rNPt8vwJsnpfe/4nFaTrGmaVrXiL+0bmK0Zr3cHmO0SARrwpP3iPQc8j1qtY20tta+GhNC8HmapNKkLjDRoyysqkduCOK7DTdAudPuNQl2Syfbbkz48kjZ8qrj3+7196v8A2O5/595f++DSWiXy/ALJ3u+r/Fnn+pWtzeab4uhsld5TeRtsjGWdRHEWUDuSAeO/StGwOl6rqVhNF4ln1Ka3JmhgzBlMoQS4SMMowcc45wOtdf8AY7n/AJ95f++DR9juf+feX/vg0K9rf1tYfLG979/zucp4WRU8O35VQC95dsxHc+Yw/kBWJZXMFp4Z8FT3c0cEKTAtJK4VV/dSdSeK9G+x3P8Az7y/98Gj7Hc/8+8v/fBoV1+H4A1Frfv+Nv8AI5PVbqDxLJZWGkSrdxpdRz3FzC26OJUO7G8cFicDaOecniumqb7Hc/8APvL/AN8Gj7Hc/wDPvL/3waXSxacb3bIaKm+x3P8Az7y/98Gj7Hc/8+8v/fBpWZXNHuQ0VN9juf8An3l/74NH2O5/595f++DRZhzR7kNFTfY7n/n3l/74NH2O5/595f8Avg0WYc0e5DRU32O5/wCfeX/vg0fY7n/n3l/74NFmHNHuQ0VN9juf+feX/vg0fY7n/n3l/wC+DRZhzR7kNJ8Rv9T4a/7GC1/9nqf7Hc/8+8v/AHwag+I3+p8Nf9jBa/8As9Wtmc9Rp1YW8zfooorpPGCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAObtP8AktB/7F//ANuK7euItP8AktB/7F//ANuK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFcZY+Nrn/hPr/RNTt4YrAXAtbO6QEEzeWr7HJOMsGO3GPukc1tRaxO/jS70l1iFtBYxXKvg7tzO6nJzjGFHajt5/5XDa/l/wAMbNFY9p4q0m8vYLSKaeOW4BMH2i0lhWbAydjOoV+OeCeOelRXXjTQbOa7jmu5C1k+y68q1lkFucA5cqpCrg/eOAcHng4AN2isa38WaLdahDaW97vedmSGQQv5MzAZKpLt2MRg8BieD6U7UPFOkaZdS291cyGSBQ0/k28kqwA8gyMikRjHPzEcc9KANeimpIksayRMrowDKynIIPQg06gAooooAKKKKACiiigAooooAKKKKACiiigAooooAK434jf6nw1/2MFr/wCz12Vcb8Rv9T4a/wCxgtf/AGeplsdGH/io36KKK0OUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5u0/5LQf+xf/APbiu3riLT/ktB/7F/8A9uK7es49TprfZ9EFFFFUYBRRRQAUUUUAcHYaNb6/qHjbT7ssqyahGUkThonEKFXU9iDg1S0TWdUPinX5LqxaXWNN0eOKSFBxcSo0hVk9nBUj647V6TRSttbt+lrj/wA/1ueWz6xHq+oeFJz4h/tS4l1KKWW1tYoxBakxvwcAujdgHck4Y4442tPRTYePCVGWvJwTjqPsyf413FFDV013v+Nv8gTs0+1vwv8A5nnd4ixeBfAYjG0Le6fjHb5KgsWg07UNfsfEHiufQ5pr+WYRSG2RbiGT7jq0sRLcDbwTjbjivS6Kpu7b73/G3+RK0SXa34X/AMyjoljBpug2NlZvM9vbwJHE04w5UDjcMDBx7D6Veooobu7glZWCiiikMKKKKACiiigAooooAKKKKACiiigAooooAK434jf6nw1/2MFr/wCz12Vcb8Rv9T4a/wCxgtf/AGeplsdGH/io36KKK0OUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5u0/5LQf+xf/APbiu3riLT/ktB/7F/8A9uK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFc14s8SXmiSW0WmW8dxIEe7uldSdttHjeVwR83zDHXoeDW1dapY2OltqV3cxxWaoJDMx4wemPXORgdTmjpcOti3RXJnxWt7430fTdOnmjilhuHube4tXhc7Quxtsihsctgjg4PpTvEnjTTrDRtXFleut3aQyos62zvDFOEJVDLtMYbOBtJ6kDGTik3ZXGld2OqorCXxRYWNlpqardN9tvLRZkjSB3eXhd21UU5OWHA59sA1YvPEumWLwxzyTmaaLzlt4rSWWZU/vNGillGeMsBzx1qmrMlO6uatFZc3iTSYbC1vPtfnRXn/AB7C3jaZ5uMnaiAs2BycDjvimyeJtLisYLp5pgtw7RwxC1lMzspIYCLbvyMHPy8YzSGa1FV7C+g1KzW5tDIY2LAeZE0bAgkEFWAIIIPUVYoAKKKKACiiigAooooAKKKKACiiigAooooAK434jf6nw1/2MFr/AOz12Vcb8Rv9T4a/7GC1/wDZ6mWx0Yf+KjfooorQ5QooooAKKKKACiiigAooooAKKKKACiiigAooooAKKK5LxVBPqPjDw1pUepX9hBefavOaynMTNsjVl56dR3Hc0pOyuaUqftJct7b/AIK51tFYH/CuY/8Aoa/FH/gx/wDsaP8AhXMf/Q1+KP8AwY//AGNTzPsaezpfz/gb9FYH/CuY/wDoa/FH/gx/+xo/4VzH/wBDX4o/8GP/ANjRzPsHs6X8/wCBv0Vgf8K5j/6GvxR/4Mf/ALGj/hXMf/Q1+KP/AAY//Y0cz7B7Ol/P+Bv0Vgf8K5j/AOhr8Uf+DH/7Gj/hXMf/AENfij/wY/8A2NHM+wezpfz/AIG/RWB/wrmP/oa/FH/gx/8AsaP+Fcx/9DX4o/8ABj/9jRzPsHs6X8/4G/RWB/wrmP8A6GvxR/4Mf/saP+Fcx/8AQ1+KP/Bj/wDY0cz7B7Ol/P8Agb9FYH/CuY/+hr8Uf+DH/wCxo/4VzH/0Nfij/wAGP/2NHM+wezpfz/gb9FYH/CuY/wDoa/FH/gx/+xo/4VzH/wBDX4o/8GP/ANjRzPsHs6X8/wCBv0Vgf8K5j/6GvxR/4Mf/ALGj/hXMf/Q1+KP/AAY//Y0cz7B7Ol/P+Bv0Vgf8K5j/AOhr8Uf+DH/7Gj/hXMf/AENfij/wY/8A2NHM+wezpfz/AIG/RWB/wrmP/oa/FH/gx/8AsaP+Fcx/9DX4o/8ABj/9jRzPsHs6X8/4G/RWB/wrmP8A6GvxR/4Mf/saP+Fcx/8AQ1+KP/Bj/wDY0cz7B7Ol/P8Agb9FYH/CuY/+hr8Uf+DH/wCxo/4VzH/0Nfij/wAGP/2NHM+wezpfz/gb9FYH/CuY/wDoa/FH/gx/+xo/4VzH/wBDX4o/8GP/ANjRzPsHs6X8/wCBv0Vgf8K5j/6GvxR/4Mf/ALGj/hXMf/Q1+KP/AAY//Y0cz7B7Ol/P+Bv0VwXjbwrL4b8H3uq2PifxFJPb+XsWbUCVO6RVOQAD0Y9672nGV3Zk1KSjFSi7p3/C3+YUUUVRiFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVy3ie41eTxNoGkaNqn9m/wBofafMl+zpN/q0DDhvxHUda6muZ1f/AJKb4O/7ff8A0SKiex0Ya3tNV0f5Ml/4RPxd/wBD1/5SIv8AGj/hE/F3/Q9f+UiL/Gu0opcq/psf1ifZf+Ax/wAji/8AhE/F3/Q9f+UiL/Gj/hE/F3/Q9f8AlIi/xrtKKOVf02H1ifZf+Ax/yOL/AOET8Xf9D1/5SIv8aP8AhE/F3/Q9f+UiL/Gu0oo5V/TYfWJ9l/4DH/I4v/hE/F3/AEPX/lIi/wAaP+ET8Xf9D1/5SIv8a7SijlX9Nh9Yn2X/AIDH/I4v/hE/F3/Q9f8AlIi/xo/4RPxd/wBD1/5SIv8AGu0oo5V/TYfWJ9l/4DH/ACOL/wCET8Xf9D1/5SIv8aP+ET8Xf9D1/wCUiL/Gu0oo5V/TYfWJ9l/4DH/I4v8A4RPxd/0PX/lIi/xo/wCET8Xf9D1/5SIv8a7SijlX9Nh9Yn2X/gMf8ji/+ET8Xf8AQ9f+UiL/ABo/4RPxd/0PX/lIi/xrtKKOVf02H1ifZf8AgMf8ji/+ET8Xf9D1/wCUiL/Gj/hE/F3/AEPX/lIi/wAa7SijlX9Nh9Yn2X/gMf8AI4v/AIRPxd/0PX/lIi/xo/4RPxd/0PX/AJSIv8a7SijlX9Nh9Yn2X/gMf8ji/wDhE/F3/Q9f+UiL/Gj/AIRPxd/0PX/lIi/xrtKKOVf02H1ifZf+Ax/yOL/4RPxd/wBD1/5SIv8AGj/hE/F3/Q9f+UiL/Gu0oo5V/TYfWJ9l/wCAx/yOL/4RPxd/0PX/AJSIv8aP+ET8Xf8AQ9f+UiL/ABrtKKOVf02H1ifZf+Ax/wAji/8AhE/F3/Q9f+UiL/Gj/hE/F3/Q9f8AlIi/xrtKKOVf02H1ifZf+Ax/yOL/AOET8Xf9D1/5SIv8aP8AhE/F3/Q9f+UiL/Gu0oo5V/TYfWJ9l/4DH/I8z8V2Pi7wx4Zu9X/4TH7V9n2fuv7LiTdudV684+9np2ruawvit/yTLVf+2P8A6OSt2iOkmh1Zc9KMmle72SXRdgooorQ5QooooAKKKKACiiigAooooAKKKKACiiigDm7T/ktB/wCxf/8Abiu3riLT/ktB/wCxf/8Abiu3rOPU6a32fRBRRRVGAUUUUAFFFFAHDaXBqviDXtZ13T7uxhtJm/s+AXVm1xviiJDMpWVMAuX7HoKxxLPZ+H7Cw1QtMPC2sxrfFEJ/0cKximK8kKAyHvjaTnivUaKFpb5fg7/5/eD1v8/yt/l9xwtzrOn6z8TNA/si4juljtLsG6gIePJCfKHHBIxkgHjIz1rHj1rTdN+EF/od46rq9vZ3EE+nsMzmQhiX2dSpzv34xjnPWvUqKTV48v8AXX/Madnc4i0jR/GfhRnUEx6JMyk9j+5GfyJ/Oqlwy6X8QNafWPEc+gx3qQyWkx+zrHOiptZA8sbDKsc7QR9/OOa9Coqm7u/r+LuSlZW9PwSPP7jTtA07SNLe18WNpdyjXNxYajdNFiXzGzJlWVUZCWBAGMjBBxzTZNSi1jRNJ1TXNbg8NazGbj7FeEKkUyZ2lgkvDIy7GxkHoQcV6FRSGYvhPVb3WfDkN7qcKRTu7gGNWVJVDELIoYkhWABGfWtqiigAooooAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv/ALPXZVxvxG/1Phr/ALGC1/8AZ6mWx0Yf+KjfooorQ5QooooAKKKKACiiigAooooAKKKKACiiigAooooAK5nV/wDkpvg7/t9/9EiumrmdX/5Kb4O/7ff/AESKie33fmdGG+N+kv8A0lnc0UUUzEKKzda1220JbE3aTP8AbbyOzj8oA4d84JyRxxz1PtWdqPjnSNK8bWPhe+8+K8vohLDMVHk5JYBC2chiVOOMdOaFqGx0dFZmua7baBDaS3iTOt3eRWaeUASHkbapOSOM9f5VD4g8U6d4cWFbzz57q5JFvZWkRmnnIGTtQc4HcnAHrQBs0VzeieMk1jVBYT6FrekzMheM6jZhEkA6gOrMM+xOazZPiZB9qvorTwx4jvorC5ktpri0s0kj3ocNjEmT+WfagDtqK56+8baTZ+Cf+EqhM19pxVGT7MoLvucIAAxHIJwQSMYNR6R4wl1bU47N/C3iDTw4J+0XttGkS4GeSJCeenSi2tg6XOlorim+JULXl9DZeF/Ed+ljcyWss9paRyRl0OGx+8yfyz7Vv6B4l0zxLZyz6XMzGBzFPBLG0csDjqrowBU0bq4PR2NaiuYk8e6ZH4DbxZ9mvHslbb5KovnE+b5XA3Y+979KdpHjCXVtTjs38LeINPDgn7Re20aRLgZ5IkJ56dKA2OlornPD/jnSPEev6to1l58V7pUrRSxzqF8wKxUumCcrkY7Hkcc1euPENtb+JF0Qw3El21i96uxQVKKwUr1zuJYYGMe9HbzDuuxq0VwknxSji1KGwk8HeKVu50aSKE2cW51XG4j970GR+dat545s7GxsXm0vVjf3ys8Okx2u+62qcMWUHaoGRyWA5o6XA6aisnQNebXoJpH0fVNKMLhPL1GBY2fjOVwzAj8a1qACiiigAooooAKKKKACiiigAooooA4/4rf8ky1X/tj/AOjkrdrC+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyiFFFFWcwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXM6v/AMlN8Hf9vv8A6JFdNXM6v/yU3wd/2+/+iRUT2+78zow3xv0l/wCks7miiimYmXr/AIk0nwxZw3Wu3YtIJplgRyjMC7AkD5QcdDyeK1K4L4r6bb6xYeHdNvU3293rkMMi+qtHIDUekeLZtE+G18NTzPq+gO2nSR85uJhhYSM8neGjOfc0ls32/wCB/n+I2tUu/wDwf8jrtP8AEelarq2o6ZYXYmvNMZVu4wjDyiwJAyRg9D0Jxjmufi+LXgyfb5Op3EgY4UrptyQecdfLxWN8NtFfQPGHiOwuJPOuhZWMt1KTnzJnEryN+LMarfDLxfLaeB9G04eF9fuFG5PtkFtGYDmRvm3GQHAzzx2NUlrb0/UV93/W1zv5/FGjWvia28P3F8seqXUPnQW7I37xOeQ2NuflPGc8dKtalq9lo8cEmozeStxcR20R2M26RzhV4Bxk9zxXnfi3w2PE/wAUrm0im+zX0OgRXFjdAcwTpcsUYe3Y+xNJrPiQ+JPCehSXUP2XUrTxHZW2oWh6wTLKNw/3T1B7gilHW3m/w5rf18glo36fja56LHq1lLrU+kxzZvoIUnki2N8qMSFOcYOSp4zniiXVrKDWbfSpZtt7dRPNDFsY7kQgMc4wMbh1PeuZ0/8A5LRrX/YHtf8A0ZLRqv8AyWPw7/2DL3/0KKha8vnf8L/5A9L+Vvxt/mdlRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/AGx/9HJW7WF8Vv8AkmWq/wDbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAKKKKACiiigAooooAKKKKACiiigDm7T/ktB/7F/8A9uK7euItP+S0H/sX/wD24rt6zj1Omt9n0QUUUVRgFFFFABRRRQAUVyGqal4nsPEmlaZHf6Sy6m0wWRtNlzEEXd08/wCbPTtW3c6xBo1tbJrV0sl3NlVS0tpGaYjklIl3vgDr1x3NHS4dTUorl9d8ShvDdvqGg3f39Qt7d2MeGXMyo6MrjKnBIIIBFa2o+INO0u5W2upZWuGQyCC2t5J5AmcbisasQueMkYzR0uBpUVkz+KdFtrSxupdQjEF/kWsigsJSFLYGB1wOh5J468UW/iGx1KK8jsJZVubaPe8NxbyQSICDhtkig4464xSbsnfoNK7RrUVyHhjxvYXfh3R31S9d7u7ijSS4+zOIWmYfdMir5YbPG3I54xW7qGv6dpl0lrcySvcunmCC2t5LiQJnG4pGrELnjJGM1TVnYlO6NKismTxRosWhprMmoRrp7sqCcg4VmbaAwxlSCcHOMc5xinWHiPTNSvmsraaVbkR+aIp7eSEumcbk3qN657rkcj1pDNSisaXxbosF49vJdtmOUQSTLBI0MchONjShditkgYLAgkCtmjzAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv8A7PXZVxvxG/1Phr/sYLX/ANnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKKKKACiiigArmdX/5Kb4O/wC33/0SK6auZ1f/AJKb4O/7ff8A0SKie33fmdGG+N+kv/SWdzRRRTMTjfiL/qvDX/YwWn82rJ8R+HrbxR8UL/SrxmjWbw5GY5k+9DILlijr7ggGuv8AEfh/+3100fafs/2HUIb3/V7t/lk/L1GM568/ShfD+3xy/iL7T97ThY/Z/L6YkL792ffGMfjRHz7v/wBJt+Y2+3Zf+lX/ACOA1TX7rVvDuk6fraCLXNJ8R2FtfoOjnzAVlX/ZdfmH4jtXSaCqXHxc8VT3QDXNrb2kFsW6pAyF2A9i+c/SpfFXw/g8SeI9I1mG9awuLCeOScLHvF2kbh1RuRggg4bnG48Vb8Q+ETq2qQaxpOpz6NrNvEYUu4UWRZIyc7JI24dc8joQeQaE9Lve7/JK/wCH3NitrZdl+bZWj8W6vD44s9B1Xw9HaW9+Zza3i34kLrEM5MYQbcgjjPGe9cdo/ivWfDdr4qms/DMt9Yxa7ePJf/alWOH5xksgDSFVHJKqeK7HTPB+qjxHaa14l8SPq1xYLItrFFZR20aeYuHJALMxOB3GMVp+HfDo0GHU42uBdLqGoT3pBj2hBIc7Opzj14z6UW/J/mv0Hdfj+jOE8Q6SND/Z7e2t7uPUWYw3HnxNiOVpLlJDtPZctx7V2eiat4ovdQ8rW/C8Gl2uwnz01RZzu7DaEHX1zWVN8O5D4I1Pwva6x5NlcXImsd1tuNknmCTy/vDeu4HHTAPfFaGl6L4utdThm1Xxdb39opPmW6aQsJkGP74kOOcHpTVr/wBdkTrb+v61Knw4/wCPfxL/ANjFe/8AoQqOBEj+N96bTgS6HG14F6FxKRGW99u78BUUPgfxPp11qR0TxnHYWt/fS3hi/shJWjaRskBmfnHHb8K6Hw74XtfDsVy8c895e3r+Zd31026WdgMDOMAADgKAABU9E+yt+Fhv7S7v9bnmkpZf2ZyY13MJwVUnGT9u6Zr0PRNW8UXuoeVrfheDS7XYT56aos53dhtCDr65rOk+H7N8MW8Ix6psYvvW8NtnB8/zfubvw+97+1XNL0Xxda6nDNqvi63v7RSfMt00hYTIMf3xIcc4PSqW7/rogexw9hot5Na6z4i8Px79b0XxHfyQxZwLqIsPMgP+8OnowHrW9pes2niH4p6PqunOXtrrw5K6ZGCP36ZUjsQcg+4rqPDfh/8A4R+PUl+0/aPt2ozX2fL2bPMIOzqc4x14z6Vk6T8P4NG+Il54lsr1lt7qB4xp3l/LE7srO6tnjJTJXHUk5pR0cb9F+PLb8/61CWvN5v8ADmuM1X/ksfh3/sGXv/oUVWfE3h/Vptcs/EXhe5t49UtIXt3t7tSYbmFiGKEjlGyoIYfjxWhd+H/tXjLTte+07fsNrNb+R5ed/mFDndnjGzpg5zUGv6Jrt9fRXXh/xM+kFY/Lkhks0uYpOSQ20kFW56g88Ulol8/zf6MfV/L8kO8MeJzr63lteafNpmqafII7yzlYNsJGVZWHDIR0PHQ8VvVgeGvDD6HNfXt/qUuq6pqDIbm7kjWMEIMIiovCqATxzySc1v1TJCiiikMKKKKACiiigAooooAKKKKAOP8Ait/yTLVf+2P/AKOSt2sL4rf8ky1X/tj/AOjkrdpR+J/L9Tef8CPrL8ohRRRVnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/8AJTfB3/b7/wCiRXTVzOr/APJTfB3/AG+/+iRUT2+78zow3xv0l/6SzuaKKKZiYHirQrrXJdDa0eFBp+qxXsvmkjciBgQuAefmHXA96zNV8BjUfiLY6+J1WxjCy3dof+W88QIgfp/Dvbv/AArwe3ZUULT8/wBAev5fqc/p2g3Vn4413WZZITb6jBbRxKpO9TGHDbhjH8QxgmuZ8N6N8RfDHh600e0i8LzQ2oZVklubjcwLFucR4716NRQtAOei0K8HxEbxBI8H2dtJWyKKx3+YJS5OMY24Prn2rE8XeALzV/Fml6zodzBbAXlvNqkMpIFwsLhkYYB+cDcvOMggZ4rvKKFo0+3+dw7+f+Vjk/EPh3Wv+Emg8R+Erqzjv1tvslzbX4bybiLduU7k5VlJPODnP5u0Lw/rLeJX8ReK7mye+W1NpbW1greTBGWDMdz/ADMxIHYAAV1VFC0/ruD1CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDj/AIrf8ky1X/tj/wCjkrdrC+K3/JMtV/7Y/wDo5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN2n/ACWg/wDYv/8AtxXb1xFp/wAloP8A2L//ALcV29Zx6nTW+z6IKKKKowCiiigAooooA5TxF/yULwj/AL13/wCiqz/EebH4i29/qOsz6NYT6cbeG8QReWsocs0bNKjKu5cEdM7Mc4ru6KOwf1+p51qlnp8PheS70vVZtVW91y0llupChV3EsSnaY1VSMKBkDqDzkGn3DLpfxA1p9Y8Rz6DHepDJaTH7Osc6Km1kDyxsMqxztBH3845r0Kij+vwS/QP6/Fv9Tza/i0rQm8HyQ3M15p41K4uPtMyg53pIxk+VQNgJLZAxjnpzWmbu213xvJqOjSpdWdlpMsE95AwaJ5HZWWMMOGKgEn03D1ro9S0b+0NY0m+8/wAv+zZnl2bM+ZujZMZzx97PfpWnSauvv/FWHezv6fnc82uI0HwAscKBttLVxjjDeYhz+da1vfWfh3x3r0uv3UVimoiCS0ublxHHIiJtKBycblbJ29cNmuzoqm7yb7kpWVjzC7gaTw1qN40TJZaj4kt57VJE274zLEpfaezEEj1Bz3rpteTPxA8LsvyuUvF3e3lqf5gV1NFLpb+tkv0H1v8A1u3+p5TolvYP4Sbw94j8V3lhdBpLe70s/ZlkdmYnKBojI+/cGBBJO7g5r1SNPLjVASQoAyepp1FO4dQooopAFFFFABRRRQAUUUUAFFFFABXG/Eb/AFPhr/sYLX/2euyrjfiN/qfDX/YwWv8A7PUy2OjD/wAVG/RRRWhyhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/8lN8Hf9vv/okV01czq/8AyU3wd/2+/wDokVE9vu/M6MN8b9Jf+ks7misqa48QLPILfTNNeIMQjPqMisy54JAgODjtk/U0z7T4k/6BOlf+DST/AOR6Lk+zfl96Niisf7T4k/6BOlf+DST/AOR6PtPiT/oE6V/4NJP/AJHoug9nLy+9GxRWP9p8Sf8AQJ0r/wAGkn/yPR9p8Sf9AnSv/BpJ/wDI9F0Hs5eX3o2KKx/tPiT/AKBOlf8Ag0k/+R6PtPiT/oE6V/4NJP8A5Houg9nLy+9GxRWP9p8Sf9AnSv8AwaSf/I9H2nxJ/wBAnSv/AAaSf/I9F0Hs5eX3o2KKx/tPiT/oE6V/4NJP/kej7T4k/wCgTpX/AINJP/kei6D2cvL70bFFY/2nxJ/0CdK/8Gkn/wAj0fafEn/QJ0r/AMGkn/yPRdB7OXl96Niisf7T4k/6BOlf+DST/wCR6PtPiT/oE6V/4NJP/kei6D2cvL70bFFY/wBp8Sf9AnSv/BpJ/wDI9H2nxJ/0CdK/8Gkn/wAj0XQezl5fejYorH+0+JP+gTpX/g0k/wDkej7T4k/6BOlf+DST/wCR6LoPZy8vvRsUVj/afEn/AECdK/8ABpJ/8j0fafEn/QJ0r/waSf8AyPRdB7OXl96Niisf7T4k/wCgTpX/AINJP/kej7T4k/6BOlf+DST/AOR6LoPZy8vvRsUVj/afEn/QJ0r/AMGkn/yPR9p8Sf8AQJ0r/wAGkn/yPRdB7OXl96Niisf7T4k/6BOlf+DST/5Ho+0+JP8AoE6V/wCDST/5Houg9nLy+9GxRWP9p8Sf9AnSv/BpJ/8AI9H2nxJ/0CdK/wDBpJ/8j0XQezl5fejH+K3/ACTLVf8Atj/6OSt2uZ+JD30nwr1c6lb29vLuhAW3naVSvnR85KLznPGPxrpqUfify/UuorUIrzl+UQooorQ5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuZ1f/kpvg7/t9/8ARIrpq5nV/wDkpvg7/t9/9Eiont935nRhvjfpL/0lnSTavexTyInh3UplViBIklsFcA9RmYHB9wD7Uz+2r/8A6FjVf+/tr/8AH62KKLeZPMv5V+P+Zj/21f8A/Qsar/39tf8A4/R/bV//ANCxqv8A39tf/j9bFFFn3Dnj/Kvx/wAzH/tq/wD+hY1X/v7a/wDx+j+2r/8A6FjVf+/tr/8AH68S+JfiTVLH4ma3ax+ItXsLeKKM28NnM2wyeTGQpG9QqkkksMn2Oa7/AOCOp3+reCbufVL65vZl1B0WS5maRgvlxnGWJ4yTx70WfcOeP8q/H/M67+2r/wD6FjVf+/tr/wDH6P7av/8AoWNV/wC/tr/8frYoos+4c8f5V+P+Zj/21f8A/Qsar/39tf8A4/R/bV//ANCxqv8A39tf/j9bFFFn3Dnj/Kvx/wAzH/tq/wD+hY1X/v7a/wDx+j+2r/8A6FjVf+/tr/8AH62KKLPuHPH+Vfj/AJmP/bV//wBCxqv/AH9tf/j9H9tX/wD0LGq/9/bX/wCP1sUUWfcOeP8AKvx/zMf+2r//AKFjVf8Av7a//H6P7av/APoWNV/7+2v/AMfrYoos+4c8f5V+P+Zj/wBtX/8A0LGq/wDf21/+P0f21f8A/Qsar/39tf8A4/WxRRZ9w54/yr8f8zH/ALav/wDoWNV/7+2v/wAfo/tq/wD+hY1X/v7a/wDx+tiiiz7hzx/lX4/5mP8A21f/APQsar/39tf/AI/R/bV//wBCxqv/AH9tf/j9bFFFn3Dnj/Kvx/zMf+2r/wD6FjVf+/tr/wDH6P7av/8AoWNV/wC/tr/8frYoos+4c8f5V+P+Zj/21f8A/Qsar/39tf8A4/R/bV//ANCxqv8A39tf/j9bFFFn3Dnj/Kvx/wAzH/tq/wD+hY1X/v7a/wDx+j+2r/8A6FjVf+/tr/8AH62KKLPuHPH+Vfj/AJmP/bV//wBCxqv/AH9tf/j9H9tX/wD0LGq/9/bX/wCP1sUUWfcOeP8AKvx/zOG+JF1Nd/CvV3uLG4sWDQgR3DRliPOj5+RmGPxzx0rpqwvit/yTLVf+2P8A6OSt2lH4n8v1Lqa0I+svyiFFUNd1NtG0C91JLZ7o2kLS+SjBSwUZPJ6ADk9TgHAJwC/R7/8AtXQ7HUfL8r7XbRz+Xu3bNyhsZ4zjPWtDmLlFFFABRRRQAUUUUAFFFFABRRRQAUUUUAc3af8AJaD/ANi//wC3FdvXEWn/ACWg/wDYv/8AtxXb1nHqdNb7PogoooqjAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuN+I3+p8Nf9jBa/8As9dlXG/Eb/U+Gv8AsYLX/wBnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKKKKACudTXr4/EiTQJIbdbIaZ9sjkUsZGbzAnPQAdeMHoDnnAueJdZk0DQZr+3snv51ZEitY2IaVmYLgYBOeSenauA/4TTUv7U/tL/hVN39v/wCfryW837u37/k56cdenFAHqtczq/8AyU3wd/2+/wDokVV8L+MtZ17Vzaaj4Qv9JhERf7TOW25BHHzIvXPbJ9sZItav/wAlN8Hf9vv/AKJFRPb7vzOjD/G/SX/pLO5ooopmIUUUUAfM3xXe1T4ra99shmlzFEIvKmEeyTyI8M2Vbco7qME+or0n4Bf8iFe/9hOT/wBFRVB49srW98e619stobjyPBk80XmxhvLkErYdc9GGeCOa6L4Tf8iFbf8AIX/g/wCQp/1yT/U/9Mf7v40AdrUS3du129qs8RuEQO8Icb1U8AleoBwefapa46N2j+J+uuhwy6NAQfQ75KTdvx/Jsdr/AIfnY7GivO4tX8R23w+03xVc6w1xMUgknsxbxLDJG7BT/DvD4bOQwGR93HFbf2rU9e8VanY2epTaXZaUscZa3ijd55nUOSTIrAKowMAZJJ56VTVnYlO6udTRXnbeIvEUXgm7uDewyapDrn2FZRCqxlfPVMbfTB+vvnmtLXdWvdCuNL0f+1L2efUJJJJb1bETyxRIoyI44o8ZJIALK2BnOeKXS/8AWyf6j62/rqv0Oyqre6la6e9st5L5bXc4t4RtJ3yEEgcDjhTyeK4w+LNU0vT9ZbZeX8cAgFhdajYPamSSVtmxxsTcFbByqjg4680uv6dqtlqXhdr3WpNRjbVohKs0EabX8t8FNijC9eG3HpzwcnVeqDo/mdjaapZ311eW1rN5ktlIIrhdpGxioYDJHPBHTNW64m98XXulWHiu7lIuDp96lvZxtHwpeOPaDtGWG58nqccDtS6bq+px69psCXWr6pBdFkvPtejPbJbnaWDo3lJgZG3axY/MOeORa2+X5XB6X+f4Ha1G1xAlxHbvNGs0oZo4ywDOBjJA6nGRn6iuDtr7xJeeEtW1sa2yTWM92ba3S2i8uRIpGwsmVychcZUrxjqeaddR3Or+OPDV7Fql5Z/bNMmmVIlhIiyIiVG6M8HPOc9OMULVr+ujf6A9P67Ox2ranYIl073tuq2f/HyxlUCDjPz8/Lwc89qq2XiXQtSulttO1rTrudgSsUF2jsQOvAOa4XUf+QN8S/8AfP8A6TJXX69pdpqvguWG9RSI7bzYpP4oZFTKyKezD1FJu0eZ9k/vQ7XdvN/gb1FZPha9uNS8I6Ve3uftE9pFJKSMZYqMn8eta1VJWdiU7q4UUUUhhRRRQAUUUUAFFFFAHH/Fb/kmWq/9sf8A0clbtYXxW/5Jlqv/AGx/9HJW7Sj8T+X6m8/4EfWX5RCiiirOYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArmdX/wCSm+Dv+33/ANEiumrmdX/5Kb4O/wC33/0SKie33fmdGG+N+kv/AElnc0UUUzEKKKKAPHviHoOmal8QNZe8s0doPCct8jKSh89HZVc7SNxAAHORgAdhXS/B02p8BR/Y7+G8/ejzfKshbeTJ5UeY2wB5jDvIcls9eK19d8Gxaxq82ox3jwTXWntpdypQOptmYs2zkFZMnhjuA/umr3hnw7F4X0gadb39/ewq2Y2vphI0a7QoRcAAKAvA7ZNAGvXmmvvLpnxUuPEEJby9Os7VbtQeDbyPIrtj/ZIVv+AmvS6whoDS+KNVvbwQy2N/YxWpiJJY7TJuyMYwQ47+tGt00HRr+t0cx8Vbp9R0O50e0YGKOyk1C8ZT0jQful/4E+D9ENaN/wCLU0yfTNGi1DTNOlexW4lutSfCRrgKqqm5d7MQf4hgDPOQKitfAt5beCdb0t7yO51DUYmt47iRjtWJU2QqSBnhRk8HknrWjPoOpWes2OsaMbSW5jshY3VtcyNGkqD5lZXVWIYNnqpyCemKFp6afk/+Agev4/p/wTLbx5eP4XurvT10+/vbTU4rEvDIfs9wHdAGUgnblX9WwQetbEWravYeKrHStZNncRajFK8EtrC8RiePBKNudtwIPB+XoeKXWNK1bXdCiguhZW90t9BcFI5XdFSORWI3lQWJCn+FRzjtk2tS0ie88UaLqUTxiGwE/mqxO5t6BRjjHUc5Iprz/rRfqJ/197/Qz7TxLeXHw/1DXXjgF1bJdsiBTsPlO4XIznnYM8+vSqZuHvPiF4XuZQoebSLiRgvQE+UTj86jn8MeIovDeq+HdOk04Wd21wYLuSVxJGkpLFGjCEHliu7d0OcEjB1IfDt3H4i0K/MkPladp0lrKAx3M7BMFeOnyHrjtSjvf+tn/wAActrL+tVYx/CPhbQdW0K8m1HR7K4nk1C8Vp3gXzf9e4GHxuBHYg5HatbwXd3LR6tpl5cSXR0u/e2inmbc7x7Vddx7kBsZ74qDRtN8TaJYz2NtbaS6SXc8yXMl5KSokkZwTEIhkjd03jOOo61t6DoyaJp7Q+c9zPNK09zcOMGaVvvNjsOwHYAChbfL8dAe/wA/8zSooooAKKKKACiiigAooooAKKKKACiiigAooooA4/4rf8ky1X/tj/6OStTUr6LS9Ku7+4V2itYXmcIAWKqpY4zjnArL+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyieW6j8YvCWq6dNY32n6w9vOuyRUCoWXuMrKDg9CM8jIPBq54X+KHhV5rDQdMttStYyfJgNyA6p6KW3s2Ow7DjoBx6NWV4mhefwtqSxDMq27vH/vqNy/qBVnMatFRwTJcW0c8ZykiB1PsRmpKACiiigAooooAKKKKACiiigAooooA5u0/5LQf+xf/APbiu3riLT/ktB/7F/8A9uK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVxvxG/wBT4a/7GC1/9nrsq434jf6nw1/2MFr/AOz1Mtjow/8AFRv0UUVocoUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjan/pPiLR7QciNpbxx7IuwZ/4FKD/wGtmsex/0nxXqlx1W1jitF9mwZG/MSR/lWxQAVzOr/wDJTfB3/b7/AOiRXTVzOr/8lN8Hf9vv/okVE9vu/M6MN8b9Jf8ApLO5ooopmIUUUUAYmreFNO1i/a9n86K4ltjZTvE/+vtiSWhYEEBSTyy4f0YVPoHhvSvC9g9loVr9lt5JTKyeY75cgAnLEnoo/KtSigArG/4R/wD4qa/1f7T/AMflklp5Xl/c2ljuznn73THatmik0mO9jnJPCXmfD+Hwz9tx5UEUX2nyuuwqc7c99vrWdq1xYaL4pvbq28U6fo11dQobu21GMOsmBhJIwXQ7sAg4LDgcZFdpRTbbdxKyVjgvDnhee+8E/Z7ie4gM2rHUEe6i/esizh13r8uCwXPQY3dO1dLruhNqs9je2d39i1HT5C9vOY/MXDDDo6ZG5WHoQc4OeK2KKPJf10/QPP8Ar+tTDutButY0W80/xBqCTi5ChWs7fyBDtOQyhmc7twByTjgcdc0pvC2raje6Tc6xr0c50u4WdEgsvKWYhSCXy7fNzwRgDn5TkY6mijrcOljnZfB9td22vW19O8kOsziYiNdjQkIijB5yQUDA4/Crmm2WuQSRDVdYtruKNMEQ2JheU4xl2MjD3woXnHbg61FGwGFZ+Gvsnhe/0f7Xv+2Nct53l42eczN0zzjd6847VFL4Ynj/ALFm0+/jhvNJtzbCSa3MiSxlVVgUDqQcqpB3ce9dFRRt/XYN/wCu5zNz4O+0WXiaD7dt/t453eTnyP3YTpu+bpntT5vD2q6lZjT9b1iCTTyoWWGxs2gaZR/AztI/ynuFAJ9QMg9HRR/X3B/X3jY40ijWOJQiIAqqowAB0Ap1FFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/AGx/9HJW7WF8Vv8AkmWq/wDbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuZ1f8A5Kb4O/7ff/RIrpq5nV/+Sm+Dv+33/wBEiont935nRhvjfpL/ANJZ3NFFFMxCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4/4rf8ky1X/tj/AOjkrdrC+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyiFIyh1KsMqwwQe4paKs5jI8KsT4WsInOXt4/szE92iJjP6rWvWPoH7qbV7T/AJ4ag5A9pFWXP5yGtigAooooAKKKKACiiigAooooAKKKKAObtP8AktB/7F//ANuK7euItP8AktB/7F//ANuK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVxvxG/1Phr/sYLX/ANnrsq434jf6nw1/2MFr/wCz1Mtjow/8VG/RRRWhyhRRRQAUUUUAFFFFABRRRQAUUUUAFFFZniK5e08OX0kHExiMcP8A10b5U/8AHiKAIfC/73RjenrfzyXQPqrMdn/jmwfhWzUNpbR2VlBawjEcEaxoPYDA/lU1ABXM6v8A8lN8Hf8Ab7/6JFdNXM6v/wAlN8Hf9vv/AKJFRPb7vzOjDfG/SX/pLO5ooopmIUUUUAFFFFABVcX9mdQNgLqA3gj802/mDzAmcbtvXGeM1YrzbXdKu774mahe6OwTVdN062uLXJwJDvkDRN/suuR9cHtSvqv66B0f9dT0RrqBLqO2eeNZ5VLRxFwGcDGSB1IGRn61LXE2+rwa74z8MajahlWawvN0bfejYGIMjehBBB+lVdd8TS6YzXumeIbrUmivESW2jsVezCtIFMZmSP5WUMOsmcjkc4p9l3/zsHmegUVybzazqnjjVtMt9YksLK1tbeRPIgiaQSPv7urDb8vIIzwMEc5wk17xIngFfFV1qqmW3lCPZR28YhmRZfLYsSC4ZuWyrADgY65FqD0/r5npNFcTqHiSe88U6hpsV9qWn22npGpfTtLe6klldd3zN5UiqoBHGASc84Fb3hbUL7UvD0E+rQSQ3YZ438yBoTIFYgPsbldwAbB9aFqrgbFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/AGx/9HJW7WF8Vv8AkmWq/wDbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuZ1f8A5Kb4O/7ff/RIrpq5nV/+Sm+Dv+33/wBEiont935nRhvjfpL/ANJZ3NFFFMxCiiigAooooAK4Px94f0ZpdGuW0ixM9zrVuk8ptk3SqSchjjLA9wa7ysTxNo1xrKaWLV4k+x6lDdyeYSMohOQMA880faXqvzDo/R/kV3uI/DfiHSdKsbCztNJ1ASoPs8Pl7LgAMo4wuGUN2zkdazNR8Y6pDbX93p9rbTwRavDp1qjhgZcsqyEtnH3iVBxxtOQa0PiAxh8JyXUDhL21nilsuMlpw42IB33Z2/QmsjxLpEuleAtF0y3nVbmPUbNTOybgZTMCzkZGcsScZH1oW6v3S+9r/hvuB7fJ/k/+B+JsNq+r6T4g0+z1trK4tdSLxxS2sLxGCRVL7W3O24FQeRt5HTmuel+JgbTJ9Zg1TQltonJTS5Js3c0Stjdu3ja5GSE2HsCeeOiXSdW1XXrC+15LK3t9ODvDb2szTebKyld7MyJtAUnCgHljzxUWi6T4g8O2I0ixTTbmwhkYW1zPcSJLFExzhowhDlckD51yAOlC/r+v608w/r+v6/A6eNxJGrr91gCPpTqKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4/4rf8AJMtV/wC2P/o5K3awvit/yTLVf+2P/o5K3aUfify/U3n/AAI+svyiFFFFWcxj237jxlqEfRbi0gmX3ZWdW/TZWxWPf/uPFmkT9BNFPan3JCyD9Im/M1sUAFFFFABRRRQAUUUUAFFFFABRRRQBzdp/yWg/9i//AO3FdvXEWn/JaD/2L/8A7cV29Zx6nTW+z6IKKKKowCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv8A7PXZVxvxG/1Phr/sYLX/ANnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKxtc/0i+0ixHIlvBNIP8AZiBcH/vsR/nWzWNH/pXjSVuq2NkqD/elfLD8BEn/AH1QBs0UUUAFczq//JTfB3/b7/6JFdNXM6v/AMlN8Hf9vv8A6JFRPb7vzOjDfG/SX/pLO5ooopmIUUUUAFFFFABWTBofk+LbvW/tG77Tax23k7Pu7GY7t2ec7umO1a1FHW4dLHMxeDIbfxyfENtdtHG8UgksgnymR9oaQHPBIVcjHJGaoSeBL9/D0WgrrypplvKrwKtn+9IWQOqSOXww7cKpyAfUHtaKFpawbmZaaN9l8S6jq3n7vtsMMXlbMbPL385zznf6DGKypPBm/wAAy+Gvt+PMZm+0+T0zL5n3d34dfeuoooDrcwJ/D15Br0+raFqMNnLdxIl3Fc2pnjlKcI4AdCrAZHUgjHHFbNpHPFaRpdzi4nA+eUIEDH2XnA7AZJx1JPNTUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/bH/ANHJW7WF8Vv+SZar/wBsf/RyVu0o/E/l+pvP+BH1l+UQoooqzmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5nV/8Akpvg7/t9/wDRIrpq5nV/+Sm+Dv8At9/9Eiont935nRhvjfpL/wBJZ3NFFFMxCiiigAooooAKKKKAK02nWVxew3k9nby3VuCIZ3iUvHnrtYjI/CpJ7aC6VVuoI5lR1kUSIGCspyGGe4PINS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHH/Fb/AJJlqv8A2x/9HJW7WF8Vv+SZar/2x/8ARyVu0o/E/l+pvP8AgR9ZflEKKKKs5jG8R/uodPu/+fbUIDn0Dt5RP5SGtmsrxNC8/hbUliGZVt3eP/fUbl/UCtGCZLi2jnjOUkQOp9iMigCSiiigAooooAKKKKACiiigAooooA5J9RsdM+MPnaleW9nE2g7BJcSrGpb7RnGSeuAfyrqP+Eu8N/8AQw6V/wCBsf8A8VVa90XS9RmE2oabZ3cqrtDzwK7AdcZI6cn86r/8Ir4e/wCgFpn/AIBx/wCFZ8sk9DqdSlJLmvdK3Q0f+Eu8N/8AQw6V/wCBsf8A8VR/wl3hv/oYdK/8DY//AIqs7/hFfD3/AEAtM/8AAOP/AAo/4RXw9/0AtM/8A4/8KdpCvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVR/wAJd4b/AOhh0r/wNj/+KrO/4RXw9/0AtM/8A4/8KP8AhFfD3/QC0z/wDj/wotIL0PP8DR/4S7w3/wBDDpX/AIGx/wDxVH/CXeG/+hh0r/wNj/8Aiqzv+EV8Pf8AQC0z/wAA4/8ACj/hFfD3/QC0z/wDj/wotIL0PP8AA0f+Eu8N/wDQw6V/4Gx//FUf8Jd4b/6GHSv/AANj/wDiqzv+EV8Pf9ALTP8AwDj/AMKP+EV8Pf8AQC0z/wAA4/8ACi0gvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVR/wAJd4b/AOhh0r/wNj/+KrO/4RXw9/0AtM/8A4/8KP8AhFfD3/QC0z/wDj/wotIL0PP8DR/4S7w3/wBDDpX/AIGx/wDxVH/CXeG/+hh0r/wNj/8Aiqzv+EV8Pf8AQC0z/wAA4/8ACj/hFfD3/QC0z/wDj/wotIL0PP8AA0f+Eu8N/wDQw6V/4Gx//FUf8Jd4b/6GHSv/AANj/wDiqzv+EV8Pf9ALTP8AwDj/AMKP+EV8Pf8AQC0z/wAA4/8ACi0gvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVR/wAJd4b/AOhh0r/wNj/+KrO/4RXw9/0AtM/8A4/8KP8AhFfD3/QC0z/wDj/wotIL0PP8DR/4S7w3/wBDDpX/AIGx/wDxVH/CXeG/+hh0r/wNj/8Aiqzv+EV8Pf8AQC0z/wAA4/8ACj/hFfD3/QC0z/wDj/wotIL0PP8AA0f+Eu8N/wDQw6V/4Gx//FUf8Jd4b/6GHSv/AANj/wDiqzv+EV8Pf9ALTP8AwDj/AMKP+EV8Pf8AQC0z/wAA4/8ACi0gvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVXLeONd0jUz4ch03VLK8lXXrVzHb3CSMF+YZwD0yR+dbH/AAivh7/oBaZ/4Bx/4U+Lw3ocEyTQaLp8csbBkdLVAykcgggcGk4yasXCpRhLmV/wNOiiitDjCiiigAooooAKKKKACiiigAooooAKx/D379dRvz/y930hX/djxEv4ER5/Gruq3o03R7y9Iz9ngeXHrhScUzRrI6bodlZscvBAiOf7zAcn8Tk0AXqKKKACuZ1f/kpvg7/t9/8ARIrpq5nV/wDkpvg7/t9/9Eiont935nRhvjfpL/0lnc0UUUzEKKKKACiiigAqve6hZ6Zam51K7gtIAQDLPII1BPQZJxViuT0yJdU+I2s3V8vmNpKw29kjjiEOm53Udi2cZ64XFHUDotP1Sw1aBptLvra9iVtjSW0yyKG64ypPPIq1XKeIol0/xj4e1KzHl3F3ctZXIQAefEY2YbvXaVBHpz61FY67qM3wy1PVpLjN9Al6Y5di/KY3kCcYxwFHbnHNJtKNxpO6Xf8A4P8AkdhRXLX2qD+zdKa51+4sLi4tlkMFjbJNcXDFQSQhjclRznanfqAKxf8AhKdam8AtdW10q6hDqyWC3E1ts8xfPVNzxkAqSDyAARzjHara8v8AW9ib6X/ra56HVK51jT7TTLrUJrqP7LaB/PkjO/y9n3gQuTkY5HWsGSbVtL8Wafpk+rT3ttqsE43yRRK9tIgDAptQDBBPDBug5655/SW1DRfh/wCJtTtdXunmt574xLLHCVV1kb95xGCWOOmdvPSp/wAr/jYpLb1seiLe2ztAgnjD3CGSFC2GkUYyQDycZGfTIqeuG1Szurv4h6BIur3luZbC4YeUkJ8vHlZA3Rnhs85z7YpW8SarY+G9dt7iTztZsr02lo5jUGXzSPIbAGOjjPH8JzVP/P8AOxK1X9drncUVHbRyRWsUc8xnlVAHlIALnHJwMAZ9qkpDCiiigAooooAKKKKACiiigAooooAKKKKAOP8Ait/yTLVf+2P/AKOSt2sL4rf8ky1X/tj/AOjkrdpR+J/L9Tef8CPrL8ohRRRVnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/8AJTfB3/b7/wCiRXTVzOr/APJTfB3/AG+/+iRUT2+78zow3xv0l/6SzuaKKKZiFFFFABRRRQAV5pr0GoR/FG+1fRQ0l5pmm28ptgf+PmEvJ5kf1IAI91Fel1iW2i3EPji+1lniNvcWUNuigneGRnJJGMY+Yd6X2k/62YdH/XVGSdRttW8c+GNQsZPMt7nTrqSNsdj5XX0PtTdf8V3vh28VrvU9ElzcIp0tQVuvKd9oZWMnzEZDEeWBwRnjNPtPBt1p/j1dUs7mJdIEUzLaknfDNKV37OMbSV3YzwSeKzj4M19PCf8Awj1v/ZUUK3KzSXvmOZLvbKHy67BtcgDLbn5XHfIa6f11B9f66G1Nq2u3ni7UtH0r+z4IbO2hmFxcwvISz7vkKh167c5zxjoc8Y8fjPxAfCKeJ57bT4bOKVY57MK7SOBJ5busm4BfmyQpVuB97njp7LSJ7bxdququ8ZgvILeKNQTuBj35yMYx8wxz61iS+D79/hjN4cE1t9skdmDlm8vmfzOu3PT260LT+vMHq/67f5jtV8ZpH4jutJttY0bSvsUaGWfVHyZHYZCIm9DgDBLZPUDHWtnwtro8R+HoNR2xq7M8cgifehZGKkq3dSRkH0Iqk2j6tpfiS91PQ0srqLUUj+0211O8OyRBtDqyo+QV4IIHQHNb9p9pNpGb8RC4Iy6wklFPoCeTjpnAz1wOgFsBNRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcf8Vv+SZar/2x/wDRyVu1hfFb/kmWq/8AbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5hGUMpVhkEYIPesjwox/4RaxiY5a2Q2rE+sTGM/qlbFY2gfurjV7Tp5F+7KPaRVlz+bt+RoA2aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAx/En762s7Adb29ijI9VU+Y4/FY2H41sVjT/6V4ytY+q2Vo8zD0aRgiH8klrZoAKKKKACuZ1f/AJKb4O/7ff8A0SK6auZ1f/kpvg7/ALff/RIqJ7fd+Z0Yb436S/8ASWdzRRRTMQooooAKKKKACsXUNAlk1pdY0e9+wah5Qhl3xebDcIDkB0ypJGTghgRnnI4raooA4t9M8R23jey1PU4otdtVhaKL7Gi232FmwGfZJId+4ADO4kc4HrJc+Cr+TR9S0W01xbfSr55XEf2PdND5hLFRJvAK7iTgrnBIyOCOwopW0sHW5zbeGL221G2v9H1OG3uY7BbGX7TamZJFU5VgBIpU5z3I56cVXh8ENF4ffTH1WSdn1QaibiSEbmIlEm0gEDJxjIx16dq6yiqu73/re/5itpb+trfkZl9o323xBpWp+fs/s/zv3WzPmeYoXrnjGPQ1ky+D7htJ1vSYtSjSw1QzOim1JkgeU5Y7t+GXJJA2g89eK6mikVcxNR0Ce4vtMv8AT72O2vNPR4g00BlSSNwAwKh1IOVUg5/OsMWtpr/xOjvdOuPPtNPgDXjQsGie5Ussa5H8Sq7kjPHy5rt6KOt/6/r9SbaWCiiigYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/bH/ANHJW7WF8Vv+SZar/wBsf/RyVu0o/E/l+pvP+BH1l+UQoooqzmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5nV/8Akpvg7/t9/wDRIrpq5nV/+Sm+Dv8At9/9Eiont935nRhvjfpL/wBJZ3NFFFMxCiiigAooooAKKKKACiiigAooooAKKKihu7e4lmignilkgbZKiOCY2xnDAdDgg80AS0UUjMFUsxAAGSSelAC0VHbXMF5bR3FpNHPBIu5JYnDKw9QRwRUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHH/ABW/5Jlqv/bH/wBHJW7WF8Vv+SZar/2x/wDRyVu0o/E/l+pvP+BH1l+UQoooqzmCsa2/ceMr+P8AhubOGZfdlZ1b9ClbNY1/+48V6RP0E0c9qfckLIP/AEU360AbNFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVHPMltbyTzHEcSF2PoAMmgDL0b/AEjVNYvuoa5FvGf9iJQD/wCPmStisrw1C8Hhuy88YmmTz5R6SSEu3/jzGtWgAooooAK5nV/+Sm+Dv+33/wBEiumrmdX/AOSm+Dv+33/0SKie33fmdGG+N+kv/SWdzRRRTMQooooAKKKKACoby9tdPtXub+5htbePG+WaQIi5OBkngcmpq5PT0Gr/ABF1eTUAJBo6wxWUT8iMum9pQP7x+7u64BHrR1sBv6drWl6v5n9k6lZ33lY8z7NOsmzPTO0nGcH8qu1ieK9V/wCEe8P3eq21tFLebUhi3jG5mfagYjnaC+cfX1rM1KfV/DDabe3WtS6lBPdRWt3BPBEijzCFDx7EDDDY4YtwT35o3YHXVElzBJJNHHPG7wECVVcExkjIDDtwQee1cfBrdzP4suLDVtcm0e6W7Is7FreNYbuAFcFXdCXZskHa4IPbjmPw3YXMXinxU76teTLFcoHjdIQsubdcFsRg8ZwNpHQZzzlX0v5X/IfWx0tr4o0C+uktbHXNNubiQ4SKG7jd278AHJrUryyym/t34R6doGnafe3F/NbxpHI9nLHDAwOfN85lCYXGcqSSeBmumM+s33jy70iLVZLSxtrCCZmhijMhkZnBwXVhg7ecg9BjHNU1rYnpc6K01Szvrq8trWbzJbKQRXC7SNjFQwGSOeCOmat1xk/im+0+z8V3LlbltPvUt7ONwFUF0iCgkDJG98nvWk4v/Dul3Wr6vrVzqK21rJLNb+REkZYDd8m1Qw6EAMzdefWpvZX8v0uVZt2Xc6GivPbPxFrLRaXepc6pfXFzNF9qsRossdskbkBjHIYgw2A53M5BwfXj0Kqs0TcKKKKQwooooAKKKKACiiigAooooAKKKKAOP+K3/JMtV/7Y/wDo5K3awvit/wAky1X/ALY/+jkrdpR+J/L9Tef8CPrL8ohRRRVnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/wDJTfB3/b7/AOiRXTVzOr/8lN8Hf9vv/okVE9vu/M6MN8b9Jf8ApLO5ooopmIUUUUAFFFFABXFnRtM1f4n6sNW060vhHp1qUFzAsmzLy5xuBx0rtK5qfTdbtPGF5q+mW1hdQ3VpDBsuLx4WQoznPETgg7/0o6r+ugdH/XVGatzb+C/FlxZxO8eiy6ZLf/ZslltnjYbvLH8KsG+6OARwBmq1t8QnI067n1PQZkvp44n0y0mD3NuJDhTvDkOVJG4bF788c7sfhd9QbUbrxDMk11qFqbPy4MiO2hOcohPLHJyXIGcDgYxTtFt/Eun2tnp14mmTQWwERvVnk8ySNeAfJ2YDEAA/OQCSeelEfP8ArV/pZfkD8v60X63/AFKun+INR1jxBfWdnd6ZamwuzDJYTxs9xJEpXMuQ42ghvl+Rh0554mi1XW9bvtQGhPYWlpYztbCW7geZriVfvYCumxQeM/MTzwO9fWfD2qeINRs2vLTS7YWd4s0OowzO9wsavuCBTGNpYAA/OR1ODwKmt9J1zQrzURoa2F3aX1w10q3k7xNbyv8AfHyo29SeQPlIyRk9aI9L/wBbf8Eb30/rf/gBNrGt3PjS50PThZQw29nFctczwvIQWZgV2h1znbwcjGD1zVG78RTaZpnjK/tLKyS40yfCMsRHnkRI2ZMHLH5sZ44ArctdIuIfGN/q0jxGC5tIIFVSdwZGckkYxj5xjmub8VaRcad4N8bXMzxlNQJniCEkqojRPmyOuVPTNLo/R/n/AJAtZL1X5f5mnd63rukf2ffaoli1heTxwS28MTiW1MhAU+YWIcBiAflXrx0qh4eXVo/E3iuS5vLKaKO4XzoxZuDIfs67cEynaMYBBDZ56Z4vy6Prms/2ba6w1jHYWc0dxJLbyO0l20ZBQFGUCME4Y/M/QD3qxBo2oWfiDV5oBbS2Oq7ZGd5mWSFxHswFCEMDtU53AjJ4NEtnbzt+H/BFB7X8v1/4Bm6d4kuG8P8Ahiz0qzsre/1i38xEWMrb2qKgZ22A5IGQAuRknqMVoW+t6lZ65caPrP2SaY2bXlrc20bRLIqkKyMjMxBBIOd2CD2xVO38J6hYaN4cezltTquiQeSVkZvJnRlCum7G5egIbBwR05q9Bol/ea1PrGsfZorj7G1nbW1vI0iRqxyzFyqliSBxtAAHfNOe7t5/rb9P+GCGyv5f8H8P6uY0PivxFH4N0/xRfxad9klELXFnDE/mBHYLvWTfgfeDbSp443d67uuTl8K3snwwt/Dglt/tkVvDEzlm8vKMpPOM4+U9q6yqla7t3JjeyuFFFFSUFFFFABRRRQAUUUUAFFFFAHH/ABW/5Jlqv/bH/wBHJW7WF8Vv+SZar/2x/wDRyVu0o/E/l+pvP+BH1l+UQoooqzmCsbxJ+6t7C7/59r+Bs+gdvKJ/KQ1s1l+JoHuPC+pJF/rRbO8f++o3L+oFAGpRUVvOlzaxTx/clQOv0IzUtABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVj+KSX0GS0X719JHacekjhW/JSx/CtisfUf9J8TaRa9VgEt431VfLUH/AL+k/wDAfagDXAwMDgUtFFABRRRQAVzOr/8AJTfB3/b7/wCiRXTVzOr/APJTfB3/AG+/+iRUT2+78zow3xv0l/6SzuaKKKZiFFFFABRRRQAViaj4fll1ldY0e++wah5YhlLxebDcIDkB0ypJGTghgRnuOK26KAMKbQb7VtPvbHxHqMF1b3Maoi2VqbcwsCTvBLuS2dpHYbRx1qIeHdRvZrL/AISDVor63sZVnjjgtDAZZF+60p3sGwecKFGcHoMV0VFAHOaj4c1HWZoodV1S3k0+G8W6SKKyKTfK+5FMnmEYHAJCAkDqM5qzBoVxaeIL++tLyNbXUdrXFvJAWfeqbAyuHGBgLkFT06jNbVFK2lg63M7w9pP9g+HbLS/O8/7JEI/N2bd+O+MnH50230byPFF7rHn7vtVtFB5OzG3YXOc55zv6Y7Vp0VV7u4dLHPSeELW6t9et76ZpYNZnEzBF2NCQiKMHJyQUDA469qnttJ1SWCS013VLfULKS3aB40svKeXcANztvIJxn7qqOfoK2qKXSwdbmBpGi6zpUNrZf23DPp9rhUDWWLho1+6jSb9p4wCdgJA7Hmt+iine4BRRRSAKKKKACiiigAooooAKKKKACiiigDj/AIrf8ky1X/tj/wCjkrdrC+K3/JMtV/7Y/wDo5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczq//ACU3wd/2+/8AokV01czq/wDyU3wd/wBvv/okVE9vu/M6MN8b9Jf+ks7miiimYhRRRQAUUUUAFFFFABRRRQAUUUjMFUsxAAGSSelAC1HcW0F5bvb3cMc8Mg2vFKgZWHoQeDRb3EN3bxz2ssc0MihkkjYMrg9CCOCKkoAAMDA6UVHcXENpbyXF1NHBDGpZ5JGCqgHUkngCnqyugdGDKwyCDkEUALRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/wAVv+SZar/2x/8ARyVu1hfFb/kmWq/9sf8A0clbtKPxP5fqbz/gR9ZflEKKKKs5gpGUMpVhkEYIPelooAx/CjH/AIRaxiY5a2Q2rE+sTGM/qtbFY2g/urrWLTtDfsyj2kRZM/8AfTt+VbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVjaf8A6T4o1a56rbrFZp7EL5jfn5qj/gNbPSsfwv8AvNCS8PW+lku8+qyOWT8kKj8KANiiiigAooooAK5nV/8Akpvg7/t9/wDRIrpq5nV/+Sm+Dv8At9/9Eiont935nRhvjfpL/wBJZ3NFFFMxCiiigAooooAKri/szqBsBdQG8Efmm38weYEzjdt64zxmrFeZ+I45bT4jX3iC1BMuj2VrNIo5L25aUTL9dvzD3UUdUHRs9GvL210+1e5v7mG1t48b5ZpAiLk4GSeByahbVrFNTttPM4NzdRNNCgUkOi4ydwGP4h371wXxKuxr2kXljZSiSysdOfUbp15V2KnyF/m//AVrorbUrqLxNoOnJLi0n0qSWSPaPmdTEFOcZ4DHv3oWrt/Wz/yB6LT+tv8AM6eorm5gs7aS4u5o4IIxueWVwqqPUk8CuEtr7xJeeEtW1sa2yTWM92ba3S2i8uRIpGwsmVychcZUrxjqeaswTx+JvHWmC/iBt7XSY9SggblfOkbG8juVAwD2JJoWv9eTf6A9P680v1Oq07WdL1cSHSdStL4R43/Zp1k2Z6Z2k46Grtcj44jSyuNF1q1UJfw6jBbCReGkikfa8Z9Qc5x2IzS6VdavqfijXVuNUkg0/TLxEhhhijy48pGZWJUnbz2w3J54FGj/AK9P8wen9ev+R095dwWFjPeXb+XBbxtLK+CdqqMk4HJ4HanwTR3NvHPC26OVA6NjGQRkGvPr+bWvEHw11TX/AO1Wto7qznki08QRmFYNrAKxK7y5UZ3BgMkcYGDPL4imjudI0S3ubyyiTS47q4uLGwe6mOQFRFARwo4Yksp7Ad6OrT8v1/yB/wCf6f5ne1FNdQW7xJPPHE0z7IldwDI2Cdoz1OATgelcHd+I/EEPhK6kgaZbuHVIbW1ur2yMBuYndAC0bKuPvFSQB0yMVoa9az2l/wCF0ur+a/kOsBjLMkakfuZOAEVRj65PuaF09Uvvt/mHf0f6/wCR2NV9Qv7bS9Pnvr6Tyra3QySvtLbVHU4GSfwrH8Wale6Mmm6hby7bKO9SO/TYCDC/ybs4yNrFTxjvWD4t1C81DS/GSJKRpthp32dUCDDzlS7tuxn5VKDGccmlfS40tUjvUdZI1dDlWGQfUUtQ2f8Ax4wf9c1/lU1U1Z2Ji7pMKKKKQwooooAKKKKACiiigDj/AIrf8ky1X/tj/wCjkrdrC+K3/JMtV/7Y/wDo5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczq//ACU3wd/2+/8AokV01czq/wDyU3wd/wBvv/okVE9vu/M6MN8b9Jf+ks7miiimYhRRRQAUUUUAFeaa+8umfFS48QQlvL06ztVu1B4NvI8iu2P9khW/4Ca9LrCGgNL4o1W9vBDLY39jFamIkljtMm7IxjBDjv60a3TQdGv63RzHxVun1HQ7nR7RgYo7KTULxlPSNB+6X/gT4P0Q1o3/AItTTJ9M0aLUNM06V7FbiW61J8JGuAqqqbl3sxB/iGAM85AqK18C3lt4J1vS3vI7nUNRia3juJGO1YlTZCpIGeFGTweSetaM+g6lZ6zY6xoxtJbmOyFjdW1zI0aSoPmVldVYhg2eqnIJ6YoWnpp+T/4CB6/j+n/BE0PxJda9ouoGzudNa7sbkwNeIS9q6gK3mABs42t93d1BG7vVbR/FT6rqGq6TJf6ZqghsxcR3mmHCEHKlGXe+GBGfvcgjgVa13RNV1/R7UXcWn/aLe9S5Ni0jPbzIvHlu5TJ6ls7MAgDBxmm2eiazL4nvtX1RrGNbnThaR29u7P5RDseWKjcDnOcDrjHGTMldNf1t/mVHRp/1v/kctL/Z/wDwgPgD+2vsv2HzofO+17fKx5D/AHt3GM461r6O1jH4wup/BqwDRY7Bvtf2MAWrXGQU27flL7c7ivbGecVftvCt9DovhOzaW3MmiyxvcEM2HCxMh2cc8sOuK6i4jM1rLGuAXQqM+4q5v4mu7/ImH2U+y/M4hNf1m++GN14h1W30x4pdNaZbBrZnVjjq5L4Knn5cdCPmNbF1rF8+qWWiaHHaxXD2gup550LR28WdoARSpYk5AG4AAE89KhPhm8/4Vb/wjXmwfbP7O+y+ZuPl79uM5xnH4VNeaJqNvrlprWjNbSXKWgs7m2uXZI5UB3KVdVYqwbP8JyDjjrTlbmfb/h/1sJX5V3/4b/glfXPE1z4ei06x1G90qPUb6SQfa58wW0Ua8lyrPknBUbd/JPUAVL4X8Uf2zqWoac97p+oPZrHIt5px/dSI+eMbm2sCpyNx4IPtRqOj6xfTaVrEf2GHWNPkk/cea5glifhkMm3cDtCndt6jpg1s6c+qSLI2rQWlucgRx20zTcdyXZU6+m3jHU54leY35F2iiigAooooAKKKKACiiigAooooAKKKKAOP+K3/ACTLVf8Atj/6OSt2sL4rf8ky1X/tj/6OSt2lH4n8v1N5/wACPrL8ohRRRVnMFFFFAGNb/uPGd6nRbmzhlX3ZGdW/QpWzWPqH7jxVo8/aZJ7Q+5ZRIP8A0Sf1rYoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMvxLO9v4bvTCcTSR+TEf+mkhCJ/48wq/bwJa2sVvCMRxIEUegAwKzNa/wBI1PR7Ech7o3Eg/wBiJSwP/fZjrYoAKKKKACiiigArmdX/AOSm+Dv+33/0SK6auZ1f/kpvg7/t9/8ARIqJ7fd+Z0Yb436S/wDSWdzRRRTMQooooAKKKKACsqLQ1TxLfarJKJFvLWK2aAx8AIXOc55zv6Y7Vq0UAcla+Abey8H6toUF4+7Ug6G5dNzRoV2xrgnkIgCjkdO2a1U0Dbrum6l9pz9hsntPL8v7+4od2c8fc6YPWtiijrf+u36h/X9fcYVn4a+yeF7/AEf7Xv8AtjXLed5eNnnMzdM843evOO1Mk8LbbfS5LG9NrqemW628d2IgyypgBkkTPzIdoOMgg8giugooD+vvMFPD93eapbXviHUIr37G2+2t7e2MEKSYI8xgXcswBwOQB6Z5q1pWjf2ZqGrXPn+b/aNyLjbsx5eI1TGc8/dznjrWpRQBx03gi/Ph678PWmu+RpE6yLEn2TdPCrc7BJvwUyTxtztONw61fm8Lzx3en3+k6glrf2doLOR5bfzYriL0ZAykEMMghuORzmuioo/r+vvDcwtQ0G81bRo7TUtSjedbuK5MsVtsQCORWCKm4kA7epYnJJ6YAk8Q6Jcau2nS2V5FaT2F0LlGmgMysdjLgqHU/wAXr2rZoo/4f+vuAx5tIvdU0S/03Xru0uUu4miDWto0OwEEZIaR8noR06VSj8IbfAt34fe/aSa8jlE960fLySEkvtz79M9AOa6WijuHbyGQx+VBHHnOxQucdcCn0UUbiSsrBRRRQMKKKKACiiigAooooA4/4rf8ky1X/tj/AOjkrdrC+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyiFFFFWcwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXM6v/AMlN8Hf9vv8A6JFdNXM6v/yU3wd/2+/+iRUT2+78zow3xv0l/wCks7miiimYhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcf8Vv+SZar/2x/wDRyVu1hfFb/kmWq/8AbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAxvEn7q1srv/AJ9b+Bs+gZ/LY/8AfMhrZrL8SwPc+F9Sii/1v2Z2j9nAyv6gVftp0urSG4j+5Miuv0IzQBLRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjQ/6V4yuZOq2NmkKn0aRizj8kj/Otmsbw5++t76/PW8vZXB9VQ+Up+hWMH8a2aACiiigAooooAK5nV/+Sm+Dv+33/wBEiumrmdX/AOSm+Dv+33/0SKie33fmdGG+N+kv/SWdzRRRTMQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOP+K3/JMtV/7Y/+jkrdrC+K3/JMtV/7Y/8Ao5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczq//JTfB3/b7/6JFdNXM6v/AMlN8Hf9vv8A6JFRPb7vzOjDfG/SX/pLO5ooopmIUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHH/Fb/AJJlqv8A2x/9HJW7WF8Vv+SZar/2x/8ARyVu0o/E/l+pvP8AgR9ZflEKKKKs5gooooAQgMpBGQRgg1keFCR4XsoWOWtVa1OfWJjGf/QK2KxtB/dXWsWn/PG/ZlHtIiyZ/wC+nb8qANmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACqWsXp03RL29Ay0EDyKP7zAcD8TgVdrG8Rfv49PsB/y+XsSsPVUzKw+hEZH40AXdJsv7N0azsgc/Z4EjJ9SFAJq5RRQAUUUUAFFFFABXM6v/yU3wd/2+/+iRXTVzOr/wDJTfB3/b7/AOiRUT2+78zow3xv0l/6SzuaKKKZiFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/wCSZar/ANsf/RyVu1hfFb/kmWq/9sf/AEclbtKPxP5fqbz/AIEfWX5RCiiirOYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArmdX/5Kb4O/wC33/0SK6auW8T2+rx+JtA1fRtL/tL+z/tPmRfaEh/1iBRy34noelRPY6MNb2mvZ/kzvKK4v/hLPF3/AEIv/lXi/wAKP+Es8Xf9CL/5V4v8KXMv6TH9Xn3X/gUf8ztKK4v/AISzxd/0Iv8A5V4v8KP+Es8Xf9CL/wCVeL/CjmX9Jh9Xn3X/AIFH/M7SiuL/AOEs8Xf9CL/5V4v8KP8AhLPF3/Qi/wDlXi/wo5l/SYfV591/4FH/ADO0ori/+Es8Xf8AQi/+VeL/AAo/4Szxd/0Iv/lXi/wo5l/SYfV591/4FH/M7SiuL/4Szxd/0Iv/AJV4v8KP+Es8Xf8AQi/+VeL/AAo5l/SYfV591/4FH/M7SiuL/wCEs8Xf9CL/AOVeL/Cj/hLPF3/Qi/8AlXi/wo5l/SYfV591/wCBR/zO0ori/wDhLPF3/Qi/+VeL/Cj/AISzxd/0Iv8A5V4v8KOZf0mH1efdf+BR/wAztKK4v/hLPF3/AEIv/lXi/wAKP+Es8Xf9CL/5V4v8KOZf0mH1efdf+BR/zO0ori/+Es8Xf9CL/wCVeL/Cj/hLPF3/AEIv/lXi/wAKOZf0mH1efdf+BR/zO0ori/8AhLPF3/Qi/wDlXi/wo/4Szxd/0Iv/AJV4v8KOZf0mH1efdf8AgUf8ztKK4v8A4Szxd/0Iv/lXi/wo/wCEs8Xf9CL/AOVeL/CjmX9Jh9Xn3X/gUf8AM7SiuL/4Szxd/wBCL/5V4v8ACj/hLPF3/Qi/+VeL/CjmX9Jh9Xn3X/gUf8ztKK4v/hLPF3/Qi/8AlXi/wo/4Szxd/wBCL/5V4v8ACjmX9Jh9Xn3X/gUf8ztKK4v/AISzxd/0Iv8A5V4v8KP+Es8Xf9CL/wCVeL/CjmX9Jh9Xn3X/AIFH/M7SiuL/AOEs8Xf9CL/5V4v8KP8AhLPF3/Qi/wDlXi/wo5l/SYfV591/4FH/ADJPit/yTLVf+2P/AKOSt2uG8V33i7xP4Zu9I/4Q77L9o2fvf7UifbtdW6cZ+7jr3ruaI6ybHVjyUoxbV7vZp9F2CiiitDlCiiigArGg/ceNLxOi3VlFKB6sjurH8mT8q2axtR/ceKdGuOglWe0PvuUSD/0SaANmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACseX/AErxnAnVbGyaRh/tSsFU/lHJ+dbFY2if6RqGr33US3fkRn/ZiUIR/wB9+ZQBs0UUUAFFFFABRRRQAVzfiTSdZutd0bVfD72An03z8rel9reYoXooyeM9x2rpKKTV1YunUdOXMv6voc39r+Ivp4X/ACuKPtfxF9PC/wCVxXSUVPJ5mvt/7q+45v7X8RfTwv8AlcUfa/iL6eF/yuK6Sijk8w9v/dX3HN/a/iL6eF/yuKPtfxF9PC/5XFdJRRyeYe3/ALq+45v7X8RfTwv+VxR9r+Ivp4X/ACuK6Sijk8w9v/dX3HN/a/iL6eF/yuKPtfxF9PC/5XFdJRRyeYe3/ur7jm/tfxF9PC/5XFH2v4i+nhf8riukoo5PMPb/AN1fcc39r+Ivp4X/ACuKPtfxF9PC/wCVxXSUUcnmHt/7q+45v7X8RfTwv+VxR9r+Ivp4X/K4rpKKOTzD2/8AdX3HN/a/iL6eF/yuKPtfxF9PC/5XFdJRRyeYe3/ur7jm/tfxF9PC/wCVxR9r+Ivp4X/K4rpKKOTzD2/91fcc39r+Ivp4X/K4o+1/EX08L/lcV0lFHJ5h7f8Aur7jm/tfxF9PC/5XFH2v4i+nhf8AK4rpKKOTzD2/91fcc39r+Ivp4X/K4o+1/EX08L/lcV0lFHJ5h7f+6vuOb+1/EX08L/lcUfa/iL6eF/yuK6Sijk8w9v8A3V9xzf2v4i+nhf8AK4o+1/EX08L/AJXFdJRRyeYe3/ur7jifEGnePPEmhXGlXzeHY4LjbvaEzhhtYMMEgjqo7V21FFNRSIqVXNKNkku3n/wwUUUVRkFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFY3iT93Z2d33tb+B8+gZxGx/75c1s1meJLdrrwxqUMX+sa2kMfs4Ulf1AoA06KhtbhbuzhuY/uTRrIv0IzU1ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQ3VzHZ2c1zMcRwxtI59ABk/wAqpeHLeS18OWKTjE7RCWYf9NH+d/8Ax5jUXij97opsh1v5o7XHqrsA/wD45uP4VsUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFIQCCCMg9QaWigDH8KEjwvZwE5NqGtTn1iYx/wDslbFY2hfurzWbTtDfs6j2kRJM/wDfTN+VbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAY1/wD6T4q0q26rbJNeN7MAI1/MSOf+A1s1j6b/AKT4k1e76iIxWaH/AHV8w4/GXH/AfatigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAxoP3HjS8T+G6sopF/3kd1Y/kyVs1j6j+48UaNcdPNE9offcgkH/omtigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKQkKCScAckntS1leJpXi8N3iwnbLOgtoiOzykRqfzYUAM8LAv4fhumHzXzPeHPXErF1H4KwH4VsUyGJIII4YhtSNQqj0AGBT6ACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDG8S/u7K0u+9rfQPn0VnCMf++XatmszxHbNd+GdSgj/1jW0nln0cKSp/PFXbO5W8sYLmP7k0ayL9CM/1oAmooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKxtY/wBI1jR7IcgztdSD/YiXj/x9462ax7f/AErxheS9VsrWO3X2dyXf9BFQBsUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFACYzwayPCh2+GLSA9bTfaH/tk5j/9krYrG0L91e6zadBFfF1H+zIiPn/vpm/KgDZooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKx/Df76zur49b68llB9VB8tD/wB8Ipqzrd62naDe3cYzJDA7Rj+8+PlH4nAqXTLJdN0m0sUOVtoUiB9dqgf0oAtUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABWPB+48aXafw3VjFIB/tI7qx/J0/KtisbUv3HifRbj/AJ6ie0P/AAJBJ/7RoA2aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMfxB+/Om2H/P1ex7h/sx5lP4fuwPxrYrGb/SvGiDqthYlj/vTPgfiBE3/fVbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFY3iX93Y2t2OtrfQSZ9FMgRj/3y7Vs1m+IrZrzwzqVvH/rJLaQIfRtp2n88UAaVFQWVyt7YW91H92eJZF+hGf61PQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRUF7dR2On3F3NxHbxNK/0UZP8qAM7Qv8ASLrVr/qJ7xokP+zEBHj/AL7Vz+NbFZvh61ksvDtjDP8A64Qq03vI3zOf++ia0qACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKOtFFAGP4U+Xwza2562he0/wC/TtH/AOy1sVjaH+5v9ate0d95ij/ZkjR8/wDfRf8AKtmgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACsfxP+90lLIdb64itiPVGYF//ABwPWxWPef6T4s02DqtrDLdt7MQI0/MPJ+VAGxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjQ/uPGt0vQXVjE4H+1G7hj+UiflWzWNqX7nxNotx/z0M9oT/vJ5n/ALRrZoAKKKKACiiigAorz/4f/D/wzrfgXT9Q1TTPPupvM8yT7RKucSso4VgOgFdH/wAKp8Gf9Ab/AMmpv/i6zUpNXt/X3HXUo0YTcHJ6abL/AOSN2isL/hVPgz/oDf8Ak1N/8XR/wqnwZ/0Bv/Jqb/4unzS7fj/wCeSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4usrwlpVlonjrxXp+lw+Raw/Y/Lj3s2MxMx5Yk9SaOZ3SaD2VNwlKEnprqrdUu77nZUUUVZzBRRRQAVj6X/pPiDWLzqEeKzQ+oRd5/8elYfhWszKiFnICqMknsKyvC6t/wjttcOCHvC12wPUGVjJg/QMB+FAGvRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjeJf3dha3Q62t9byE+imQIx/wC+Xatms7xBate+G9Rt4/8AWSW0gQjs207T+eKtWN0t9p9vdp92eJZF+jAH+tAE9FFFABRRRQBhfCn/AJJlpX/bb/0c9dhXH/Cn/kmWlf8Abb/0c9dhWcPhR1Yn+PP1f5hRRXI+FfH0HiXxPruhvZmzudJuGjTMu/7SiuULj5RjBHI5xkc1W7sc+yuddRXLfEDxxD4E0FL97X7dcSy+XDaiXyy+AWY5wcAKCenoO9HifxXqOjaro+m6Noiard6oszKj3gtwgjVSeSrZ4b26UAdTRXMx+ItdtfDur6p4g8OxacdPtnniij1AT+ftVmIyEG3oB0PX2p/gjxhb+M9B+2pbtZXUbbLmzkbc0LYyOcDIKkEHHINHcDo6K5OXxx5Wg+KdS/s/P/CPXE0Pl+f/AMfHlor5zt+XO7HQ9K6NLzfpK3uzG6AS7M/7OcZpXSV/R/eO2tv60LNFcjF468zwp4c1r+zsf25cwQeT5/8AqPNzzu2/NjHTAz7UzXfiHb+H/H1j4evrFha3UCSvqIk+WBndkRWXHALKBuz1YcVVne3nb5k3Vr+V/kdjRWN4l8Qf8I9bWEv2b7R9s1CCyx5mzZ5jbd3Q5x6d/UVB4g8VrpF9BpWnWE2r6zcxmWKxgZUxGDgySO3CJnjJ6ngA0v6/UZ0FFc3omu+IrvVBa694UbS4mQsl1Dfpcx5H8LYClT+GKx4vHPifULnU/wCxPBkd/a6fezWZl/tZI3kaNsEhGTv6Z/GgDvKK5K+8fW8Xw3Pi3TbKS6T5ALSZ/JYOZRGyMcHBVic8HpVjSNY8WXepxw6x4Tt9OtGB33KassxXjj5Agzk8dadtbB0udLRXCQeN/E+pXepDRPBsV9a2F9NZmY6ukTSNGcEhWj4zx3/Gt7w54qg8QNd2z2lxp2pWLBbuwugBJFkcMCCQynswODS3VwejsbtFcZJ8QGX4Yt4uj0vewfYtmbnGT5/lff2/j93296vaRrHiy71OOHWPCdvp1owO+5TVlmK8cfIEGcnjrQB0tFcd4Z+Idvr/AIs1bQLixaxuLKeWK3dpNy3axttdl4GCOCV54PWtW58RPD4yj8PxWfmvJpsl8spl25KuqBMY77uuePSjt5/8OHVrsblFefTeO/GFvr1ro8nga3F5dwyTRKNbXBVCA3PlcfeFal74w1O0+wafH4da58Q3kLztp0d4gjgjVgpZpiAMcjGFJzxijpcDraKydAv9av4Jm1/RE0iVHAjRL1bkSLjrkKMc8YIrWoAKKKKACuG0j/kpvjH/ALcv/RJrua4bSP8AkpvjH/ty/wDRJqXuv66G9L4Knp/7dE6aiiitDmCiiigDI8USMvhu7iiYrJdBbWMjqGlYRgj6bs/hWrGixRrHGoVEAVQOwFZOrf6RrmjWQ5Alku3HqsabR/4/Ih/CtigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKx/CnyeGra3/wCfRpLTHp5TtGP0WtisbQ/3Ooa1a9o73zEH+zJGjZ/76L0AbNFFFABRRRQBhfCn/kmWlf8Abb/0c9dhXH/Cn/kmWlf9tv8A0c9dhWcPhR1Yn+PP1f5hXjdojaQdQ8X24OdJ8T3sd4AfvWkrqsnHfadrj/dNeyVj6f4X0/T9P1SyXzJ4NVuZ7i4ScggmX76jAHy+xz9aeqd12/VGGjVn3/R/5nmnjp18TaV4w10HfY6Ra/2ZYns0hZWncevO1M/7JroPGo1ZvHvgweH3skvvKvdhvkdoseXHnIUg9Pet5fAmkx/D/wD4Q+FriLTTD5RdHXzTltxbJGMk89O9P8R+DbfxHd6ddNqmpabc6cJFhm0+VEbDgBsllbso6Y709mrdP+D+rFvq/P8AS35GfrK+IF+G3iX/AISiTTZJ/wCz7jyzp0ciKF8puu9ic5rCMEnhPT9C8a6ejNavp1tb65Ag+/CEUJOB/ejzz/sk9MV1Vp4LWHStT0+98Qa1qcOo27W7/brhHMSsCCUwgwcHvnoK27TToLTR4NMA822hgW3AlAbegXb83GDkdeKNrteX6/5hvZPz/G3+R5gxF14A+KDWxEqyXt2yMhyGH2eM5B+lehW9xC3gqK5WRTCdPDhweNvl5z+VVPCngjSvB+kXmmaaZ5rS7naZ47plfG5QuwcD5cKBg5+tZSfCvSkhNkusa8NIJI/sn7efs2z+5jG/b7bqmSvFxXVJfcrDTtLm7Nv72c5CjR/Cv4cq42n+0dPOD6HJH6Vs6vo1n4g+K+oaVqcXm2t14ZVJF7j/AElsEehBwQexArq9U8OWOq22nW8m+CLTbqK6gSDCgNH91SCD8vsMU8aFbDxU3iAPN9rayFkUyPL2By+cYznJ9cY7Vbacm/Nv742JV1FLyX4SueZ6lq19/ZmmeGvEMm/WNH8QaejTHj7ZAZf3c4+oGG9GBrq9AZI/i34siuiBcy29nJbBvvNAEIO32D5zjua0vEfgjSvE+r6Tqd+Z4rvSZ1mhkgZV34YNsfIOVyoOOPY81N4h8Jad4ke3nuWubS+td32a+spjDPDnqAw6g+hBHtSTdtd7v8l/l+Y7K+nb9WzIOueKNP8AiHpukar/AGRJpuqfaWgNtFKJkWJQw3Fm25+YZwPXpXJ6Tf8AjPTbPxZdeHLPTbixi1y8Z95ke6HzjcUj+VXwOQu4Eniu40jwJZ6ZrEWq3eq6xrF7bqy28upXfmCAMMNtVQq8jqSD0FamiaFbaCl8tm8zi9vZb2TzSDh5DkgYA4446n3ot+T/ADX6Dv8An+jPPPEFnp1n+zxs8PXf2y2kEE0d1IP9a73KMzMO3zE5Xt07V2miJ4zXUM+I59Bez2HiwhmWTd25dyMde1V5/h7pc3h3VNDS7vobDUrn7T5ccif6M28ORFlTtBZc4Oepxin6X4Mm0zU4bx/FniK+ERJNvd3UbRPxj5gIwT1z1p31/rsien9f1qVPhx/x7+Jf+xivf/QhUdsY7r42X8tmdy2miRwXjL0WRpSyKf8Aa25P0NK/w0hF5fTWfijxJYR31zJdSwWl4kaB3OWxiPI/PPvXRaH4e03w5YNaaTb+SjuZJXZi8kznq7uclmPqTU9F5K34WG+vm/1ueWy7/wDhmc+VtD+eNu4cZ+3cZr0PRE8ZrqGfEc+gvZ7DxYQzLJu7cu5GOvaopPAWmSeA28J/abxLJm3ecrr5wPm+bwduPve3SjS/Bk2manDeP4s8RXwiJJt7u6jaJ+MfMBGCeuetUt3/AF0B7HH6b4duNZ0bXr3RnWHXNL8TXtzp0zdN+4bo2/2HHykfQ9qv6Br8Hib4n6VqUCNCz+H50mgf70Eq3CB429wQRXa6JoVtoKXy2bzOL29lvZPNIOHkOSBgDjjjqfeqFn4I0qw8dXfiu0M8d9eQGGaIMvknJUl9uM7jsGTnn0zSjo4+S/HlsEtb+v4c1zP1X/ksfh3/ALBl7/6FFV/xL4Wm1e+tdV0fUpNK1myVkiuVQOkkbEExyIfvKSAfUHkVoXGhW1z4lstcd5hc2cEsEaAjYVkKkkjGc/KMc+tUtf8ACMWu30V7HrGr6Tcxx+WZNNuvL8xckgMpDKcEnnGeaWyXz/Nj6v5fkiPwp4h1DUrrUdJ8QWsFtq+lsgn+zOWhmRwSkiZ5AOD8p5GOtdJWP4d8MWPhuK4+ySXVzcXcgkubu8mMs07AYG5j6DgAYArYqmSFFFFIYVw2kf8AJTfGP/bl/wCiTXc1w2kf8lN8Y/8Abl/6JNS91/XQ3pfBU9P/AG6J01FFFaHMFFFFAGNaf6T4u1CfqtpbxWy+ztmR/wBDF+VbNY3hn99ps98et9dSzg+q7tqf+OKlbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVjRfuPGtyva7sI3A943YMfykT8q2ax9T/c+JdFuP+ejTWhP+8nmfzhFAGxRRRQAUUUUAYXwp/5JlpX/AG2/9HPXYVw3w3S+k+FekDTbi3t5d0xLXEDSqV86TjAdec45z+FdD9m8Sf8AQW0r/wAFcn/yRWUX7qOzERTrz16vv39DYorH+zeJP+gtpX/grk/+SKPs3iT/AKC2lf8Agrk/+SKq77GPJH+Zfj/kbFFYN5JrWnWj3Woa/olrbx43zT6e6IuTgZJuMDkgfjWdD4kNzcRwW/jbwtLNKwSOOODczsTgAAXOSSe1F32Dkj/Mvx/yOvorH+zeJP8AoLaV/wCCuT/5Io+zeJP+gtpX/grk/wDkii77ByR/mX4/5GxRWP8AZvEn/QW0r/wVyf8AyRR9m8Sf9BbSv/BXJ/8AJFF32Dkj/Mvx/wAjYorH+zeJP+gtpX/grk/+SKPs3iT/AKC2lf8Agrk/+SKLvsHJH+Zfj/kbFFY/2bxJ/wBBbSv/AAVyf/JFH2bxJ/0FtK/8Fcn/AMkUXfYOSP8AMvx/yNiisf7N4k/6C2lf+CuT/wCSKPs3iT/oLaV/4K5P/kii77ByR/mX4/5GxRWP9m8Sf9BbSv8AwVyf/JFH2bxJ/wBBbSv/AAVyf/JFF32Dkj/Mvx/yNiisf7N4k/6C2lf+CuT/AOSKPs3iT/oLaV/4K5P/AJIou+wckf5l+P8AkbFFY/2bxJ/0FtK/8Fcn/wAkUfZvEn/QW0r/AMFcn/yRRd9g5I/zL8f8jYorH+zeJP8AoLaV/wCCuT/5Io+zeJP+gtpX/grk/wDkii77ByR/mX4/5GxRWP8AZvEn/QW0r/wVyf8AyRR9m8Sf9BbSv/BXJ/8AJFF32Dkj/Mvx/wAjYorH+zeJP+gtpX/grk/+SKPs3iT/AKC2lf8Agrk/+SKLvsHJH+Zfj/kbFFY/2bxJ/wBBbSv/AAVyf/JFH2bxJ/0FtK/8Fcn/AMkUXfYOSP8AMvx/yNiuG0j/AJKb4x/7cv8A0Sa6SG38QLPGbjU9NeIMC6pp0isy55AJnODjvg/Q1zekf8lN8Y/9uX/ok0uq/roawSUKlnfT/wBuidNRRRWhyBWfr149h4fvrmHmZIG8oerkYUfixArQrG1/9/NpdgOftF6juP8AZiBlz/30ij8aANHT7NNP022sovuW8KRL9FAA/lViiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKxvE37vTra6HW1vbeQn0UyBWP8A3yzVs1neILVr3w3qVtH9+S1kVCOzbTg/nigDRoqvY3S32nW12n3Z4lkGPRgD/WrFABRRRQBhfCn/AJJlpX/bb/0c9dhXH/Cn/kmWlf8Abb/0c9dhWcPhR1Yn+PP1f5hRRRVHOcb8WoZbj4WaykETysFicqikkKsqMx47AAknsATXgnguGbVPHugf2dpePstzbPP9lWR8qsq7pnyW29RkjCjjgV9JeNP+RC1//sGXP/opq8phFqfHvgT7ZYQ3n/FPaf5Xm3otvJk83iRckeYw7RjJbPTigD3Gq+oX9tpenT31/J5VtbxmSV9pbao6nAyT+FWK57x9/wAk913/AK8Zf/QaT0VxxV2kbOn6ha6rp0F9p8wntrhBJFIoI3A+x5H0NV7XXtNvdavNJtbkSX1iFNxEEb92GGRzjB/A1yOnatH4Ig1XTrgM0EcA1HTowPvLIQrQr9JSMe0grM0dLjwlrWv3NyPtN9BoSXtxzxJMXldvwzx9BTdk/LX8n/lqTG7Xnp+n+Z6jRXnenL4gltdJv7Oz12W+kkhlurm6voTbTRNjzMQiYqo2kldqAjA983tO06bXde8UJe6rqSwwXixW0VveSQiA+ShLAoQTyfunK98cmh6X8v0t/mO6ev8AXX/I7amu6xxs7nCqCSfQVwWk61qV/a+B5ru6kMt1PPHdbG2rPsikGWAwDyoP1rWubuc+PtQtPPkNuuiLJ5O87A5kcbtvTOABmlP3b28/wVxxV3r5fi7HQadqFrq2mwX+ny+dbXCB4pNpXcp74IBH41ZrzHS2un8GfD+1tL+4sluZAkzW77S6eS5Kn64/DqOQK1Lm/ufCmua1Baz3N3ZxaKdRihu7h5ykqMVIDOS204HGfXGKqVot/P8ABXJjdpfL8XY7qqf9r2X9p3Gn+f8A6VbQrPLHsb5UJIBzjB+6eBzXC63p95p3gK21ddd1T+05HtpLiX7a5SXzJU3KI87EX5uNgBwMZwTnUH/JS9f/AOwLB/6FLUyvH8fwVyo2l+H4uxox+OdBktY7oz3UVpLtK3c+n3EUGGOATKyBADkck4roFZXQOjBlYZBByCK53wOkcnw30VJlVo20+MOrDII2cg+1Q/Dgv/wg1opJaFJJkt2JzmESsE59NuMe2KuStJrsQndJnU0UUVJQUUUUAFFFFABRRRQAUUUUAFcNpH/JTfGP/bl/6JNdzXDaR/yU3xj/ANuX/ok1L3X9dDel8FT0/wDbonTUUUVocwVj/wDH1409VsLH/wAemf8AmBD/AOPe9bFY2gfv5dUv+v2m9dUP+zEBF+WUY/jQBs0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBj+FPk8N29t3s2ktMf9cnaMfoorYrG0T9zqWtWvZLwSoP8AZkjRv/Qt9bNABRRRQBhfCn/kmWlf9tv/AEc9dhXH/Cn/AJJlpX/bb/0c9dhWcPhR1Yn+PP1f5hRRRVHOMmhiubeSC4iSWGVSkkcihldSMEEHggjtWXD4U0GC4jnGlW0k0LBoJJ081oMHKrGWyY1U/dVcKvYCteigAqnq+mQ61o13pt00iQ3cTRO0ZAYAjBxkEZ/CrlFAbGTqPhrTtVuNLnvEZ5NLlEsByOSBjDccjIU9uVFPGg2n9uXeqP5kkt5apayxOQYyilj0xnJ3HPNadFAbGFpvhj+yhBDa6zqf2C3bMVi7xmNFByqb9nmFRxgFzwADkcVe0/SINOvNQuYHkZ9QnE8ocghWCKmFwOmFHXNX6KAOf/4Q2xXQrDTILm7hOnzefa3Ubr50b5Yk5K7SCGYEFSCDT7DwnbWWrXWpyXt7d3l5bC3nluHT5lBJBCqoCkA44AHHTOSd2igDDtfCdjaWOiWsctwU0Vt1uWZcudjJ8/HPDHpirF1olpLqs2qTJJPI9ibN4OCjx7txGPU9OuK1KKHqC0PMrvRU1fTbDR9JfxFLClxC6QajbPBDYxq4Yne0amQhcqql36j0yO6/sG2/ty81XfN593arayLkbQqliCBjOfmPetOih6r+uugLQ5m18FJb6VDpUmuatcaZFGIvsjvCisgGApeONXx/wLnocjIro4YY7eBIbeNYoo1CoiLhVA4AA7Cn0U7gFFFFIAooooAKKKKACiiigAooooAK4bSP+Sm+Mf8Aty/9Emu5rhtI/wCSm+Mf+3L/ANEmpe6/rob0vgqen/t0TpqKKK0OYrajeJp2mXV7L9y3heVvooJ/pUGhWb2Hh+xtpv8AWxwL5p9XIyx/76Jqt4m/fadBYjrfXUUBHqm7c4/74R62aACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMaP9x42nXtd2EbAe8UjAn8pV/IVs1jap+58SaJcf8APR5rQn2ePzP5witmgAooooAwvhT/AMky0r/tt/6Oeuwrj/hT/wAky0r/ALbf+jnrsKzh8KOrE/x5+r/MKKKKo5wooooAKr6hf22l6dPfX8nlW1vGZJX2ltqjqcDJP4VYrnvH/wDyTvXv+vGX/wBBNTJ2i2VFXaRvxSpNCksZ3I6hlOOoPSnVxNzHcXvinw5p41C7trOTSpZJ4reZo/Nx5QAJBBHXqMHtnk1Ru9Y1Pw/Z+KrGyupZjZz2yWMt1IZmg+0bQcs+SwUkkbiew6VclZ2X9a2Ii7r+u1z0SiuG1jS5NA1Dw0bHWNUYTalHBcpcX0kouBsc5IYnHI6LhTnkcDE+iQT6h4s8RT3t/fSQWN8gtbZLmREQ+SjHIUjcDkfKcr145paa+X6W/wAxv+vx/wAjsqK8vsbzxBr3hr+2rCx12TVbhmmtZY76FLRMMdsflecAUwNpLJuPJ9MdFbpP4l8VatBqF1eW1rpgihitrS6eDLugdnZo2DN1AAJ28HjJo1A1Y/E0E1hcXVtZX1wLe+axeOGIO+4PsLAA/cGc59B0rZrzCylutO8F3YhvpmlXxR5TXCttaQG6UNnbgcjOR0Oa1L2/vNX8ZatZPZazc2emrFFHFpl4lt87pvMjt5sbE8gAcrwT1NC2T/rZP9Q6tf1u1+h3dBIVSScAck159OfEDWPhWy1S7vLC6m1KSGdklTzJYQkhUOUJXcVC5I6Hkc1ejtXtfFuo6D9svpdNuNMW7VZL2VpIpA5U7ZS28AjHG719TSlonbz/AAVw/r8bHQN4j0pdEt9XN1/oNyyLFN5bfMXYKvGMjJI6imXvibTbDUn0+U3cl0kayvHbWM8+1WJAJMaMBnB/KvPksYrX4M6NNE9wzTz2LOJbiSRQfPX7qsxCD2UCuwsP+Soaz/2DbX/0OWqa1t5tfhcXS/kvzsbWl6xYazA8um3KzCNzHKhBV4mHVXRgGU+xANXa5Qr5fxcQ2mB52kMbwL3xKBGW9+XAPpmurpdE/wCt7D62/rYKKKKACiiigAooooAKKKKACuG0j/kpvjH/ALcv/RJrua4bSP8AkpvjH/ty/wDRJqXuv66G9L4Knp/7dEf428QXfhvQ4ryxhiZpLqOB5p1ZorZGPMjhecDAHUcsPoeUHxG1M6pPZNdaLDaQmXy9ZkhlNtdFVBEaAN97nn5m9geM9B4r1HxxaarHH4S0eyvrIwhnkuHAYSbmyOZF4xt7d+tYn9t/Fn/oWNK/7+r/APH60OY2PDusXPiiTQb2+tfssqWct3LFg4DM3lRsM84ZfMYexHJ612Nc74Ug1uSO41PxTb29rqV0FjMFu2VSNC23ueSXY8E9R9B0VABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGN4m/d6bBdDra3lvKT6L5iq3/jrNWzWdr9q194b1G2T78ttIqEdm2nB/PFWbC6W+022u0+7cQpKPowB/rQBYooooAwvhT/AMky0r/tt/6Oeuwrj/hT/wAky0r/ALbf+jnrsKzh8KOrE/x5+r/MKKKKo5wooooAKp6xpcOtaNd6ZdNIkN3E0TtGQGAIwcZBGfwq5RSaurAtNTNXQrZdWstQDy+bZWr2sa5G0qxUknjr8g9O9UtU0DTVt9cu7y2ur6PUo0+020Q3MRGuB5YGDnv1zkcVv0U3qC0PPY9JfV/EGhG3uddvotMn8+S51W3NusShGARVMcfmMx25YhiADyM89np+kQabeahcwPIz6hOJ5Q5BCsEVMLgdMKOuav0U/wCv6+4DAt/CcdlJKmm6tqVlZSzGVrGF4/LBY5YKWQugJycKwwScYqW88NRT6vJqdlqF7pl1PGIrhrRkxOo+7uV0YZHIDDBwetbVFIDnLfwRplroZ0qCW6Ft9uF8C0oZw4kDgbiCSMgdcnHfPNWr3w3Hcaw2qWOoXmmXkkQimktDGRMoPy7lkRlyOcEAHkjNbNFH9foBlSaBBO2lvcXN1NLpsxnjkdwWlcqykvx/tk4GAOMYAxUx0e3PiA6uWkM5tfshQkbNm7dnGM5z71fooA5weCrP+xRpLX981jHPHNBCWj/cbH3hFOzJXOB8xJwOCKsXfhrz9cm1a01e/wBPuJ4UhkFuIWVlQsRxJG2D8x6Vt0UAZ+laLa6T5zwmSa5uGDXF1O++WZgMAk9AB2UAKOwFaFFFABRRRQAUUUUAFFFFABRRRQAVw2kf8lN8Y/8Abl/6JNdzXDaR/wAlN8Y/9uX/AKJNS91/XQ3pfBU9P/bonTUUUVocwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeUReOdS0WO80mM6fZpo0Hl20d7HI0moFJGjxEQQMfLt6NyPTO31evP9Um8f6RruoReGNEsbzTZp/PiluJFDEsqlhjzV43bu1AFbTviJq13qelxSQWIlvdQa1n0hY5ftdog/5aOxOMD7x+QDH47fSa82/tv4s/9CxpX/f1f/j9d7pUl/NpFrJq8EVvfNEpniibcqPjkA//AFz9T1IBnfCn/kmWlf8Abb/0c9dhXH/Cn/kmWlf9tv8A0c9dhWcPhR1Yn+PP1f5hRRRVHOFFFFABVfUL+20vTp76/k8q2t4zJK+0ttUdTgZJ/CrFc94+/wCSe67/ANeMv/oNJ6K44q7SNnT9QtdV06C+0+YT21wgkikUEbgfY8j6Gq9rr2m3utXmk2tyJL6xCm4iCN+7DDI5xg/ga5HTtWj8EQarp1wGaCOAajp0YH3lkIVoV+kpGPaQVmaOlx4S1rX7m5H2m+g0JL2454kmLyu34Z4+gpuyflr+T/y1Jjdrz0/T/M9RorzvTl8QS2uk39nZ67LfSSQy3VzdX0Jtpomx5mIRMVUbSSu1ARge+b2nadNruveKEvdV1JYYLxYraK3vJIRAfJQlgUIJ5P3Tle+OTQ9L+X6W/wAx3T1/rr/kdtTXdY42dzhVBJPoK4LSda1K/tfA813dSGW6nnjutjbVn2RSDLAYB5UH61rXN3OfH2oWnnyG3XRFk8nedgcyON23pnAAzSn7t7ef4K44q718vxdjQk8WaPHplhf/AGiWS31E4tDDayyNKcFuEVS3QE8jtUth4j0zUr5rKCaSO7CeZ9mureS3kZf7ypIqlhx1AIrh9E/5Fv4b/wDXb/23lrofHi4j0KaDH26PV7cW5H3juOHA9im7I9BVtWlbzt+RCd438r/n/kdXRRRUlBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXDaR/yU3xj/wBuX/ok13NcNpH/ACU3xj/25f8Aok1L3X9dDel8FT0/9uidNRRRWhzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGF8Kf+SZaV/22/wDRz12Fcf8ACn/kmWlf9tv/AEc9dhWcPhR1Yn+PP1f5hRRRVHOFFFFABVPV9Mh1rRrvTbppEhu4midoyAwBGDjIIz+FXKKA2MnUfDWnarcaXPeIzyaXKJYDkckDGG45GQp7cqKeNBtP7cu9UfzJJby1S1licgxlFLHpjOTuOea06KA2MLTfDH9lCCG11nU/sFu2YrF3jMaKDlU37PMKjjALngAHI4q9p+kQadeahcwPIz6hOJ5Q5BCsEVMLgdMKOuav0UAc/wD8IbYroVhpkFzdwnT5vPtbqN186N8sScldpBDMCCpBBp9h4TtrLVrrU5L29u7y8thbzy3Dp8ygkghVUBSAccADjpnJO7RQBzp8GWiaPo9ha399a/2O262njMZkztK/NuQqeGPardn4ehg1BL++vLrU7yJSsM12U/cqeoREVUBPdtu4jjOOK16Kd9bh0sFFFFIAooooAKKKKACiiigAooooAKKKKACiiigArhtI/wCSm+Mf+3L/ANEmu5rhtI/5Kb4x/wC3L/0Sal7r+uhvS+Cp6f8At0TpqKKK0OYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvhT/AMky0r/tt/6Oeuwrj/hT/wAky0r/ALbf+jnrsKzh8KOrE/x5+r/MKKKKo5wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4bSP+Sm+Mf+3L/0Sa7muG0j/kpvjH/ty/8ARJqXuv66G9L4Knp/7dE6aiiitDmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAML4U/8ky0r/tt/wCjnrsK4/4U/wDJMtK/7bf+jnrsKzh8KOrE/wAefq/zCiiiqOcKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuG0j/kpvjH/ALcv/RJrua4bSP8AkpvjH/ty/wDRJqXuv66G9L4Knp/7dE6aiiitDmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAML4U/8ky0r/tt/6Oeuwrj/AIU/8ky0r/tt/wCjnrsKzh8KOrE/x5+r/MKKKKo5wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4bSP+Sm+Mf+3L/wBEmu5rhtI/5Kb4x/7cv/RJqXuv66G9L4Knp/7dE6aiiitDmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAML4U/8AJMtK/wC23/o567CuP+FP/JMtK/7bf+jnrsKzh8KOrE/x5+r/ADCiiiqOcKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuG0j/AJKb4x/7cv8A0Sa7muG0j/kpvjH/ALcv/RJqXuv66G9L4Knp/wC3ROmooorQ5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDC+FP8AyTLSv+23/o567CuP+FP/ACTLSv8Att/6OeuwrOHwo6sT/Hn6v8woooqjnCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArhtI/5Kb4x/7cv/RJrua4bSP+Sm+Mf+3L/wBEmpe6/rob0vgqen/t0TpqKKK0OYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvhT/yTLSv+23/AKOeuwrj/hT/AMky0r/tt/6OeuwrOHwo6sT/AB5+r/MKKKKo5wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4bSP+Sm+Mf8Aty/9Emu5rhtI/wCSm+Mf+3L/ANEmpe6/rob0vgqen/t0TpqKKK0OYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA43SvCXibRNMi0/S/GfkWsOfLj/suJsZYseWJPUmrn9keM/+h5/8pEP+NdNRUezj/TZ0vFVG7u3/AIDH/I5n+yPGf/Q8/wDlIh/xo/sjxn/0PP8A5SIf8a6aijkX9Ni+sT7L/wABj/kcz/ZHjP8A6Hn/AMpEP+NH9keM/wDoef8AykQ/4101FHIv6bD6xPsv/AY/5HM/2R4z/wCh5/8AKRD/AI0f2R4z/wCh5/8AKRD/AI101FHIv6bD6xPsv/AY/wCRzP8AZHjP/oef/KRD/jR/ZHjP/oef/KRD/jXTUUci/psPrE+y/wDAY/5HM/2R4z/6Hn/ykQ/40f2R4z/6Hn/ykQ/4101FHIv6bD6xPsv/AAGP+RzP9keM/wDoef8AykQ/40f2R4z/AOh5/wDKRD/jXTUUci/psPrE+y/8Bj/kcz/ZHjP/AKHn/wApEP8AjR/ZHjP/AKHn/wApEP8AjXTUUci/psPrE+y/8Bj/AJHM/wBkeM/+h5/8pEP+NH9keM/+h5/8pEP+NdNRRyL+mw+sT7L/AMBj/kcz/ZHjP/oef/KRD/jR/ZHjP/oef/KRD/jXTUUci/psPrE+y/8AAY/5HM/2R4z/AOh5/wDKRD/jR/ZHjP8A6Hn/AMpEP+NdNRRyL+mw+sT7L/wGP+RzP9keM/8Aoef/ACkQ/wCNH9keM/8Aoef/ACkQ/wCNdNRRyL+mw+sT7L/wGP8Akcz/AGR4z/6Hn/ykQ/40f2R4z/6Hn/ykQ/4101FHIv6bD6xPsv8AwGP+RzP9keM/+h5/8pEP+NH9keM/+h5/8pEP+NdNRRyL+mw+sT7L/wABj/kcz/ZHjP8A6Hn/AMpEP+NH9keM/wDoef8AykQ/4101FHIv6bD6xPsv/AY/5HM/2R4z/wCh5/8AKRD/AI1P4e8PXuk6nqeoapq39p3Wo+V5kn2ZYceWpUcKSOhHp0rfopqCTuKWInKLjpr2SX5IKKKKowCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAP/2Q==)

Many meals in the menu will have many ingredients assigned to them. Therefore, my solution was to use a link table which takes the MenuID (Primary Key) and the IngredientID (Primary Key) and references them in the link table as foreign keys with MenuIngID as the primary key for the table. This solution allows me to access the multiple ingredients that are assigned to an individual meal item, which is important for the development of the stock replenishment aspect of the program.

The stock replenishment area will reference the many ingredients assigned to the meals in the customer’s order and compare the quantity in stock to the threshold level. A re-order will be processed accordingly if the quantity in stock is less than the threshold level.

## Data Flow Diagram – Level 1

Diagram

Description automatically generated

## Prototype Designs

### Example Menu Form

Graphical user interface

Description automatically generated with low confidence

User input from Manager / Head Chef to create menu / insert existing menu.

A visual representation in a list box of the menu the user has entered into the system.

Buttons for basic functions of the form.

### Example Menu Form – Client Feedback

A screenshot of a computer

Description automatically generated with medium confidence

User chooses the ingredients related to the meal, so a link is established.

After showing the client my prototype design for the Menu Form, I received feedback in relation to the layout of the form and a function that could be added to improve the ease of use for the relevant members of staff. These were highlighted in the above screenshot.

The client suggested an option that would be added to the form to choose the ingredients that are required for the individual meal items; this would improve the ease of use within the system for other users. This feature was easy to accommodate for, as the link between the meals and the ingredients would be needed when a user takes a customer’s order for the system to check the real time stock levels and update the user accordingly.

### Example Take Order Form

Graphical user interface, text

Description automatically generated

List box to show the order.

Drop-down combo box used to select the meal item from menu and waiter enters the quantity.

Basic Functions of the form.

### Example Take Order Form – Client Feedback

A screenshot of a computer

Description automatically generated with medium confidence

Deletes selected order from the list.

After showing the client my prototype design for the Take-Order Form, I received feedback regarding the design of the form and a function that could be added which would improve the efficiency of the waiter taking an order. This was highlighted in the above screenshot.

The client suggested a button be added to the form which allowed the waiter to select an order in the list and delete only that order. Thereby saving time as with my design, the waiter would have to clear the whole order from the list and ask the customers again for their order which could be annoying for the customer resulting in them leaving a negative review, affecting the reputation of the restaurant.

The client also suggested a chronological view of all the processes in the form. I decided to shift the design of the form in a vertical format so that data is entered in such a flow that the waiter can follow, should there be any confusion when entering in this data.

## Use of Language

I have chosen to use Python in combination with MySQL to code a solution to this problem. This is primarily due to Python being a multi-paradigm and object-orientated programming language. I feel that Python will offer an elegant solution to this problem.

# Design

## Design Overview

This program will have 4 major inputs (Create Menu, Populate ingredients, Take Order, Record Suppliers). I will be using a top-down design methodology approach to develop my program. I will do this by breaking down each of the 4 major inputs into their individual tasks and their subsequent processes as well. An example form would be the ‘Create Menu’ form. The form will require user input to create/ populate the system with their existing food menu, this will require the user to: select if the meal is a starter/main/desert, set the name of the meal, the description, the price, select the ingredients required for the meal.

Throughout the program, I will be using many classes and functions to repeat certain tasks and procedures without the need of coding them again.

### Top-Level Structure Chart

### System Flow Chart

![Diagram

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAqACoAAD/4RDaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAFAAAISodpAAQAAAABAAAIUJydAAEAAAAKAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFJhamEAAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAM0MgAAkpIAAgAAAAM0MgAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMjowMzowOCAwMToxMDozNwAyMDIyOjAzOjA4IDAxOjEwOjM3AAAAUgBhAGoAYQAAAP/hCxdodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTAzLTA4VDAxOjEwOjM3LjQyMDwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5SYWphPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIBVcCsAMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDJ1/VrvTLcDTNOfULt0Z0iVwoAXGT6n7w4HWuX8EeJ9e1m81ZtStWleFo1EC4iEH3uMNzz788V19x/yMFj/1wm/nHV0RosjSKih2ADMByQOmT+J/Oos3K9zhlRqVK6mqjSj0+X/B63KX22//AOgU/wD3/T/Gj7bf/wDQKf8A7/p/jV+iqsdXJL+Z/h/kUPtt/wD9Ap/+/wCn+NH22/8A+gU//f8AT/Gr9FFg5JfzP8P8ih9tv/8AoFP/AN/0/wAaPtt//wBAp/8Av+n+NX6KLByS/mf4f5FD7bf/APQKf/v+n+NH22//AOgU/wD3/T/Gr9FFg5JfzP8AD/Iofbb/AP6BT/8Af9P8aPtt/wD9Ap/+/wCn+NX6KLByS/mf4f5FD7bf/wDQKf8A7/p/jR9tv/8AoFP/AN/0/wAav0UWDkl/M/w/yKH22/8A+gU//f8AT/Gj7bf/APQKf/v+n+NX6KLByS/mf4f5FD7bf/8AQKf/AL/p/jR9tv8A/oFP/wB/0/xq/RRYOSX8z/D/ACKH22//AOgU/wD3/T/Gj7bf/wDQKf8A7/p/jV+iiwckv5n+H+RQ+23/AP0Cn/7/AKf40fbb/wD6BT/9/wBP8av0UWDkl/M/w/yKH22//wCgU/8A3/T/ABo+23//AECn/wC/6f41foosHJL+Z/h/kUPtt/8A9Ap/+/6f40fbb/8A6BT/APf9P8av0UWDkl/M/wAP8ih9tv8A/oFP/wB/0/xo+23/AP0Cn/7/AKf41foosHJL+Z/h/kUPtt//ANAp/wDv+n+NH22//wCgU/8A3/T/ABq/RRYOSX8z/D/Iofbb/wD6BT/9/wBP8aPtt/8A9Ap/+/6f41foosHJL+Z/h/kUPtt//wBAp/8Av+n+NH22/wD+gU//AH/T/Gr9FFg5JfzP8P8AIofbb/8A6BT/APf9P8aPtt//ANAp/wDv+n+NX6KLByS/mf4f5FD7bf8A/QKf/v8Ap/jR9tv/APoFP/3/AE/xq/RRYOSX8z/D/Iofbb//AKBT/wDf9P8AGj7bf/8AQKf/AL/p/jV+iiwckv5n+H+RQ+23/wD0Cn/7/p/jR9tv/wDoFP8A9/0/xq/RRYOSX8z/AA/yKH22/wD+gU//AH/T/Gj7bf8A/QKf/v8Ap/jV+iiwckv5n+H+RQ+23/8A0Cn/AO/6f40fbb//AKBT/wDf9P8AGr9FFg5JfzP8P8ih9tv/APoFP/3/AE/xo+23/wD0Cn/7/p/jV+iiwckv5n+H+RQ+23//AECn/wC/6f40fbb/AP6BT/8Af9P8av0UWDkl/M/w/wAih9tv/wDoFP8A9/0/xo+23/8A0Cn/AO/6f41foosHJL+Z/h/kUPtt/wD9Ap/+/wCn+NH22/8A+gU//f8AT/Gr9FFg5JfzP8P8ih9tv/8AoFP/AN/0/wAaPtt//wBAp/8Av+n+NX6KLByS/mf4f5FD7bf/APQKf/v+n+NH22//AOgU/wD3/T/Gr9FFg5JfzP8AD/Iofbb/AP6BT/8Af9P8aPtt/wD9Ap/+/wCn+NX6KLByS/mf4f5FD7bf/wDQKf8A7/p/jR9tv/8AoFP/AN/0/wAav0UWDkl/M/w/yKH22/8A+gU//f8AT/Gj7bf/APQKf/v+n+NX6KLByS/mf4f5FD7bf/8AQKf/AL/p/jR9tv8A/oFP/wB/0/xq/RRYOSX8z/D/ACKH22//AOgU/wD3/T/Gj7bf/wDQKf8A7/p/jV+iiwckv5n+H+RQ+23/AP0Cn/7/AKf40fbb/wD6BT/9/wBP8av0UWDkl/M/w/yKH22//wCgU/8A3/T/ABqK41S7tbd559MkEcYyxEyHA/OtSqGu/wDIBvP+uRpO9iJxlGDak/w/yL9FFFUbhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFC4/5GCx/wCuE3846v1QuP8AkYLH/rhN/OOr9JGcPil6/ogooopmgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVQ13/kA3n/XI1fqhrv/ACAbz/rkaT2M6v8ADl6Mv0UUUzQKKKKACiiigAooooAKKKKACiiigAooooAoXH/IwWP/AFwm/nHV+qFx/wAjBY/9cJv5x1fpIzh8UvX9EFFFFM0CiiigAooooAKKKKACiiigAooqnqzFdLmKkg8cj6inFXdhSdlcuUVxnny/89X/AO+jR58v/PV/++jXT9Xfc5frK7HZ0Vxnny/89X/76NHny/8APV/++jR9XfcPrK7HZ0Vxnny/89X/AO+jR58v/PV/++jR9XfcPrK7HZ0Vxnny/wDPV/8Avo0efL/z1f8A76NH1d9w+srsdnRXGefL/wA9X/76NHny/wDPV/8Avo0fV33D6yux2dFcZ58v/PV/++jR58v/AD1f/vo0fV33D6yux2dFcZ58v/PV/wDvo0efL/z1f/vo0fV33D6yux2dFNi/1Kf7op1cp1hRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFUNd/5AN5/1yNX6oa7/AMgG8/65Gk9jOr/Dl6Mv0UUUzQKKKKACiiigAooooAKKKKACiiigAooooAoXH/IwWP8A1wm/nHV+qFx/yMFj/wBcJv5x1fpIzh8UvX9EFFFFM0CiiigAooooAKKKKACiiigAqlrH/IJm/D/0IVdqlrH/ACCZvw/9CFXD4kRP4GcrRRRXpHlhRRRQAUUUUAFFFFABRRRQAUUUUAFKOtJSjrQB2kX+pT/dFOpsX+pT/dFOryj1lsFFFFAwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKoa7/yAbz/rkav1Q13/AJAN5/1yNJ7GdX+HL0ZfooopmgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBQuP+Rgsf8ArhN/OOr9ULj/AJGCx/64Tfzjq/SRnD4pev6IKKKKZoFFFFABRRRQAUUUUAFFFFACMyopZ2CqoySTgAVzGpeMdBupjpFnqMV1ey9Et8yAbeTlh8o6eta+r3UYh+wLbpeXF0pVbZ/usvQs/oo7/kOa4nSPhvD4U1PUNTjnWaKS3RIV2kGNi4L/AIcDHfBI9zcPiRE/gZoUUUV6R5YUUUUAcRoz39rpD63cwS3KQxzy7zq9w7PtLYHkEbB0x146099S1RdSkSa7gkmcQBGhEixIWjmbOzfz0GcnBwDxxjpIvD+jW90LqDSLGK4BLCVLZA4J6nIGe9SW+kabaKFtdPtYADuAjgVcHnngf7TfmfWseSXKlfsa86vc5iSfUpvA3h6SK7la9mETl0dlMp8pmCscknJAznr6dqr3PiW5XVZdVsZHntZrYRWsBf8Adk+bEvmFSQM7pGHJGQoGR1rtI7G0hhihhtYY4oDuiRYwFjPPKjt1PT1pv9m2PlGL7Fb+WUaMp5S4KscsuMdCeSO9W4tyb/rawlJW2/q9zmV8Q667R2pis7a72XLSNLGHB8sRkfJHK20kPggsTwD7VZ0/xPc3jR+ZHDH5t/HbqvOdjW4l9eTknn07VuQ6ZYW0ccdvY20SRKyxqkKqEDHLAADgHv60i6TpyXS3KWFqtwoCrKIVDgAYABxngcUJSvv2/wCD94nKLW3ct0UUVZAUfrRSjrQB0MevWsMSDUI57A7Rk3MeEH/AxlP1rShmiuIxJBIkqHoyMCD+IpYv9Sn+6KozaFp8shlW3+zzHrLbMYnP1KkZ/GvKPWWxoUVmfYtUtv8Ajz1ITqP+Wd5EG/AOu0j6kGj+07y34v8AS5gO8lowmX8uH/8AHTQM06Kp2ur2F5J5dvdRmXvEx2yD6qcEflVygAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqhrv8AyAbz/rkav1Q13/kA3n/XI0nsZ1f4cvRl+iiimaBRRRQAUUUUAFFFFABRRkUZFABRRkUZFABRRkUZFAFC4/5GCx/64Tfzjq/Xm2vXfiuH4jWunWV9+5uctbSGBD5cbEFx93nbt784A9a9IHCgFskDqe9RGV2ziw1dVZ1EotWdtfRC0UZFGRVnaFFGRRkUAFFGRRkUAFFFFABVHUdRNqUt7WPz72bPlQ5wMd2Y9lHc/gOaNR1E2pS3tY/PvZs+VDnAx3Zj2Udz+A5pdO04WYeWaTz7ubBmnIwWPYAdlHYf1yaADTtOFmHlmk8+7mwZpyMFj2AHZR2H9cmjWP8AkEzfh/6EKu1S1j/kEzfh/wChCrh8SIn8DOVooor0jywooooAKKKKACiiigAooooAKKKKAClHWkpR1oA7SL/Up/uinU2L/Up/uinV5R6y2CiiigZBdWVrex+XeW8U6ekiBsfnVP8AsUQ86de3Vp6IJPMT/vl84H0xWnRQBmeZrVt9+G1v09YmML/98tkH/voUf29axcX6T6ee5uo9qj/gYyn61p0detADIpo54xJBIsiHoyMCD+IpEnilllijkVpISBIoPKkjIz+BqnLoWnySGWOD7NKestsxiY/UrjP45rzmy8J+OofHt3r1ncW9vHLLtK3k+7z4V4UMEB52gemDQB6tRWIPERAxJaAN/EFlyAfrij/hIx/z6/8AkT/61a+xn2MvbU+5t0Vif8JGP+fX/wAif/Wo/wCEjH/Pr/5E/wDrUexn2F7an3NuisT/AISMf8+v/kT/AOtR/wAJGP8An1/8if8A1qPYz7B7an3NuisT/hIx/wA+v/kT/wCtR/wkY/59f/In/wBaj2M+we2p9zborE/4SMf8+v8A5E/+tR/wkY/59f8AyJ/9aj2M+we2p9zborE/4SMf8+v/AJE/+tR/wkY/59f/ACJ/9aj2M+we2p9zborE/wCEjH/Pr/5E/wDrUf8ACRj/AJ9f/In/ANaj2M+we2p9zboqtYXf22283Zs+YjGc1ZrNpp2ZqmmroKKKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKoa7/wAgG8/65Gr9UNd/5AN5/wBcjSexnV/hy9GX6KKKZoFFFFABRRRQAVl6w8sYjaN2UdDg9a1Kp6pF5lm3qvNc+Ji5UZJbmlJpTVzD+1T/APPZ/wA6PtU//PZ/zqGivl/a1P5n956/JHsTfap/+ez/AJ0fap/+ez/nUNFHtan8z+8OSPYm+1T/APPZ/wA6PtU//PZ/zqGij2tT+Z/eHJHsOZ2adJmYmSNSqOTyoOM4+uB+VSfap/8Ans/51DRS9rU/mf3h7OHYm+1T/wDPZ/zo+1T/APPZ/wA6hop+1qfzP7w5I9ib7VP/AM9n/Oj7VP8A89n/ADqGij2tT+Z/eHJHsTfap/8Ans/51NaXM5uUBkdsnGM1Tq/pUW+7DdlGa7ME51K6TbstTGvyxpvQ3h90ZqlqOom1KW9rH597Nnyoc4GO7MeyjufwHNGo6ibUpb2sfn3s2fKhzgY7sx7KO5/Ac0unacLIPLNJ593NgzTkYLHsAOyjsP65NfSHlBp2nCyDyzSefdzYM05GCx7ADso7D+uTV2iigApk0KXELRSjKN1GafRQG5R/sax/54/+PGj+xrH/AJ4/+PGr1FXzy7kezh2KP9jWP/PH/wAeNH9jWP8Azx/8eNXqKOeXcPZw7FH+xrH/AJ4/+PGj+xrH/nj/AOPGr1FHPLuHs4dij/Y1j/zx/wDHjR/Y1j/zx/8AHjV6ijnl3D2cOxR/sax/54/+PGj+xrH/AJ4/+PGr1FHPLuHs4dij/Y1j/wA8f/HjR/Y1j/zx/wDHjV6ijnl3D2cOxR/sax/54/8Ajxo/sax/54/+PGr1FHPLuHs4dhANqgDoBilooqCwooooAKKKKACiiigAooooA4qT/WN9TTadJ/rG+ppteoeSFFFFMQUUUUAFFFFABRRRQAUUUUAFFFFAHS6F/wAg3/gZrSrN0L/kG/8AAzWlXm1PjZ6dP4EFFFFQaBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVQ13/AJAN5/1yNX6oa7/yAbz/AK5Gk9jOr/Dl6Mv0UUUzQKKKZNNHb28k0zbY41Lu2OgAyTQA+isxdftGUFYNQIIyD/Z0/P8A45S/27b9rXUD/wBuMv8A8TQBpU2Rd8bA9xWf/bkXaz1A/wDbnJ/hXH+OPiNc+FrzS3ttPeS3uPNE8VzG0TNt2YKt68nsaANeRNkrL/dOKZUNpqo1q1TUEsrqzWYZ8q6j2N9R6j0Pepq+QrQ9nUlDse3TlzRTCiiisSwrKPibSF0V9Wa8AsY38tpTG3DbtuNuM9SO3v0rVrgpPDGrNo72AtsxNatMV81f+PnYYwnXGNpVvTIralGEn7zt/WpnOTS0R3tFed6h4U1WafV5I4LqS7njuglwJbdY5UdT5cedvmnGVG1iFG3IPAFakmg3OnatHdabpu62gv1uBb27RpuBtmjZgCwAO5gT0J5PNW6UEviFzyvsdhRWd4etp7Pw/Z295GI5448OgYNtPpkda0awkrOxondXCiiipGFXbe7NnGI7aITXtwT5UOeMDgsx7KO5/Ac1l3119jspJwFyoGN5wuScDJ7DJ5PpXQ6Fpa6fZ75H8+6mAMs5H3vQD0Udh/XJr28rp/FP5HBi5bRJ9O04WYeWaTz7ubBmnIwWPYAdlHYf1yau0UV7RwBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcVJ/rG+pptOk/1jfU02vUPJCgnAJooIypAOPcdqBGDo3iOXVQ8xtIFtVVizw3YlkgI/glj2gqxHYbsEEVdtde0+7k8uOSVJNzLsnt5ImBChjw6g9CD7846GqD+HLy5laa91KJrlbd4Ibu3tfJnUMCMs4c7sZzgBRnnioIvBoSwuYBdxQyXEyyl7W28tV+TY4A3E/MpPJJOTnmo9809y5q2/iDTbq4hhgmkZpgCjGCQISV3BSxXaGxztJz7Uz+3VTxDcabNCI44bcT/aDJ19RjHGBznPrVRPCcEXiFtST7HtaUTYayRplYKFwsp6LwDjGfQipNU8NDU7ieU3RiMzR5Ajz8oVldevO5XI9uDzT96wvduR6b4vs7rTorm+iks5JHcGII0vlBXKbnZVwg46tgdeeDVo+J9JWSdGuXXyFlLsYJAv7vO8BtuGIweAScc1lXPgaKe7acy2kjSNJvNzYJOVVpWkHl7jhWG9hkhgeMrxVt/CiSWj20l0TG8t1IcR44nDDHXtu698dqV522Kap336/gbMF5BcTSRQybnjVWYbSMBgSv8AI1PWXo+lXOnzTzXt5HdSzJGmY4PKVQgIHG5uufX/AAGpWhkFFFFAHS6F/wAg3/gZrSrn9K0bT7y0M1zaRySM5yzdTwKvf8I7pP8Az4xfrXm1PjZ6dP4EaVFZv/CO6T/z4xfrR/wjuk/8+MX61BoaVFZv/CO6T/z4xfrR/wAI7pP/AD4xfrQBpUVm/wDCO6T/AM+MX60f8I7pP/PjF+tAGlRWb/wjuk/8+MX60f8ACO6T/wA+MX60AaVFYf8AZ1pp/iLTzZQLD5kcwfbn5hhetblABRRRQAUUUUAFFFFABVDXf+QDef8AXI1fqhrv/IBvP+uRpPYzq/w5ejL9FFFM0CqOt/8AIv6j/wBesv8A6AavVR1v/kX9R/69Zf8A0A0AWrb/AI9Yv9wfyqSo7b/j1i/3B/KpKACql1pdje3ltdXlpFPPa7vIeRd3l7sZIz3+Uc1booAytaiyiSDscVjV0uoRebZuO+Miuar53MqfLW5u56eFleFuwUUUV5h1hRRRQAUUUUAFFFFABRRRQBNbQie4WNgGVj8wIzkVeQt4eYI5LaUxwrHk2h9D/se/8P06N0eLfcF/7orbZQ6lWAZSMEEcEV9TgqfJQj56nkV5c1Rig5GRyKKxgW8PuFcltKY4VjybQ+h/6Z+/8P06bIORkciuwwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4qT/WN9TTadJ/rG+ppteoeSFFFFMQUUUUAFFFFABRRRQAUUUUAFFFFAHS6F/wAg3/gZrSrN0L/kG/8AAzWlXm1PjZ6dP4EFFFFQaBRRRQAUUUUAFFFFAGbef8jBpn+5N/Ja0qzbz/kYNM/3Jv5LWlQAUUUUAFFFFABRRRQAVQ13/kA3n/XI1fqhrv8AyAbz/rkaT2M6v8OXoy/RRRTNAqjrf/Iv6j/16y/+gGr1Udb/AORf1H/r1l/9ANAFq2/49Yv9wfyqSo7b/j1i/wBwfyqSgAooooARhuUiuXuI/KuHTGMHiuprMvdMaeYyRkAnrmuDG4eVeC5d0dGHqqnJ3MSitH+xpv76/lR/Y0399fyryv7Or+R2fWqZnUVo/wBjTf31/Kj+xpv76/lR/Z1fyD61TM6itH+xpv76/lR/Y0399fyo/s6v5B9apmdRWj/Y0399fyo/sab++v5Uf2dX8g+tUzOpa0P7Gm/vr+VOXRptwy649hTWW1762D61TLekR7bXf/eOa0KZDEIYVQdhT6+iSSVkeW9RGUOpVgGUjBBHBFY4LeHnCuS2lMcKx5NofQ/9M/f+H6dNmkZQ6lXAZSMEEZBFMBQcjI5FFYwLeHnCuS2lMcKx5NofQ/8ATP3/AIfp02QcjI5FABRRVXUpng0+WSJtrrjBx7imld2E3ZXLVFct/bF9/wA9v/HRR/bF9/z2/wDHRW/sJHP9YgdTRXLf2xff89v/AB0Uf2xff89v/HRR7CQfWIHU0Vy39sX3/Pb/AMdFH9sX3/Pb/wAdFHsJB9YgdTRXLf2xff8APb/x0Uf2xff89v8Ax0UewkH1iB1NFct/bF9/z2/8dFH9sX3/AD2/8dFHsJB9YgdTRXLf2xff89v/AB0Uf2xff89v/HRR7CQfWIHU0Vy39sX3/Pb/AMdFH9sX3/Pb/wAdFHsJB9YgdTRTYyTGpPUgU6uc6QooooAKKKKACimTTRW8TSzyJFGoyzuwAH4ms7+2HuuNIs5LoHpPJ+6h/wC+iMt/wEH60AalcFJ8WdOg1+TRp9I1M30c5g8uBEfc+ccfMMg11P8AZdzec6tevIv/AD722Yo/oSDub8Tg+lVbPwbo1j4nk1y2tUjuWhWJVVQFjxkFgPUjAz7e5oArf2PfSfOYdm7nazjI9uDR/Yl7/wA81/77FdPRW/t5HP8AV4HMf2Je/wDPNf8AvsUf2Je/881/77FdPRR7eYfV4HMf2Je/881/77FH9iXv/PNf++xXT0Ue3mH1eBzH9iXv/PNf++xR/Yl7/wA81/77FdPRR7eYfV4HMf2Je/8APNf++xR/Yl7/AM81/wC+xXT0Ue3mH1eBzH9iXv8AzzX/AL7FH9iXv/PNf++xXT0Ue3mH1eBzH9iXv/PNf++xR/Yl7/zzX/vsV09FHt5h9XgUtKtpLWz8uYYbcTwc1doorFu7ubxXKrIKKKKQwooooAKKKKACiiigDNvP+Rg0z/cm/ktaVZt5/wAjBpn+5N/Ja0qACiiigAooooAKKKKACqGu/wDIBvP+uRq/VDXf+QDef9cjSexnV/hy9GX6KKKZoFUdb/5F/Uf+vWX/ANANXqo63/yL+o/9esv/AKAaALVt/wAesX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFACModSrgMpGCCMgiscFvDzhXJbSmOFY8m0Pof+mfv/AA/Tps0jKHUq4DKRggjIIoAUHIyORVLWP+QTN+H/AKEKpgt4fcK5LaUxwrHk2p9D/se/8P06WNWuIXsLiFJUaVUR2QNyFLcHHocH8quHxIifwM5miiivSPLCiiigAooooAKKKKACiiigAooooAKUdaSlHWgDtIv9Sn+6KdTYv9Sn+6KUkKpLEAAZJPavKPWWwtFZja3FMxTS4pNQcHBMOBGp95D8v4DJ9qT7FqN7zqF59mjP/LCyJB/GQ8n/AICFoGWbzVLOxcRzzDzm+7CgLyN9FGSaredqt9/x7wJp8R/5aXGHkP0QHA/En6Vcs9PtLBCtnAkW7liB8zH1J6k+5qxQBnRaJarKs92XvrhTkS3J3bT6qv3V/ACtGiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAzbz/kYNM/3Jv5LWlWbef8AIwaZ/uTfyWtKgAooooAKKKKACiiigAqhrv8AyAbz/rkav1Q13/kA3n/XI0nsZ1f4cvRl+iiimaBVHW/+Rf1H/r1l/wDQDV6qOt/8i/qP/XrL/wCgGgC1bf8AHrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQ3aSyWcyW3liVkITzV3JnHGR3FeT6J4V8V+GfFFxc6oI7yyvYTFLcQyAqmCCnynBGNu0YGBmvXqpax/wAgmb8P/QhVw+JET+BnK0UUV6R5YVg+bf31zqFwmpmwisZjGkRjQxsFVWLSFgWwcn7rLgYrerOudB067vGuZ4XLvjzFWZ1SXHTegIV+OPmB4AFS07lJoyrjxbNErmLThNm6e2h2SSPvKZLMRHGxAwBjAPJ5wOaW58XPB9hIsObjyhLDI8izQF32DcnlnAz3crnBxmtWbQtOnhMbwsoM7XAaOZ0dZG6srKQy5yRwRwcVE3hnSWkib7My+UqKqrM6rhG3KSoOCQeckE1KU9Lvt/wR3j2MCz8ZX6WNqtxpz3t0yNNP9nWV8J5jKoXZEQT8p4baOBzycaut6nd2WvaSsExS0dZHuY9gO9d0agkkZGPMzx6c1YbwtpDqitbPtTcNonkAZWbcUYbvmXP8JyvXirt3ptpfMGuofMIieEfMR8j43Dg99o/KhKSWu47wvscsviy9t7693xNeLPMn2GFY3wke1uSY0djkJu+6fvDoOavL4suHkU/2U0cS/Z/P86QxyRmZtgAQpzg9clePyrRfw3pTxon2Zk2BAjRTOjIEBVdrKQV4JHB5B5zT00HTI4zGlqFRhFkB2H+rbcnfseffvmmlJPcV49iDT9f+3XFnF9m8v7Stw2fMzt8qQJ6c53Z9vetis600HTrG+N5bQMsxMmCZnYLvYM+1SSFBIzgAVo1SvbUTtfQKM45opR1pknQR32oXkKDTrLyI9o/f3oK/lGPmP47aeNEjnYPqs8moOOdkvEQ+kY4/PJ960Yv9Sn+6KdXlHrLYRVCqFUAKBgADpS0UUDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAM28/5GDTP9yb+S1pVm3n/IwaZ/uTfyWtKgAooooAKKKKACiiigAqhrv/IBvP8Arkav1Q13/kA3n/XI0nsZ1f4cvRl+iiimaBVHW/8AkX9R/wCvWX/0A1eqjrf/ACL+o/8AXrL/AOgGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVS1j/kEzfh/6EKu1S1j/AJBM34f+hCrh8SIn8DOVooor0jywooooAKKKKACiiigAooooAKKKKAClHWkpR1oA7SL/AFKf7op1Ni/1Kf7op1eUestgooooGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFMmmit4XmuJEiiQbmd2Cqo9ST0rnbz4jeELFikniGxlkUgGO1k+0OCexWPcf0pNpbjSb2Olori5vipoKjNpa6xd84/d6ZLH/6MC1Wk+KaFT9m8La3IccFzbRqfzmz+lYvEUY7zX3o1WHrS2g/uZ3tFefP8Tb8/6nwrP/22vol/9B3VEPiXrZ/5lW3HPfVf/tVZvGYdfbRosHiH9hi+I/iNbaN42g0+40q9ee2LIoj2nzvMC7SvNehxlmjUuuxiASuc4PpmvINS1+71PxRpuuXHhO3+06erBR/anD5+7n91/CSSPc1tn4l62OnhW3PPbVf/ALVR9dw386H9SxH8jPRqK89T4m34/wBd4Vn/AO2N9E3/AKFtqaP4poFH2nwtrcZxyYzbSKPymz+lUsXh39tfeS8LXX2H9x3lFcXF8U9BIzd2usWnOP3mmSyf+iw1X7L4i+EL5gkXiCyikYkCO6k+zuT6BZNpP5VtGpCXwtMxlTnH4lY6WimxyJNGskLrIjDKspyCPUGnVZAVQ13/AJAN5/1yNX6oa7/yAbz/AK5Gk9jOr/Dl6Mv0UUUzQKo63/yL+o/9esv/AKAavVR1v/kX9R/69Zf/AEA0AWrb/j1i/wBwfyqSo7b/AI9Yv9wfyqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKhu7cXVq8LMVDY5H1zU1FNOzuJq6szH/4R2L/nu/8A3yKP+Edi/wCe7/8AfIrYorT2s+5n7GHYx/8AhHYv+e7/APfIo/4R2L/nu/8A3yK2KKPaz7h7GHYx/wDhHYv+e7/98ij/AIR2L/nu/wD3yK2KKPaz7h7GHYx/+Edi/wCe7/8AfIo/4R2L/nu//fIrYoo9rPuHsYdjH/4R2L/nu/8A3yKP+Edi/wCe7/8AfIrYoo9rPuHsYdjH/wCEdi/57v8A98ij/hHYv+e7/wDfIrYoo9rPuHsYdjH/AOEdi/57v/3yKP8AhHov+e7/APfIrYoo9rPuHsYdhFXagXrgYpaKKyNQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArjvin4vm8FeAbrUrExi/lkjtrTzR8vmOwGSTwNq7m+bj5cHrXY1wfxksBf8Aw8k+cJJDeWzxlkDqSZlQhlPDDDnipk7RbKirySPL11LQtQnQ+J724n1DdkjxASrbv9lWxH/37GK6eBYVhUWwQRY+XywNv4YriZNCudOjEX2C9t4FJYtpFyJoSc55tpsgDJzhQcYHNZUllYWrEi70m3ctndf6ZNp0n4urIPxC18lOCrPm5m/x/wCGPrIT9ircqX4f8Oen0V53HDeTRounyRznIwbDxTMwPfAV1I7dKuQr4njXEkPiCMDPEVzYT/q6qaweH/vL52/zZusR/dfy/pHcUVxD32uxck+JMc/esLR//QTTF17WQxG7WuP+emg7s/8AfLCl9Xk9mv6+Q/rEez/r5ndUVw//AAkGt5+/f4/7FyXP/oymtr2ssQM60cn/AJZ6Dtx/30xo+rS7/n/kH1iPb8v8zuqK4hL3XpWyG8SY4wFsLNP/AEI0ky+J5FxHD4glBI4lubCAY+qKxo+ru+sl94fWP7r+47io7hYHgYXSxtFj5hIBtx75rz+WG7giYX80Vuckk3/imZQPqqKAfpmqsVjYXRV/tOmXL7sh7DS5r+T8JHZ1/EirWHW9/wAL/kyHiHtb8bfmjo21TQ9OndvC99dwX2d23w+xbJyPvquYvr5gxXrXws8YT+NfAdnqN/s+3oXguvL+60iMRuBAwQww3y8fNgdK8Yg0K51CED+z7iWP76Prc4EYJ9LaLC/mFr1f4NWwtvA65bfLJdXDSuBtDMJWXIUcKMKBgccV7mWz1lDmvbzueHmMdIztb5WPQqoa7/yAbz/rkav1Q13/AJAN5/1yNeu9jxKv8OXoy/RRRTNAqjrf/Iv6j/16y/8AoBq9VHW/+Rf1H/r1l/8AQDQBatv+PWL/AHB/KpKjtv8Aj1i/3B/KpKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqK6n+zWrzbd2wZxnGalqpqn/ILn/wB3+tVFXkkTJ2i2Z/8AwkX/AE7f+P8A/wBaj/hIv+nb/wAf/wDrVh0V2+xh2OD21Tubn/CRf9O3/j//ANaj/hIv+nb/AMf/APrVh0Uexh2D21Tubn/CRf8ATt/4/wD/AFqP+Ei/6dv/AB//AOtWHRR7GHYPbVO5uf8ACRf9O3/j/wD9aj/hIv8Ap2/8f/8ArVh0Uexh2D21Tubn/CRf9O3/AI//APWo/wCEi/6dv/H/AP61YdFHsYdg9tU7m5/wkX/Tt/4//wDWo/4SL/p2/wDH/wD61YdFHsYdg9tU7m5/wkX/AE7f+P8A/wBaj/hIv+nb/wAf/wDrVh0Uexh2D21TudrE/mQo+MblBx6Zp1RWv/HnD/1zX+VS1wvc9BbBRRRSGFFFFABRRRQAUUUUAFFFFABVe9v7fT4Vlui4VnCKI42kZmPQBVBJqxWbrH+s07/r9T/0FqAE/wCEgsv+ed//AOC64/8AiKP+Egsv+ed//wCC64/+IrTooAzP+Egsv+ed/wD+C64/+Io/4SCy/wCed/8A+C64/wDiK06KAMz/AISCy/553/8A4Lrj/wCIo/4SCy/553//AILrj/4itOigDM/4SCy/553/AP4Lrj/4ij/hILL/AJ53/wD4Lrj/AOIrTooAzP8AhILL/nnf/wDguuP/AIij/hILL/nnf/8AguuP/iK06KAMz/hILL/nnf8A/guuP/iKP+Egsv8Annf/APguuP8A4itOigDk/FfjZdD8N3GoWFtPLNCyYS5s5o0ILgHLFQBwTjnr61yPiLx5Y+Mfhvd/ZrS6tp47m08xHjLRg/aYukgG388H2r0/VNKs9a097LU4RPbOys8ZJAbawYZx7gVyvxLtoLP4bT29nDHBDHc2gSOJAqqPtMXQCoqfA/Qun8a9Tk6KKK+CPvClc6Lpd4c3mm2c/wD11gVv5iqZ8JaDjEemQwDOcW+Yv/QSK2aKtVJrZsh04PdGP/wi2l9lu1/3b6cfyej/AIRewH+rn1NP93VLnH5eZitiin7Wp/Mxezh2Rjf8I1BuJ/tHViMfd/tCX/HNL/wi9gfvzalJ/v6pckfl5mK2KKPaz7h7OHYx/wDhFdK7pdN/vX05/m9A8JaDtAk0u3mGMYnXzf8A0LNbFFHtan8z+8PZU/5UU7bR9MssfY9OtLfHTyoFX+Qq5RRUNt7lpJbBW/8AC68itPBMJlWY7rm5A8qF5P8AlvJ/dBxWBXT/AAm/5EyP/r4uf/R8le3k/wAc/Q8POb+zjY6v+2rX/nnef+AM3/xFVNV1KG60q4gghvGkkQqo+xTDJ+pXFbVFfR2Z8vKFSSabWvl/wQooopmwVR1v/kX9R/69Zf8A0A1eqjrf/Iv6j/16y/8AoBoAtW3/AB6xf7g/lUlR23/HrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVTVP+QXP/u/1q3VTVP8AkFz/AO7/AFqofEiZ/Czk6KKK9M8oKKz9Y1X+ybVJVgadmfBRTjCgFnboeiqTjucDvUGp+I7TTnSJd087vCNiq21RI4UEuAVHUkAkE44pXRXKzXorKs/EFrqGriysw0i+S8hlKMoO1lX5cgbgcn5gSOKh/wCEpsYhL9t8yFkmmjVI4nmLrEQGbCqSPvA+1LmXcOV3tY26KzJPEelR3SwNcks2wBlidkywyo3gbckHOM5xz0qMeKNJaGKVZ5THMu9XFtLgKejsdvyrz944HB54NO6FZmvRWFfeLLO1uTBD+9KGQSSOHSNSiknDBCGwRg7c4PX0q1L4i0yCaeOadk8hXZ3ML7PkGWAfbtZgAcqCTweODS5kPlfY06Kw5/Funx+R5PnSeZdLbvugkTy9ylg5BX7uBweh9eDV/TdXs9WV2snkOwAkSQvEcEZBAcAkHsRwcUJp7CaaLtFFFUI7K1/484f+ua/yqWorX/jzh/65r/Kpa8t7nrLYKKKKQwooooAKKKKACiiigAooooAKzdY/1mnf9fqf+gtWlWbrH+s07/r9T/0FqANKiiigAooooAKKKKACiiigAooooAKKKKACuN+KzrF8Ob6WQ7Y4praR2/uqtxGSfwAJrsqZLEk0TRyKGVhggjIIpSXMmhxfK0zx9HWSNXjYOjAFWU5BHqDTq6a++FOjSTPNpUt5pLuSxWyuGSMH1ERzGP8AvmsuX4deIbcn7F4iSdewvbJWJ/GMp/KvmamUVk/caZ9LTzei176aM2ipZPC/jS3P/HtpN4PUXEkB/LY/86qy6d4ugU7/AA3HKQOBBfqc/wDfSrXNLLsUvs/ijpjmOFf2vwZLRVYr4iT7/hLUW/65z2x/nKKaZdZH3vCusD/vwf5S1n9RxP8AIzT69hv50W6Kp/adV6f8IxrGf+ucX/xyl87WT93wprB/78D+ctL6lif5GP67h/50W6KqgeIn+54S1Ff+uk9sP5SmnxWHi2dRt8NJESORNfqMf98q1UsBiX9gl4/DL7ZPRTo/DPjS4P8Ax6aTaD1NxJOfy2J/OrsXw88RXJH2zxAluuORZWSqR+Mhf+VbRyvEy3SXzMZZpho7Nv5f5mc7rHGzyMERQSzMcAAdya6f4RMJfAtvKmSks08iHGNytM5B/EEGmWXwn0lZVl1Wa81R1O7F7cM6Z/65jCf+O13FpaRWUCxQIERRgADAAr2sDgnhruTu2eNjsasTZRVkieiiivTPMCiiigAqjrf/ACL+o/8AXrL/AOgGr1Udb/5F/Uf+vWX/ANANAFq2/wCPWL/cH8qkqO2/49Yv9wfyqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoopGZUQu7BVUZJJwAKAI7lnW1laF443CEq8oyinHU8jj8a8j0nW/GGueNJLXW7gw2dpA1w0MEflxSoflQgnllJIIyT0r0pVbXnEkoK6YpykZGDdHszD+56Dv1PHW1qiJ/Z877V37Nu7HOM9KqHxImfws5WiiivTPKMzUNEi1S/jmu57gRRRFEignkh5YjJLIwJ4AGPrWYPCMyRCCLUgLdmgMqvAWZvJcFAG3cfKqqeDnGeOldNRU8qK5mjE0jQJtMuoHkvVmhtbdra3jEGwqhZSNzbjuI24zgZ9B3anhnZfG5+15ybo7fL/57FT1z22/jntW7RS5U1Z/1cOZ3ucYfD+ow6jHY2iyfYDNDNNPIke0ukaqSp8wtg7F+Up1yd2KsTeB4pYrBTLaStbWUdm73NgkxKp0aMMcI3Lddw5GQcV1dFPlQ+d9DmJ/B8ssT2y6iqWYad4Y/s+WQy7t2W3cgFjjgccHPWo5/A0MtxfsktpGt2JiJPsCGdWlUg5lJyVBYkAAHoN2OD1dFLkiwU5LYwrnw201ys8d4I2WSBxmHd/q1ZT37hj9PepdA0FtFa5JuI2WYrtht4jFDHjPITcwDHPJXAOBx67FFOyu33Ju7WCiiiqEdBbeHtJNpETYQ5KL/D7VL/wjukf9A+H/AL5q7a/8ecP/AFzX+VS15b3PWWxm/wDCO6R/0D4f++aP+Ed0j/oHw/8AfNaVFIZm/wDCO6R/0D4f++aP+Ed0j/oHw/8AfNaVFAGb/wAI7pH/AED4f++aP+Ed0j/oHw/981pUUAZv/CO6R/0D4f8Avmj/AIR3SP8AoHw/981pUUAZv/CO6R/0D4f++aP+Ed0j/oHw/wDfNaVFAGb/AMI7pH/QPh/75p8OhaXbzpNDYwrJGcowXlT0yKv0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB5xrGva7/wAJRrVrpV7qT3Fnc2qWdjBpyy27h1jLiWXyjsHLcmRcDkVLNrmsP4Z1LxWmpNGlldTKmliGMxNHFKYyrEr5m9gpOQwAJHBHB7i3061tL27u7eLZPesrzvuJ3lVCjgnA4AHFZ8nhHRJdSe9e0bzJJhcSRC4kEMkoxh2hDeWzcA5Kk5APUULp/XYb1MO51PVLLx7HFql3cWWm3EyJZKLRJba4BQAxtIBvjl3kkZO0hcAGptV/tO+8bXOmWWsXGmwQ6UlyiwQwNukMjrlvMjY4wo4BFbD+FNGk1oaq1oxuhMJ/9fJ5ZlC7BIYt2wuF43bc+9GqeFdK1i+N5epdCdoPs7Pb3s0G+PJO1hG6hhlj19aXRfP8v8w6v5fn/keXP8XZZNd0F3t0hthAh1BAfld5FGSOpwvBHfqK3IviLqNh4ZW/1PS7O6maW7fZb3TEiKGUrwqxM3GCNxATABZ1LAVu33grw83irTZxpqIxy5SOR0jLRhdh2KQvGMdORwcircnw88Myo6NYzBHEoZUvZ1BWU7pEwHHyE87Pu55ApsStazMSx8Z3sOq6k99bJPpS6tFZxymQJJbiWOPYAgX5xufklgRnjOKqX3xXFrYag50qNJI4Fks3S4MiS7pNnLbApI6ny2kXgjdxXXR+C9Biv/tiWT+aZRMwa5lKPIFCqzIW2sQAMEg46jmq6fDzwxHG0Y052UxCFQ93M3loGDKqZf5ACMjbjHOMZNH9fh/mLX+vX/I4rwv8VbiTw/aWY0u51XVLeN/tAiSVmeNCoBHlxvliGA+bauQcsMivV4pPNhSQKyb1DbXGGGexHY1yOi+CPD154X0vz9P5+z5dkmkRpfMwziQqwLqT1Vsj2rsAAqgKAABgAdqelh63FqnI7DXLdAx2m2lJXPBIaPn9TVyqMv8AyMFr/wBes3/ocVSzOpt81+ZeooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo63/AMi/qP8A16y/+gGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRVW8vVs9u5Sd3pUznGEeaWw4xcnZFqisv+24/+ebUf23H/wA82rm+uYf+Y19hU7GpRWX/AG3H/wA82o/tuP8A55tR9cw/8wewqdjUorL/ALbj/wCebUf23H/zzaj65h/5g9hU7GpRWX/bcf8Azzaj+24/+ebUfXMP/MHsKnY1KKy/7bj/AOebUf23H/zzaj65h/5g9hU7GpRWX/bcf/PNqP7bj/55tR9cw/8AMHsKnY1KKy/7bj/55tT49YjkkC7GGe5qo4qhJ2UtROjUSu0aNFAORmkZlRC7sFVRkknAArpMgZlRC7sFVRkknAArJVW15xJKCumKcpGRg3R/vMP7noO/U8dRVbXnEkoK6YpykZGDdH+8w/ueg79Tx116ACqmqf8AILn/AN3+tW6r38Tz2MsUYyzDAGfeqj8SJl8LOQorQ/sS9/uL/wB9Cj+xL3+4v/fQr0PaQ7nnezn2M+itD+xL3+4v/fQo/sS9/uL/AN9Cj2kO4ezn2M+itD+xL3+4v/fQo/sS9/uL/wB9Cj2kO4ezn2M+itD+xL3+4v8A30KP7Evf7i/99Cj2kO4ezn2M+itD+xL3+4v/AH0KP7Evf7i/99Cj2kO4ezn2M+itD+xL3+4v/fQo/sS9/uL/AN9Cj2kO4ezn2M+itD+xL3+4v/fQo/sS9/uL/wB9Cj2kO4ezn2Oitf8Ajzh/65r/ACqWo4FKW8aNwVQA/lUlec9z0lsFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBm3n/IwaZ/uTfyWtKs28/wCRg0z/AHJv5LWlQAUUUUAZvhz/AJFfTP8Ar1j/APQRWlWb4c/5FfTP+vWP/wBBFaVABVGX/kYbX/r1m/8AQ4qvVRl/5GG1/wCvWb/0OKkzOpt81+ZeooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo63/AMi/qP8A16y/+gGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABWfq8W+03DqpzWhUdwgkgZT3FZ1Ye0puPcqEuWSZytFKwKsQeoOKSvj9j2wooopDCiiigAooooAKKKKACiiigAqxZR+ZdoPfJqvWlpKKryTSHaqLyWOAO+a9DL6fPXT7anNiZWp27m0WWOMs7BVUZJJwAKylVtecSSgrpinKRkYN0f7zD+56Dv1PHUVW15xJKCulqcpGRg3R/vMP7noO/U8ddevpTygooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAM28/5GDTP9yb+S1pVm3n/IwaZ/uTfyWtKgAooooAzfDn/Ir6Z/16x/+gitKs3w5/yK+mf9esf/AKCK0qACqEv/ACMVr/16zf8AocVX6oS/8jFa/wDXpN/6HFSZnU2+a/Mv0UUUzQKKKKACqOt/8i/qP/XrL/6AavVR1wZ8PaiP+nWX/wBANAFq2/49Yv8AcH8qkrJt9A042sRMUvKDpcSDt/vVJ/wj+n/3bgfS7l/+KoA0qztT13TdHntodUu47U3ZKxNKcKWGOM9B170n9gWP/T2PpezD/wBnrl/GXw5HidtPhtryS2ghd2neaaSZsEDAUMxA7+n40Aat8gW7YryrfMCDnOarUlr4XtPC+mxWthJcSxjgtPMX59h0X6AClr5XF0/Z15I9ijLmppnH6vrOsQ6+1rp0qsFvoUWFkXDoYHkZM4yNxXr2+nFSx63d6leSWtneGKO4vxDFOI1LQxi2WUgAjG4nI+YHGTxxit59HsXvxevBm4Eqy797ffCFAcZx91iKjPh/TDFcxi1Ci6uPtMhR2VvNwBvUg5VuByuP1qVUp226fjoU4y6P+tf6/wCGOXvdduoLm3sV1jUCv27y5Jl00/afLMMjAbDDhvnThkTp+dJ/bOrXEGnrZ3eo3ENzeTpHNDDbx3E8SJkErKqouGDDopIHSuot/D2m200U0cMjzRS+cs008kjlthTJZmJICsQASQM8Uy58NaZdTGZ4545DKZt0F1LEQ5XaSNjDGR1x171XtaXb8F/X9fMjkm+plx/2rdata6e2qalZD7E87+ZHamZm8wAbtqMnQ/w+2eayrnxLrD2t0I7hllijt0Bt448s5unhdl35GWCjAY4FdPJ4X02Ro2Y33mRoUWQajcBypOSCwfJGfUmny+GtJmgaE2mxGjjixFI8eFjYsgG0jGGJORzRGpTVrr8F3Hyyu9f60/4Jh3V7rtnDYCF9Q8+a+CGLUltSZUETuUBh4GSuMnkH24Jd6lfX1jq19YavcW0drcIsCxRQkFWjiOG3oxyCzdx156VvQaBYwSROPtUrQyCWM3F5NNsbaVyN7HHDHj/AU9dE05IbyJLYKl7N59woYje/HzdePujpS9pTvt+Hmv0Hyytv/WpiasdZ06UbtR1E2MUOftkNtBM2/c2TNGEDFQNoHlqDjdkjrXTwuJYI5FdXDKGDKMBsjqKpXmhWF9d/aZ0mWYqEdobiSLzFGcBwjAOOTw2ep9avqoRQqgKoGAAOAKylJSiu5STUrikhQSxAA5JJ6VZ0m3bWYFlmUrpudyRt1uT2Lf7HoO/U8daVxa/bbaS2DbfNG3OAcZ9jwfp3roNKvGZfsV4iw3kK5ZB92Rf76eo/UHg+/s5XTtCU+5w4uXvKJpUUUV65xBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZt5/yMGmf7k38lrSrNvP+Rg0z/cm/ktaVABRRRQBm+HP+RX0z/r1j/8AQRWlWb4c/wCRX0z/AK9Y/wD0EVpUAFUJf+Rjtf8Ar0m/9Diq/VCX/kY7T/r0n/8AQ4qTM6m3zX5l+iiimaBRRRQAVR1v/kX9R/69Zf8A0A1eqjrf/Iv6j/16y/8AoBoAtW3/AB6xf7g/lUlR23/HrF/uD+VSUAFFFFAFLVIvMs2I6rzXPV1cq74mU9xXLyJ5cjIf4TivCzSnaUZ/I9DCS0cRlFFFeOdwUUUUAFFFFABRRRQAUUUUAW9Oj828T/Z5rZvrBL2JMO0M8R3Qzp96NvX3HqOhFUtFi5eQ/QVr19Zhafs6MYnjVpc1Rso2F+8sjWl6ixXsQyyj7si/309R+oPB971Vb+wS+jX52iniO6GdPvRt6+49R0IqOwv3mka0vUWK9iGWQfdkX++nqv6g8H36TIvUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGbef8jBpn+5N/Ja0qzbz/kYNM/3Jv5LWlQAUUUUAZvhz/kV9M/69Y//AEEVpVm+HP8AkV9M/wCvWP8A9BFaVABVCX/kY7T/AK9J/wD0OKr9UJv+RjtP+vSf/wBDipMzqfD81+aL9FFFM0CiiigAqjrf/Iv6j/16y/8AoBq9VHW/+Rf1H/r1l/8AQDQBatv+PWL/AHB/KpKjtv8Aj1i/3B/KpKACiiigArA1K2ZbtmA+Vua36ayK33gDWFehGvHlkaU6jpu6OW8tv8mjy2/ya6jyY/7g/KjyY/7g/KuH+y6XdnR9bn2OX8tv8mjy2/ya6jyY/wC4Pyo8mP8AuD8qP7Lpd2H1ufY5J5EjnihkdVkmz5ak8tgZOKl8tv8AJrC8T6H4kvfHNpfWC26RwH/REaXGQuC2eO+fyr0OONWjVniCMQCV67T6VMctpNtXZyUMyq1ZzjKFuV6ef9flY5ny2/yaPLb/ACa6jyY/7g/KjyY/7g/Kq/sul3Z1/W59jl/Lb/JoETE9K6jyY/7g/KjyY/7op/2ZS7sX1ufYr6dCYbRQw+Y8mrdHSivUOQKq39gl9GvztFPEd0M6fejb+o9R0Iq1RQBRsL95ZGtL1FivYhlkH3ZF/vp6j9QeD73qq39gl9GvztFPEd0M6fejb19x6joRUdhfvLI1peosV7EMsg+7Iv8AfT1H6g8H3AL1RzzLbwNLJnavJxUlVNU/5Bc/+7/WnFXaRMnZNkH9u2f/AE0/75o/t2z/AOmn/fNc1RXb7CBxfWJnS/27Z/8ATT/vmj+3bP8A6af981zVFHsIB9YmdL/btn/00/75o/t2z/6af981zVFHsIB9YmdL/btn/wBNP++aP7ds/wDpp/3zXNUUewgH1iZ0v9u2f/TT/vmj+3bP/pp/3zXNUUewgH1iZ0v9u2f/AE0/75o/t2z/AOmn/fNc1RR7CAfWJnS/27Z/9NP++aP7ds/+mn/fNc1RR7CAfWJnaxuJI1dejAEZp1RWv/HnD/1zX+VS1xPc7lsFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBm3n/IwaZ/uTfyWtKs28/wCRg0z/AHJv5LWlQAUUUUAZvhz/AJFfTP8Ar1j/APQRWlWb4c/5FfTP+vWP/wBBFaVABVCb/kY7P/r0n/8AQ4av1Qm/5GOz/wCvSf8A9DhpMzqfD81+aL9FFFM0CiiigAqjrf8AyL+o/wDXrL/6AavVR1v/AJF/Uf8Ar1l/9ANAFq2/49Yv9wfyqSo7b/j1i/3B/KpKACiiigAooooAKKKKACiiigChdf8AIb0//cl/ktX6oXP/ACHbD/rnN/7JV+kjOHxS9f0QUUUUzQKKKKACiiigAooooAKq39gl7GvztFPEd0Myfejb19x6joRVqigCjYX7yyNaXqLFexDLIPuyL/fT1H6g8H3fqn/ILn/3f60zVreCWzM8832VrYGRLkHBhIHJ+nqOhFeeaN8Tj4l8RXOktHHHbG2YQuuczSKQS3PIBXJA7YqofEiZ/Czaooqtf6hbaZZm5vHZIgyrlI2cksQAAqgkkkgcCvS2PL32LNFVLDVLXUhJ9laTfEQHjmheJ1z0yjgMAexxzUq3cLXr2ivmeONZGTB4ViQDnp1U/lQFmTUVD9rh+2i03/vzH5uzB+7nGfzp088dtbSTzttiiQu7YzgAZJ4ourXDrYkopsUizQpLGco6hlOOoNOpiCiiq630D3T2yMzSxuEcCNiFJXcMnGBx3/DrSAsUUVCt3A99JZq+Z441kdMHhWJAOenVT+VAE1FFFMDsrX/jzh/65r/Kpaitf+POH/rmv8qlry3uestgooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAM28/wCRg0z/AHJv5LWlWbef8jBpn+5N/Ja0qACiiigDN8Of8ivpn/XrH/6CK0qzfDn/ACK+mf8AXrH/AOgitKgAqhN/yMdn/wBes/8A6HDV+qE3/IxWf/XrP/6HFSZnU+H5r80X6KKKZoFFFFABVHW/+Rf1H/r1l/8AQDV6qOt/8i/qP/XrL/6AaALVt/x6xf7g/lUlR23/AB6xf7g/lUlABRRRQAUUUUAFFFFABRRRQBQuP+Q/Y/8AXGb+aVfqhP8A8jBZf9e8/wDOOr9JGcPil6/ogooopmgUUUUAFFFFABRRRQAUUUUAFYWr+HdIdm1P+zrdb6EEpcIm18ng5I68E9a3aqap/wAguf8A3f61UPiRM/hZydZPiW2ubrRwtnA1xKlxBL5SMqlgkqscFiBnAPU1rUV6T1PLTs7nG61pOp63JJeGxnt4cwxvZloGlnRS5JIYtEcF1IDH+E9Diqr+GdT+wNEkFx5XkQKYmlgaRkWaVjHgqI8hXX5SNn8OSBmu8oqORF+0djz5vCd+kDfZbKVGezeJWkkgMir54fyjhQg3JuAUAoM4JxW3pmjXFr4T1Kyjt54jOsvkQTvCGXKYxiICNMnJwCeuSckgdNRR7NWa76B7R3v53OKPhq882e6+zpFdrcW7QXJZd0SCFEkIOeBkHI7471laTYQ6u6w6dYwNDClmt4Yp45Y5WWRjIxZWIbI5OfmIPI5r0qijkXNzfMOd2t5WOAl8KambzUGgjuEuJPPC3AkgSKWNvuRZVfN6YXDEKu3IPAFTt4e1CT7R/Zmnf2SskrtAnmIvk5tTGDhCQvzn+HPrXcUUezVh+0d7nCWHhWbbaxyadMlp9tSSe2ujbbdoikVm2QqFwSyjuTjkDHMY8J3CpcpNpbMj2sUamFoS3yTOQmHyrfL5fDfLhcZBArv6KHBP+vKwvaP+vW5meHba4s9AtoLyGOCVA37qIYVBuOBjcwHGOASB0HGK06KK0MzoLbSJTaRH+1tQHyLwHT0/3Kl/siX/AKC+of8Afaf/ABFXbX/jzh/65r/Kpa8t7nrLYzf7Il/6C+of99p/8RR/ZEv/AEF9Q/77T/4itKikMzf7Il/6C+of99p/8RR/ZEv/AEF9Q/77T/4itKigDN/siX/oL6h/32n/AMRR/ZEv/QX1D/vtP/iK0qKAM3+yJf8AoL6h/wB9p/8AEUf2RL/0F9Q/77T/AOIrSooAzf7Il/6C+of99p/8RR/ZEv8A0F9Q/wC+0/8AiK0qKAM3+yJf+gvqH/faf/EUf2RL/wBBfUP++0/+IrSooAzf7Il/6C+of99p/wDEUf2RL/0F9Q/77T/4itKigDN/siX/AKC+of8Afaf/ABFH9kS/9BfUP++0/wDiK0qKAM3+yJf+gvqH/faf/EUf2RL/ANBfUP8AvtP/AIitKigDN/siX/oL6h/32n/xFH9kS/8AQX1D/vtP/iK0qKAM3+yJf+gvqH/faf8AxFH9kS/9BfUP++0/+IrSooAzf7Il/wCgvqH/AH2n/wARR/ZEv/QX1D/vtP8A4itKigDN/siX/oL6h/32n/xFH9kS/wDQX1D/AL7T/wCIrSooAzf7Il/6C+of99p/8RR/ZEv/AEF9Q/77T/4itKigDN/siX/oL6h/32n/AMRR/ZEv/QX1D/vtP/iK0qKAM3+yJf8AoL6h/wB9p/8AEUf2RL/0F9Q/77T/AOIrSooAz4NIEV9HdS3t1cvErKgmZSBuxnoo9K0KKKACiiigDN8Of8ivpn/XrH/6CK0qzfDn/Ir6Z/16x/8AoIrSoAKoTf8AIxWX/XrP/wChRVfqhP8A8jFZf9es/wD6FFSZnU+H5r80X6KKKZoFFFFABVHW/wDkX9R/69Zf/QDV6qOt/wDIv6j/ANesv/oBoAtW3/HrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAUJ/+Risv+vWf/wBCiq/VCb/kY7P/AK9J/wD0OKr9JGcPil6/ogooopmgUUUUAFFFFABRRRQAUUUUAFVNU/5Bc/8Au/1q3VXUkaTTplRSzFeABknmqj8SJn8LOSoqf7Ddf8+0v/fBo+w3X/PtL/3wa9HmXc8zlfYgoqf7Ddf8+0v/AHwaPsN1/wA+0v8A3waOZdw5X2IKKn+w3X/PtL/3waPsN1/z7S/98GjmXcOV9iCip/sN1/z7S/8AfBo+w3X/AD7S/wDfBo5l3DlfYgoqf7Ddf8+0v/fBo+w3X/PtL/3waOZdw5X2IKKn+w3X/PtL/wB8Gj7Ddf8APtL/AN8GjmXcOV9iCip/sN1/z7S/98Gj7Ddf8+0v/fBo5l3DlfY6u1/484f+ua/yqWo7YFbWIMMEIAQe3FSV5r3PUWwUUUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZvhz/AJFfTP8Ar1j/APQRWlWb4c/5FfTP+vWP/wBBFaVABVCf/kYrL/r2n/8AQoqv1Qn/AORhsv8Ar2n/APQoqTM6nw/Nfmi/RRRTNAooooAKo63/AMi/qP8A16y/+gGr1Udb/wCRf1H/AK9Zf/QDQBatv+PWL/cH8qkqO2/49Yv9wfyqSgAooooAKKKKACiiigAooooAoy/8jDa/9es3/ocVXqpT6RZXN0biaNzKRjcJXXjj0PsPypv9iWP9yX/wIk/+KpamKVRN2S18/wDgF+iqH9iWP9yX/wACJP8A4qj+xLH+5L/4ESf/ABVGpV6nZff/AMAv0VQ/sSx/uS/+BEn/AMVR/Ylj/cl/8CJP/iqNQvU7L7/+AX6Kof2JY/3Jf/AiT/4qj+xLH+5L/wCBEn/xVGoXqdl9/wDwC/RVD+xLH+5L/wCBEn/xVH9iWP8Acl/8CJP/AIqjUL1Oy+//AIBfoqh/Ylj/AHJf/AiT/wCKo/sSx/uS/wDgRJ/8VRqF6nZff/wC/RVD+xLH+5L/AOBEn/xVH9iWP9yX/wACJP8A4qjUL1Oy+/8A4Bfoqh/Ylj/cl/8AAiT/AOKo/sSx/uS/+BEn/wAVRqF6nZff/wAAv0VQ/sSx/uS/+BEn/wAVR/Ylj/cl/wDAiT/4qjUL1Oy+/wD4Bfoqh/Ylj/cl/wDAiT/4qj+xLH+5L/4ESf8AxVGoXqdl9/8AwC/RVD+xLH+5L/4ESf8AxVH9iWP9yX/wIk/+Ko1C9Tsvv/4Bfoqh/Ylj/cl/8CJP/iqP7Esf7kv/AIESf/FUahep2X3/APAL9FUP7Esf7kv/AIESf/FUf2JY/wByX/wIk/8AiqNQvU7L7/8AgF+iqH9iWP8Acl/8CJP/AIqj+xLH+5L/AOBEn/xVGoXqdl9//AL9FUP7Esf7kv8A4ESf/FUf2JY/3Jf/AAIk/wDiqNQvU7L7/wDgF+iqH9iWP9yX/wACJP8A4qj+xLH+5L/4ESf/ABVGoXqdl9//AAC/RVD+xLH+5L/4ESf/ABVH9iWP9yX/AMCJP/iqNQvU7L7/APgF+iqH9iWP9yX/AMCJP/iqP7Esf7kv/gRJ/wDFUahep2X3/wDAL9FUP7Esf7kv/gRJ/wDFUf2JY/3Jf/AiT/4qjUL1Oy+//gF+iqH9iWP9yX/wIk/+Ko/sSx/uS/8AgRJ/8VRqF6nZff8A8Av0VQ/sSx/uS/8AgRJ/8VR/Ylj/AHJf/AiT/wCKo1C9Tsvv/wCAX6Kof2JY/wByX/wIk/8AiqP7Esf7kv8A4ESf/FUahep2X3/8Av0VQ/sSx/uS/wDgRJ/8VR/Ylj/cl/8AAiT/AOKo1C9Tsvv/AOAX6Kof2JY/3Jf/AAIk/wDiqP7Esf7kv/gRJ/8AFUahep2X3/8AAL9FUP7Esf7kv/gRJ/8AFUf2JY/3Jf8AwIk/+Ko1C9Tsvv8A+AX6Kof2JY/3Jf8AwIk/+Ko/sSx/uS/+BEn/AMVRqF6nZff/AMAv0VQ/sSx/uS/+BEn/AMVR/Ylj/cl/8CJP/iqNQvU7L7/+AX6Kof2JY/3Jf/AiT/4qj+xLH+5L/wCBEn/xVGoXqdl9/wDwC/RVD+xLH+5L/wCBEn/xVH9iWP8Acl/8CJP/AIqjUL1Oy+//AIBfoqh/Ylj/AHJf/AiT/wCKo/sSx/uS/wDgRJ/8VRqF6nZff/wC/RVD+xLH+5L/AOBEn/xVH9iWP9yX/wACJP8A4qjUL1Oy+/8A4Bfoqh/Ylj/cl/8AAiT/AOKo/sSx/uS/+BEn/wAVRqF6nZff/wAAv0VQ/sSx/uS/+BEn/wAVR/Ylj/cl/wDAiT/4qjUL1Oy+/wD4Azw5/wAivpn/AF6x/wDoIrSrn/D+j2cnhvTndJNzW0ZOJ3H8I960f7Esf7kv/gRJ/wDFUahep2X3/wDAL9UJ/wDkYbL/AK9p/wD0KKj+xLH+5L/4ESf/ABVVlsYLPxFaG3VwWtpgd0jN/FH6k0nczm52V0t11815GxRRRVHQFFFFABVHW/8AkX9R/wCvWX/0A1eqjrf/ACL+o/8AXrL/AOgGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVX1Cd7axkmjxuXGM/UCrFUtY/wCQTN+H/oQqo6ySJm7RbMn+37v+7H+VH9v3f92P8qy6K7/Zw7Hne1n3NT+37v8Aux/lR/b93/dj/Ksuij2cOwe1n3NT+37v+7H+VH9v3f8Adj/Ksuij2cOwe1n3NT+37v8Aux/lR/b93/dj/Ksuij2cOwe1n3NT+37v+7H+VH9v3f8Adj/Ksuij2cOwe1n3NT+37v8Aux/lR/b93/dj/Ksuij2cOwe1n3NT+37v+7H+VH9vXf8Adj/KsulHWj2cOwe1n3O0Qlo1J6kA06mxf6lP90U6vOPTQUUUUAFFFFABRRRQAUUUUAFZl5rcVvIY4l81h1IPAqTV7w2tnhDh5PlX29TXL10UqSkrs5q1VxfLE1/+EhuP+eMf60f8JDcf88Y/1rIoro9lDsc3tZ9zX/4SG4/54x/rR/wkNx/zxj/Wsiij2UOwe1n3Nf8A4SG4/wCeMf60f8JDcf8APGP9ayKKPZQ7B7Wfc1/+EhuP+eMf60f8JDcf88Y/1rIoo9lDsHtZ9zX/AOEhuP8AnjH+tH/CQ3H/ADxj/Wsiij2UOwe1n3Nf/hIbj/njH+tTWuq3t5Jtgt4zjqTnArDALMAOpOBXXWNotnarGAN2MsfU1lUUILY2pOc3q9CSHz9v+kGPd6ID/WpKKK5DsCiiigAooooAKKKKACiiigAooooAKKKKAM3w5/yK+mf9esf/AKCK0qzfDn/Ir6Z/16x/+gitKgAqhP8A8jBZf9e8/wDOOr9ULj/kYLL/AK95/wCcdJmdT4fmvzRfooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo63/AMi/qP8A16y/+gGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVS1j/AJBM34f+hCrtUtYONJmz7fzFXD4kRP4GcrRRRXpHlhRRRQAUVyekeJru5Q3eo3EcVtGJXkRdJuFCqm7pOWKHpnpz0pbjxdcxTTf6BLBtRGjgljUyOTHK+CRJtA/djnnuCPTP2keXmNPZyvY6uisSXWrn+w9MmihRL3UjHGiyjKRsyFiSAeQACcZ54GR1qE+IbjTb86dqsH2u6ZoxA1lGEEodXI+V3+UgxMPvEdDxzi72diVFtXXqdDRXMXHjOCC5SWSCSLTzDNJ58oUecUdEXYd2ACXI+bHY8DmrGn+MLHU5reKxhmmlnd1KxvE4j2bNxZlcqRh1Pyk9+/FJST2HyStc36KKKogKUdaSlHWgDtIv9Sn+6KdTYv8AUp/uinV5R6y2CiimtLGhw7qp9CcUDHUVH9oh/wCe0f8A30KPtEP/AD2j/wC+hQBJRUf2iH/ntH/30KPtEP8Az2j/AO+hQBJRUf2iH/ntH/30K56Xx9oNr4mn0O/uhaXEW3bJLxHJuUHhugPOOcUALr0u/UNnaNQPz5rMqzqEgl1CZ1IZS3BByCKw9d1j+xbOKbZETLKIg9xN5MUeQTl3wdo4x0OSQO9ejG0YI8yd5TdjTorKOupaaalzq9vJbE7iRbo9yu0fxho1PykYOWA+gqZ9c06OPzHuQqeY0ZYq2AyqWOeOBgZz06eoq7oizL9FUYdYtbmxuLmDzituCZI3gdJBgZxsZQ3I6cc9qpWPivT7qysZZma3kvIY5fLMbsIt/Chm24XJyAWxu7UXQWZt0VRv9ZsNMkRL2YozDd8sbOEXONzFQdi/7TYHvUDeJdLUuBNK5WUw4S2kbLgkMBhecbTkjOByeCKXMg5WatFY2n+KLG9t7JpC0Ut3HG+0I7pGXGVVpAu0E9gSCcjjkUv/AAlekFEcTzFJMkOLWUqAMZYnbgKCQNx4zkZ4NO6HytGxRWVqGv2+m6vBZXEcm2W3knaZY3YIEKjsp/vHnPGB6ikbxDbHWrbToUllM7SKZVjfYpQAkbtuD1xnOARjrxS5kHKzpNHh83Uo8jhPmP4V1Nc/4eH+lSn0TH610Fcdd3mduHVoBRRRWB0BRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGb4c/5FfTP+vWP/0EVpVm+HP+RX0z/r1j/wDQRWlQAVQuP+Rgsf8ArhN/OOr9ULj/AJGCx/64TfzjpMzqfD81+aL9FFFM0CiiigAqjrf/ACL+o/8AXrL/AOgGr1Udb/5F/Uf+vWX/ANANAFq2/wCPWL/cH8qkqO2/49Yv9wfyqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooqG8vILG1e4un2Rr7ZJPYAdyegFAEOrTJBpU7vdvZ/IQs0ahmVjwNqkEE56DBzXlGgeHfEv/CXXd74rmu5xZW4kt5J3JUmQ7RgZIB278gdDXp1nZz3t0mo6omxl5trUnIgH95vVz+nQdybGsf8AIJm/D/0IVcPiRE/gZytFFFekeWFFFFAGJb+GVgtWs21S+msWDq1pIsOxlbORkRhu5/izTR4TtmffdXt7dSfLl5WTJAR0A+VQOkh/IH1zu0VHJG1iuaRnto0D6Pb2DzTH7ME8mfIEiMowrAgYz+GD0IIOKrx+G7cXMV1cXNzc3Uc6zGeUpucqjKqkKoAUB24AHJz3Odiiqsr3Em0rIwH8H2TnDXV4Y0R1gjDqFt9zq+UwuchkUjJOMY6cVcttDSC+gvJr27uriESASTup3b9ueAoAxsGAoA6nknNadFJRSHzMKKKKokKUdaSlHWgDtIv9Sn+6KdTYv9Sn+6KdXlHrLYKxprG0vPFUn2y1hn22SbfNjDY+d/Wtms1P+Rqm/wCvKP8A9DegZJ/Yek/9Auz/APAdP8KP7D0n/oF2f/gOn+FXqKAKP9h6T/0C7P8A8B0/wo/sPSf+gXZ/+A6f4VeooAo/2HpP/QLs/wDwHT/Cubn+GGgXviefWL+HzlfZ5VmqhIo8KByB97JGew56GuyooA4+8t4rS8kgt4khiRsJGihVUegA6VnX8F5NEn9n3i2sqNkmSHzUcYxhlyD78MOR6cVua7F5eob+0ig/0rNr0YawR5k9Js5W58FC405LNbm3jiIl82EWn7jc+PmSPfhWXHBO7BJPerdz4Uiu2uRNct5dxa+SUCDCuQoaTnIOQiDB9D61v0U+SIueRkaNoS6VZ3EWLNWuD832KzW3QDGPugkk9Tkk+2Ko2/hOe3s4LUakpgEEENwPs/MgiPylTu+QkcHO72xXS0U+VC5mY+saFLqUsrQXgt1ubc2tyGh3loyT905G1uW5O4c9Kgj8NS2oieyvVjniuZ5leSDeuyViWQruHI4wc9unat+ilyr+vvDmdrf12OTsvAsNndW0vm2k3lLD5kk1gjzFo1Cgo5J2AhVyMEjnBBORNceEGm0ywtEvURrRSouPIIlTJzujdXDRt26sD3BrpqKfKh88r3uZN7pN3c6lBeQX0cEkMUsP+oLfK5U5HzfeG3ryDnpUNn4cNhc2BtroC2sTIsMLRliI3UfLu3dQRkHHTjHetyijlQuZ2sbHh5v9JmX1TP61v1y+jS+VqSAnAcFa6iuOuvfO7Du8AooorA3CiiigAooooAKKKKACiiigAooooAKKKKAM3w5/yK+mf9esf/oIrSrN8Of8ivpn/XrH/wCgitKgAqhcf8h+x/64Tfzjq/VC4/5D9j/1xm/mlJmdT4fmvzRfooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo63/AMi/qP8A16y/+gGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRVHUL57QpsQHdnOTWdSpGnFzlsVGLk7IvUVif21N/zzX86P7am/55r+dcn9oYfv8AgbfVqvY26KxP7am/55r+dH9tTf8APNfzo/tDD9/wD6tV7G3RWJ/bU3/PNfzo/tqb/nmv50f2hh+/4B9Wq9jborE/tqb/AJ5r+dH9tTf881/Oj+0MP3/APq1XsbdFYn9tTf8APNfzo/tqb/nmv50f2hh+/wCAfVqvY26KxP7am/55r+dH9tTf881/Oj+0MP3/AAD6tV7G3RWJ/bU3/PNfzqSDV5JJlV0UAnHWrhjaE5KMXq/IUqFSKu0a9FAORUN5eQWFq9xdPsjX2ySewA7k9AK7DALy8gsbV7i6fZGvtkk9gB3J6AVQs7Oe9uk1HVE2MvNtak5EA/vN6uf06DuSWdnPe3SajqibGXm2tSciAf3m9XP6dB3J1aACqWsf8gmb8P8A0IVdqC9tzd2bwqwUtjk/XNVF2kmTJXi0jj6K2f8AhHZP+fhf++aP+Edk/wCfhf8Avmu72sO55/sZ9jGorZ/4R2T/AJ+F/wC+aP8AhHZP+fhf++aPaw7h7GfYxqK2f+Edk/5+F/75o/4R2T/n4X/vmj2sO4exn2MaitDUdJksNOnuvNWTykLbcYz+NWf+Edk/5+F/75o9rDuHsanYxqK2f+Edk/5+F/75o/4R2T/n4X/vmj2sO4exn2Maitn/AIR2T/n4X/vmj/hHZP8An4X/AL5o9rDuHsZ9jGpR1rY/4R2T/n4X/vmj/hHZP+fhf++aPaw7h7GfY3Iv9Sn+6KdSINsag9QMUteeekgrNT/kapv+vKP/ANDetKs1P+Rqm/68o/8A0N6ANKiiigAooooAKKKKAM7WbQ3NnvQZeLke471zNdvWReaEs0hktnEZPVSOP/rV00aqirM5a1JyfNE5+itP+wbz/pn/AN9Uf2Deesf/AH1XR7SHc5vZz7GZRWn/AGDeesf/AH1R/YN56x/99Ue0h3D2c+xmUVp/2Deesf8A31R/YN56x/8AfVHtIdw9nPsZlFaf9g3nrH/31R/YN56x/wDfVHtIdw9nPsZlFaf9g3nrH/31R/YN56x/99Ue0h3D2c+xnIxRwynBU5FddZ3K3dqsq9/vD0NYX9g3nrH/AN9VZs9P1GykLRGIqeqljg1lV5JrRm1Lng9VobdFRwtKy/v0VG/2WzmpK4ztCiiigAooooAKKKKACiiigAooooAKKKKAM3w5/wAivpn/AF6x/wDoIrSrN8Of8ivpn/XrH/6CK0qACqFx/wAh6x/64zfzSr9ULn/kPWP/AFxm/wDZKTM6nw/Nfmi/RRRTNAooooAKo63/AMi/qP8A16y/+gGr1Udb/wCRf1H/AK9Zf/QDQBatv+PWL/cH8qkqO2/49Yv9wfyqSgArP1eLfabh1U5rQqO4QSQOp7is6sPaU3HuVCXLJM5WilYFWIPUHFJXx+x7YUUUUhhRRRQAUUUUAFFFFABRRRQAVb02LzLxfReTVStGxngsbWW8un2RrxnHJPoB3JPAHevSy6nzVubscuKlaFu5rXd3BYWrXF0+yNPbJJ7ADuT0AqjZ2c99dJqOqJsZeba1JyIB/eb1c/p0Hcks7Oe+uk1HVE2MvNtak5EA/vN6uf06DuTq19GeWFFFFABRRRQAUUUUAFFFFABRRRQBm+Iv+Rcvv+uRrSrN8Rf8i5ff9cjWlQAUUUUAFFFFABRRRQAUUUUAFZqf8jVN/wBeUf8A6G9aVZqf8jVN/wBeUf8A6G9AGlRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZvhz/kV9M/69Y/8A0EVpVm+HP+RX0z/r1j/9BFaVABVC5/5D1j/1ym/9kq/VC5/5D1h/1ym/9kpMzqfD81+aL9FFFM0CiiigAqjrn/Iv6j/16y/+gGr1Udc58Paj/wBesv8A6AaAH217a/ZYv9Jh+4P+Wg9Kl+2W3/PxD/32KrW+lacbWImwtT8g/wCWK+n0p/8AY+m/9A60/wC/C/4UAT/arc9J4v8AvsVh+IPGuj+Gruxh1WVljvd4WZBvWMrt+9jnnd6dq1P7G0s9dNtP+/C/4VgeIvh1oviS8sZLmP7PDab90Nsix+du24ywHQbf16igBZLi2u2+0WM8dxBL8ySROGU/iKZVmbQ9P0S2jh0mzitYRwVjXGfcnqT7mq1fKYun7OvJHsUZc1NMwPFloslnBcCe8hkF1bw5t7yWEFXmRWBCMAcgkZ61njVUsvEsccM981pDKtiyyRzyRcgkuZWBUtvKJ8zZGD611ksMc6BZo0kUMGAdQQCDkH6ggGm/ZLbyGg+zxeUzFmj2DaSTkkj1J5+tRColHll/W3/B+8uUb6r+v60OI0vUb65sdup3FxAFhuZLJlmP+kOruDvYHOVGMJ0xk84wur4Qs71LS2u72NsTWiHzW1i4uS5IByY5AFU98gnHTpW+2n2T26QPZwNCjb1jMSlVbk5AxgHk8+9QWWg6Pptx5+naVY2k2CvmQWyI2PTIGap1YtNWtf8A4JHI7ry/4H+Rz17I81rr2pT6jc213p0zrbpHOypEFUFAYwdr7ic/MDndgdBWXd3V5Je6vI0erLOt7DFBcx3xS2tWaKI4ePzB8oZiT8hBzXdT6Xp9zexXlzY2011Djyp5IVZ0wcjDEZHPpUhs7Vo50a2hKXJJnUxjEuQF+b14AHPYU41lHp/Wn+QSpuXX+tfyORk3xWWqasL68W8t9ReOBTeSGNvnAWPyy2zBzt6Z5455pPD9vqNy89/MJJI1u7oCdtYuMgLLIoHkY8vAwABnoM9eK6aLQdHgvReQ6VYx3SnInS2QOOMfexnpxTR4d0Rbz7Wuj2Aud5k84Wqb93XduxnPvQ6seXl/r8/67D5HzX/rqcn4SnuZZtBLLqsMk9l51xNfXxliuxsH3FMjYbcQ3RSBn6V3tQCxtBFbxC1h8u2IMCeWMREDA2j+HA44qes61RVJXSHTg4KwyeeO2geaYkIgy2Bk/gO59q0tD06a4EWoaom115t7Y9Ic/wAR9XP6dB3Jz3tFvl+zuXUORhkOCpByCD68Vu6fdywyrp+o4FwFzFKBhbhR3How7r+I4r2ssp2pufc4cXK8kuxpUUUV6pxhRRRQAUUUUAFFFFABRRRQAUUUUAZviL/kXL7/AK5GtKs3xF/yLl9/1yNaVABRRRQAUUUUAFFFFABRRRQAVmp/yNU3/XlH/wChvWlWan/I1Tf9eUf/AKG9AGlRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZvhz/kV9M/69Y//AEEVpVm+HP8AkV9M/wCvWP8A9BFaVABVC5/5Dth/1ym/9kq/VC5/5Dth/wBc5v8A2SkzOp8PzX5ov0UUUzQKKKKACqOt/wDIv6j/ANesv/oBq9VHW/8AkX9R/wCvWX/0A0AWrb/j1i/3B/KpKjtv+PWL/cH8qkoAKKKKAKWqReZZsR1Xmuerq5V3xMp7iuXkTy5GU/wnFeFmlO0oz+R6GElo4jKKKK8c7gooooAKKKKACiiigAooooAvaVFvuwT/AAjNbN5ZxX1uYZwcZDK6nDIw6Mp7EVT0aLbEzkYJNadfXYen7OlGJ4tSXNNsz7O8ljuBYakR9pwTFKBhbhR3How7r+I4rQqC8s4r63MM4OMhldThkYdGU9iKrWd5LHcCw1Ij7TgmKUDC3CjuPRh3X8RxW5maFFFFABRRRQAUUUUAFFFFABRRRQBm+Iv+Rcvv+uRrSrkfiD4qs/Dmj+TfW904vUZI5YUUqGHZssCPyra8O69b+JdGj1OzguIbeRiI/tChWYA4yACeM5H4UAalFFFABRRRQAUUUUAFFFFABWan/I1Tf9eUf/ob1pVmp/yNU3/XlH/6G9AGlRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZvhz/AJFfTP8Ar1j/APQRWlWb4c/5FfTP+vWP/wBBFaVABVC5/wCQ7Yf9c5v/AGWr9ULr/kOWH/XOb/2WkzOp8PzX5ov0UUUzQKKKKACqOt/8i/qP/XrL/wCgGr1Udb/5F/Uf+vWX/wBANAFq2/49Yv8AcH8qkqO2/wCPWL/cH8qkoAKKKKACsO/sZPtLOgyG9q3KMZrGtRhWjyzLhUlTd4nM/Y5f7p/I0fY5f7p/I10uB6UYHpXJ/Z1DzNvrVQ5r7HL/AHT+Ro+xy/3T+RrpcD0owPSj+zqHmH1qoc19jl/un8jR9jl/un8jXS4HpRgelH9nUPMPrVQ5r7HL/dP5Gj7HL/dP5GulwPSjA9KP7OoeYfWqhzX2OX+6fyNAs5ifun8jXS4HpRgelNZfQTvqH1moQ2cPkWqJ6Cp6KK9A5gqC8s4r63MM4OMhldThkYdGU9iKnooAz7O8ljuBYakR9pwTFKBhbhR3How7r+I4rQqC8s4r63MM4PUMrqcMjDoynsRVazvJY7gWGpEfacExSgYW4Udx6MO6/iOKANCoby4+yWrzbd+3HGcZ5xU1UtY/5BM34f8AoQqoq8kmTJ2i2il/wkQ/59j/AN9//Wo/4SIf8+x/77/+tWFRXb7GHY4Pb1O5u/8ACRD/AJ9j/wB9/wD1qP8AhIh/z7H/AL7/APrVhUUexh2D29Tubv8AwkQ/59j/AN9//Wo/4SIf8+x/77/+tWFRR7GHYPb1O4/xYLbxX4dm0y5gMZYh45d2TG46Hp9R9Ca07HVrfTrCCztLPy4LeMRxqH6ADA7Vk0Uexh2D29Tubv8AwkQ/59j/AN9//Wo/4SIf8+x/77/+tWFRR7GHYPb1O5u/8JEP+fY/99//AFqP+EiH/Psf++//AK1YVFHsYdg9vU7m7/wkQ/59j/33/wDWo/4SIf8APsf++/8A61YVKOtHsYdg9tU7naq25A3TIzS02L/Up/uinVwHohWan/I1Tf8AXlH/AOhvWlWan/I1Tf8AXlH/AOhvQBpUUUUAFFFFABRRRQAVBeXItLV5T1A+Uepqeud1y8865ECH5I+vua0px5pWM6k+SNw/4SC5/wCecX5H/Gj/AISC5/55xfkf8ayqK7fZQ7HD7Wfc1f8AhILn/nnF+R/xo/4SC5/55xfkf8ayqKPZQ7B7Wfc1f+Eguf8AnnF+R/xo/wCEguf+ecX5H/Gsqij2UOwe1n3NX/hILn/nnF+R/wAaP+Eguf8AnnF+R/xrKoo9lDsHtZ9zV/4SC5/55xfkf8aP+Eguf+ecX5H/ABrKoo9lDsHtZ9zYTxDMG/eQoR/s5FbNrdRXcIkhOfUdxXHVd0u7NreLk/u3O1h/Ws6lFWvE0p1pJ2kdVRRRXEdwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBm+HP+RX0z/r1j/8AQRWlWb4c/wCRX0z/AK9Y/wD0EVpUAFULr/kOWH+5L/Jav1Quv+Q5Yf7kv8lpMzqfD81+aL9FFFM0CiiigAqjrf8AyL+o/wDXrL/6AavVR1v/AJF/Uf8Ar1l/9ANAFq2/49Yv9wfyqSo7b/j1i/3B/KpKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACoLyzivrcwzg9QyupwyMOjKexFT0UAZ9neSx3AsNSI+04JilAwtwo7j0Yd1/EcU/WP+QTN+H/oQpdWihl02U3EU0gjHmIIFJlDDoUx/F6f4V5boHjvWNU8V32l6/HJaia2C29vLF5bKyNnJB7kbifoOwq4fEiJ/Azp6KKK9I8srSanYRX6WMt7bpdyDKW7SqJGHqFzk9D+VWa5iF/sa3thc6XPeXU968yZt28qYFwyOZQCq7QAOTkbOAeM5l1/bPk3n2U6sL3y7j7Qfn8sHePK8rPy52/3O2d3NZ82hpyandUVxGoWOqW7XjWU2rN5VyywDz5XBja2yepO4eYeCc4IwMdKkNjqUMyyRz6qxR7KQBppGBZnxNkdCNvVei9QBRzt9O34g4banXQXdvdDNtPFMNobMbhuD0PHY9qiTVLCTUGsI762a8QZa3WZTIo68rnPcVzvg21uobq6m1BLxbie0tSz3BkIZghDfe43A9R1/Oobe3m/sjTtG+w3A1K2u45ZZzAyxqVk3PKJcbTuXdwCSd+COuHzO/wDXcOVXZ2EkiRRtJK6oiAszMcBQOpJpysGUMpBBGQR3rgLi11H+xIlvRrEpnsJjtjaZ2NzwFDgdBt6A/L1zyamij1sa05ae+jK5CQpbSmMxeVx8/mCMc+i+ZuHpUe08v6/r8NR+z8/6/r8TuaKx9Otr9PCtskNw0d88CPJJeh5iHIBbILKRznjIA9O1WNDh1GDR4I9YmjmuQihiiFSPlHDEs245zlsjPpW3Voz6XNClHWkpR1oEdpF/qU/3RTqbF/qU/wB0U6vKPWWwVmp/yNU3/XlH/wChvWlWan/I1Tf9eUf/AKG9AzSooooAKKKKACijNc3ZePdBu9butIkuxaX1tO0JjuPlEhBIyrdDn06+1AG3f3QtLN5P4uij3rkiSzEk5J5JrR1q7+0Xflofki4+p71m13UYcsbnn1p80rdgoorn4PEd46tczadCmnrdtamVLotICJDGGKFAMbsdGOAe9bXV7GKTaudBRWTB4k0+SWKF5SJZG2jy4pGjUlmVQX2gKSVPBx7Z4zEPFmnTpCbB2uDLJEF3RvGGV3C7lLLhgCecdDwcUk09g5WbdFZo8Qad9pa3aSaOQMFHmW0iBssEBUlQGG5gMjI5HrSahrcdrod1qNoouhbsU2ligZg+0jOD0OecHpRzK1w5WadFYsevS213cW+t20Nq8USSqbaZrjeGYqBjYrbsjgAHOeKlbxLpSRxO1w4Eu7A8iTKBThi425jAJwS2AKOZMfK0atFZU/iOwjW5EcuXt0d8yRyLG+3htr7SGweDt3Y9KiufFemQxXRjkklkt0lJRYZAGaPO5A23BYY6DnHOMc0OUUr3BRb6G1RUNlci9sILlVZBNGrhWBBGRnGCAamqtiTsLGXz7GGQ9SvNT1n6I27TF9mIrQrzJq0mj1YO8UwoooqSgooooAKKKKACiiigAooooAKKKKAM3w5/yK+mf9esf/oIrSrN8Of8ivpn/XrH/wCgitKgAqhdf8hzT/8Acl/ktX6oXX/Ib0//AHZf5LSZnU+H5r80X6KKKZoFFFFABVHW/wDkX9R/69Zf/QDV6qOt/wDIv6j/ANesv/oBoAtW3/HrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFZ+twxSac8kkaM8eCjMoJUkgHB7cVoVS1j/kEzfh/wChCrh8SIn8DOVooor0jyzjbvVdSbWtUitL3UDNb3Mcdtaw2IeBwUQ4eTyztyWOSXGBzWjD4kvLtUFppsLSzSzJEsl0VBWJirsx2HHIGAAc55xituCzgtpriWFNr3LiSU5J3MFC59uFHSqcnh/TZLRLcwuiJI8qmKd0dWckth1IYAljkZxWdpW0ZpzRb2M638Wtd2s2oQaex023iDzSmYCVT5Ql4jxggBgM7s5PTHNQr4vvDpstw+iTI0bJkus6xBGBJYsYQ3G3nCEDI5xkjWPhzSftCS/Y1GxQoiDMIjhdoJjztJC8ZIzjimR+GNKitzFHDMMlSJPtUvmrtBChZN25QASMAgYY+pptS1C8OxSi8VtJqNhA1nCkN4iFZzdfKzN1EbBdkmD23hj1CmqOva1Paa7qEMmq6hZQW9nFJEtpZLMpdjIDvJibH3VAyV71vp4f02OeGVIHUwBdiCZ9hK9GZM7WYf3mBPTnirJ0+1a5uJ2hDSXMSxTFiSHRd2AR0/ib86TUnGwKUU9jCXxTcx6lFYzaZNI8YiS7kijlby5GUMcbUKbRuGSXB68HjI/iu6i0qC+k0n5b0p9iCStJ5gYFvnCRllIVc4VXHPXrjSTw3paXEUwgkLxBAA08jK2z7hZS2HI7FgSMD0FMHhbSBG6LbOAxUqRPIDFtJIEZ3ZjAJPCYFN8wJw7Gcniq4eZSNOnWWRIFFtM3lBWeWRM/MgbHy5yeox8oOcy6Xrmoah4nNpNbxQQxW0nmxrLvxKsm3IO0ZGOnTryKnu/Cenz28cVurW+14izK7lmVHL43bshiWY787snOTV2z0PT7CeOa1hZJI0ZA5ldiwZtzbsn5iTzk5NCUua721/IJONtPIv0daKUdaszOhj8PaWYkzag/KOrt/jT/APhHdK/58kP1J/xrQi/1Kf7op1eUestjN/4R3SP+fCI/UUn/AAjejbt39m2+7GM7BnFadFAzN/4RzRv+gZan6xCj/hHNF/6BVmfrAv8AhWlRQBm/8I3on/QIsT9bZP8ACgeHNEHTRtP/APAVP8K0qKAM7/hHtFHTSLD/AMBk/wAK5yy+FugQ67darfQi9lmnaSOB1AhhBOQAg4OBxzx7Cu0ooA5fVbFbK6HlIFhcZQKMBfaqFdZqVp9rs2QD51+ZPrXKEYODXfRnzRPOrQ5Zeolc/B4cvEVrWbUYX09rtroxJalZCTIZApcuRjdjooyB2roKK1sr3Mk2lY5oeEmS8tJ0vUU277jIkBSZhvL7BIrD5DnBVgwPbHatpHha/aw0ttWukSaxiiSOFIfuBXR2DMGIYnYoBGMehrrqKSik7orndrHJt4Vmsvtl4GGoXEsEkSrHEI5ZCzqyM8rOclcdeMADAGMHVn0IXPhX+x5Jhl4lWSUpu3NkFiRxnJz+da9FJQilYTk27nNXfgqydbiLT1tbO2uDHI9t9kV4zKhG1tuQNpHBXv1BByS+38MXFgqNpl9bWUxVkmMFgiptLZ+RAcKR2Lb/AHBroqKfKk7hzNnKr4HhQ3gjltEWdZQkq2CCcGQ5O+TOWAyQMbeOpNJbeGr66juI9RuUhtxc3ckESxAuPNLqGL7sEbXJ24ByeT2rq6KXJG1vkPnkQ2UD21hBBK6yPFGqM6rtDEDGcZOPzNTUUVe5B02hrjTFz3YmtGq9hF5Onwoeu3J/HmrFeZN3k2epBWikFFFFSWFFFFABRRRQAUUUUAFFFFABRRRQBm+HP+RX0z/r1j/9BFaVZvhz/kV9M/69Y/8A0EVpUAFULr/kN6f/ALsv8hV+qF3/AMhrT/8Adl/kKTM6nw/Nfmi/RRRTNAooooAKo63/AMi/qP8A16y/+gGr1Udb/wCRf1H/AK9Zf/QDQBatv+PWL/cH8qkqO2/49Yv9wfyqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKpax/wAgmb8P/QhV2qupQvcafLFENztjAzjuKqGkkTPWLOSoq/8A2Lff88R/32P8aP7Fvv8AniP++x/jXoc8e55vs59ihRV/+xb7/niP++x/jR/Yt9/zxH/fY/xo549w9nPsUKKv/wBi33/PEf8AfY/xo/sW+/54j/vsf40c8e4ezn2KFFWbzT7qys5bmaE+XEu5trAnH51N/Yt9/wA8R/32P8aOeHcPZz7FCir/APYt9/zxH/fY/wAaP7Fvv+eI/wC+x/jRzx7h7OfYoUVf/sW+/wCeI/77H+NH9i33/PEf99j/ABo549w9nPsUKUdavf2Lff8APEf99j/Gj+xb7/niP++x/jRzx7h7OfY6aL/Up/uinU2MFY1B6gAGnV5p6i2CiiigAooooAKKKKACiiigArm9atPIu/NQfJLz9D3rpKrX9qLyzeL+Lqp9DWlOfLK5lVhzxscjRWj/AGHe/wB1f++hR/Yd7/dX/voV2+0h3OH2c+xnUVo/2He/3V/76FH9h3v91f8AvoUe0h3D2c+xnUVo/wBh3v8AdX/voUf2He/3V/76FHtIdw9nPsZ1FaP9h3v91f8AvoUf2He/3V/76FHtIdw9nPsZ1FaP9h3v91f++hR/Yd7/AHV/76FHtIdw9nPsZ1W9NtTd3irj5F+Zj7VZj0G6ZvnKIPXOa27KyjsodkfJP3mPU1nUqxS0NKdGTd5bFiiiiuI7wooooAKKKKACiiigAooooAKKKKACiiigDN8Of8ivpn/XrH/6CK0qzfDn/Ir6Z/16x/8AoIrSoAKoXf8AyGtP+kv8hV+qF3/yGtP+kv8A6CKTM6nw/Nfmi/RRRTNAooooAKo63/yL+o/9esv/AKAavVR1v/kX9R/69Zf/AEA0AWrb/j1i/wBwfyqSo7b/AI9Yv9wfyqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAM3xF/yLl9/wBcjWlWb4i/5Fy+/wCuRrSoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAM3w5/yK+mf9esf/oIrSrN8Of8AIr6Z/wBesf8A6CK0qACqF3/yGtP+kv8A6CKv1Qu/+Qzp3/bX/wBBFJmdT4fmvzRfooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo63/AMi/qP8A16y/+gGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZviL/kXL7/rka0qgvrNL+wmtJWdEmQqWQjcPpnNU/wCybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqP7Juv+g7qP8A3zB/8aoA06KzP7Juv+g7qP8A3zB/8ao/sm6/6Duo/wDfMH/xqgDTorM/sm6/6Duo/wDfMH/xqj+ybr/oO6j/AN8wf/GqANOisz+ybr/oO6j/AN8wf/GqzfEUsnh7w/d6pc69qBW3jJVdsHzt0Vf9V3OBQBp+HP8AkV9M/wCvWP8A9BFaVc54B1iDWvBGnXFueY4hBKpOSroMEf1+hFdHQAVQu/8AkM6d/wBtf/Qav1QvP+Qzp3/bX/0GkzOp8PzX5ov0UUUzQKKKKACqOt/8i/qP/XrL/wCgGr1Udb/5F/Uf+vWX/wBANAFq2/49Yv8AcH8qkqO2/wCPWL/cH8qkoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKyvEOgad4i0w22rwNPBGfMVBK6DcAcE7SM9e9atMn/AOPeT/dP8qAMHwX4f0zQvD8DaXbmA3cMUsw812DPtHOGJx17YroaoaF/yLum/wDXpF/6AKv0AFULz/kMad9ZP/Qav1QvP+Qxp31k/wDQaTM6nw/Nfmi/RRRTNAooooAKo63/AMi/qP8A16y/+gGr1Udb/wCRf1H/AK9Zf/QDQBatv+PWL/cH8qkqO2/49Yv9wfyqSgAoooJwM0AFFZp1qEHGx/yFJ/bcP9x/yFcv1uh/MjX2NTsadFZn9tw/3H/IUf23D/cf8hR9bofzIfsanY06KzP7bh/uP+Qo/tuH+4/5Cj63Q/mQexqdjTorM/tuH+4/5Cj+24f7j/kKPrdD+ZB7Gp2NOisz+24f7j/kKP7bh/uP+Qo+t0P5kHsanY06KzP7bh/uP+Qo/tuH+4/5Cj63Q/mQexqdjTorM/tuH+4/5Cj+24f7j/kKPrdD+ZB7Gp2NOiq9rdpdLuQEexqxXSmpK6MmmnZhRRRTEFFFRXNwtrbtNICVXGQOvXFC10E3ZXJaKyv+Egtf7kv5D/Gj/hILX+5L+Q/xrT2c+xHtYdzVorK/4SC1/uS/kP8AGj/hILX+5L+Q/wAaPZz7B7WHc1aKyv8AhILX+5L+Q/xo/wCEgtf7kv5D/Gj2c+we1h3NWisr/hILX+5L+Q/xo/4SC1/uS/kP8aPZz7B7WHc1aKyv+Egtf7kv5D/Gj/hILX+5L+Q/xo9nPsHtYdzVorK/4SC1/uS/kP8AGj/hILX+5L+Q/wAaPZz7B7WHc1aKyv8AhILX+5L+Q/xpf+Egtv7kv5D/ABo9nPsHtYdzUopFO5Q3qM0tZmgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTJ/wDj3k/3T/Kn0yf/AI95P90/yoAp6F/yLum/9ekX/oAq/VDQv+Rd03/r0i/9AFX6ACqF5/yGNO+sn/oNX6oXn/IX07/ek/8AQKTM6nw/Nfmi/RRRTNAooooAKo63/wAi/qP/AF6y/wDoBq9VHW/+Rf1H/r1l/wDQDQBatv8Aj1i/3B/KpKjtv+PWL/cH8qkoAKCMgiiigDmb2PyryQe+ar1p6zFtmSQdxisyvk8VT9nWlE9mjLmpphRRRXMahRRRQAUUUUAFFFFABRRRQAUtJSqMsAM5PHFXTi5zUV1JlLli2bukR7bXef4jmtCoraPy7dFHYVLX2KSSsjw3qFFFFMAqlrH/ACCZvw/9CFXapax/yCZvw/8AQhVw+JET+BnK0UUE4GTwK9E8sKKrWWpWOpI76de292qHazQSq4U+hweKs0AFFFNWWN5HRJFZ4yA6hslSRkZ9OKAHUUwTRtM0KyKZUUMyBhuUHOCR6HB/I08nAyeBQAUVVstTsNSV2069trtYzhzBKr7T6HB4qw8scRQSSKhkbagZsbj1wPU8GgB1FFNeRI9vmOqbmCruOMk9vrTAdSjrSUo60AdpF/qU/wB0U6mxf6lP90U6vKPWWwUUUUDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigApk//HvJ/un+VPpk/wDx7yf7p/lQBT0L/kXdN/69Iv8A0AVfqhoX/Iu6b/16Rf8AoAq/QAVQvP8AkL6b/vSf+gVfqhe/8hfTf96T/wBANJmdT4fmvzRfooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo65/yL2o9v8ARZf/AEA0AWrb/j1i/wBwfyqSsmDTLw20X/E8vlGwcCOD0946k/sq5/i1vUD+EI/lHQBpUVm/2RJ/Fq2oH/gaD+S0f2Nn72pagf8Atvj+QoAryajYa5pZudLuorpEPJjblT6EdQfY1l1zvhX4UTabqEmq6rqM0czMzLbWkhTgno7jr7gce5rpHUo5U9QcV4OaU7TjPuejhJaOI2iiivIO05bV9evbLxNNYRErA9tARO0YMdqXkdS7dznCgDpnk4GTUF9daoum+IWi1i6jk0nmF1ihJkxbo/z5jI5Yk8Y6+mK6aTTLOa4nmmgWR7iEQS78kPGM4Ujp/E351DFoOnQ2N1ZpAxhu12Th5XYuNgTlic/dAGc10xqU10/rQiz5r9DOVNRl1QaUus3kaxWy3D3XlQGWQsxAX/V7Ao2nPy55HPrXTXdRh1ya0WGTV0jso2ZrHyVVZN8isTvcf3QMAnBB6Vt32j2WovG9wkqyRAqksE8kLhT1XcjA4OBxnHA9KfbaVY2cpktbdYm8lYPlJA2KSQMdP4j+dTzx5dvwEov+vkcbpura7qskeJ9WcfYbWVvsCWYAd0JYt5ozyR/DxVyTX9St11kTTjat2YrOTYo2bdmYzxzkEsD7N6CtkeFNKVlMK3dvtiSEfZ7+eLKoMKDtcZwO55qxLoOmzWc1rNbb4ZphcSKzsS0gIIbOc5+UVpKrScttPTzRHJO2/wDWpoUUUVyG4VZsI/Nu0HYHNVqs6do8d1cyTyTXaHgYiuXQdPQHivRy+nzV79jmxMrU7dzpRwKKzf7Dg/5+9Q/8Dpf/AIqj+w4P+fvUP/A6X/4qvpDyjSorN/sOD/n71D/wOl/+Ko/sOD/n71D/AMDpf/iqALl3dQ2NnNd3biOCCNpJHIJ2qBknj2qjd31rqPh+S5sLiK5gcLtkicMp5HcVS1zwz9u8P6haWlzeNPPbSRxiW9kKbmUgbsnpk81zPh34bJ4R0ye6n1O4uLpwu+OJykPUdV/i+p/Krh8SIn8DNCs3xDay3mgXMEEZlZgpMQx+9UMCyc8fMARz61pUV6LV1Y8xOzuc1qF7Jf2k8ukafdxlYVSW4a3eCbZvUsiKyhidm85HQ4xknjIa31S5muVt5NYSwW1unsyzzJIWHlbNxb5yd3mYDckdiK7yiocblqduhw1ybiy1WCC+uNYNnPMGYRPMzuTASwG35sbhnCdD6DNVhBrgsd05vYBLNEbmUQSySFRAACREyufmAB2nr14zXfvDFJJHJJGjPGSUZlBKEjBwe3HFY2oT61FrVtbWl5YJBc7yolsndk2gHkiUA5z6CplFXuxxl0OYez1iGa5mVtQ86e0sxLcGCUkoHcP8iPkMBsJVW3csRyTW4tlqF34EntWee6nfdsE8ZheWPfnYQ7swyuQN5zgjdjmrNr4mW4hj8q1ubhjEryTxwgRxFgdoZd5bnHbcBnkiqlr44tP+EbtNTvUP75I13oyJG8zLuZFZ3AG3ByWIHbJPFC5Wnr/X9fn5jfNdaf1/X9aFbWHuNTmW60S1v7YwW4ikl+zyQOd0seEVSATgBiSOBnryaZNa3q62kYTUpJ4L7MJdpZIFgEJCks2VJ3Hkk7s5zxWvoniZdc1aWK0h/wBDFpHMk2VOWLOrKSGPQrjjuDzjGd6nyX1v/V/6/pk8zj7v9bHAWEGtyW6pNdamxlmtxcr5E8JQl/3mHaRjjGcmMBMYIxXXXELy6vYRBGNvbq8zOcn58BFGT14Zj68CtGirUbKxLld3ClHWkpR1qiTtIv8AUp/uinVlR32p+Un/ABKD90f8vKU/7dqn/QHP/gSleUestjSorN+3ap/0Bz/4EpR9u1T/AKA5/wDAlKBmlRWb9u1T/oDn/wACUo+3ap/0Bz/4EpQBpUVm/btU/wCgOf8AwJSj7dqn/QHP/gSlAGlRWb9u1T/oDn/wJSj7dqn/AEBz/wCBKUAaVFZv27VP+gOf/AlKPt2qf9Ac/wDgSlAGlRWb9u1T/oDn/wACUo+3ap/0Bz/4EpQBpUVm/btU/wCgOf8AwJSj7dqn/QHP/gSlAGlRWb9u1T/oDn/wJSj7dqn/AEBz/wCBKUAaVFZv27VP+gOf/AlKPt2qf9Ac/wDgSlAGlRWb9u1T/oDn/wACUo+3ap/0Bz/4EpQBpUVm/btU/wCgOf8AwJSj7dqn/QHP/gSlAGlRWb9u1T/oDn/wJSj7dqn/AEBz/wCBKUAaVFZv27VP+gOf/AlKPt2qf9Ac/wDgSlAGlRWb9u1T/oDn/wACUo+3ap/0Bz/4EpQBpUVm/btU/wCgOf8AwJSj7dqn/QHP/gSlAGlRWb9u1T/oDn/wJSj7dqn/AEBz/wCBKUAaVFZv27VP+gOf/AlKPt2qf9Ac/wDgSlAGlTJ/+PeT/dP8qofbtU/6A5/8CUqOa+1PyJP+JQfun/l5T0oAn0L/AJF3Tf8Ar0i/9AFX6oaDk+G9NyMH7JFx/wAAFX6ACqF7/wAhbTf9+T/0A1fqhe/8hbTf9+T/ANANJmdT4fmvzRfooopmgUUUUAFUdb/5F/Uf+vWX/wBANXqo63/yL+o/9esv/oBoAtW3/HrF/uD+VSVHbf8AHrF/uD+VSUAFFFFAB1rm9Ri8q9f0bmukrH1qL7kg+hrz8wp89BvtqdOGlap6mTRRRXzR6oUUUUAFFFFABRRRQAUUUUAKOa6DSovLs1OMFuTWDGu+QKP4jiuoiXZEqjsK9/LKdoOfc87FyvJRH0UUV6xxBRRRQAVS1j/kEzfh/wChCrtUtY/5BM34f+hCrh8SIn8DOVooor0jywooooAKrzWUc17b3Tlg9uHCAHg7hg5/KrFFJq4GJF4Xt4PLW3vb2GNURZI45FAm2/dLHbnv2IB6EEU2LwnaW9rDDaXV3bm3EZikjZNyOiFN/KkEsh2nIIPoDzW7RSUUtiuZmdZaNHY3xu1urmaVrdYJDM4bftZmDHjOcsemBjoBgVo0UUxN3CiiimIKUdaSlHWgDtIv9Sn+6KdTYv8AUp/uinV5R6y2CiiigYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFMn/495P90/yp9Mn/AOPeT/dP8qAKehf8i7pv/XpF/wCgCr9UNC/5F3Tf+vSL/wBAFX6ACqF7/wAhbTf9+T/0A1fqhe/8hbTf9+T/ANANJmdT4fmvzRfooopmgUUUUAFUdb/5F/Uf+vWX/wBANXqo63/yL+o/9esv/oBoAtW3/HrF/uD+VSVHbf8AHrF/uD+VSUAFFFFABVTUofOs2A6jkVboIz1qZxU4uL6ji3F3RyZUg4I5pMGup8iM/wAA/Kj7PF/cH5V5n9l0/wCZnX9bn2OWwaMGup+zxf3B+VH2eL+4Pyo/sun/ADMPrc+xy2DTIZY7iESwOskbZwyng84rY8R216+g3MWiQK95KvloSwUJnq2T6D9cVz3w30vVdN0uW11WBDaljJbyK4bBzhlx168/nUf2bT5rXZyTzOpHERo8mjT11t/X/ALmDRg11P2eL+4Pyo+zxf3B+VX/AGXT/mZ1/W59jlsGjBrqfs8X9wflR9ni/uD8qP7Lp/zMPrc+xh6bAXulYj5V5roKasap91QKdXoUaSpQUI9DmnNzlzMKKKK1ICiiigAqpqkbzabKkSlmOMAd+RVuimnZ3E1dWOS/sy8/593/ACo/sy8/593/ACrraK3+sS7HP9Xj3OS/sy8/593/ACo/sy8/593/ACrraKPrEuwfV49zkv7MvP8An3f8qP7MvP8An3f8q62ij6xLsH1ePc5L+zLz/n3f8qP7MvP+fd/yrraKPrEuwfV49zkv7MvP+fd/yo/sy8/593/Kutoo+sS7B9Xj3OS/sy8/593/ACo/sy8/593/ACrraKPrEuwfV49zkv7MvP8An3f8qX+zLz/n3f8AKusoo+sS7B9Xj3GxjESg9dop1FFc50hRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFMn/wCPeT/dP8qfTJ/+PeT/AHT/ACoAp6F/yLum/wDXpF/6AKv1Q0L/AJF3Tf8Ar0i/9AFX6ACqF7/yFdN/66Sf+izV+qF9/wAhXTP+ukn/AKLakzOp8PzX5ov0UUUzQKKKKACqOt/8i/qP/XrL/wCgGr1Udb/5F/Uf+vWX/wBANAFq2/49Yv8AcH8qkqO2/wCPWL/cH8qkoAKKKKACiiigAooooAKKKKAGTR+dC0Zd03DG5GwR9DVCLQ4IIxHFc3iIOgFy3H61pUUrIiVOMndoof2Sn/P3e/8AgS1H9kp/z93v/gS1X6KLIXsodih/ZKf8/d7/AOBLUf2Sn/P3e/8AgS1X6KLIPZQ7FD+yU/5+73/wJaj+yU/5+73/AMCWq/RRZB7KHYof2Sn/AD93v/gS1H9kp/z93v8A4EtV+iiyD2UOxQ/slP8An7vf/AlqP7JT/n7vf/Alqv0UWQeyh2KH9kp/z93v/gS1H9kp/wA/d7/4EtV+iiyD2UOxQ/slP+fu9/8AAlqP7JT/AJ+73/wJar9FFkHsodih/ZKf8/d7/wCBLUf2Sn/P3e/+BLVfoosg9lDsUP7JT/n7vf8AwJaj+yU/5+73/wACWq/RRZB7KHYof2Sn/P3e/wDgS1U9KsPtOlwTTXt6zsuWP2lvWtuqGh/8gaD/AIEP/HjSsrmbpw50rdH+gf2Sn/P3e/8AgS1H9kp/z93v/gS1X6Kdkaeyh2KH9kp/z93v/gS1H9kp/wA/d7/4EtV+iiyD2UOxQ/slP+fu9/8AAlqP7JT/AJ+73/wJar9FFkHsodih/ZKf8/d7/wCBLUf2Sn/P3e/+BLVfoosg9lDsUP7JT/n7vf8AwJaj+yU/5+73/wACWq/RRZB7KHYof2Sn/P3e/wDgS1H9kp/z93v/AIEtV+iiyD2UOxQ/slP+fu9/8CWo/slP+fu9/wDAlqv0UWQeyh2KH9kp/wA/d7/4EtR/ZKf8/d7/AOBLVfoosg9lDsUP7JT/AJ+73/wJaj+yU/5+73/wJarryJEuZXVBnGWOKj+123/PxF/32KfL5C9nT7Fb+yU/5+73/wACWo/slP8An7vf/Alqs/a7b/n4i/77FH2u2/5+Iv8AvsUcvkL2dPsVv7JT/n7vf/AlqP7JT/n7vf8AwJarP2u2/wCfiL/vsUfa7b/n4i/77FHL5B7On2K39kp/z93v/gS1H9kp/wA/d7/4EtVn7Xbf8/EX/fYo+123/PxF/wB9ijl8g9nT7Fb+yU/5+73/AMCWo/slP+fu9/8AAlqs/a7b/n4i/wC+xR9rtv8An4i/77FHL5B7On2K39kp/wA/d7/4EtR/ZKf8/d7/AOBLVZ+123/PxF/32KPtdt/z8Rf99ijl8g9nT7Fb+yU/5+73/wACWo/slP8An7vf/Alqs/a7b/n4i/77FH2u2/5+Iv8AvsUcvkHs6fYrf2Sn/P3e/wDgS1H9kp/z93v/AIEtVoXVuSAJ4iT0AcVLRy2H7Om+hQ/slP8An7vf/AlqP7JT/n7vf/Alqv0UrIfsodih/ZKf8/d7/wCBLUf2Sn/P3e/+BLVfoosg9lDsUP7JT/n7vf8AwJaj+yU/5+73/wACWq/RRZB7KHYof2Sn/P3e/wDgS1H9kp/z93v/AIEtV+iiyD2UOxQ/slP+fu9/8CWo/slP+fu9/wDAlqv0UWQeyh2KH9kp/wA/d7/4EtTJtJQQSf6Xe/dP/Ly3pWlTJ/8Aj3k/3T/KiyD2UOxi6LpaPoGnsbq8G61jOFuGAHyir39kp/z93v8A4EtRoX/Iu6b/ANekX/oAq/RZB7KHYof2Sn/P3e/+BLVVlsVttY011nuJMySDEsxcf6tuxrZqhff8hTTP+ur/APotqTSM6lOCSaXVfmi/RRRVHQFFFFABVHW/+Rf1H/r1l/8AQDV6qOt/8i/qP/XrL/6AaALVt/x6xf7g/lUlR23/AB6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFUNE/5BEXszj/AMfNX6oaJ/yCk9pJR/5Eal1M3/EXo/0L9FFFM0CiiigAooooAKKKKACiiigAooooAKKKKACiiigDL8Qf8g9P+ug/ka5yuj8Qf8g9P+ug/ka5yu6h8B59f4wooorcwCiiigAooooAKKKKACiiigAooooAmtf+PyH/AHx/OuxrjrX/AI/If98fzrsa48Rujtw2zCiiiuY6gooooAKKKKACiiigAooooAKZP/x7yf7p/lT6ZP8A8e8n+6f5UAU9C/5F3Tf+vSL/ANAFX6oaF/yLum/9ekX/AKAKv0AFUL7/AJCmmf8AXV//AEW1X6oX3/IU0z/rq/8A6LakzOp8PzX5ov0UUUzQKKKKACqOt/8AIv6j/wBesv8A6AavVR1v/kX9R/69Zf8A0A0AWrb/AI9Yv9wfyqSo7b/j1i/3B/KpKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKoaL/yDPpNMP/IrVbuLmC0hMt1NHDGOC8jBQPxNZfh/UbK4tWgt7y3llE07+Wkqs23zWwcA9ORz7il1MZSiqqTfR/obFFFFM2CiiigAooooAKKKKACiiigAooooAKKKKACiisLxh4ik8MeH5NSht4rl42GYXm8ssvcrwckdcemaALPiD/kHp/10H8jXOVR0Txrf+M9NkuH0UWNlFIFE5uN/mPg/KBtHQdTn0q9XdQ+A8+v8YUVQ17/kXdS/69Zf/QDXGafdWWmaM02k6h4f+0tFCsr6baok0EZdRJI/ztkKCScjGRk+lac3vW/rqZqN1c9CorzK7uDLb6h5Wo+ZEXeQ3ipHudRLb/Nu24xg5OMKeuMV1Hi2OxkjsTqF5YxKrM0a6nD5ltKdvRjkKG54JP8AewD2XP7tw5Nbep0tFcIviGTTrJoYXj0+E2CtYweYJQziV1JjZhl127COMBSDgCo5vE18lpcS2+sLPcILoSwCFD9lVJCEcgDd2A5ODnijnX9eQcjvY7+ivPdU1OSeOVor9NTgt3m8i6aONlbNvkrwNjAE+nsckVJP4i1lLrViL22ja3iusWe8NJCEUmOTy/KyM4U5Zyp3cDkCj2iRSpt7HfUVzuqCUabopnneeRr+BmkcKCcknGFAGBnH4d+tYF1dalqGgiO81KaRL6z8118qIBMTIhA+TkEMcg5/AcUOdm1ba/4K5CjfX+t7HoNFcnpmozx69FYR3a/JPLA+mLHGotoEU+XKAAGGcJyTt+fAA4rrKtO6uglFxdmTWv8Ax+Q/74/nXY1xcXmecnk7fM3Dbv6Z966LGuf39PH/AABz/WuTEbo68NszSorN2a5/z208f9sXP/s1Hl65/wA/Wnj/ALdnP/tSuY6jSorN8nXP+f3Tx/25uf8A2rR5Guf9BHTx9LB//j1AGlRWb9m1rvqVl+Fi3/x2j7LrPfVLb8LI/wDxygDSorN+yav31WH8LP8A+yo+xar31ZfwtR/jQBpUVm/YdT76wfwtko+wal31mT8LeP8AwoA0qg+0Q3NnI9vKkqYZdyMCMjqOO9VP7O1DvrVx+EEX/wATXnPh34f+JbTxFqGqSavLpdrJcSv5cZDPcKWJBZfujI9QT7CgD0nQv+Rd03/r0i/9AFX6oaDx4c03v/okXP8AwAVfoAKoX3/IU0z/AK7P/wCi2q/VC/8A+Qnpn/XZ/wD0U9JmdT4fmvzRfooopmgUUUUAFUdb/wCRf1H/AK9Zf/QDV6qOt/8AIv6j/wBesv8A6AaALVt/x6xf7g/lUlR23/HrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBUOq2QODOMj2NJ/a1l/z8L+Rrl5f9c/8AvGmV2ewj3OH6xLsdX/a1l/z8L+Ro/tay/wCfhfyNcpRR9Xj3D6xLsdX/AGtZf8/C/kaP7Wsv+fhfyNcpRR9Xj3D6xLsdFqU2l6pptxY3UytDcRlG4PGe/wBR1rl/h7o9r4as7uW9mT7XNKyA4P8Aq1JAx9eT9MVNRUvCwbuc8+WdaNZr3o3t8zq/7Wsv+fhfyNH9rWX/AD8L+RrlKKr6vHudH1iXY6v+1rL/AJ+F/I0f2tZf8/C/ka5Sij6vHuH1iXY6v+1rL/n4X8jR/a1l/wA/C/ka5Sij6vHuH1iXY7OC4iuY98D71zjOKkrM0H/kHH/rof6Vp1yyXLJo7IS5ophRRRUlBRRVK71a1tJRCWaa5IyLeFd8h/AdB7nAoAu1VvNStbDatxL+8f7kSAs7/RRyaq+Xquof66QabAf4IiHmP1b7q/gD9RVqz021sNxtogHf78rEs7/7zHk/jQBVLatqH+rUaZAf4nAkmI9h91fx3fQVJFoljGkoki+0PMhSWWdi7upGCCT0HsMCtCigDAvdNtdH8O2un6fEIre3YIij0weT6k9Sax66PxB/yD0/66D+RrnK7qHwHn1/jCiiitzAKKKKACqX9kWP9mS6f5H+iys7PHvbksxZuc55JJq7RSGFFFFMQUUUUAFFFFAE1r/x+Q/74/nXY1x1r/x+Q/74/nXY1x4jdHbhtmFFFFcx1BRRRQAUUUUAFFFFABRRRQAUyf8A495P90/yp9Mn/wCPeT/dP8qAKehf8i7pv/XpF/6AKv1Q0L/kXdN/69Iv/QBV+gAqhf8A/IT0z/rs/wD6Ker9UL//AJCWl/8AXd//AEU9JmdT4fmvzRfooopmgUUUUAFUdb/5F/Uf+vWX/wBANXqo63/yL+o/9esv/oBoAtW3/HrF/uD+VSVHbf8AHrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBxcv+uf/eNMp8v+uf8A3jTK9RbHkvcKKKqX2radpmz+0r+1s/Mzs+0TLHux1xk89aL2CzexboqG1vbW9jMllcw3CDGWikDjkAjkexB+hqSSWOIKZZFQMwVdzYyT0A96Yh1FVbzVdP0+SOO/v7a1eX/VrPMqF/oCeeoq1SAKKKie7t47gQPcRLM2MRlwGOc44687W/I+lAEtFFFMAooooA6TQf8AkHH/AK6H+ladZmg/8g4/9dD/AEpZNaieRodNie/mU4YQkbEP+05+UfTk+1edU+Nnp0/gRpVQuNYt4p2t7dXvLpesFuNxX/ePRfxIqL+zr2+51W72xn/l1tCUX6M/3m/DaPar9vbQWkCw2sKQxr0RFAArM0M/7JqV/wA31z9jhP8AywtG+c/70nX/AL5A+pq9aWNtYReXZwJEpOTtHLH1J6k+5qeigAooooAKKKKAMvxB/wAg9P8AroP5GucrtXjSVcSIrjOcMM1H9ktv+feL/vgV0U6qhG1jmqUXOV7nHUV2P2S2/wCfeL/vgUfZLb/n3i/74FX9YXYz+rPucdRXY/ZLb/n3i/74FH2S2/594v8AvgUfWF2D6s+5x1Fdj9ktv+feL/vgUfZLb/n3i/74FH1hdg+rPucdRXY/ZLb/AJ94v++BR9ktv+feL/vgUfWF2D6s+5x1Fdj9ktv+feL/AL4FH2S2/wCfeL/vgUfWF2D6s+5x1Fdj9ktv+feL/vgUfZLb/n3i/wC+BR9YXYPqz7nKWv8Ax+Q/74/nXY1ELW3BBEEQI6EIKlrGpU52b0qbgmFFFFZGwUUUUAFFFFABRRRQAUUUUAFMn/495P8AdP8AKn0yf/j3k/3T/KgCnoX/ACLum/8AXpF/6AKv1Q0L/kXdN/69Iv8A0AVfoAKoX/8AyEtL/wCu7/8Aop6v1Qv/APkJaX/13f8A9FPSZnU+H5r80X6KKKZoFFFFABVHW/8AkX9R/wCvWX/0A1eqjrf/ACL+o/8AXrL/AOgGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV5/4q+KkWgau+kWmj3VzqAYKomxGjE9Cp5LA/h9a7LUtS+x+XBbx/aL2fIhgBxn1Zj2Udz+HUgVmv4P0+9uLO+1cG71G2nFwLnphh0UDsgwML7Z65NAGS27ed+N2ecdM0lPl/wBc/wDvGmV6i2PJe4VgatYXV54q057a4urREtZw1xbojYJaPCkurAZwe2eK36KGrgnY5C5s9SufEE9vLNqX2Pc+HjlkjDYgi2ncmMfPu6YGd3uKpwJqdxeWbapHqT3YuLR0ASTyViCIWLD7gYPvzn5un8Nd3RUqNnf0/ApyureVjmPEEezUpJYFvY7ia2ER/wBAN3bXK5b924UEr9490HzdTggUnuNVgmnt5Ib2JvtYlHlK/kxQ/ZgNof7u0ODwD15xXaUjKHUqwDKRggjgipcG07P+r3Gp9zzuK9nMGnS/aNb+xzi3F0zmbfJIUfcI/wCIgjGfL46FeatWUOsjVIpkjvgGCLG8yvuMQe5KCQnnhTHndzyM81232W38uKPyI9kJBiXYMRkDA2jtgelS0+T87hz6Wt0OAsINbkt1Sa61NjLNbi5XyJ4ShL/vMO0jHGM5MYCYwRiu+VQiBRnAGBkkn8zS0VUVZWIk7sKKKKoRYtbC8lR5pg2oafu/48o38srwMkjpJn0Yge1dFp19Y3Mfk2RWNohhrcp5bx/VDgj+VQ6D/wAg4/8AXQ/0qzeada34X7TFl0+5KpKuh9VYcj8K82p8bPTp/Ai1RWVt1XTv9Wf7Ttx/CxCTqPY8K/47T7mrNnqlrfO0cTlJ0GXglUpIn1U849+hqDQuUUUUAFFFFABRRRQAUU2SWOJd0rqgzjLHFR/bLb/n5h/77FOzFdImoqH7Zbf8/MP/AH2KPtlt/wA/MP8A32KLMOZE1FQ/bLb/AJ+Yf++xR9stv+fmH/vsUWYcyJqKh+2W3/PzD/32KPtlt/z8w/8AfYosw5kTUVD9stv+fmH/AL7FH2y2/wCfmH/vsUWYcyJqKh+2W3/PzD/32KPtlt/z8w/99iizDmRNRUP2y2/5+Yf++xR9stv+fmH/AL7FFmHMiaioRd2xIAuIiT0AcVNRZoLphRRRSGFFFFABRRRQAUUUUAFFFFABTJ/+PeT/AHT/ACp9Mn/495P90/yoAp6F/wAi7pv/AF6Rf+gCr9UNC/5F3Tf+vSL/ANAFX6ACqF//AMhLS/8Ar4f/ANFPV+qGof8AIS0v/r4f/wBFPSZnU+H5r80X6KKKZoFFFFABVHW/+Rf1H/r1l/8AQDV6qOt/8i/qP/XrL/6AaALVt/x6xf7g/lUlR23/AB6xf7g/lUlABRRTXdY13OwUepOKNgHUVB9tt/8Ansn/AH0KPttv/wA9k/76FR7SHcrll2J6Kg+22/8Az2T/AL6FH223/wCeyf8AfQo9pDuHLLsT0VB9tt/+eyf99Cj7bb/89k/76FHtIdw5ZdieioPttv8A89k/76FH223/AOeyf99Cj2kO4csuxPRUH223/wCeyf8AfQo+22//AD2T/voUe0h3Dll2J6Kg+22//PZP++hR9tt/+eyf99Cj2kO4csuxPRUH223/AOeyf99Cj7bb/wDPZP8AvoUe0h3Dll2J6KQMGGRyKWrJCqOpal9j8uC3j+0Xs+RDADjPqzHso7n8OSQKNS1L7H5cFvH9ovZ8iGAHGfVmPZR3P4ckgUabpv2PfPcSfaL2fBmnIxn0VR2Udh/MkmgA03TfsfmT3En2i9nwZpyMZ9FUdlHYfjySTV6iigDi5f8AXP8A7xplPlB85+P4jTcH0r1FseS9xKKXB9KMH0oEJRS4PpRg+lACUUuD6UYPpQAlFLg+lGD6UAJRS4PpRg+lACUUuD6UYPpQB0eg/wDIOP8A10P9K06zNB/5Bx/66H+QrTrz6nxs9On8CCq17p1rqCKt3CHKnKOCVZD6qw5U+4NWaKzNDL8rVNP/ANRJ/aUA/wCWcxCTL9H6N+OPrViz1W1vJDCjNFcKMtbzLskX3weo9xke9XKr3lha38YS7hWQKcqTwyH1BHIPuKALFFZfkanp/wDx6y/2hAP+WNw22Vfo/Q/Rh+NT2mrW13MYMtBcgZNvOuyQe+O49xke9AF2iiigDL8Qf8g9P+ug/ka5yuj8Qf8AIPT/AK6D+RrnK7qHwHn1/jCiiitzAKKKKACiiigAooooAKKKKACiiigCa1/4/If98fzrsa461/4/If8AfH867GuPEbo7cNswooormOoKKKKACiiigAooooAKKKKACmT/APHvJ/un+VPpk/8Ax7yf7p/lQBT0L/kXdN/69Iv/AEAVfqhoX/Iu6b/16Rf+gCr9ABVDUP8AkI6X/wBfDf8AomSr9UNQ/wCQjpf/AF8t/wCiZKTM6nw/Nfmi/RRRTNAooooAKo63/wAi/qP/AF6y/wDoBq9VHW/+Rf1H/r1l/wDQDQBatv8Aj1i/3B/KpKjtv+PWL/cH8qkoAKgvY/NtHXrkVPSMMqRSlFSTTGnZ3OTpKmuo/KupF96hr42UXGTi+h7cXdXCiiipKCiiigAooooAKKKKACiiigApR14Gfakqa1j824ReuTW+Hp+0qxiZ1JcsGzobNClqgPXFQalqX2Py4LeP7Rez5EMAOM+rMeyjufw5JApuoaj9hWO3to/tF7NxDADjPqzHso7n8OSQKdpum/Y/MnuJPtF7PgzTkYz6Ko7KOw/mSTX1x4oabpv2PzJ7iT7Rez4M05GM+iqOyjsP5kk1eoooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqC7sba/i8u8hSVQcjcOVPqD1B9xU9FAGX9m1LT+bKf7dAP8AlhdNiQD/AGZO/wDwIH6inR69ZEOLp2spY0LvDcjYwUDJI7MB6qSK0qxPFvh9/E/h+XS47tbPzWG6UweaQo6gDIwT6+mfWgCG51az1zw3a6jp0olt52DKe44OQfQg8EVk1S0bwRd+C9PkhGtm+s5pA32drbZsfB+YHecccEY549Ku13UPgPPr/GFFFclrv9hf8JhF/wAJJ/Z/k/2efL+3bNu7zP4d3fHpWkpWt/XS5klc62ivP4dY1i2uLK1S6jtowqGCK7kZZLpGkYAbTE7uQgUcMpGcmo/7Z1SzkitrO7t7SPfK8Qnk2/aXNzICgHlOXwAvyoVb5u+Rg51crkZ6JRXmqavqWnQPb2mpwwD7VeOZbqRV3yiY4jwIn3HBDbBtY7uD6dL4qW1ms7A6nd2NrhywTUYDJayNsPyvkqAe4yex4PZKacboHTtK3qdLRXBjxC2maa8cDR6fEbCRrKESiVXkWVhmJmGWXbtIGMBSMACo73xPfwQ3slvrCyXMUt4j2vlRn7PHGX2SEAbuqquScHOMZ5p86/ry0D2bPQKK8+1PU5Z1uGiv01OC3juDFctHGylvszFkIA2sAccEHrg5on8R6zHPqRW9tomt4rjbZ7gZI1SMlJPLEWQDhTuZyp3YAyQKXOkNU3JKx6DRVfTxINPhM87zyMu5pHCgnPOMKAMDOPw79asVo9HYyJrX/j8h/wB8fzrsa4uLzPOTydvmbht39M5710WNc/v6f/3w/wDjXHiN0duG2ZpUVm41z+/p/wD3w/8AjRjXP7+n/wDfD/41zHUaVFZuNc/v6f8A98P/AI0Y1z+/p/8A3w/+NAGlRWbjXP7+n/8AfD/40Y1z+/p//fD/AONAGlRWbjXP7+n/APfD/wCNGNc/v6f/AN8P/jQBpUVm41z+/p//AHw/+NGNc/v6f/3w/wDjQBpUyf8A495P90/yqhjXP7+n/wDfD/41554xi8bt48tm8Plt4sV85oARBt3v9/dxnr7+lAHo2hf8i7pv/XpF/wCgCr9ZPhYXQ8KaZ9vMRm+zR58kELjaMde+K1qACqGof8hDSv8Ar5b/ANEyVfqhqH/IQ0r/AK+W/wDRMlJmdT4fmvzRfooopmgUUUUAFUdb/wCRf1H/AK9Zf/QDV6qOuf8AIvajjn/RZf8A0A0AWrb/AI9Yv9wfyqSsm3v9SNrFt0eT7g5a4jHb2JqT7Xq5+7pUI/3rzH8kNAGlRWb5+tnpp9gv1vnP/tKsDxw+vHwXqJjjt4nCKUa1nkaUMHUjb8o5zigDU1iLZcq4/iFZ1ZHhaLxpJpAk8WmIQqMxCVf9IP8AvY4A+vNa9fM4+nyV2++p6uGlzU/QKwdQ8RTW2u/2ZZ2lvPMI1k8ua8EMswO7IiQqQ5AU5yVHvW9WLrWiXesCa2e9tzp9wgV4J7MSMhGfmjcMNp6EFg2CMj0rkp8vN72xvK9tBza/t0/Urr7N/wAeNyYNvmffxt5zjj73TnpVVvEt4k0sr6dB9givhZGUXZ80sXCBhH5eMZYcbs4z16U+fw5cSSXMMGoJHYXc6zzwtb7pSRtyFk3AAHYOqk8nnpiew8M6dZ39xfSWlrPeS3Lzrctbr5ibv4Q3J49a0Xslq9fv8v8Agke//Xz/AOAZyeLrxLG3vrvS4EtboSiAxXheQsiO+GUxgAERnkE444rVn1ryYdKf7Pn+0ZAmN/8Aq8xs+enP3cdutVLHwfp1hpEttDb2q3ksMkT3yWyrI2/OST1PX17URaDqDjT1v9RtZU0+RXiEFm0ZbEbJgkyN/eB4A6e/Dfsne36+f/AGufqZ1j47N5pc18I9JPlWj3JtYdV8ycbVztZPLG30Jyce9XZfEt5YrMdU02FGWye7jFrdGXcqlQQ2Y12/eHPI6+lNg8N6qnh1tEm1WzezNo1qCtgyyAFSoOfNI469OfapovCsWnTyS+HpINKaaDypvJtEIdl+44HABBJyOcg9iAauXsLu36/1+DJ9+39eX/BLmiapcapA8ssFqIuDFcWd4LiGUcggNtU5BGDxjpgnnGnWVpWjyWV/d313LbSXV0EWQ2tsYEYLnBILMS3zYyT0AGOOdWuepy83u7Gkb21CnxXv2a4WO3i+0XkoIhhBxn1Zj2UcZP8AMkCmVY8JxwpLdtOpXUZG3S7+rR5+Tb/sY/XOea9HLKd6jn2OXFytFLubGm6b9j8ye4k+0Xs+DNORjPoqjso7D+ZJNXqKK+gPNCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAy/EH/IPT/roP5Gucro/EH/IPT/roP5GucruofAefX+MKKKK3MAooooAKKKKACqTaPYtp1zYtBm2umkaZN7fMXJLc5yMknpV2ikO7QAYGBRRRTEFFFFAE1r/x+Q/74/nXY1x1r/x+Q/74/nXY1x4jdHbhtmFFFFcx1BRRRQAUUUUAFFFFABRRRQAUyf8A495P90/yp9Mn/wCPeT/dP8qAKehf8i7pv/XpF/6AKv1Q0L/kXdN/69Iv/QBV+gAqhqH/ACENK/6+m/8ARMlX6oah/wAhDSv+vpv/AETJSZnU+H5r80X6KKKZoFFFFABVHW/+Rf1H/r1l/wDQDV6qOt/8i/qP/XrL/wCgGgC1bf8AHrF/uD+VSVHbf8esX+4P5VJQAUUUUARXMfm27qe4rlyMEg9RXWkZFczfR+VeOPU5FePmlO8Yz+R24SWriV6KKK8I9EKKKKACiiigAooooAKKKKAFAJOB17VuS6YtxaQ+W5guYeYZ1HKH+oPcd/yNZVjH5l2i++TXTAYGK+jy6ny0ebueXipXqW7FKwv2uGe2u0EN7CP3kYOQw7Op7qf06GrtVL+wW9VHRzDcwndDOo5Q/wBQe47/AJGm2F+1wz212ghvYR+8jB4YdnU91P6dDXpHKXaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDL8Qf8g9P+ug/ka5yu1eNJFxIiuOuGGaZ9lt/wDnhH/3wK6KdVQjaxzVKLnK9zjaK7L7Lb/88I/++BR9lt/+eEf/AHwKv6wuxH1Z9zjaK7L7Lb/88I/++BR9lt/+eEf/AHwKPrC7B9Wfc42iuy+y2/8Azwj/AO+BR9lt/wDnhH/3wKPrC7B9Wfc42iuy+y2//PCP/vgUfZbf/nhH/wB8Cj6wuwfVn3ONorsvstv/AM8I/wDvgUfZbf8A54R/98Cj6wuwfVn3ONorsvstv/zwj/74FH2W3/54R/8AfAo+sLsH1Z9zk7X/AI/If98fzrsaiFtADkQxg/7gqWsalTnZtSpuCYUUUVkbBRRRQAUUUUAFFFFABRRRQAUyf/j3k/3T/Kn0yf8A495P90/yoAp6F/yLum/9ekX/AKAKv1Q0L/kXdN/69Iv/AEAVfoAKoaj/AMf+lf8AX03/AKJlq/VDUf8Aj/0r/r7b/wBES0mZ1Ph+a/NF+iiimaBRRRQAVR1v/kX9R/69Zf8A0A1eqjrf/Iv6j/16y/8AoBoAtW3/AB6xf7g/lUlR23/HrF/uD+VSUAFFFFABWLrUWJEkHfg1tVFPbpcJtcVhiKPtqbgaU58kuY5ait7+x4PVvzo/seD1b8zXlf2XL+Y7Pri7GDRW9/Y8Hq35mj+x4PVvzNH9ly/mD64uxg0Vvf2PB6t+Zo/seD1b8zR/Zcv5g+uLsYNFb39jwerfmaP7Hg9W/M0f2XL+YPri7GDRW9/Y8Hq35mj+x4PVvzo/suX8wfXF2KujRZlaQjoMVtVFBbpbptSpa9inBU4KC6HDKXNJsKqX9gt6qOjmG5hO6GdRyh/qD3HerdFaElKwv2uGe2u0EN7CP3kYPDDs6nup/Toau1Uv7Bb1UdHMNzCd0M6jlD/UHuO9NsL9rhntrtBDewj95GDww7Op7qf06GgC7RRRQA3zE/vL+dHmJ/eX8642Unzn/wB403Jrq+r+ZyfWfI7TzE/vL+dHmJ/eX864vJoyaf1fzF9Z8jtPMT+8v50eYn95fzri8mjJo+r+YfWfI7TzE/vL+dHmJ/eX864vJoyaPq/mH1nyO08xP7y/nR5if3l/OuLyaMmj6v5h9Z8jtPMT+8v50eYn95fzri8mjJo+r+YfWfI7TzE/vL+dHmJ/eX864vJoyaPq/mH1nyO1BDdCD9KWszQf+Qcf+uh/kK065pLldjqjLmimFFFFSUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTJ/+PeT/AHT/ACp9Mn/495P90/yoAp6F/wAi7pv/AF6Rf+gCr9UNC/5F3Tf+vSL/ANAFX6ACqGo/8f2lf9fbf+iJav1Q1H/j+0r/AK+2/wDRMtJmdT4fmvzRfooopmgUUUUAFUdb/wCRf1H/AK9Zf/QDV6qOt/8AIv6j/wBesv8A6AaALVt/x6xf7g/lUlR23/HrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFVL+wW9VHRzDcwndDOo5Q/1B7jvVuigClYX7XDPbXaCG9hH7yMHhh2dT3U/p0NXaqX9gt6qOjmG5hO6GdRyh/qD3Hf8jWPc+MrHS7yy03WM2+pXU6wLCoypycCQH+4fzzx2NAGdL/rn/wB40yny/wCuf/eNMr1FseS9wooopiCiiigAooooAKKKKACiiigAooooA6TQf+Qcf+uh/pWnWZoP/IOP/XQ/0rTrzqnxs9On8CCiiiszQKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACmT/8AHvJ/un+VPpk//HvJ/un+VAFPQv8AkXdN/wCvSL/0AVfqhoX/ACLum/8AXpF/6AKv0AFUNR/4/tK/6+z/AOiZav1Q1H/j90v/AK+z/wCiZKTM6nw/Nfmi/RRRTNAooooAKo63/wAi/qP/AF6y/wDoBq9VHW/+Rf1H/r1l/wDQDQBatv8Aj1i/3B/KpKjtv+PWL/cH8qkoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuE8T/AAr0/wAQanJqkGoXdnqDMG8wt5qAjpweRj0BAFd3RQBxTbt534LZ5x602ny/65/940yvUWx5L3MvxIZhoE5gMgwU8zyt2/y948zbt5zt3dOawZ73SrOIDwc8MImnt4pprBUaBd8qrjAyvmYY9s4Az/DXZUUuXW5SlpY4my13VRrl1HdXsTpB56vZhg06ogOyTyxEME4U5Z9p3cAZAGYusXt5e2UjakbpIb1Xtz5kT+ZuglwrMkarnK7cLnBLDcTjHpNFTyu25XOr3scv4P1W/wBSec3t9b3aeUj4ik3tC5zlCREgXt8hyw7nkVjwPa2eoTPpMljf3zC4ImgUxXsDbXP79CT5i5AX5goB2/KcjHoFFDi2txKaTehwsvit7ia2MGtxQWcht0lulEZVN0Uxf5iCAcovXgEdOoqOHxLfu7h9YAljSI2kHlRg32ZZFB6ZO5VU/LjGc9OK7i4s4LqSB7hN7W8vmxHJG1sFc8deGPX1oSzgjvprxExPMiJI+T8yrnaMdONx/OnZt38w5o2tY8//ALVvdPixZyQ26ssayXMhWPyVae45MhR8DKqPmUgbu2c1ueHtS1LUdWjju79ZIo7JJsQKCk5MkqBizRqSCFU5UKCRkcHnqqKIxa6hKaa2CiiirMzY0qXU1tGFnaWksW84aW6aNug7CNv51d8/XP8AoHaf/wCB7/8Axmk0H/kHH/rof6Vp151T42enT+BGb5+uf9A7T/8AwPf/AOM0efrn/QO0/wD8D3/+M1pUVmaGb5+uf9A/T/8AwPf/AOM0efrn/QP0/wD8Dn/+M1pUUAZvn65/0D9P/wDA5/8A4zR52uf8+Gn/APgc/wD8arSooAzfO1z/AJ8NP/8AA1//AI1R52uf8+On/wDga/8A8arSooAzfO1z/nx0/wD8DX/+NUedrn/Pjp//AIGP/wDGq0qKAM3ztc/58dP/APAx/wD41R52uf8APlp//gY//wAarSooAzfN1z/ny0//AMC3/wDjdHm65/z56f8A+Bb/APxutKigDN83XP8Anz0//wAC3/8AjdHm65/z56f/AOBT/wDxutKigDN83XP+fTT/APwKf/43R5uuf8+mn/8AgU//AMbrSooAzfM1z/n10/8A8CX/APjdHma5/wA+un/+BL//AButKigDM8zXP+fXT/8AwJf/AOIpfM1z/n20/wD8CH/+IrSooAzfM1z/AJ9tP/8AAh//AIijzNc/59tP/wDAh/8A4itKigDN8zXP+ffT/wDv+/8A8RRv1z/n30//AL/v/wDEVpUUAZu/XP8An30//v8Av/8AEUb9c/54af8A9/3/APiK0qKAM3frn/PDT/8Av8//AMTRv1z/AJ46f/3+f/4mtKigDN365/zx0/8A7+v/APE0btc/546f/wB/X/8Aia0qKAM3drn/ADx0/wD7+v8A/E0btb/55af/AN/X/wDia0qKAM3drf8Azy0//v6//wATXmvhefx+2vaisCK+ji5mBN8SI1G4/wCqP3j7Y4r1ymT/APHvJ/un+VAFPQv+Rd03/r0i/wDQBV+qGhf8i7pv/XpF/wCgCr9ABVDUf+P3S/8Ar7P/AKJkq/VDUv8Aj80v/r7P/omSkzOp8PzX5ov0UUUzQKKKKACqOt/8i/qP/XrL/wCgGr1Udb/5F/Uf+vWX/wBANAFq2/49Yv8AcH8qkqO2/wCPWL/cH8qkoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDi5f9c/+8aZT5f8AXP8A7xpleotjyXuFFFFMQUUUUAFFFFABRRRQAUUUUAFFFFAHSaD/AMg4/wDXQ/0rTrM0H/kHH/rof6Vp151T42enT+BBRRRWZoFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTJ/+PeT/AHT/ACp9Mn/495P90/yoAp6F/wAi7pv/AF6Rf+gCr9UNC/5F3Tf+vSL/ANAFX6ACqGpf8fml/wDX2f8A0VJV+qGpf8fml/8AX2f/AEVJSZnU+H5r8y/RRRTNAooooAKo63/yL+o/9esv/oBq9VHW/wDkX9R/69Zf/QDQBatv+PWL/cH8qkqO2/49Yv8AcH8qkoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCudPtCcm3j/ACpP7OtP+feP8qs0VXNLuTyx7Fb+zrT/AJ94/wAqP7OtP+feP8qs0Uc0u4csexW/s60/594/yo/s60/594/yqzRRzS7hyx7Fb+zrT/n3j/Kj+zrT/n3j/KrNFHNLuHLHsVv7OtP+feP8qP7OtP8An3j/ACqzRRzS7hyx7Fb+zrT/AJ94/wAqP7OtP+feP8qs0Uc0u4csexW/s60/594/yo/s60/594/yqzRRzS7hyx7DIoY4E2woEXOcCn0UVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTJ/wDj3k/3T/Kn0yf/AI95P90/yoAp6F/yLum/9ekX/oAq/VDQv+Rd03/r0i/9AFX6ACqGpf8AH3pn/X3/AO0pKv1Q1L/j60z/AK+//aUlJmdT4fmvzL9FFFM0CiiigAqjrf8AyL+o/wDXrL/6AavVR1v/AJF/Uf8Ar1l/9ANAFq2/49Yv9wfyqSo7b/j1i/3B/KpKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACmT/wDHvJ/un+VPpk//AB7yf7p/lQBT0L/kXdN/69Iv/QBV+qGhf8i7pv8A16Rf+gCr9ABVDUv+PrTP+vv/ANpSVfqhqX/H1pv/AF9/+03pPYzq/D935l+iiimaBRRRQAVR1v8A5F/Uf+vWX/0A1eqjrf8AyL+o/wDXrL/6AaALVt/x6xf7g/lUlR23/HrF/uD+VSUAFFFFABmjIrE1WSaK6GyR1Vh0DcVS+1T/APPaT/vo15tXMI0puDi9DqhhnOKkmdRkUZFcv9qn/wCe0n/fRo+1T/8APaT/AL6NZf2pD+Vl/VJdzqMijIrl/tU//PaT/vo0fap/+e0n/fRo/tSH8rD6pLudRkUZFcv9qn/57Sf99Gj7VP8A89pP++jR/akP5WH1SXc6jIoyK5f7VP8A89pP++jR9qn/AOe0n/fRo/tSH8rD6pLudRkUZFcv9qn/AOe0n/fRo+1T/wDPaT/vo0f2pD+Vh9Ul3OoyKMiuX+1T/wDPaT/vo0fap/8AntJ/30aP7Uh/Kw+qS7nUZorD0yeZ7razswxk5OcVuV6VGp7WCna1zlnHklyhRRRWpAUUUUAFFci97dCRgLmXqf4zTft11/z8zf8AfZrp+rvucv1hdjsKK4/7ddf8/M3/AH2aPt11/wA/M3/fZo+rvuH1ldjsKK4/7ddf8/M3/fZo+3XX/PzN/wB9mj6u+4fWV2Oworj/ALddf8/M3/fZo+3XX/PzN/32aPq77h9ZXY7CiuP+3XX/AD8zf99mj7ddf8/M3/fZo+rvuH1ldjsKK4/7ddf8/M3/AH2aPt11/wA/M3/fZo+rvuH1ldjsKK4/7ddf8/M3/fZo+3XX/PzN/wB9mj6u+4fWV2OworP0WV5dP3SuztvPLHNaFYSXK7HRF8yuFFFFSUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTJ/+PeT/AHT/ACp9Mn/495P90/yoAp6F/wAi7pv/AF6Rf+gCr9UNC/5F3Tf+vSL/ANAFX6ACqGpf8fOm/wDX3/7Ter9UNT/4+dN/6+//AGm9JmdX4fu/Mv0UUUzQKKKKACqOt/8AIv6j/wBesv8A6AavVR1v/kX9R/69Zf8A0A0AWrb/AI9Yv9wfyqSo7b/j1i/3B/KpKACiiigDM1mLdCsgHKmsSunvYvNtXX24rma+fzOny1VLuelhZXhbsJRVPUNWtNMMS3TSl5ifLjggeZ2x1O1ATgcZOMcj1qlJ4s0mP7OQ91KtyAYXhsZ5VkyC2AVQgnAOR1GDnpXmqEmrpHU5Jbs2aKy5fEelw3X2eWeRWBVXbyJPLiZsbVd9u1GORwxB5HqKfPr+nW16bWWd96sFkZYXaOInoHkA2oeRwxHUeoo5J9g5o9zRoqiNasGWMpOW8ydrdFWNizOrFWGAM4BByegHOcc1OL23N9JZiT9/FGsrptPCsSAc9Oqn8qXK10HdE9FYreLtHUBhPO6eUk7SJZzMiI4yrMwTCjHPJGO9S/8ACTaULx7czTApKIWlNtKIVc4wpl27M8jv3FV7OfZi5o9zVoqnLqtpDdTWzO7Tww+c8ccTu2zOMgKDk+wyenHIqiPF2kGCSYteLHE4jYtp9wvzlgoUZTltxAwOaShJ7IHJLdm1RVax1CHUY2e3S4QKcEXFrJAfwDqCfqKtfXpS5Xfl6jurXNbRYuXkOPQVr1U02Ly7Nc9Tyat19hTioQUV0PElLmbYUUUVZIUUUUAcVJ/rG+pptOk/1jfU02vUPJCiiimIKKKKACiiigAooooAKKKKACiiigDpdB/5Bv8AwM1pVm6D/wAg3/gZrSrzanxs9On8CCiiioNAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKZP/wAe8n+6f5U+mT/8e8n+6f5UAU9C/wCRd03/AK9Iv/QBV+qGhf8AIu6b/wBekX/oAq/QAVQ1P/j403/r7H/ot6v1Q1P/AI+NO/6+x/6Lek9jOr8P3fmX6KKKZoFFFFABVHXDjw9qP/XrL/6AavVR1v8A5F/Uf+vWX/0A0AV7fXrEW0Q/0okIPu2cx7ey1J/b1n2ivz9NOn/+Iq9bf8esX+4P5VJQBm/27b9rXUD/ANuMo/mtH9tp2sdQP/bqw/nWlRQB5l4i+Jt/oPjQacNLku7OaJGWBkKTqxyDt656dCPxFbyTG4QTNbzW3mDd5M6hXTPYgE4NdMNNsxqTagLaL7YyCMzlcvtHYHsOTWXrEW25Vx/EK83MqfNR5ux1YWVp27nJ+IbR7iS3ePTru5ZA4Wewu1hngJx03MqspxyCSMhcqezRZalKmhPeZmmtrlpLhztBCmKRQSBgE/MoO3jOccVu0V4KqNRsely63OUu9O1JNP1bR4bA3CalLKyXhkTy41l67wSGyuTgKCCAvI5xZgi1XSzeWNppxuftE7Sw3jyoYl39fMBYPwc8KDkY5HboqKftXa1v6/pCcbnJ6RomoaRfPqCwtNJcXk6zwNIp2wvKWV48nC4+8VyMgnOSFFXr+O9sdfn1C006bUEubNYNkEkalGRnIzvZeDv6jJ46VvUUnVbd2hqKV7dTjtM8JXqedbXl9c21s1jbWzi1Me2fahVxllLDrjI2nmnNp+pC11jSIdKmWO/uWMd2ZYvKjjZVG7G/fkAHA29cdOtdfRVe3k5cz/r+rEeyVrHMRabrianHqchtTm8ZntlhxKIm/d8y+ZtOFCtgL1XFR6po99N4durdLWWWRtVW5EUMyo7RidXJVtwwdoOOQfpXV0UlWkmn2/QpwTTX9df8zL0NDFbyp9h1KzG7ONQuxcM3HY+bIQOOmRWhJM0UkarbzTlzjESg4+uSKkq9pcfmXgPZRmujCR9riE36/cZVnyUmXI9VdIwo0nUOB/zzT/4qnf2vJ/0CdQ/79p/8VWlRX055Jm/2vJ/0CdQ/79p/8VR/a8n/AECdQ/79p/8AFVpUUAZv9ryf9AnUP+/af/FVwGn/ABQ1RfG19otzo89/Et06RfZo/wB9Eu7oyjggeuR9TXqNV7WwtLJ5ntLaOF55DJKyKAZGJyST3PNAHmfi9rlrG1S1MiSTX0SFBcvAWBJypdMkD6ZrNur290BUjjiVZmjUlZb2e6UFp44/vOQTw2enB9s5664himk/fRpJsk3ruUHaw6Eeh96ims7a4bdcW8UrYAy6BuhDDr7gH6gV6Ki07/1tY8vm6f1ucrqHibVLLT7hVa1kvrV5i4S1YrJHGFJbBlUIBvGcsx9Aamnv7l/CXiO5WaVJI/OMTCQ5j/dKQFPbBPat+fStOuWDXNhazMrmQGSFWwxGC3I64A59ql+xWot5IBbQ+TLkSR+WNr5GDkdDxxQovW76WHzK602dzjotTn0iS8uIrbUYFt7ASPb6le+d5rlgFdCZHAUfNuOR1Ga19L1+6Kyw6zaSJdJJsXyINwI2hhvEbyCLr/E3I5+m1LZ207o81vFI0asqM6AlVYYYD0BHX1otLK10+3EFhbQ20IJIjhjCKCfYcU+V9H/VxcytqjlR4m1XyrTH2FpNQWCWErG2IFkcLhhu+cjOcjbnB4FRXPiPUInTUJmhMELXgFrGGR28lWB3ksQQSueny5HXrXVRaVp0Bcw2FrGZJBK5SFRucHIY8ctnv1py6ZYJePeJZW63Mn35xEoduMctjJ44pcrsPmjfY5m18Ra5Le2ljcwWsE1zOFEjop+QxO/3Emc/wcEsM56cVqXyfb/Eiafc3E8NsLXzo0guGhaV92GJZCGwoxxnHzcjpWhbaTp1kqLZ6fa24RzIgihVdrkbSwwODjjPpTr3TrHUolj1Gzt7uNW3Kk8SuAfUAjrTa/r5C5lc55/EFxCr/YLmG5tbGKN5HuEPmXIaRk+UggDGw4bDBiR9TXfxHrjWxmi/s9d9vdTRhoXO3yHC4PzjO4H2x79K6mTTbGaaCaazt5Jbb/UO0Slov904+XoOlL9gs9oX7JBtCuoHljADnLD6E9fWk4y6MalFPVHNxa/rN3Kn2f7BFHNczW6B4nYoUUsGOHGeFxt49c9qp2Ot6hqeq6TKsq28c12nnRKXYOGshJjlsAAk8AYzg9Qc9illaR7fLtoV2sXXbGBhiME/UgnmmLpdggQJY2yiN1dMQqNrKu0EccEKAAew4qkmvwBSVnp3Oi0rUXt7Qxrp95MA5+eJFKngerCrv9sP/wBArUP+/a//ABVGg/8AIN/4Ga0q8+p8bO+n8CM3+2H/AOgVqH/fpf8A4qj+2G/6Beof9+h/8VWlRUGhm/2yf+gZqH/fkf40f2yf+gZqH/fkf41pUUAZv9s/9Q3UP+/H/wBej+2f+obqH/fj/wCvWlRQBm/20P8AoHah/wCA5/xo/tof9A/UP/AY1pUUAZv9tL/0D9Q/8BjR/bSf8+Gof+ArVpUUAZv9tp/z46h/4CtR/bcf/PjqH/gI/wDhWlRQBm/23H/z5ah/4CP/AIUf25F/z5ah/wCAcn+FaVFAGb/bkP8Az56h/wCAcn+FH9uQ/wDPpqH/AIBSf4VpUUAZv9uQf8+mof8AgDL/APE0f27B/wA+uof+AMv/AMTWlRQBm/27b/8APrqH/gBN/wDE0f27bf8APtqH/gBN/wDE1pUUAZv9u23/AD76h/4L5v8A4ij+3rX/AJ99Q/8ABfP/APEVpUUAZv8Ab1r/AM8NQ/8ABdP/APEUf29af88NQ/8ABdcf/EVpUUAZv9vWf/PHUP8AwW3H/wARR/b9n/zyv/8AwXXH/wARWlRQBm/2/Z/8877/AMF0/wD8RR/b9l/cvv8AwXz/APxFaVFAGb/b9l/cvf8AwAn/APiKP7fsf7t5/wCAM/8A8RWlRQBm/wBv2Ppd/wDgDN/8RR/b9h/09f8AgFN/8RWlRQBm/wBv2P8A09f+Ac3/AMTS/wBvWP8A09f+Ac3/AMTWjRQBnf27Y/8ATz/4CS//ABNcRpnxcsbvV7rSNUtJbeUTSQwzQozrJgkDK43KfwP4V6RWVpnhnSNHlnmsbKNJ7l2eacjMjliSfmPOOeg4oAm0H/kXNN/69Iv/AEAVfpscaQxJHEoREUKqqMAAdAKdQAVQ1P8A4+NO/wCvsf8AoD1fqhqf+v07/r7H/oD0nsZ1fh+78y/RRRTNAooooAKo63/yL+o/9esv/oBq9VHW/wDkX9R/69Zf/QDQBatv+PWL/cH8qkqO2/49Yv8AcH8qkoAKKKKACs/WIt9ruHVTmtCormPzLd1PcVlWh7Sm4dy4S5ZJnLUUpGGIPUUlfIHthRRRSAKKKKACiiigAooooAK2tGixG8hzycc1jD2GfaulsYvKtEXvivayun8U/kcGLltEsUUUV7ZwBRRRQAUUUUAcVJ/rG+pptOk/1jfU02vUPJCiiimIKKKKACiiigAooooAKKKKACiiigDpdB/5Bv8AwM1pVm6D/wAg3/gZrSrzanxs9On8CCiiioNAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqhqf+v07/r7H/oD1fqhqf+u0/wD6+x/6A9J7GdX4fu/Mv0UUUzQKKKKACqOt/wDIv6j/ANesv/oBq9VHW/8AkX9R/wCvWX/0A0AWrb/j1i/3B/KpKjtv+PWL/cH8qkoAKKKKACgjIoooAybjSGeZmjYAE55qL+xpf76/lW3RXO8LRbu4o19rUXUxP7Gl/vr+VH9jS/31/Ktuil9VofyoPbVO5if2NL/fX8q5bTtci1Dxtd6GkibY12wvjh3XO8f5/umvQ2G5SDnBGODiuXsPCGhWviWaS309Y3t44ZYmEj5Vyz5Oc/7IqZYWldWijjxNXF89P2UtL636q239dbF7+xpf76/lR/Y0v99fyrboqvqtD+VHZ7ap3MT+xpf76/lR/Y0v99fyrboo+q0P5UHtqncx4tHYSAyMCAc8CtdRtUAUtFbQpxpq0FYiUnJ3YUUUVZIUUUUAFFFFAGadCtCST5mT/tUf2DZ/9NP++q0qKv2k+5n7OHYzf7Bs/wDpp/31R/YNn/00/wC+q0qKPaT7h7OHYzf7Bs/+mn/fVH9g2f8A00/76rSoo9pPuHs4djN/sGz/AOmn/fVH9g2f/TT/AL6rSoo9pPuHs4djN/sGz/6af99Uf2DZ/wDTT/vqtKij2k+4ezh2M3+wbP8A6af99Uf2DZ/9NP8AvqtKij2k+4ezh2M3+wbP/pp/31R/YNn/ANNP++q0qKPaT7h7OHYhtbWOzh8uLdtznk5qaiipbbd2WkkrIKKKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqhqf+u0//AK+x/wCgNV+qGqf63T/+vtf/AEFqT2M6vw/d+ZfooopmgUUUUAFUdb/5F/Uf+vWX/wBANXqo63/yL+o/9esv/oBoAtW3/HrF/uD+VSVHbf8AHrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABVCD/kYL3/r3g/8AQpKv1nzFrTUZJ47S5uTNGinytm1dpb+8wOfmNJmdTRp9n+jNCiqH9pT/APQJvvzi/wDi6P7Sn/6BN9+cX/xdFw9rHz+5l+iqH9pT/wDQJvvzi/8Ai6P7Sn/6BN9+cX/xdFw9rHz+5l+iqH9pT/8AQJvvzi/+Lo/tKf8A6BN9+cX/AMXRcPax8/uZfoqh/aU//QJvvzi/+Lo/tKf/AKBN9+cX/wAXRcPax8/uZfoqh/aU/wD0Cb784v8A4uj+0p/+gTffnF/8XRcPax8/uZfoqh/aU/8A0Cb784v/AIuj+0p/+gTffnF/8XRcPax8/uZfoqh/aU//AECb784v/i6P7Sn/AOgTffnF/wDF0XD2sfP7mX6Kof2lP/0Cb784v/i6P7Sn/wCgTffnF/8AF0XD2sfP7mX6Kof2lP8A9Am+/OL/AOLo/tKf/oE335xf/F0XD2sfP7mX6Kof2lP/ANAm+/OL/wCLo/tKf/oE335xf/F0XD2sfP7mX6Kof2lP/wBAm+/OL/4uj+0p/wDoE335xf8AxdFw9rHz+5l+iqH9pT/9Am+/OL/4uj+0p/8AoE335xf/ABdFw9rHz+5l+iqH9pT/APQJvvzi/wDi6P7Sn/6BN9+cX/xdFw9rHz+5l+iqH9pT/wDQJvvzi/8Ai6P7Sn/6BN9+cX/xdFw9rHz+5l+iqH9pT/8AQJvvzi/+Lo/tKf8A6BN9+cX/AMXRcPax8/uZfoqh/aU//QJvvzi/+Lo/tKf/AKBN9+cX/wAXRcPax8/uZfoqh/aU/wD0Cb784v8A4uj+0p/+gTffnF/8XRcPax8/uZfoqh/aU/8A0Cb784v/AIuj+0p/+gTffnF/8XRcPax8/uZfoqh/aU//AECb784v/i6P7Sn/AOgTffnF/wDF0XD2sfP7mX6Kof2lP/0Cb784v/i6P7Sn/wCgTffnF/8AF0XD2sfP7mX6Kof2lP8A9Am+/OL/AOLo/tKf/oE335xf/F0XD2sfP7mX6Kof2lP/ANAm+/OL/wCLo/tKf/oE335xf/F0XD2sfP7mX6Kof2lP/wBAm+/OL/4uj+0p/wDoE335xf8AxdFw9rHz+5l+iqH9pT/9Am+/OL/4uj+0p/8AoE335xf/ABdFw9rHz+5l+iqH9pT/APQJvvzi/wDi6P7Sn/6BN9+cX/xdFw9rHz+5l+iqH9pT/wDQJvvzi/8Ai6P7Sn/6BN9+cX/xdFw9rHz+5l+iqH9pT/8AQJvvzi/+Lo/tKf8A6BN9+cX/AMXRcPax8/uZfoqh/aU//QJvvzi/+Lo/tKf/AKBN9+cX/wAXRcPax8/uZfoqh/aU/wD0Cb784v8A4uj+0p/+gTffnF/8XRcPax8/uZfoqh/aU/8A0Cb784v/AIuj+0p/+gTffnF/8XRcPax8/uZfoqh/aU//AECb784v/i6P7Sn/AOgTffnF/wDF0XD2sfP7mX6Kof2lP/0Cb784v/i6P7Sn/wCgTffnF/8AF0XD2sfP7mX6Kof2lP8A9Am+/OL/AOLo/tKf/oE335xf/F0XD2sfP7mX6oap/rdP/wCvtf8A0FqP7Sn/AOgTffnF/wDF1UvLyWe609HsLmAfalO+TZj7rejE0mzOpUi429Oj7m1RRRVHQFFFFABVHW/+Rf1H/r1l/wDQDV6qOt/8i/qP/XrL/wCgGgC1bf8AHrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVDVP9ZYf9fa/+gtV+qGqf6yw/wCvtf8A0FqT2M6vwF+iiimaBRRRQAVR1v8A5F/Uf+vWX/0A1eqjrf8AyL+o/wDXrL/6AaALVt/x6xf7g/lUlR23/HrF/uD+VSUAFFFFABRTJZo4VBlYKD0yah+323/PVfzqHUhF2bKUZPZFmiq32+2/56r+dH2+2/56r+dL2tP+ZfeHJLsWaKrfb7b/AJ6r+dH2+2/56r+dHtaf8y+8OSXYs0VW+323/PVfzo+323/PVfzo9rT/AJl94ckuxZoqt9vtv+eq/nR9vtv+eq/nR7Wn/MvvDkl2LNFVvt9t/wA9V/Oj7fbf89V/Oj2tP+ZfeHJLsWaKrfb7b/nqv50fb7b/AJ6r+dHtaf8AMvvDkl2LNFVxfW7EASrk+9WAcjIqoyjLZiaa3CiiiqEFFFFABRRRQAUVVk1K0ikZJJgGU4IweKb/AGtY/wDPwPyP+FVyy7E88e5coqn/AGtY/wDPwPyP+FH9rWP/AD8D8j/hRyS7Bzx7lyiqf9rWP/PwPyP+FH9rWP8Az8D8j/hRyS7Bzx7lyiqf9rWP/PwPyP8AhR/a1j/z8D8j/hRyS7Bzx7lyiqf9rWP/AD8D8j/hR/a1j/z8D8j/AIUckuwc8e5coqoNVsicC4X8QRVlHSRd0bBl9Qc0nFrcaknsx1FFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFUNU+/Yf9fa/yar9UNU+/Y/9fafyak9jOr8BfooopmgUUUUAFUdb/wCRf1H/AK9Zf/QDV6qOt/8AIv6j/wBesv8A6AaALVt/x6xf7g/lUlR23/HrF/uD+VSUAFFFFAFHVo99mSP4Tmufrqp08yFlPcVy7KUcqeoOK8HNKdpxn3PRwkvdcRtFFFeQdoUVS1HWLPS3hS7Mxecny0gtpJmbHJ4RSe9QN4k0tbaCZZpZRcbvLjhtpJJDtOGzGqlhtPByBg8HmrUJNXSJ5kupqUVky+J9Ki8nbNNcedD56fZbWWf5M43HYpxz61Yh1rT7iCeeC5WSK3iWWR1BICMu4Ecc5XnjNDhJK7QcybsXqKr2moW188y2su8wMqyDaRtJUMOo9GB/GoJNb0+KC/mkuMR6fk3J2MfLwMntzx6Z5BHUGlyyvaw7ov0VQsNbstRuGgtzcJMqeYY7i1lgJXOMgSKMjPpV+hxcXZgmnsFFFFSMkhBaZAOpOBXURjbGB7VgaXH5l4PRRk10NfTZfT5KCffU8nEy5qnoFFFFd5zhRRRQAVXvbkWlo8p6gYUepqxXOa3eefdeSh+SLg+5rSnHmlYzqz5I3MxmLsWY5JOSaSiivQPMCiiimAUUUUAFFFFABRRRQAVc06/eynBzmNjhl/rVOipaTVmNNxd0dsrBlDKcgjIPrS1naJP51gEJy0Zx+HatGvOkuV2PUjLmimFFFFSUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFUNU+9Y/9fafyNX6oar96x/6+0/kaT2M6vwF+iiimaBRRRQAVR1v/kX9R/69Zf8A0A1eqjrn/Ivajj/n1l/9ANAFq2/49Yv9wfyqSsi3g1z7LFjUNPxsGP8AQX9P+u1S+Rrn/QQ0/wD8AH/+PUAaVFZvka5/0ENP/wDAB/8A49R5Guf9BDT/APwAf/49QBpH+dc3qMXlXrjGAeRWB8SdL8UX2h2Eemyrc3S36NGLG3eF4z5cnzFzIQBzjPHUc1b0rT/EtnpcR8VX1vdT4wPLj+df95+A35fia8/MKfPQb7anThpWqepaooor5o9Uxda0u8v9Z0uW0uJrRIBN5k8Hlll3KABh1YHOPSqsekz+H9TW8sLa41OOSAxTASRibzDI0hk+YqpDFmzgjGFwMdOkrmvEl5qUeuadZ6c98EmhmkkSwW38wlSgBzP8uPmPTnmt6cpSagvP9WRJKzZU07w1qian58l5NpwlgkaQ2ZiYK8k7SeX86N0DdQBk1JNpF1pMOoWOk6bLdQXtrHBC6zIBCVjMf7wswbHQ5UMevHrYjOp6hfXNnHqtzZfYYIvmMULSSu6lsyfKVwMAYTHO7npivpmpal4mDNFqB0zyLaJ8W8aOJJHUksd6n5OOAMHrz0xq3J3btbrv6ozVr3V7k2nJqWj6jqSDSLm6S4kjaKeGSEIdsKJyGkDDlT2rMvfDuvjRruP/AEW6lvtPmgljt4fJKyNucFmaQhvnZxwB9/PTpZ02/wBU8Q3zmPVJdPVdOt7hY7eOJozI5kDE70LFfkGMEcVG+s6vNa22q3NxdWFjJawzCW1to5oFYjc/mqcy7e2VwAOSRyav31LpfTv20/pC0t1sa2n6XfWXiHz7y5udSie28uKebylNtyCylUVc7sLzg424reoHIorjlJy3NoxUVoFFFI4cxsIiqvg7Sy7gD9MjP5ilGLk0kNuyubWix/I8h9cCtWsSxtNbitUC39gOO9i5P/o2rPka5/0ENP8A/AB//j1fYwioRUV0PDk7u5pUVm+Rrn/QQ0//AMAH/wDj1Hka5/0ENP8A/AB//j1UI0qTcNxXIyBkis7yNc/6CGn/APgA/wD8ergfHGkeMr3xVpTaDcZuY4W3XFshgjjUt0fc7A9Dx39KAPRtQuxZ2bSfxHhfrXJkkkk8k9at3EGsQ6baDXbqC6uApDyW8RRc/Tv9ePpVOu6jFKNzgrybnbsFYvipnGkwJHLLD5t7bRM0MrRsVaVQw3KQRkEjg1tVDd2dtf27W99bxXMLYLRzIHU45HB4rZ6mCdmcvNdyaRrb2NrPJPbI8Lf6TdO3lOyTEqXLZxhFOGJAznHSm2niPWZ7i3sZvscF7NNsbzrZ18tTG7AgCQq/KcFXwRn7pFdNb6ZYWcKQ2ljbQRRv5iJFCqqrYxuAA4PvWbf+HtBitEluLK2t7Ozdrl4kgQRudhUl12/NwfrkCs7NX1LumZVr4gvZo7mdJFFzKII4YxE0yM/7zOxdy43bc5LAAck8VY0zxLf3iWUcsMCzXjJ5ZCHaVDP5vRjyAmRyQN68nvZS40q6U2uqaH9hgeISKL6GHy3RMY6M23bkcNgjPHQ4lhl0NNYtnh1O1ErRMtrapNGFwzZZkUckkryeeh6c01e+4aW2Ir61XUPFj29zc3cUEdgsgWC8khAYuwLHYwzwO9Za+KNQis9OMMkdzuMK3DPb5JWSXYkm/eoG4c4VW98A1pateeFbtRc30Wm6m0bxRFtsUzRCRgFJJ+6vOfpzVqefw1NLaXlzLpUkk48u1nkaMmQA/dRj1wT0Hekl59f+CP1RzsXiPVbDw1ayieO7uVSSacNbscx+aVzvMgC+n8R54XiuqvJnfVLC1iZl3F55NpxlEGMH/gTrx7GnPoekybfM0uzfYXK7rdDt3/exx37+tXPLTzPM2Lv27d2OcemfSqimlqS2ug6iiirINnw8586ZOxUH/P51vVz3h7/j8k/65/1FdDXBW+M9Ch8AUUUVibhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVDVetj/19p/I1fqhqvWy/6+0/rSexnV+Bl+iiimaBRRRQAVR1v/kX9R/69Zf/AEA1eqjrf/Iv6j/16y/+gGgC1bf8esX+4P5VJUdt/wAesX+4P5VJQAUUUUAFU9Ti8yzf1HNXKbIu6Mj1FROKnFxfUqL5WmcnRT5U8uVl9DimV8c007M9tO6uFQvZwSX0V48eZ4UaON8nhWwWGOn8I/KpqKLtbDM/UNDsNTmEt1HJ5mzy2aGd4i6/3W2MNy9eDkcn1NNvPD+m3pTzYGj2R+V/o8zw7o+yNsI3KOynIGT6mtKiqU5LZi5UzKufDOl3dwZpIZo2MKwMsF1LCjRrnapVGCkDceCO9LL4c0ua484wPGSqq0cM8kUcgUYAeNWCuMcYYHjjpxWpRT9pPuxcsewUUUVmUFS28ZkuEXHVhUVaGkxb7rcR90da7sDT5668tTnxEuWmzdQbUA9qWiivpzyQooooAKKKKAIL22F3aPEepGVPoa5F1KOVYYZTgiu1rn9ds/LmFyg+WThvY100J2fKzlxELrmRkUUUV2HEFVdTshqWmXFmzmPzkKhwM7T2OO/NWqKW4HJ6roOt6zHciZ7e1aS1khxHfTPHMzLtBMZUKgHXgMffvVjVNE1O91iKVJYzZwywyohndNpRhvGxVw+QMgsTjGAB1rpKKlwTK5mcHpnhrU7/AEnR2ure2s1s4oQIiW3OBNHI+9So2n93wOcljkir9/4Xv5Uf7G1vHLJPO3m+ay7EkfdyhVklHqjAc9GHWutoo5UU6jbAcCiiirMwooooA2vDsfzzSdgAtbtUNHtzBp6lhhpDuP8ASr9edVd5s9KlG0EgooorM1CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACqGq/8uX/X2n9av1Q1X/ly/wCvtP60nsZ1fgZfooopmgUUUUAFUdb/AORf1H/r1l/9ANXqo63/AMi/qP8A16y/+gGgC1bf8esX+4P5VJUdt/x6xf7g/lUlABRRRQAUUUUAc9qkXl3hPZhmqddPPbRz/fFQf2XB6H868irlvtKjmpWv5HbDFcsUmjn6K6D+y4PQ/nR/ZcHofzrP+y3/AD/gV9cXY5+iug/suD0P51xPjfxJB4b1bTbS3G87xNdKD/yy6bfqeT+AqZZZyq7n+BhiM0p4an7SotDWordhsbS4gjmhbfHIodGB4YEZBp/9lweh/Oq/st/z/gbrGJ62OforoP7Lg9D+dH9lweh/Oj+y3/P+AfXF2Ofrb0eLbAznqxqYaZAD0P51ajjWJAqDAFdmFwf1dtt3uYVq/tElYdRRRXec4UUUUAFFFFABUV1brdWzxN/EOD6GpaKadtRNXVji5I2ilaNxhlOCKZXZtbwuxZ4Y2J6kqDmk+y2//PCL/vgV1fWF2OT6s+5xtFdl9lt/+eEX/fAo+y2//PCL/vgUfWF2D6s+5xtFdl9lt/8AnhF/3wKPstv/AM8Iv++BR9YXYPqz7nG0V2X2W3/54Rf98Cj7Lb/88Iv++BR9YXYPqz7nG0V2X2W3/wCeEX/fAo+y2/8Azwi/74FH1hdg+rPucbWjpmmvdSiSQFYVOST/ABe1dCLaAHIhjB9QgqWplXurJFRw6Tu2AGBgcCiiiuY6gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKoar/y5f8AX3H/AFq/VDVeln/19x/1pPYzq/Ay/RRRTNAooooAKo63/wAi/qP/AF6y/wDoBq9VHW/+Rf1H/r1l/wDQDQBatv8Aj1i/3B/KpKjtv+PWL/cH8qkoAKKKKACiiigAooooAKKKKACsXWNL0+e9spZ7G2kklugsjvCpLjy24JI56D8hW1VDU/8Aj407/r7H/oD0nsY1oqULNdvzLkMMdvCsUEaRRqMKiKAAPYCn0UUzbYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqhqvSz/AOvuP+dX6oat92z/AOvuP+dJ7GdX4GX6KKKZoFFFFABVHW/+Rf1H/r1l/wDQDV6qOt/8i/qP/XrL/wCgGgC1bf8AHrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAFUdTguJRavaIkjwziQq77QRtYdcH1q9RQTKPMrFD7Rqv8A0D7f/wACz/8AEUfaNV/6B9v/AOBZ/wDiKv0UrE8kv5n+H+RQ+0ar/wBA+3/8Cz/8RR9o1X/oH2//AIFn/wCIq/RRYOSX8z/D/IofaNV/6B9v/wCBZ/8AiKPtGq/9A+3/APAs/wDxFX6KLByS/mf4f5FD7Rqv/QPt/wDwLP8A8RR9o1X/AKB9v/4Fn/4ir9FFg5JfzP8AD/IofaNV/wCgfb/+BZ/+Io+0ar/0D7f/AMCz/wDEVfoosHJL+Z/h/kUPtGq/9A+3/wDAs/8AxFH2jVf+gfb/APgWf/iKv0UWDkl/M/w/yKH2jVf+gfb/APgWf/iKPtGq/wDQPt//AALP/wARV+iiwckv5n+H+RQ+0ar/ANA+3/8AAs//ABFH2jVf+gfb/wDgWf8A4ir9FFg5JfzP8P8AIofaNV/6B9v/AOBZ/wDiKPtGq/8AQPt//As//EVfoosHJL+Z/h/kUPtGq/8AQPt//As//EUfaNV/6B9v/wCBZ/8AiKv0UWDkl/M/w/yKH2jVf+gfb/8AgWf/AIij7Rqv/QPt/wDwLP8A8RV+iiwckv5n+H+RQ+0ar/0D7f8A8Cz/APEUfaNV/wCgfb/+BZ/+Iq/RRYOSX8z/AA/yKH2jVf8AoH2//gWf/iKPtGq/9A+3/wDAs/8AxFX6KLByS/mf4f5FD7Rqv/QPt/8AwLP/AMRR9o1X/oH2/wD4Fn/4ir9FFg5JfzP8P8ih9o1X/oH2/wD4Fn/4ij7Rqv8A0D7f/wACz/8AEVfoosHJL+Z/h/kUPtGq/wDQPt//AALP/wARR9o1X/oH2/8A4Fn/AOIq/RRYOSX8z/D/ACKH2jVf+gfb/wDgWf8A4ij7Rqv/AED7f/wLP/xFX6KLByS/mf4f5FD7Rqv/AED7f/wLP/xFH2jVf+gfb/8AgWf/AIir9FFg5JfzP8P8ih9o1X/oH2//AIFn/wCIo+0ar/0D7f8A8Cz/APEVfoosHJL+Z/h/kUPtGq/9A+3/APAs/wDxFH2jVf8AoH2//gWf/iKv0UWDkl/M/wAP8ih9o1X/AKB9v/4Fn/4ij7Rqv/QPt/8AwLP/AMRV+iiwckv5n+H+RQ+0ar/0D7f/AMCz/wDEUfaNV/6B9v8A+BZ/+Iq/RRYOSX8z/D/IofaNV/6B9v8A+BZ/+Io+0ar/ANA+3/8AAs//ABFX6KLByS/mf4f5FD7Rqv8A0D7f/wACz/8AEUfaNV/6B9v/AOBZ/wDiKv0UWDkl/M/w/wAih9o1X/oH2/8A4Fn/AOIo+0ar/wBA+3/8Cz/8RV+iiwckv5n+H+RQ+0ar/wBA+3/8Cz/8RR9o1X/oH2//AIFn/wCIq/RRYOSX8z/D/IofaNV/6B9v/wCBZ/8AiKPtGq/9A+3/APAs/wDxFX6KLByS/mf4f5FD7Rqv/QPt/wDwLP8A8RR9o1X/AKB9v/4Fn/4ir9FFg5JfzP8AD/IofaNV/wCgfb/+BZ/+Io+0ar/0D7f/AMCz/wDEVfoosHJL+Z/h/kUPtGq/9A+3/wDAs/8AxFH2jVf+gfb/APgWf/iKv0UWDkl/M/w/yKH2jVf+gfb/APgWf/iKPtGq/wDQPt//AALP/wARV+iiwckv5n+H+RQ+0ar/ANA+3/8AAs//ABFH2jVf+gfb/wDgWf8A4ir9FFg5JfzP8P8AIofaNV/6B9v/AOBZ/wDiKPtGq/8AQPt//As//EVfoosHJL+Z/h/kUPtGq/8AQPt//As//EVBcLqd3JbK9nbxpHOkjMLkscA84GwVrUUWE6bas5P8P8gooopmoUUUUAFUdb/5F/Uf+vWX/wBANXqo63/yL+o/9esv/oBoAtW3/HrF/uD+VSVHbf8AHrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVR1v8A5F/Uf+vWX/0A1eqjrf8AyL+o/wDXrL/6AaALVt/x6xf7g/lUlR23/HrF/uD+VSUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVR1v/AJF/Uf8Ar1l/9ANXqo63/wAi/qP/AF6y/wDoBoAtW3/HrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVHW/+Rf1H/r1l/wDQDV6qOt/8i/qP/XrL/wCgGgC1bf8AHrF/uD+VSVHbf8esX+4P5VJQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVHW/+Rf1H/r1l/8AQDV6qOt/8i/qP/XrL/6AaALVt/x6xf7g/lUlR23/AB6xf7g/lUlABRRRQAUUVx/j/wAXf8I9pwt7Qg3twML/ALC+tXCDnLlQFvxL450zw4DE7faLrHEMZ5H19K851H4pa9dyH7IYrSPsEXJ/M1xs00k8zyzOXkc5ZmPJNMr1KeHhBa6myprqb7eOfErNk6tN+AH+FJ/wm/iT/oLz/p/hWDRWvJHsVyR7G9/wm/iT/oLz/p/hR/wm/iT/AKC8/wCn+FYNFHJHsHJHsb3/AAm/iT/oLz/p/hR/wm/iT/oLz/p/hWDRRyR7ByR7G9/wm/iT/oLz/p/hSr458Sq2Rq034gf4VgUUckewckex2mn/ABR160kH2poruPuHXB/MV6N4Z8daZ4jAiVvs133hkPX6HvXgtPilkglWWF2R0OVZTgg1lPDwmtFYl010Pp2iuN+H/i//AISCwNresPt0A5/219a7KvLnBwlysxCiiioAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACqOt/8AIv6j/wBesv8A6AavVR1w48Paiev+iy/+gGgC1bf8esX+4P5VJWTb6tOLWIf2PqB+Qc4i9P8AfqT+1p/+gNqH5Rf/ABdAGlRWb/a0/wD0BtQ/KL/4uj+1p/8AoDah+UX/AMXQBosQqkngAZNfPXizVn1nxNd3LElA5SMeijivbb7Vp/7PuP8AiU36fu2+ZhHgcdeHr55ZizFm5JOTXfg4rVmlNa3Eooor0DYKKKKACiu78tJPCAm0zTrOC4s7SOS4jvtPHmMMn99HIRhw2cbW7KcVrixs21tbm6sdOFhZabFcvEYIIVlnkUqoZiAMHJOCcZArB1rO1iOf+vu/zPLaK9Nk8PQafHriwf2bEZLqB7K5u7dZY1jkJwoyrADtnGOKim+yxat4ogh0yzjSwtDJCsthFlZBtBYZUnaTkgHjBHFL26d2l/Wn+Y+a+39a2PN6KdJIZZXkYKGdixCqFHPoBwB7Cm10FBRRRQBr+FtVfR/EtndISF8wI49VPBr6IVg6Bl5DDIr5hBwcjg19K6WxfSbRm6mFf5V5+MitGY1Ny1RRRXAZhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBHPH51tJEf40K/mK+bNQtWstSuLZxgxSMuD7GvpevI/ij4Ze21Aazax5hn4mwPut612YSaUnF9S4O0jzuiiivTNwooooAtvquoSWIsnvrlrQYAtzMxjGDkfLnFJLql/PAYJ765kiYKDG8zFSF+7wT27elVaKVkFkWpNTv5rUW0t7cvAFVRE0rFcL90YzjA7elK+q6hK0jS39y5kj8py0zHen9088j2qpRRZAFFFFMAooooAnsrZry/gtkGTLIqYHua+lLaLyLWKL/nmgX8hXkvwu8NPd6l/bF0mILfiLI+83r+Fev15uLmnJRXQwm7yCiiiuIgKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACobu1hvbWS3uo1kikG1lYZzU1FGwHjPiv4b3ulyvc6QjXVoTnYOWj/xFcO6NG5SRSrKcFWGCK+nqzr3w9pGotm9063mPqyCu6ni2laSLU2j5wor3t/h94akbJ01B7KxFN/4V34Z/6Bw/77b/ABrb63Ar2nkeDUV7z/wrvwz/ANA4f99t/jR/wrvwz/0Dh/323+NH1umHtPI8Gor3n/hXfhn/AKBw/wC+2/xo/wCFd+Gf+gcP++2/xo+t0w9p5Hg1Fe8/8K78M/8AQOH/AH23+NOT4feGkbI01T7MxNL63APaeR4MiNI4SNSzMcBVGSa7bwr8N73VZEudWRrWzznaeHf/AAr1ay8PaRpzbrLTreFvVUGa0qyqYttWgiXNshs7OCws47a0jWOKNdqqo6VNRRXDuQFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH/9k=)

## Key Algorithms

### Menu Form – SAVE (New record)

This algorithm will save the record (meal) the user has created into the database and redisplay this meal in the list box full of meals that are currently in the menu. This is done by initially validating all the data. All the input boxes on the form are checked to see if they are empty. If the validation for this has passed, then the algorithm will check to see if there is a digit present in the ‘name’ field. This is done by using a built-in library within python which maps an existing comparison module to the ‘name’ field. This module returns a Boolean, and this is stored in a variable. This process is repeated to check if there is a character in the ‘price’ field as well.

If both checks have passed, then the program will check to see if the ‘meal type’ selected by the user was a beverage. If this is the case then, the record will be saved in the ‘menu’ table using an SQL command. Then the program will execute another SQL command to update a link table within the database with the new meal item’s ID. This is for use later in the program for the waiter to check that there is enough stock to take an order. A message then alerts the user that the record has been saved and the input boxes are cleared to allow for another entry. This section of the check allows the user to leave the ingredients selection input blank as it is not applicable for the meal type.

If the meal type was not a beverage but e.g., a starter then the program will execute another validation check to see if the ingredients selection input has been left blank. If this validation has passed, then the program saves the entry in the database. It will then insert a record into the link table for this new meal item and store its ID alongside the many ingredients ID’s that are linked to that one meal. This is also for later use in the program where the waiter checks to see if there is enough stock to take an order. After the saves are complete, a message alerts the user to confirm the records being saved.

After this, a repeated function within the class is used to refresh the list box on the form to show the newly added record to the database and display it to the user.

This whole process of validation uses nested if statements, list comprehension and for-loop

techniques.

### Menu Form – SAVE (Update existing record)

This algorithm is like the previous one above, however the main difference is that the global variable that was declared in the ‘EDIT’ algorithm is used here to check to see if the record being saved, is to be updated or saved as a new entry in the table. All the validation checks are repeated here again before the save is complete.

Once those checks are complete, the program will then execute the SQL command to update that record in the ‘menu’ table. The program will then update the link table for that meal with its new selection of ingredients. It does this by using an SQL command to query the ‘ingredients’ table for it to return their individual id’s and stores it in an array. Then another SQL query is executed on the link table to return the primary keys where the meal id matches that in the table and store it in an array. At this point a simple for-loop is used to execute an SQL command to update the link table with those new ingredient id’s. The for-loop works by iterating through both arrays at the same time and parsing through their values into the SQL command as parameters.

Once all the relevant tables are updated in the database, the program will then alert the user with a message to say that the meal record was successfully updated.

The pseudo code for both above algorithms is shown below:

**IF NUM\_PRESENT IN NAME:**

MSGBOX(Error)

**ELSE**:

**IF CHAR\_PRESENT IN PRICE:**

MSGBOX(Error)

**ELSE**:

**IF MENU\_TYPE == "BEVERAGE":**

**IF EDITED\_RECORD:**

GLOBAL ID = SQL(SELECT MENUID)

SQL(UPDATE MENU WITH VALUES WHERE MENUID = ID)

EDITED\_RECORD = FALSE

**ELIF NOT EDITED\_RECORD:**

SQL(INSERT INTO MENU VALUES FROM INPUT BOXES)

**ELIF NO NONE SELECTED INGREDIENTS FOR MEAL:**

**IF EDITED\_RECORD:**

SQL(UPDATE MENU WITH VALUES WHERE MENUID = ID)

FOR EACH SELECTED INGREDIENT ITEM:

INGRED\_IDS.APPEND(SQL(SELECT INGREDIENT ID))

P\_KEYS.APPEND(SQL(SELECT P\_KEY FROM LINK TBALE WHERE MENUID = ID))

FOR X,Y IN (INGRED\_IDS, P\_KEYS):

SQL(UPDATE LINK TABLE WITH X,Y)

**ELIF NOT EDITED\_RECORD:**

SQL(INSERT INTO MENU VALUES FROM INPUT BOXES)

FOR EACH SELECTED INGREDIENT ITEM:

ID = SQL(SELECT ID FROM INGREDIENTS WHERE NAME = INGREDIENT ITEM)

SQL(INSERT INTO LINK TABLE (INGREDIENT ID, MEALID))

### Take Order Form – SAVE AN ORDER / CREATE TICKET

This algorithm will save the order the waiter has created for the customer by creating a order ticket which is sent through digitally to the kitchen who have the correct order in an efficient manner. The algorithm also implements the use of a real time stock level system which will alert the waiter if there are enough ingredients to cook the meal. This saves the waiter coming back to the customer, after they confirmed It, and telling the customer to choose another meal as the one they wanted is not available. This stops the customer leaving a negative review.

The algorithm works by looping through each of the meal items in the order and initially checking if the item is a beverage or a food item. If it is a beverage, then the loop skips to the next item in the order. If the item is a food item, then the ingredient id’s that are related to that food item is added to an array. Then by looping through each of the ingredient id’s the current level of stock is compared to the minimum required amount of stock for that ingredient. If it is greater or less than, then a boolean is set to true or false accordingly. Then if there is no stock left, a text file is created if it doesn’t already exist, and the item is added to the text file. If the item already exists in the text file, then the item is not added to the file. This file will then be used in another form, where the manager reviews these items and processes a re-order.

However, if there is stock of that ingredient, then the whole order is saved to a file with the name of the waiter, date, time, and the order Id. This file is used so that a receipt can be displayed to the customer straight after the order is processed. The file is also used later in the program for the manager to view the most popular products sold at the restaurant.

The pseudo code for the above algorithm is shown on the next page.

ORDERS = ""

**FOR EACH ENTRY IN ORDERS:**

FOOD\_ITEMS.APPEND(ENTRY)

ORDERS += ENTRY

ISSTOCK = FALSE

**FOR ITEM IN FOOD\_ITEMS:**

RESULT = SQL(SELECT MENUID,TYPE FROM MENU WHERE NAME = ITEM)

FOOD\_ID, FOOD\_TYPE = RESULT[0]

**IF FOOD\_TYPE == "BEVERAGE":**

CONTINUE

**ELSE:**

INGRED\_IDS.APPENED(SQL(SELECT INGREDID FROM LINK TABLE WHERE MENUID = FOOD\_ID))

**FOR INGRED\_ID IN INGRED\_IDS:**

SQL(SELECT QUANTITY, MINORDERLVL FROM INGREDIENTS WHERE INGRED\_ID = I NGRED\_ID)

ISSTOCK = QUANTITY > MINORDERLVL

NAME = SQL(SELECT NAME FROM INGREDIENTS WHERE INGRED\_ID = INGRED\_ID)

**IF NOT ISSTOCK:**

**IF STOCK\_REPLENISHMENT.TXT EXISTS:**

OPEN FILE

READ FILE

IF NAME EXISTS IN FILE:

PASS

**ELSE:**

WRITE NAME OF INGREDIENT TO BE RE-ORDERED

**ELSE:**

OPEN FILE

WRITE NAME OF INGREDIENT TO BE RE-ORDERED

**IF ISSTOCK:**

**IF ORDERS.TXT FILE EXISTS:**

OPEN FILE

READ LAST LINE

NEW\_ORDER\_ID = LINE.SPLIT(',')[0] + 1

ORDER = NEW\_ORDER\_ID, FULL NAME OF WAITER, DATE AND THE FULL ORDER

WRITE ORDER TO FILE

MSGBOX(ORDER)

**ELSE:**

OPEN FILE

NEW\_ORDER\_ID += 1

ORDER = NEW\_ORDER\_ID, FULL NAME OF WAITER, DATE AND THE FULL ORDER

WRITE ORDER TO FILE

MSGBOX(ORDER)

### Suppliers Form – FIND(Search supplier contacts)

This algorithm will search the ‘suppliers’ table in the database, for the name of the supplier the manager has entered, by using an SQL query. It will then set corresponding variables equal to the result from the query. This is done using list comprehension. It will set the relevant input fields equal to the text in the variables and then it will disable input to those fields. The ‘edit’ and ‘delete’ button will be made visible for the manager to then manipulate the data related to the supplier as needed.

If the record of the supplier doesn’t exist then, the try-except clause will catch the error and display a message box alerting the manager that there isn’t contact information for the supplier.

The pseudo code for the above algorithm is shown below.

supplier\_name = txtFindName.text()

**if supplier\_name != “”:**

result = SQL(SELECT \* FROM SUPPLIERS WHERE Name=supplier\_name)

**try:**

s\_id, s\_name, s\_address, s\_telephone = [result[i] for I in range(0, len(result))]

txtName.setText(s\_name)

txtAddress.setText(s\_address)

txtPhone.setText(s\_telephone)

txtName.enabled = False

txtAddress.enabled = False

txtPhone.enabled = False

txtFindName.clear()

btnEdit.visible = True

btnDelete.visible = True

btnSave.visible = False

**except IndexError:**

MSGBOX(“Record doesn’t exist”)

**Else:**

MSGBOX(“Enter the name of the supplier”)

### Kitchen View Form – REFRESH (Using Bubble Sort)

This ‘refresh’ method in the ‘KitchenViewWin’ class updates the list box with the orders that have been made on that day. In short, this is done by taking the dates at the end of the order file, sorting them, and then comparing those dates to the system’s date and then updating the list box with those dates that are the same as the system’s date.

The bubble sort algorithm is used when the program stores the dates from the file in an array. The algorithm will then compare the current value in the array to the next value and swap them if the next value in the array is larger than the current value. This is done until the length of the array has been reached.

The pseudo code for the algorithm is shown below.

**For x in range(len(DATES)):**

Swapped = True

J = 0

**while J < len(DATES) – 1:**

**if DATES[J + 1] > DATES[J]:**

DATES[J], DATES[J+1] = DATES[J+1], DATES[J]

Swapped = True

J +=1

**If not Swapped:**

break

## Forms / Functions

|  |  |
| --- | --- |
| Form / Function Name | Description |
| Splash Screen | Initial loading screen |
| Login | Main login form to access the system |
| Home | Main menu used to access all areas of the system based on the current user’s role |
| Menu | Form used to add, edit, and delete meals to the menu |
| Ingredients | Form used to add and delete ingredients to the available stock list |
| TakeOrder | Form used to note the customer’s order |
| Kitchen | Kitchen staff view the order tickets |
| Suppliers | Search, Edit or Add supplier contacts |
| Stock Update | View ingredients for re-order |
| Popular Foods | Graph of popular foods on the menu |

## Data Structures

Throughout the program, many 1-D arrays are used for the same 2 purposes, storing the ingredient id’s, and storing the meal id’s from their tables in the database. They are declared and used in smaller functions, such as saving a meal away in a table in the database, so that they can be iterated in a for-loop or using list comprehension.

### Ingredient ID Array

|  |  |
| --- | --- |
| **Position in array** | **Ingredient ID’s** |
| 0 | 27 |
| 1 | 28 |
| 2 | 29 |

### Meal ID Array

|  |  |
| --- | --- |
| **Position in array** | **Meal ID’s** |
| 0 | 56 |
| 1 | 57 |
| 2 | 58 |

## Data Requirements

**MENU (**MenuID, Type, Name, Description, MealPrice**)**

**INGREDIENTS (**IngredientID, Name, Quantity, CostPrice, MinOrderLevel**)**

**MENUINGREDIENTS (**MenuIngID, IngredientID, MenuID**)**

**SUPPLIERS (**SupplierID, Name, Address, Telephone**)**

All the tables in my database are fully normalised. This means there are no repeated fields, no partial key dependencies, and no none-key dependencies.

The table (MENUINGREDIENTS) is a link table. This stores the ingredient and meal id’s. Both of those id’s are foreign keys pulled from their own respective table in the database. This is all stored in a link table for use later where the program will query the table when taking an order to notify the waiter whether there are enough ingredients in stock to prepare the meal. This prevents the customer perceiving the restaurants reputation in a negative manner.

## File organisation and processing

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | File Type | Description | Master/Transaction |
| Data | SQL Lite Database | Stores records of menu, ingredients available etc | Master |
| Order Receipt | Text File | When a waiter takes the order, it is saved as a text file so that it can be sent to the kitchen to be prepared. The data is used to generate a graph to show the most popular meal item on the menu. | Master |

## Table/Record Structures

### Menu Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Purpose | Field Size | Data Type | Validation |
| MenuID | A unique number to identify each menu option | 1 | Integer | Primary Key |
| Type | A unique string to identify the type of meal in the menu (starter, main etc) | 15 | String | Required |
| Name | Stores the name of the meal | 40 | String | Required |
| Description | Stores a brief description of the meal | 50 | String | Required |
| MealPrice | Stores the price of the meal | 5 | Decimal | Required |

### Ingredients Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Purpose | Field Size | Data Type | Validation |
| IngredientID | A unique number to identify each ingredient. | 1 | Integer | Primary Key |
| Name | Stores the name of the ingredient | 40 | String | Required |
| Quantity | Stores the amount of ingredient in stock | 20 | String | Required |
| CostPrice | Stores the cost of the ingredient | 2 | Decimal | Required |
| MinOrderLvl | Stores the min amount of ingredient needed in stock | 20 | String | Required |

### Menu-Ingredients Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Purpose | Field Size | Data Type | Validation |
| MenuIngID | A unique number to identify each row / record in the table | 1 | Integer | Primary Key |
| IngredientID | A unique number to identify each ingredient. | 1 | Integer | Foreign Key |
| MenuID | A unique number to identify each menu option | 1 | Integer | Foreign Key |

### Suppliers Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Purpose | Field Size | Data Type | Validation |
| SupplierID | A unique number to identify each supplier in the table | 1 | Integer | Primary Key |
| Name | Stores the name of the supplier | 40 | String | Required |
| Address | Stores the address of the supplier | 75 | String | Required |
| Telephone | Stores the telephone number of the supplier | 11 | String | Required |

## Data Model

![Diagram

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAqACoAAD/4RDaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAFAAAISodpAAQAAAABAAAIUJydAAEAAAAKAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFJhamEAAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAMwMQAAkpIAAgAAAAMwMQAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMjowMjoyNyAyMjo0MDo0NAAyMDIyOjAyOjI3IDIyOjQwOjQ0AAAAUgBhAGoAYQAAAP/hCxdodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTAyLTI3VDIyOjQwOjQ0LjAxMjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5SYWphPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIA5MEkQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APfqKKKokKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKK5/WPFsGkayulrpeqahdNbi422FuJcJuK5IyD1Hp3FVv8AhNpP+hS8Uf8Agt/+yqeeKN1h6sldI6miuW/4TaT/AKFLxR/4Lf8A7Kj/AITaT/oUvFH/AILf/sqXPEf1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqP+E2k/6FLxR/4Lf/ALKjniH1ar2Oporlv+E2k/6FLxR/4Lf/ALKj/hNpP+hS8Uf+C3/7KjniH1ar2Oporlv+E2k/6FLxR/4Lf/sqRfHcK3lpBd6Br1iLu4S3jlu7IRpvY4AyW/H8DRzxD6tV7HVUUUVZzhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN2n/JaD/2L/wD7cV29cRaf8loP/Yv/APtxXb1nHqdNb7PogoooqjAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuN+I3+p8Nf9jBa/+z12Vcb8Rv8AU+Gv+xgtf/Z6mWx0Yf8Aio36KKK0OUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5u0/5LQf8AsX//AG4rt64i0/5LQf8AsX//AG4rt6zj1Omt9n0QVwfja/kPi3SNG1HWrrQtGvYJCbu1k8lp7gMu2HzcHZ8uTxgnpmu8rlvF3ibwrphGleMgiWt1HuU3Vq0kEnJBXdggMMZwcdRim7XVzFbMoeINJl8J/DPxJJp+taxczfZJJY57y+aWSFgnGx+q+vXrXUQyyHwzHKXYyGzDF8852dc+tec6Dpra34R8ZaR4aa6k8PXEJi0Y3W8De0Z3pGX58oNtAJ4zuxV63+KOh/8ACJrYv9pXX0tPIbR/sshuPPCY2bcevfpilO/LJdbL9QjZOL6Xf6GdJd6hqXwv8ArJq2owzalfWsN1dW926TSKyPnLg5PQdfStmC1u/B3xA0PTbXXdU1Oy1lZ1mttSuDcNEY03CRHPzKOxGcHNZOveHpbL4f8AgDQb2ae2nTU7OGaS2l2SRt5b7trjoQc8iu40PwbpWg3r30Jur3UJI/Ka+1C5e4m2ddoZicDPYYrV25pPzf5IzXwxXkvzZxug+HLjxPJ4hvbrxT4is5YNau4IRbak6xRIjfKAhyuBnp0xRB4m1XVPBfgfUJ7yRbm61yK3uZYCYhcoplUkgcFW2BsdPameFPBlj4hn8Sz6lfaqYG168jksYb+SK3lXePvIpGc559a0/iPJp3h3S/Ckj+VY6dY65bE7VwkUapJ2HYCoi7KF/wC7+hctXO3979TrPE80lv4S1eaCRopY7KZkdGIZSEJBBHQ1xGuT3t74C8Cxf2pqFtJqV5ZQ3NxbXTxzSK8LFsuDk5PPPete+8beHPE/hbX7bQNVhvZotNnkdIw3yrsIzyB3Nct4t/sz/hVfgL+3/L/sz7XY/avMzt8vyGznHNJb694/mxvbTtL8kd9ofhOPQLqW4j1nW78vGU8vUNQedF5zkK3Q8da880LRbq5+FKeK5/GPiK2v47aW58x9SZ4QyM2A0b5BX5RkHrXS+Brr4cR6lcW3gOSzF5NDulS335ZFPU7vQn9a5Hwv8OrLVPhXpmtaRCP7ejDXEQuZGmt53WRvkeJyUwQMZABBwQeKHezfp8twjudJ4h8SajJ4S8J3Oo3s2g2mreWNVvoF2tbbotwUMQfLDPgbj09RXUeHPDUOiySXNtrer6nHcRqFF/ftcoB1DLnpnPUVlN8RPDdx4UsdQ11WgstQDQzrNbtLHBKuA8MuAdpzkcjBxWX4Bk04+M9QXwO07eFntA8gKutvHdb+kO4d1yWC8dKv7bt5/wBf5evzI+yr/wBa/wBfcekUUUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVxvxG/1Phr/ALGC1/8AZ67KuN+I3+p8Nf8AYwWv/s9TLY6MP/FRv0UUVocoUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBzdp/yWg/9i//AO3FdvXEWn/JaD/2L/8A7cV29Zx6nTW+z6IKKKKowCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv8A7PXZVxvxG/1Phr/sYLX/ANnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAObtP+S0H/sX/AP24qxfaNoVtePFH4c0YquMFrFM9AfSq9p/yWg/9i/8A+3FaWrf8hSb/AID/AOgiueWz9T1aWs4r+6jO/s3RP+hb0X/wAT/Cj+zdE/6FvRf/AAAT/CpaKzudnKiL+zdE/wChb0X/AMAE/wAK881D4gaZZX+pQxfDnSbiHT52hluFgXauHKgt+7IXJHGTXpFeE+LfDVwniDxNexTxvDYzxzS7sqx887gFHIOC2Dkjpn2q4Wb1OXEc0IpwPXPDVzoniHw9bap/wiei2/n7/wB19jR9u1yvXAz0z0rV/s3RP+hb0X/wAT/CsXwRZ/YPBtjbbblNnmcXUHkyDMjHlMnHX16YNb9S3qbwjeKb3Iv7N0T/AKFvRf8AwAT/AAo/s3RP+hb0X/wAT/CsW31PUtXvtQj0yWztIrGc25FxC0zyMBktgOu0c8dc9eOlasd9Gt1DY3Mi/bZITKURTtIGAxB9MkdeaNR+7/X3Ev8AZuif9C3ov/gAn+FH9m6J/wBC3ov/AIAJ/hVWfWtPtlvmmuNo08A3J2MfLBGR2549M0yHxBps93FbxztumJELtC6xykc4RyNrfgTRqHuLr+Jd/s3RP+hb0X/wAT/Cj+zdE/6FvRf/AAAT/Cs+XxJpcM08RmlkltnKTJDbyStGcA5IVSQMHr096h1DVmNzoT6dcq9tfXJVmQBhInluw57cgdKFdifIuv4mt/Zuif8AQt6L/wCACf4Uf2bon/Qt6L/4AJ/hVKzvF+1ao02pLLFbygMjRCMWo2AkFv4hzuz2zjtS2evadf3CQW8snmSIXjEsDxiVR1KFlAccjpmjUfu/0y5/Zuif9C3ov/gAn+FH9m6J/wBC3ov/AIAJ/hVCLxBYXbyxWMrzyRl1bbBIVRlyCGbGF5B6kZ7VlReIdSu9F8PzwfZYbnVX2yM8TOifIzcKHB/h9aFd/wBdxNwX4/h/w50n9m6J/wBC3ov/AIAJ/hR/Zuif9C3ov/gAn+FY8up6jpF9ax6x9luLS7lWFLm2jaIxSHOAyFmyDwNwPB7d63aBpJ6EX9m6J/0Lei/+ACf4Uf2bon/Qt6L/AOACf4VLRSuVyoi/s3RP+hb0X/wAT/Cj+zdE/wChb0X/AMAE/wAKloouHKiL+zdE/wChb0X/AMAE/wAKP7N0T/oW9F/8AE/wqWii4cqIv7N0T/oW9F/8AE/wo/s3RP8AoW9F/wDABP8ACpaKLhyoi/s3RP8AoW9F/wDABP8ACj+zdE/6FvRf/ABP8KloouHKiL+zdE/6FvRf/ABP8Kb48s7awsPC9tY28VtAniC22RQoEVc7ycAcDkk1PSfEb/U+Gv8AsYLX/wBnq1sznqK1SCXn+Rv0UUV0njBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN2n/JaD/2L/8A7cVpat/yFJv+A/8AoIrNtP8AktB/7F//ANuK19Ttp5NSlaOGRlOMFUJB4Fc8tn6nq0mlON/5TOoqb7Hc/wDPvL/3waPsdz/z7y/98Gs7M7OaPchrzzxJoep3Nx4qht7OSR9Y+x/Yyoyr+UAX3N0THbeRntmvSfsdz/z7y/8AfBo+x3P/AD7y/wDfBpq6ImoTVm/62MTw1Dd2/h62i1FLlLld+9bq5FxIPnOMyAANxj6DA7Vq1N9juf8An3l/74NH2O5/595f++DRqUpRStc4rXZNBkuLq6hvzpmuQAorRN5c8rDBVTGf9apwuBg5HApTfPZ65omp+IDHZefpzwyu52okxKNtJPC8BsZPbFdp9juf+feX/vg0fY7n/n3l/wC+DTV1/X9dyGot3v8A1dP9Dze+vIr/AEvxvcW+WiZIwjY4cCIDcPUHqD3HNaep6hY6xZ6VpulOst2LqCQwx/ftVRgWLr1TA+XnHJxXT61oNzq+i3en7JYftEZTzPJLbc98cZq6tlcqgX7PKcDH+rNPt8vwJsnpfe/4nFaTrGmaVrXiL+0bmK0Zr3cHmO0SARrwpP3iPQc8j1qtY20tta+GhNC8HmapNKkLjDRoyysqkduCOK7DTdAudPuNQl2Syfbbkz48kjZ8qrj3+7196v8A2O5/595f++DSWiXy/ALJ3u+r/Fnn+pWtzeab4uhsld5TeRtsjGWdRHEWUDuSAeO/StGwOl6rqVhNF4ln1Ka3JmhgzBlMoQS4SMMowcc45wOtdf8AY7n/AJ95f++DR9juf+feX/vg0K9rf1tYfLG979/zucp4WRU8O35VQC95dsxHc+Yw/kBWJZXMFp4Z8FT3c0cEKTAtJK4VV/dSdSeK9G+x3P8Az7y/98Gj7Hc/8+8v/fBoV1+H4A1Frfv+Nv8AI5PVbqDxLJZWGkSrdxpdRz3FzC26OJUO7G8cFicDaOecniumqb7Hc/8APvL/AN8Gj7Hc/wDPvL/3waXSxacb3bIaKm+x3P8Az7y/98Gj7Hc/8+8v/fBpWZXNHuQ0VN9juf8An3l/74NH2O5/595f++DRZhzR7kNFTfY7n/n3l/74NH2O5/595f8Avg0WYc0e5DRU32O5/wCfeX/vg0fY7n/n3l/74NFmHNHuQ0VN9juf+feX/vg0fY7n/n3l/wC+DRZhzR7kNJ8Rv9T4a/7GC1/9nqf7Hc/8+8v/AHwag+I3+p8Nf9jBa/8As9Wtmc9Rp1YW8zfooorpPGCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAObtP8AktB/7F//ANuK7euItP8AktB/7F//ANuK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFcZY+Nrn/hPr/RNTt4YrAXAtbO6QEEzeWr7HJOMsGO3GPukc1tRaxO/jS70l1iFtBYxXKvg7tzO6nJzjGFHajt5/5XDa/l/wAMbNFY9p4q0m8vYLSKaeOW4BMH2i0lhWbAydjOoV+OeCeOelRXXjTQbOa7jmu5C1k+y68q1lkFucA5cqpCrg/eOAcHng4AN2isa38WaLdahDaW97vedmSGQQv5MzAZKpLt2MRg8BieD6U7UPFOkaZdS291cyGSBQ0/k28kqwA8gyMikRjHPzEcc9KANeimpIksayRMrowDKynIIPQg06gAooooAKKKKACiiigAooooAKKKKACiiigAooooAK434jf6nw1/2MFr/wCz12Vcb8Rv9T4a/wCxgtf/AGeplsdGH/io36KKK0OUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5u0/5LQf+xf/APbiu3riLT/ktB/7F/8A9uK7es49TprfZ9EFFFFUYBRRRQAUUUUAcHYaNb6/qHjbT7ssqyahGUkThonEKFXU9iDg1S0TWdUPinX5LqxaXWNN0eOKSFBxcSo0hVk9nBUj647V6TRSttbt+lrj/wA/1ueWz6xHq+oeFJz4h/tS4l1KKWW1tYoxBakxvwcAujdgHck4Y4442tPRTYePCVGWvJwTjqPsyf413FFDV013v+Nv8gTs0+1vwv8A5nnd4ixeBfAYjG0Le6fjHb5KgsWg07UNfsfEHiufQ5pr+WYRSG2RbiGT7jq0sRLcDbwTjbjivS6Kpu7b73/G3+RK0SXa34X/AMyjoljBpug2NlZvM9vbwJHE04w5UDjcMDBx7D6Veooobu7glZWCiiikMKKKKACiiigAooooAKKKKACiiigAooooAK434jf6nw1/2MFr/wCz12Vcb8Rv9T4a/wCxgtf/AGeplsdGH/io36KKK0OUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5u0/5LQf+xf/APbiu3riLT/ktB/7F/8A9uK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFc14s8SXmiSW0WmW8dxIEe7uldSdttHjeVwR83zDHXoeDW1dapY2OltqV3cxxWaoJDMx4wemPXORgdTmjpcOti3RXJnxWt7430fTdOnmjilhuHube4tXhc7Quxtsihsctgjg4PpTvEnjTTrDRtXFleut3aQyos62zvDFOEJVDLtMYbOBtJ6kDGTik3ZXGld2OqorCXxRYWNlpqardN9tvLRZkjSB3eXhd21UU5OWHA59sA1YvPEumWLwxzyTmaaLzlt4rSWWZU/vNGillGeMsBzx1qmrMlO6uatFZc3iTSYbC1vPtfnRXn/AB7C3jaZ5uMnaiAs2BycDjvimyeJtLisYLp5pgtw7RwxC1lMzspIYCLbvyMHPy8YzSGa1FV7C+g1KzW5tDIY2LAeZE0bAgkEFWAIIIPUVYoAKKKKACiiigAooooAKKKKACiiigAooooAK434jf6nw1/2MFr/AOz12Vcb8Rv9T4a/7GC1/wDZ6mWx0Yf+KjfooorQ5QooooAKKKKACiiigAooooAKKKKACiiigAooooAKKK5LxVBPqPjDw1pUepX9hBefavOaynMTNsjVl56dR3Hc0pOyuaUqftJct7b/AIK51tFYH/CuY/8Aoa/FH/gx/wDsaP8AhXMf/Q1+KP8AwY//AGNTzPsaezpfz/gb9FYH/CuY/wDoa/FH/gx/+xo/4VzH/wBDX4o/8GP/ANjRzPsHs6X8/wCBv0Vgf8K5j/6GvxR/4Mf/ALGj/hXMf/Q1+KP/AAY//Y0cz7B7Ol/P+Bv0Vgf8K5j/AOhr8Uf+DH/7Gj/hXMf/AENfij/wY/8A2NHM+wezpfz/AIG/RWB/wrmP/oa/FH/gx/8AsaP+Fcx/9DX4o/8ABj/9jRzPsHs6X8/4G/RWB/wrmP8A6GvxR/4Mf/saP+Fcx/8AQ1+KP/Bj/wDY0cz7B7Ol/P8Agb9FYH/CuY/+hr8Uf+DH/wCxo/4VzH/0Nfij/wAGP/2NHM+wezpfz/gb9FYH/CuY/wDoa/FH/gx/+xo/4VzH/wBDX4o/8GP/ANjRzPsHs6X8/wCBv0Vgf8K5j/6GvxR/4Mf/ALGj/hXMf/Q1+KP/AAY//Y0cz7B7Ol/P+Bv0Vgf8K5j/AOhr8Uf+DH/7Gj/hXMf/AENfij/wY/8A2NHM+wezpfz/AIG/RWB/wrmP/oa/FH/gx/8AsaP+Fcx/9DX4o/8ABj/9jRzPsHs6X8/4G/RWB/wrmP8A6GvxR/4Mf/saP+Fcx/8AQ1+KP/Bj/wDY0cz7B7Ol/P8Agb9FYH/CuY/+hr8Uf+DH/wCxo/4VzH/0Nfij/wAGP/2NHM+wezpfz/gb9FYH/CuY/wDoa/FH/gx/+xo/4VzH/wBDX4o/8GP/ANjRzPsHs6X8/wCBv0Vgf8K5j/6GvxR/4Mf/ALGj/hXMf/Q1+KP/AAY//Y0cz7B7Ol/P+Bv0VwXjbwrL4b8H3uq2PifxFJPb+XsWbUCVO6RVOQAD0Y9672nGV3Zk1KSjFSi7p3/C3+YUUUVRiFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVy3ie41eTxNoGkaNqn9m/wBofafMl+zpN/q0DDhvxHUda6muZ1f/AJKb4O/7ff8A0SKiex0Ya3tNV0f5Ml/4RPxd/wBD1/5SIv8AGj/hE/F3/Q9f+UiL/Gu0opcq/psf1ifZf+Ax/wAji/8AhE/F3/Q9f+UiL/Gj/hE/F3/Q9f8AlIi/xrtKKOVf02H1ifZf+Ax/yOL/AOET8Xf9D1/5SIv8aP8AhE/F3/Q9f+UiL/Gu0oo5V/TYfWJ9l/4DH/I4v/hE/F3/AEPX/lIi/wAaP+ET8Xf9D1/5SIv8a7SijlX9Nh9Yn2X/AIDH/I4v/hE/F3/Q9f8AlIi/xo/4RPxd/wBD1/5SIv8AGu0oo5V/TYfWJ9l/4DH/ACOL/wCET8Xf9D1/5SIv8aP+ET8Xf9D1/wCUiL/Gu0oo5V/TYfWJ9l/4DH/I4v8A4RPxd/0PX/lIi/xo/wCET8Xf9D1/5SIv8a7SijlX9Nh9Yn2X/gMf8ji/+ET8Xf8AQ9f+UiL/ABo/4RPxd/0PX/lIi/xrtKKOVf02H1ifZf8AgMf8ji/+ET8Xf9D1/wCUiL/Gj/hE/F3/AEPX/lIi/wAa7SijlX9Nh9Yn2X/gMf8AI4v/AIRPxd/0PX/lIi/xo/4RPxd/0PX/AJSIv8a7SijlX9Nh9Yn2X/gMf8ji/wDhE/F3/Q9f+UiL/Gj/AIRPxd/0PX/lIi/xrtKKOVf02H1ifZf+Ax/yOL/4RPxd/wBD1/5SIv8AGj/hE/F3/Q9f+UiL/Gu0oo5V/TYfWJ9l/wCAx/yOL/4RPxd/0PX/AJSIv8aP+ET8Xf8AQ9f+UiL/ABrtKKOVf02H1ifZf+Ax/wAji/8AhE/F3/Q9f+UiL/Gj/hE/F3/Q9f8AlIi/xrtKKOVf02H1ifZf+Ax/yOL/AOET8Xf9D1/5SIv8aP8AhE/F3/Q9f+UiL/Gu0oo5V/TYfWJ9l/4DH/I8z8V2Pi7wx4Zu9X/4TH7V9n2fuv7LiTdudV684+9np2ruawvit/yTLVf+2P8A6OSt2iOkmh1Zc9KMmle72SXRdgooorQ5QooooAKKKKACiiigAooooAKKKKACiiigDm7T/ktB/wCxf/8Abiu3riLT/ktB/wCxf/8Abiu3rOPU6a32fRBRRRVGAUUUUAFFFFAHDaXBqviDXtZ13T7uxhtJm/s+AXVm1xviiJDMpWVMAuX7HoKxxLPZ+H7Cw1QtMPC2sxrfFEJ/0cKximK8kKAyHvjaTnivUaKFpb5fg7/5/eD1v8/yt/l9xwtzrOn6z8TNA/si4juljtLsG6gIePJCfKHHBIxkgHjIz1rHj1rTdN+EF/od46rq9vZ3EE+nsMzmQhiX2dSpzv34xjnPWvUqKTV48v8AXX/Madnc4i0jR/GfhRnUEx6JMyk9j+5GfyJ/Oqlwy6X8QNafWPEc+gx3qQyWkx+zrHOiptZA8sbDKsc7QR9/OOa9Coqm7u/r+LuSlZW9PwSPP7jTtA07SNLe18WNpdyjXNxYajdNFiXzGzJlWVUZCWBAGMjBBxzTZNSi1jRNJ1TXNbg8NazGbj7FeEKkUyZ2lgkvDIy7GxkHoQcV6FRSGYvhPVb3WfDkN7qcKRTu7gGNWVJVDELIoYkhWABGfWtqiigAooooAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv/ALPXZVxvxG/1Phr/ALGC1/8AZ6mWx0Yf+KjfooorQ5QooooAKKKKACiiigAooooAKKKKACiiigAooooAK5nV/wDkpvg7/t9/9EiumrmdX/5Kb4O/7ff/AESKie33fmdGG+N+kv8A0lnc0UUUzEKKzda1220JbE3aTP8AbbyOzj8oA4d84JyRxxz1PtWdqPjnSNK8bWPhe+8+K8vohLDMVHk5JYBC2chiVOOMdOaFqGx0dFZmua7baBDaS3iTOt3eRWaeUASHkbapOSOM9f5VD4g8U6d4cWFbzz57q5JFvZWkRmnnIGTtQc4HcnAHrQBs0VzeieMk1jVBYT6FrekzMheM6jZhEkA6gOrMM+xOazZPiZB9qvorTwx4jvorC5ktpri0s0kj3ocNjEmT+WfagDtqK56+8baTZ+Cf+EqhM19pxVGT7MoLvucIAAxHIJwQSMYNR6R4wl1bU47N/C3iDTw4J+0XttGkS4GeSJCeenSi2tg6XOlorim+JULXl9DZeF/Ed+ljcyWss9paRyRl0OGx+8yfyz7Vv6B4l0zxLZyz6XMzGBzFPBLG0csDjqrowBU0bq4PR2NaiuYk8e6ZH4DbxZ9mvHslbb5KovnE+b5XA3Y+979KdpHjCXVtTjs38LeINPDgn7Re20aRLgZ5IkJ56dKA2OlornPD/jnSPEev6to1l58V7pUrRSxzqF8wKxUumCcrkY7Hkcc1euPENtb+JF0Qw3El21i96uxQVKKwUr1zuJYYGMe9HbzDuuxq0VwknxSji1KGwk8HeKVu50aSKE2cW51XG4j970GR+dat545s7GxsXm0vVjf3ys8Okx2u+62qcMWUHaoGRyWA5o6XA6aisnQNebXoJpH0fVNKMLhPL1GBY2fjOVwzAj8a1qACiiigAooooAKKKKACiiigAooooA4/4rf8ky1X/tj/AOjkrdrC+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyiFFFFWcwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXM6v/AMlN8Hf9vv8A6JFdNXM6v/yU3wd/2+/+iRUT2+78zow3xv0l/wCks7miiimYmXr/AIk0nwxZw3Wu3YtIJplgRyjMC7AkD5QcdDyeK1K4L4r6bb6xYeHdNvU3293rkMMi+qtHIDUekeLZtE+G18NTzPq+gO2nSR85uJhhYSM8neGjOfc0ls32/wCB/n+I2tUu/wDwf8jrtP8AEelarq2o6ZYXYmvNMZVu4wjDyiwJAyRg9D0Jxjmufi+LXgyfb5Op3EgY4UrptyQecdfLxWN8NtFfQPGHiOwuJPOuhZWMt1KTnzJnEryN+LMarfDLxfLaeB9G04eF9fuFG5PtkFtGYDmRvm3GQHAzzx2NUlrb0/UV93/W1zv5/FGjWvia28P3F8seqXUPnQW7I37xOeQ2NuflPGc8dKtalq9lo8cEmozeStxcR20R2M26RzhV4Bxk9zxXnfi3w2PE/wAUrm0im+zX0OgRXFjdAcwTpcsUYe3Y+xNJrPiQ+JPCehSXUP2XUrTxHZW2oWh6wTLKNw/3T1B7gilHW3m/w5rf18glo36fja56LHq1lLrU+kxzZvoIUnki2N8qMSFOcYOSp4zniiXVrKDWbfSpZtt7dRPNDFsY7kQgMc4wMbh1PeuZ0/8A5LRrX/YHtf8A0ZLRqv8AyWPw7/2DL3/0KKha8vnf8L/5A9L+Vvxt/mdlRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/AGx/9HJW7WF8Vv8AkmWq/wDbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAKKKKACiiigAooooAKKKKACiiigDm7T/ktB/7F/8A9uK7euItP+S0H/sX/wD24rt6zj1Omt9n0QUUUVRgFFFFABRRRQAUVyGqal4nsPEmlaZHf6Sy6m0wWRtNlzEEXd08/wCbPTtW3c6xBo1tbJrV0sl3NlVS0tpGaYjklIl3vgDr1x3NHS4dTUorl9d8ShvDdvqGg3f39Qt7d2MeGXMyo6MrjKnBIIIBFa2o+INO0u5W2upZWuGQyCC2t5J5AmcbisasQueMkYzR0uBpUVkz+KdFtrSxupdQjEF/kWsigsJSFLYGB1wOh5J468UW/iGx1KK8jsJZVubaPe8NxbyQSICDhtkig4464xSbsnfoNK7RrUVyHhjxvYXfh3R31S9d7u7ijSS4+zOIWmYfdMir5YbPG3I54xW7qGv6dpl0lrcySvcunmCC2t5LiQJnG4pGrELnjJGM1TVnYlO6NKismTxRosWhprMmoRrp7sqCcg4VmbaAwxlSCcHOMc5xinWHiPTNSvmsraaVbkR+aIp7eSEumcbk3qN657rkcj1pDNSisaXxbosF49vJdtmOUQSTLBI0MchONjShditkgYLAgkCtmjzAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv8A7PXZVxvxG/1Phr/sYLX/ANnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKKKKACiiigArmdX/5Kb4O/wC33/0SK6auZ1f/AJKb4O/7ff8A0SKie33fmdGG+N+kv/SWdzRRRTMTjfiL/qvDX/YwWn82rJ8R+HrbxR8UL/SrxmjWbw5GY5k+9DILlijr7ggGuv8AEfh/+3100fafs/2HUIb3/V7t/lk/L1GM568/ShfD+3xy/iL7T97ThY/Z/L6YkL792ffGMfjRHz7v/wBJt+Y2+3Zf+lX/ACOA1TX7rVvDuk6fraCLXNJ8R2FtfoOjnzAVlX/ZdfmH4jtXSaCqXHxc8VT3QDXNrb2kFsW6pAyF2A9i+c/SpfFXw/g8SeI9I1mG9awuLCeOScLHvF2kbh1RuRggg4bnG48Vb8Q+ETq2qQaxpOpz6NrNvEYUu4UWRZIyc7JI24dc8joQeQaE9Lve7/JK/wCH3NitrZdl+bZWj8W6vD44s9B1Xw9HaW9+Zza3i34kLrEM5MYQbcgjjPGe9cdo/ivWfDdr4qms/DMt9Yxa7ePJf/alWOH5xksgDSFVHJKqeK7HTPB+qjxHaa14l8SPq1xYLItrFFZR20aeYuHJALMxOB3GMVp+HfDo0GHU42uBdLqGoT3pBj2hBIc7Opzj14z6UW/J/mv0Hdfj+jOE8Q6SND/Z7e2t7uPUWYw3HnxNiOVpLlJDtPZctx7V2eiat4ovdQ8rW/C8Gl2uwnz01RZzu7DaEHX1zWVN8O5D4I1Pwva6x5NlcXImsd1tuNknmCTy/vDeu4HHTAPfFaGl6L4utdThm1Xxdb39opPmW6aQsJkGP74kOOcHpTVr/wBdkTrb+v61Knw4/wCPfxL/ANjFe/8AoQqOBEj+N96bTgS6HG14F6FxKRGW99u78BUUPgfxPp11qR0TxnHYWt/fS3hi/shJWjaRskBmfnHHb8K6Hw74XtfDsVy8c895e3r+Zd31026WdgMDOMAADgKAABU9E+yt+Fhv7S7v9bnmkpZf2ZyY13MJwVUnGT9u6Zr0PRNW8UXuoeVrfheDS7XYT56aos53dhtCDr65rOk+H7N8MW8Ix6psYvvW8NtnB8/zfubvw+97+1XNL0Xxda6nDNqvi63v7RSfMt00hYTIMf3xIcc4PSqW7/rogexw9hot5Na6z4i8Px79b0XxHfyQxZwLqIsPMgP+8OnowHrW9pes2niH4p6PqunOXtrrw5K6ZGCP36ZUjsQcg+4rqPDfh/8A4R+PUl+0/aPt2ozX2fL2bPMIOzqc4x14z6Vk6T8P4NG+Il54lsr1lt7qB4xp3l/LE7srO6tnjJTJXHUk5pR0cb9F+PLb8/61CWvN5v8ADmuM1X/ksfh3/sGXv/oUVWfE3h/Vptcs/EXhe5t49UtIXt3t7tSYbmFiGKEjlGyoIYfjxWhd+H/tXjLTte+07fsNrNb+R5ed/mFDndnjGzpg5zUGv6Jrt9fRXXh/xM+kFY/Lkhks0uYpOSQ20kFW56g88Ulol8/zf6MfV/L8kO8MeJzr63lteafNpmqafII7yzlYNsJGVZWHDIR0PHQ8VvVgeGvDD6HNfXt/qUuq6pqDIbm7kjWMEIMIiovCqATxzySc1v1TJCiiikMKKKKACiiigAooooAKKKKAOP8Ait/yTLVf+2P/AKOSt2sL4rf8ky1X/tj/AOjkrdpR+J/L9Tef8CPrL8ohRRRVnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/8AJTfB3/b7/wCiRXTVzOr/APJTfB3/AG+/+iRUT2+78zow3xv0l/6SzuaKKKZiYHirQrrXJdDa0eFBp+qxXsvmkjciBgQuAefmHXA96zNV8BjUfiLY6+J1WxjCy3dof+W88QIgfp/Dvbv/AArwe3ZUULT8/wBAev5fqc/p2g3Vn4413WZZITb6jBbRxKpO9TGHDbhjH8QxgmuZ8N6N8RfDHh600e0i8LzQ2oZVklubjcwLFucR4716NRQtAOei0K8HxEbxBI8H2dtJWyKKx3+YJS5OMY24Prn2rE8XeALzV/Fml6zodzBbAXlvNqkMpIFwsLhkYYB+cDcvOMggZ4rvKKFo0+3+dw7+f+Vjk/EPh3Wv+Emg8R+Erqzjv1tvslzbX4bybiLduU7k5VlJPODnP5u0Lw/rLeJX8ReK7mye+W1NpbW1greTBGWDMdz/ADMxIHYAAV1VFC0/ruD1CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDj/AIrf8ky1X/tj/wCjkrdrC+K3/JMtV/7Y/wDo5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN2n/ACWg/wDYv/8AtxXb1xFp/wAloP8A2L//ALcV29Zx6nTW+z6IKKKKowCiiigAooooA5TxF/yULwj/AL13/wCiqz/EebH4i29/qOsz6NYT6cbeG8QReWsocs0bNKjKu5cEdM7Mc4ru6KOwf1+p51qlnp8PheS70vVZtVW91y0llupChV3EsSnaY1VSMKBkDqDzkGn3DLpfxA1p9Y8Rz6DHepDJaTH7Osc6Km1kDyxsMqxztBH3845r0Kij+vwS/QP6/Fv9Tza/i0rQm8HyQ3M15p41K4uPtMyg53pIxk+VQNgJLZAxjnpzWmbu213xvJqOjSpdWdlpMsE95AwaJ5HZWWMMOGKgEn03D1ro9S0b+0NY0m+8/wAv+zZnl2bM+ZujZMZzx97PfpWnSauvv/FWHezv6fnc82uI0HwAscKBttLVxjjDeYhz+da1vfWfh3x3r0uv3UVimoiCS0ublxHHIiJtKBycblbJ29cNmuzoqm7yb7kpWVjzC7gaTw1qN40TJZaj4kt57VJE274zLEpfaezEEj1Bz3rpteTPxA8LsvyuUvF3e3lqf5gV1NFLpb+tkv0H1v8A1u3+p5TolvYP4Sbw94j8V3lhdBpLe70s/ZlkdmYnKBojI+/cGBBJO7g5r1SNPLjVASQoAyepp1FO4dQooopAFFFFABRRRQAUUUUAFFFFABXG/Eb/AFPhr/sYLX/2euyrjfiN/qfDX/YwWv8A7PUy2OjD/wAVG/RRRWhyhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/8lN8Hf9vv/okV01czq/8AyU3wd/2+/wDokVE9vu/M6MN8b9Jf+ks7misqa48QLPILfTNNeIMQjPqMisy54JAgODjtk/U0z7T4k/6BOlf+DST/AOR6Lk+zfl96Niisf7T4k/6BOlf+DST/AOR6PtPiT/oE6V/4NJP/AJHoug9nLy+9GxRWP9p8Sf8AQJ0r/wAGkn/yPR9p8Sf9AnSv/BpJ/wDI9F0Hs5eX3o2KKx/tPiT/AKBOlf8Ag0k/+R6PtPiT/oE6V/4NJP8A5Houg9nLy+9GxRWP9p8Sf9AnSv8AwaSf/I9H2nxJ/wBAnSv/AAaSf/I9F0Hs5eX3o2KKx/tPiT/oE6V/4NJP/kej7T4k/wCgTpX/AINJP/kei6D2cvL70bFFY/2nxJ/0CdK/8Gkn/wAj0fafEn/QJ0r/AMGkn/yPRdB7OXl96Niisf7T4k/6BOlf+DST/wCR6PtPiT/oE6V/4NJP/kei6D2cvL70bFFY/wBp8Sf9AnSv/BpJ/wDI9H2nxJ/0CdK/8Gkn/wAj0XQezl5fejYorH+0+JP+gTpX/g0k/wDkej7T4k/6BOlf+DST/wCR6LoPZy8vvRsUVj/afEn/AECdK/8ABpJ/8j0fafEn/QJ0r/waSf8AyPRdB7OXl96Niisf7T4k/wCgTpX/AINJP/kej7T4k/6BOlf+DST/AOR6LoPZy8vvRsUVj/afEn/QJ0r/AMGkn/yPR9p8Sf8AQJ0r/wAGkn/yPRdB7OXl96Niisf7T4k/6BOlf+DST/5Ho+0+JP8AoE6V/wCDST/5Houg9nLy+9GxRWP9p8Sf9AnSv/BpJ/8AI9H2nxJ/0CdK/wDBpJ/8j0XQezl5fejH+K3/ACTLVf8Atj/6OSt2uZ+JD30nwr1c6lb29vLuhAW3naVSvnR85KLznPGPxrpqUfify/UuorUIrzl+UQooorQ5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuZ1f/kpvg7/t9/8ARIrpq5nV/wDkpvg7/t9/9Eiont935nRhvjfpL/0lnSTavexTyInh3UplViBIklsFcA9RmYHB9wD7Uz+2r/8A6FjVf+/tr/8AH62KKLeZPMv5V+P+Zj/21f8A/Qsar/39tf8A4/R/bV//ANCxqv8A39tf/j9bFFFn3Dnj/Kvx/wAzH/tq/wD+hY1X/v7a/wDx+j+2r/8A6FjVf+/tr/8AH68S+JfiTVLH4ma3ax+ItXsLeKKM28NnM2wyeTGQpG9QqkkksMn2Oa7/AOCOp3+reCbufVL65vZl1B0WS5maRgvlxnGWJ4yTx70WfcOeP8q/H/M67+2r/wD6FjVf+/tr/wDH6P7av/8AoWNV/wC/tr/8frYoos+4c8f5V+P+Zj/21f8A/Qsar/39tf8A4/R/bV//ANCxqv8A39tf/j9bFFFn3Dnj/Kvx/wAzH/tq/wD+hY1X/v7a/wDx+j+2r/8A6FjVf+/tr/8AH62KKLPuHPH+Vfj/AJmP/bV//wBCxqv/AH9tf/j9H9tX/wD0LGq/9/bX/wCP1sUUWfcOeP8AKvx/zMf+2r//AKFjVf8Av7a//H6P7av/APoWNV/7+2v/AMfrYoos+4c8f5V+P+Zj/wBtX/8A0LGq/wDf21/+P0f21f8A/Qsar/39tf8A4/WxRRZ9w54/yr8f8zH/ALav/wDoWNV/7+2v/wAfo/tq/wD+hY1X/v7a/wDx+tiiiz7hzx/lX4/5mP8A21f/APQsar/39tf/AI/R/bV//wBCxqv/AH9tf/j9bFFFn3Dnj/Kvx/zMf+2r/wD6FjVf+/tr/wDH6P7av/8AoWNV/wC/tr/8frYoos+4c8f5V+P+Zj/21f8A/Qsar/39tf8A4/R/bV//ANCxqv8A39tf/j9bFFFn3Dnj/Kvx/wAzH/tq/wD+hY1X/v7a/wDx+j+2r/8A6FjVf+/tr/8AH62KKLPuHPH+Vfj/AJmP/bV//wBCxqv/AH9tf/j9H9tX/wD0LGq/9/bX/wCP1sUUWfcOeP8AKvx/zOG+JF1Nd/CvV3uLG4sWDQgR3DRliPOj5+RmGPxzx0rpqwvit/yTLVf+2P8A6OSt2lH4n8v1Lqa0I+svyiFFUNd1NtG0C91JLZ7o2kLS+SjBSwUZPJ6ADk9TgHAJwC/R7/8AtXQ7HUfL8r7XbRz+Xu3bNyhsZ4zjPWtDmLlFFFABRRRQAUUUUAFFFFABRRRQAUUUUAc3af8AJaD/ANi//wC3FdvXEWn/ACWg/wDYv/8AtxXb1nHqdNb7PogoooqjAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuN+I3+p8Nf9jBa/8As9dlXG/Eb/U+Gv8AsYLX/wBnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKKKKACudTXr4/EiTQJIbdbIaZ9sjkUsZGbzAnPQAdeMHoDnnAueJdZk0DQZr+3snv51ZEitY2IaVmYLgYBOeSenauA/4TTUv7U/tL/hVN39v/wCfryW837u37/k56cdenFAHqtczq/8AyU3wd/2+/wDokVV8L+MtZ17Vzaaj4Qv9JhERf7TOW25BHHzIvXPbJ9sZItav/wAlN8Hf9vv/AKJFRPb7vzOjD/G/SX/pLO5ooopmIUUUUAfM3xXe1T4ra99shmlzFEIvKmEeyTyI8M2Vbco7qME+or0n4Bf8iFe/9hOT/wBFRVB49srW98e619stobjyPBk80XmxhvLkErYdc9GGeCOa6L4Tf8iFbf8AIX/g/wCQp/1yT/U/9Mf7v40AdrUS3du129qs8RuEQO8Icb1U8AleoBwefapa46N2j+J+uuhwy6NAQfQ75KTdvx/Jsdr/AIfnY7GivO4tX8R23w+03xVc6w1xMUgknsxbxLDJG7BT/DvD4bOQwGR93HFbf2rU9e8VanY2epTaXZaUscZa3ijd55nUOSTIrAKowMAZJJ56VTVnYlO6udTRXnbeIvEUXgm7uDewyapDrn2FZRCqxlfPVMbfTB+vvnmtLXdWvdCuNL0f+1L2efUJJJJb1bETyxRIoyI44o8ZJIALK2BnOeKXS/8AWyf6j62/rqv0Oyqre6la6e9st5L5bXc4t4RtJ3yEEgcDjhTyeK4w+LNU0vT9ZbZeX8cAgFhdajYPamSSVtmxxsTcFbByqjg4680uv6dqtlqXhdr3WpNRjbVohKs0EabX8t8FNijC9eG3HpzwcnVeqDo/mdjaapZ311eW1rN5ktlIIrhdpGxioYDJHPBHTNW64m98XXulWHiu7lIuDp96lvZxtHwpeOPaDtGWG58nqccDtS6bq+px69psCXWr6pBdFkvPtejPbJbnaWDo3lJgZG3axY/MOeORa2+X5XB6X+f4Ha1G1xAlxHbvNGs0oZo4ywDOBjJA6nGRn6iuDtr7xJeeEtW1sa2yTWM92ba3S2i8uRIpGwsmVychcZUrxjqeaddR3Or+OPDV7Fql5Z/bNMmmVIlhIiyIiVG6M8HPOc9OMULVr+ujf6A9P67Ox2ranYIl073tuq2f/HyxlUCDjPz8/Lwc89qq2XiXQtSulttO1rTrudgSsUF2jsQOvAOa4XUf+QN8S/8AfP8A6TJXX69pdpqvguWG9RSI7bzYpP4oZFTKyKezD1FJu0eZ9k/vQ7XdvN/gb1FZPha9uNS8I6Ve3uftE9pFJKSMZYqMn8eta1VJWdiU7q4UUUUhhRRRQAUUUUAFFFFAHH/Fb/kmWq/9sf8A0clbtYXxW/5Jlqv/AGx/9HJW7Sj8T+X6m8/4EfWX5RCiiirOYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArmdX/wCSm+Dv+33/ANEiumrmdX/5Kb4O/wC33/0SKie33fmdGG+N+kv/AElnc0UUUzEKKKKAPHviHoOmal8QNZe8s0doPCct8jKSh89HZVc7SNxAAHORgAdhXS/B02p8BR/Y7+G8/ejzfKshbeTJ5UeY2wB5jDvIcls9eK19d8Gxaxq82ox3jwTXWntpdypQOptmYs2zkFZMnhjuA/umr3hnw7F4X0gadb39/ewq2Y2vphI0a7QoRcAAKAvA7ZNAGvXmmvvLpnxUuPEEJby9Os7VbtQeDbyPIrtj/ZIVv+AmvS6whoDS+KNVvbwQy2N/YxWpiJJY7TJuyMYwQ47+tGt00HRr+t0cx8Vbp9R0O50e0YGKOyk1C8ZT0jQful/4E+D9ENaN/wCLU0yfTNGi1DTNOlexW4lutSfCRrgKqqm5d7MQf4hgDPOQKitfAt5beCdb0t7yO51DUYmt47iRjtWJU2QqSBnhRk8HknrWjPoOpWes2OsaMbSW5jshY3VtcyNGkqD5lZXVWIYNnqpyCemKFp6afk/+Agev4/p/wTLbx5eP4XurvT10+/vbTU4rEvDIfs9wHdAGUgnblX9WwQetbEWravYeKrHStZNncRajFK8EtrC8RiePBKNudtwIPB+XoeKXWNK1bXdCiguhZW90t9BcFI5XdFSORWI3lQWJCn+FRzjtk2tS0ie88UaLqUTxiGwE/mqxO5t6BRjjHUc5Iprz/rRfqJ/197/Qz7TxLeXHw/1DXXjgF1bJdsiBTsPlO4XIznnYM8+vSqZuHvPiF4XuZQoebSLiRgvQE+UTj86jn8MeIovDeq+HdOk04Wd21wYLuSVxJGkpLFGjCEHliu7d0OcEjB1IfDt3H4i0K/MkPladp0lrKAx3M7BMFeOnyHrjtSjvf+tn/wAActrL+tVYx/CPhbQdW0K8m1HR7K4nk1C8Vp3gXzf9e4GHxuBHYg5HatbwXd3LR6tpl5cSXR0u/e2inmbc7x7Vddx7kBsZ74qDRtN8TaJYz2NtbaS6SXc8yXMl5KSokkZwTEIhkjd03jOOo61t6DoyaJp7Q+c9zPNK09zcOMGaVvvNjsOwHYAChbfL8dAe/wA/8zSooooAKKKKACiiigAooooAKKKKACiiigAooooA4/4rf8ky1X/tj/6OStTUr6LS9Ku7+4V2itYXmcIAWKqpY4zjnArL+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyieW6j8YvCWq6dNY32n6w9vOuyRUCoWXuMrKDg9CM8jIPBq54X+KHhV5rDQdMttStYyfJgNyA6p6KW3s2Ow7DjoBx6NWV4mhefwtqSxDMq27vH/vqNy/qBVnMatFRwTJcW0c8ZykiB1PsRmpKACiiigAooooAKKKKACiiigAooooA5u0/5LQf+xf/APbiu3riLT/ktB/7F/8A9uK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVxvxG/wBT4a/7GC1/9nrsq434jf6nw1/2MFr/AOz1Mtjow/8AFRv0UUVocoUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjan/pPiLR7QciNpbxx7IuwZ/4FKD/wGtmsex/0nxXqlx1W1jitF9mwZG/MSR/lWxQAVzOr/wDJTfB3/b7/AOiRXTVzOr/8lN8Hf9vv/okVE9vu/M6MN8b9Jf8ApLO5ooopmIUUUUAYmreFNO1i/a9n86K4ltjZTvE/+vtiSWhYEEBSTyy4f0YVPoHhvSvC9g9loVr9lt5JTKyeY75cgAnLEnoo/KtSigArG/4R/wD4qa/1f7T/AMflklp5Xl/c2ljuznn73THatmik0mO9jnJPCXmfD+Hwz9tx5UEUX2nyuuwqc7c99vrWdq1xYaL4pvbq28U6fo11dQobu21GMOsmBhJIwXQ7sAg4LDgcZFdpRTbbdxKyVjgvDnhee+8E/Z7ie4gM2rHUEe6i/esizh13r8uCwXPQY3dO1dLruhNqs9je2d39i1HT5C9vOY/MXDDDo6ZG5WHoQc4OeK2KKPJf10/QPP8Ar+tTDutButY0W80/xBqCTi5ChWs7fyBDtOQyhmc7twByTjgcdc0pvC2raje6Tc6xr0c50u4WdEgsvKWYhSCXy7fNzwRgDn5TkY6mijrcOljnZfB9td22vW19O8kOsziYiNdjQkIijB5yQUDA4/Crmm2WuQSRDVdYtruKNMEQ2JheU4xl2MjD3woXnHbg61FGwGFZ+Gvsnhe/0f7Xv+2Nct53l42eczN0zzjd6847VFL4Ynj/ALFm0+/jhvNJtzbCSa3MiSxlVVgUDqQcqpB3ce9dFRRt/XYN/wCu5zNz4O+0WXiaD7dt/t453eTnyP3YTpu+bpntT5vD2q6lZjT9b1iCTTyoWWGxs2gaZR/AztI/ynuFAJ9QMg9HRR/X3B/X3jY40ijWOJQiIAqqowAB0Ap1FFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/AGx/9HJW7WF8Vv8AkmWq/wDbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuZ1f8A5Kb4O/7ff/RIrpq5nV/+Sm+Dv+33/wBEiont935nRhvjfpL/ANJZ3NFFFMxCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4/4rf8ky1X/tj/AOjkrdrC+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyiFIyh1KsMqwwQe4paKs5jI8KsT4WsInOXt4/szE92iJjP6rWvWPoH7qbV7T/AJ4ag5A9pFWXP5yGtigAooooAKKKKACiiigAooooAKKKKAObtP8AktB/7F//ANuK7euItP8AktB/7F//ANuK7es49TprfZ9EFFFFUYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVxvxG/1Phr/sYLX/ANnrsq434jf6nw1/2MFr/wCz1Mtjow/8VG/RRRWhyhRRRQAUUUUAFFFFABRRRQAUUUUAFFFZniK5e08OX0kHExiMcP8A10b5U/8AHiKAIfC/73RjenrfzyXQPqrMdn/jmwfhWzUNpbR2VlBawjEcEaxoPYDA/lU1ABXM6v8A8lN8Hf8Ab7/6JFdNXM6v/wAlN8Hf9vv/AKJFRPb7vzOjDfG/SX/pLO5ooopmIUUUUAFFFFABVcX9mdQNgLqA3gj802/mDzAmcbtvXGeM1YrzbXdKu774mahe6OwTVdN062uLXJwJDvkDRN/suuR9cHtSvqv66B0f9dT0RrqBLqO2eeNZ5VLRxFwGcDGSB1IGRn61LXE2+rwa74z8MajahlWawvN0bfejYGIMjehBBB+lVdd8TS6YzXumeIbrUmivESW2jsVezCtIFMZmSP5WUMOsmcjkc4p9l3/zsHmegUVybzazqnjjVtMt9YksLK1tbeRPIgiaQSPv7urDb8vIIzwMEc5wk17xIngFfFV1qqmW3lCPZR28YhmRZfLYsSC4ZuWyrADgY65FqD0/r5npNFcTqHiSe88U6hpsV9qWn22npGpfTtLe6klldd3zN5UiqoBHGASc84Fb3hbUL7UvD0E+rQSQ3YZ438yBoTIFYgPsbldwAbB9aFqrgbFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/AGx/9HJW7WF8Vv8AkmWq/wDbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuZ1f8A5Kb4O/7ff/RIrpq5nV/+Sm+Dv+33/wBEiont935nRhvjfpL/ANJZ3NFFFMxCiiigAooooAK4Px94f0ZpdGuW0ixM9zrVuk8ptk3SqSchjjLA9wa7ysTxNo1xrKaWLV4k+x6lDdyeYSMohOQMA880faXqvzDo/R/kV3uI/DfiHSdKsbCztNJ1ASoPs8Pl7LgAMo4wuGUN2zkdazNR8Y6pDbX93p9rbTwRavDp1qjhgZcsqyEtnH3iVBxxtOQa0PiAxh8JyXUDhL21nilsuMlpw42IB33Z2/QmsjxLpEuleAtF0y3nVbmPUbNTOybgZTMCzkZGcsScZH1oW6v3S+9r/hvuB7fJ/k/+B+JsNq+r6T4g0+z1trK4tdSLxxS2sLxGCRVL7W3O24FQeRt5HTmuel+JgbTJ9Zg1TQltonJTS5Js3c0Stjdu3ja5GSE2HsCeeOiXSdW1XXrC+15LK3t9ODvDb2szTebKyld7MyJtAUnCgHljzxUWi6T4g8O2I0ixTTbmwhkYW1zPcSJLFExzhowhDlckD51yAOlC/r+v608w/r+v6/A6eNxJGrr91gCPpTqKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4/4rf8AJMtV/wC2P/o5K3awvit/yTLVf+2P/o5K3aUfify/U3n/AAI+svyiFFFFWcxj237jxlqEfRbi0gmX3ZWdW/TZWxWPf/uPFmkT9BNFPan3JCyD9Im/M1sUAFFFFABRRRQAUUUUAFFFFABRRRQBzdp/yWg/9i//AO3FdvXEWn/JaD/2L/8A7cV29Zx6nTW+z6IKKKKowCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArjfiN/qfDX/YwWv8A7PXZVxvxG/1Phr/sYLX/ANnqZbHRh/4qN+iiitDlCiiigAooooAKKKKACiiigAooooAKxtc/0i+0ixHIlvBNIP8AZiBcH/vsR/nWzWNH/pXjSVuq2NkqD/elfLD8BEn/AH1QBs0UUUAFczq//JTfB3/b7/6JFdNXM6v/AMlN8Hf9vv8A6JFRPb7vzOjDfG/SX/pLO5ooopmIUUUUAFFFFABWTBofk+LbvW/tG77Tax23k7Pu7GY7t2ec7umO1a1FHW4dLHMxeDIbfxyfENtdtHG8UgksgnymR9oaQHPBIVcjHJGaoSeBL9/D0WgrrypplvKrwKtn+9IWQOqSOXww7cKpyAfUHtaKFpawbmZaaN9l8S6jq3n7vtsMMXlbMbPL385zznf6DGKypPBm/wAAy+Gvt+PMZm+0+T0zL5n3d34dfeuoooDrcwJ/D15Br0+raFqMNnLdxIl3Fc2pnjlKcI4AdCrAZHUgjHHFbNpHPFaRpdzi4nA+eUIEDH2XnA7AZJx1JPNTUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/bH/ANHJW7WF8Vv+SZar/wBsf/RyVu0o/E/l+pvP+BH1l+UQoooqzmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5nV/8Akpvg7/t9/wDRIrpq5nV/+Sm+Dv8At9/9Eiont935nRhvjfpL/wBJZ3NFFFMxCiiigAooooAKKKKAK02nWVxew3k9nby3VuCIZ3iUvHnrtYjI/CpJ7aC6VVuoI5lR1kUSIGCspyGGe4PINS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHH/Fb/AJJlqv8A2x/9HJW7WF8Vv+SZar/2x/8ARyVu0o/E/l+pvP8AgR9ZflEKKKKs5jG8R/uodPu/+fbUIDn0Dt5RP5SGtmsrxNC8/hbUliGZVt3eP/fUbl/UCtGCZLi2jnjOUkQOp9iMigCSiiigAooooAKKKKACiiigAooooA5J9RsdM+MPnaleW9nE2g7BJcSrGpb7RnGSeuAfyrqP+Eu8N/8AQw6V/wCBsf8A8VVa90XS9RmE2oabZ3cqrtDzwK7AdcZI6cn86r/8Ir4e/wCgFpn/AIBx/wCFZ8sk9DqdSlJLmvdK3Q0f+Eu8N/8AQw6V/wCBsf8A8VR/wl3hv/oYdK/8DY//AIqs7/hFfD3/AEAtM/8AAOP/AAo/4RXw9/0AtM/8A4/8KdpCvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVR/wAJd4b/AOhh0r/wNj/+KrO/4RXw9/0AtM/8A4/8KP8AhFfD3/QC0z/wDj/wotIL0PP8DR/4S7w3/wBDDpX/AIGx/wDxVH/CXeG/+hh0r/wNj/8Aiqzv+EV8Pf8AQC0z/wAA4/8ACj/hFfD3/QC0z/wDj/wotIL0PP8AA0f+Eu8N/wDQw6V/4Gx//FUf8Jd4b/6GHSv/AANj/wDiqzv+EV8Pf9ALTP8AwDj/AMKP+EV8Pf8AQC0z/wAA4/8ACi0gvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVR/wAJd4b/AOhh0r/wNj/+KrO/4RXw9/0AtM/8A4/8KP8AhFfD3/QC0z/wDj/wotIL0PP8DR/4S7w3/wBDDpX/AIGx/wDxVH/CXeG/+hh0r/wNj/8Aiqzv+EV8Pf8AQC0z/wAA4/8ACj/hFfD3/QC0z/wDj/wotIL0PP8AA0f+Eu8N/wDQw6V/4Gx//FUf8Jd4b/6GHSv/AANj/wDiqzv+EV8Pf9ALTP8AwDj/AMKP+EV8Pf8AQC0z/wAA4/8ACi0gvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVR/wAJd4b/AOhh0r/wNj/+KrO/4RXw9/0AtM/8A4/8KP8AhFfD3/QC0z/wDj/wotIL0PP8DR/4S7w3/wBDDpX/AIGx/wDxVH/CXeG/+hh0r/wNj/8Aiqzv+EV8Pf8AQC0z/wAA4/8ACj/hFfD3/QC0z/wDj/wotIL0PP8AA0f+Eu8N/wDQw6V/4Gx//FUf8Jd4b/6GHSv/AANj/wDiqzv+EV8Pf9ALTP8AwDj/AMKP+EV8Pf8AQC0z/wAA4/8ACi0gvQ8/wNH/AIS7w3/0MOlf+Bsf/wAVR/wl3hv/AKGHSv8AwNj/APiqzv8AhFfD3/QC0z/wDj/wo/4RXw9/0AtM/wDAOP8AwotIL0PP8DR/4S7w3/0MOlf+Bsf/AMVXLeONd0jUz4ch03VLK8lXXrVzHb3CSMF+YZwD0yR+dbH/AAivh7/oBaZ/4Bx/4U+Lw3ocEyTQaLp8csbBkdLVAykcgggcGk4yasXCpRhLmV/wNOiiitDjCiiigAooooAKKKKACiiigAooooAKx/D379dRvz/y930hX/djxEv4ER5/Gruq3o03R7y9Iz9ngeXHrhScUzRrI6bodlZscvBAiOf7zAcn8Tk0AXqKKKACuZ1f/kpvg7/t9/8ARIrpq5nV/wDkpvg7/t9/9Eiont935nRhvjfpL/0lnc0UUUzEKKKKACiiigAqve6hZ6Zam51K7gtIAQDLPII1BPQZJxViuT0yJdU+I2s3V8vmNpKw29kjjiEOm53Udi2cZ64XFHUDotP1Sw1aBptLvra9iVtjSW0yyKG64ypPPIq1XKeIol0/xj4e1KzHl3F3ctZXIQAefEY2YbvXaVBHpz61FY67qM3wy1PVpLjN9Al6Y5di/KY3kCcYxwFHbnHNJtKNxpO6Xf8A4P8AkdhRXLX2qD+zdKa51+4sLi4tlkMFjbJNcXDFQSQhjclRznanfqAKxf8AhKdam8AtdW10q6hDqyWC3E1ts8xfPVNzxkAqSDyAARzjHara8v8AW9ib6X/ra56HVK51jT7TTLrUJrqP7LaB/PkjO/y9n3gQuTkY5HWsGSbVtL8Wafpk+rT3ttqsE43yRRK9tIgDAptQDBBPDBug5655/SW1DRfh/wCJtTtdXunmt574xLLHCVV1kb95xGCWOOmdvPSp/wAr/jYpLb1seiLe2ztAgnjD3CGSFC2GkUYyQDycZGfTIqeuG1Szurv4h6BIur3luZbC4YeUkJ8vHlZA3Rnhs85z7YpW8SarY+G9dt7iTztZsr02lo5jUGXzSPIbAGOjjPH8JzVP/P8AOxK1X9drncUVHbRyRWsUc8xnlVAHlIALnHJwMAZ9qkpDCiiigAooooAKKKKACiiigAooooAKKKKAOP8Ait/yTLVf+2P/AKOSt2sL4rf8ky1X/tj/AOjkrdpR+J/L9Tef8CPrL8ohRRRVnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/8AJTfB3/b7/wCiRXTVzOr/APJTfB3/AG+/+iRUT2+78zow3xv0l/6SzuaKKKZiFFFFABRRRQAV5pr0GoR/FG+1fRQ0l5pmm28ptgf+PmEvJ5kf1IAI91Fel1iW2i3EPji+1lniNvcWUNuigneGRnJJGMY+Yd6X2k/62YdH/XVGSdRttW8c+GNQsZPMt7nTrqSNsdj5XX0PtTdf8V3vh28VrvU9ElzcIp0tQVuvKd9oZWMnzEZDEeWBwRnjNPtPBt1p/j1dUs7mJdIEUzLaknfDNKV37OMbSV3YzwSeKzj4M19PCf8Awj1v/ZUUK3KzSXvmOZLvbKHy67BtcgDLbn5XHfIa6f11B9f66G1Nq2u3ni7UtH0r+z4IbO2hmFxcwvISz7vkKh167c5zxjoc8Y8fjPxAfCKeJ57bT4bOKVY57MK7SOBJ5busm4BfmyQpVuB97njp7LSJ7bxdququ8ZgvILeKNQTuBj35yMYx8wxz61iS+D79/hjN4cE1t9skdmDlm8vmfzOu3PT260LT+vMHq/67f5jtV8ZpH4jutJttY0bSvsUaGWfVHyZHYZCIm9DgDBLZPUDHWtnwtro8R+HoNR2xq7M8cgifehZGKkq3dSRkH0Iqk2j6tpfiS91PQ0srqLUUj+0211O8OyRBtDqyo+QV4IIHQHNb9p9pNpGb8RC4Iy6wklFPoCeTjpnAz1wOgFsBNRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcf8Vv+SZar/2x/wDRyVu1hfFb/kmWq/8AbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5hGUMpVhkEYIPesjwox/4RaxiY5a2Q2rE+sTGM/qlbFY2gfurjV7Tp5F+7KPaRVlz+bt+RoA2aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAx/En762s7Adb29ijI9VU+Y4/FY2H41sVjT/6V4ytY+q2Vo8zD0aRgiH8klrZoAKKKKACuZ1f/AJKb4O/7ff8A0SK6auZ1f/kpvg7/ALff/RIqJ7fd+Z0Yb436S/8ASWdzRRRTMQooooAKKKKACsXUNAlk1pdY0e9+wah5Qhl3xebDcIDkB0ypJGTghgRnnI4raooA4t9M8R23jey1PU4otdtVhaKL7Gi232FmwGfZJId+4ADO4kc4HrJc+Cr+TR9S0W01xbfSr55XEf2PdND5hLFRJvAK7iTgrnBIyOCOwopW0sHW5zbeGL221G2v9H1OG3uY7BbGX7TamZJFU5VgBIpU5z3I56cVXh8ENF4ffTH1WSdn1QaibiSEbmIlEm0gEDJxjIx16dq6yiqu73/re/5itpb+trfkZl9o323xBpWp+fs/s/zv3WzPmeYoXrnjGPQ1ky+D7htJ1vSYtSjSw1QzOim1JkgeU5Y7t+GXJJA2g89eK6mikVcxNR0Ce4vtMv8AT72O2vNPR4g00BlSSNwAwKh1IOVUg5/OsMWtpr/xOjvdOuPPtNPgDXjQsGie5Ussa5H8Sq7kjPHy5rt6KOt/6/r9SbaWCiiigYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/5Jlqv/bH/ANHJW7WF8Vv+SZar/wBsf/RyVu0o/E/l+pvP+BH1l+UQoooqzmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5nV/8Akpvg7/t9/wDRIrpq5nV/+Sm+Dv8At9/9Eiont935nRhvjfpL/wBJZ3NFFFMxCiiigAooooAKKKKACiiigAooooAKKKihu7e4lmignilkgbZKiOCY2xnDAdDgg80AS0UUjMFUsxAAGSSelAC0VHbXMF5bR3FpNHPBIu5JYnDKw9QRwRUlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHH/ABW/5Jlqv/bH/wBHJW7WF8Vv+SZar/2x/wDRyVu0o/E/l+pvP+BH1l+UQoooqzmCsa2/ceMr+P8AhubOGZfdlZ1b9ClbNY1/+48V6RP0E0c9qfckLIP/AEU360AbNFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVHPMltbyTzHEcSF2PoAMmgDL0b/AEjVNYvuoa5FvGf9iJQD/wCPmStisrw1C8Hhuy88YmmTz5R6SSEu3/jzGtWgAooooAK5nV/+Sm+Dv+33/wBEiumrmdX/AOSm+Dv+33/0SKie33fmdGG+N+kv/SWdzRRRTMQooooAKKKKACoby9tdPtXub+5htbePG+WaQIi5OBkngcmpq5PT0Gr/ABF1eTUAJBo6wxWUT8iMum9pQP7x+7u64BHrR1sBv6drWl6v5n9k6lZ33lY8z7NOsmzPTO0nGcH8qu1ieK9V/wCEe8P3eq21tFLebUhi3jG5mfagYjnaC+cfX1rM1KfV/DDabe3WtS6lBPdRWt3BPBEijzCFDx7EDDDY4YtwT35o3YHXVElzBJJNHHPG7wECVVcExkjIDDtwQee1cfBrdzP4suLDVtcm0e6W7Is7FreNYbuAFcFXdCXZskHa4IPbjmPw3YXMXinxU76teTLFcoHjdIQsubdcFsRg8ZwNpHQZzzlX0v5X/IfWx0tr4o0C+uktbHXNNubiQ4SKG7jd278AHJrUryyym/t34R6doGnafe3F/NbxpHI9nLHDAwOfN85lCYXGcqSSeBmumM+s33jy70iLVZLSxtrCCZmhijMhkZnBwXVhg7ecg9BjHNU1rYnpc6K01Szvrq8trWbzJbKQRXC7SNjFQwGSOeCOmat1xk/im+0+z8V3LlbltPvUt7ONwFUF0iCgkDJG98nvWk4v/Dul3Wr6vrVzqK21rJLNb+REkZYDd8m1Qw6EAMzdefWpvZX8v0uVZt2Xc6GivPbPxFrLRaXepc6pfXFzNF9qsRossdskbkBjHIYgw2A53M5BwfXj0Kqs0TcKKKKQwooooAKKKKACiiigAooooAKKKKAOP+K3/JMtV/7Y/wDo5K3awvit/wAky1X/ALY/+jkrdpR+J/L9Tef8CPrL8ohRRRVnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVzOr/wDJTfB3/b7/AOiRXTVzOr/8lN8Hf9vv/okVE9vu/M6MN8b9Jf8ApLO5ooopmIUUUUAFFFFABXFnRtM1f4n6sNW060vhHp1qUFzAsmzLy5xuBx0rtK5qfTdbtPGF5q+mW1hdQ3VpDBsuLx4WQoznPETgg7/0o6r+ugdH/XVGatzb+C/FlxZxO8eiy6ZLf/ZslltnjYbvLH8KsG+6OARwBmq1t8QnI067n1PQZkvp44n0y0mD3NuJDhTvDkOVJG4bF788c7sfhd9QbUbrxDMk11qFqbPy4MiO2hOcohPLHJyXIGcDgYxTtFt/Eun2tnp14mmTQWwERvVnk8ySNeAfJ2YDEAA/OQCSeelEfP8ArV/pZfkD8v60X63/AFKun+INR1jxBfWdnd6ZamwuzDJYTxs9xJEpXMuQ42ghvl+Rh0554mi1XW9bvtQGhPYWlpYztbCW7geZriVfvYCumxQeM/MTzwO9fWfD2qeINRs2vLTS7YWd4s0OowzO9wsavuCBTGNpYAA/OR1ODwKmt9J1zQrzURoa2F3aX1w10q3k7xNbyv8AfHyo29SeQPlIyRk9aI9L/wBbf8Eb30/rf/gBNrGt3PjS50PThZQw29nFctczwvIQWZgV2h1znbwcjGD1zVG78RTaZpnjK/tLKyS40yfCMsRHnkRI2ZMHLH5sZ44ArctdIuIfGN/q0jxGC5tIIFVSdwZGckkYxj5xjmub8VaRcad4N8bXMzxlNQJniCEkqojRPmyOuVPTNLo/R/n/AJAtZL1X5f5mnd63rukf2ffaoli1heTxwS28MTiW1MhAU+YWIcBiAflXrx0qh4eXVo/E3iuS5vLKaKO4XzoxZuDIfs67cEynaMYBBDZ56Z4vy6Prms/2ba6w1jHYWc0dxJLbyO0l20ZBQFGUCME4Y/M/QD3qxBo2oWfiDV5oBbS2Oq7ZGd5mWSFxHswFCEMDtU53AjJ4NEtnbzt+H/BFB7X8v1/4Bm6d4kuG8P8Ahiz0qzsre/1i38xEWMrb2qKgZ22A5IGQAuRknqMVoW+t6lZ65caPrP2SaY2bXlrc20bRLIqkKyMjMxBBIOd2CD2xVO38J6hYaN4cezltTquiQeSVkZvJnRlCum7G5egIbBwR05q9Bol/ea1PrGsfZorj7G1nbW1vI0iRqxyzFyqliSBxtAAHfNOe7t5/rb9P+GCGyv5f8H8P6uY0PivxFH4N0/xRfxad9klELXFnDE/mBHYLvWTfgfeDbSp443d67uuTl8K3snwwt/Dglt/tkVvDEzlm8vKMpPOM4+U9q6yqla7t3JjeyuFFFFSUFFFFABRRRQAUUUUAFFFFAHH/ABW/5Jlqv/bH/wBHJW7WF8Vv+SZar/2x/wDRyVu0o/E/l+pvP+BH1l+UQoooqzmCsbxJ+6t7C7/59r+Bs+gdvKJ/KQ1s1l+JoHuPC+pJF/rRbO8f++o3L+oFAGpRUVvOlzaxTx/clQOv0IzUtABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVj+KSX0GS0X719JHacekjhW/JSx/CtisfUf9J8TaRa9VgEt431VfLUH/AL+k/wDAfagDXAwMDgUtFFABRRRQAVzOr/8AJTfB3/b7/wCiRXTVzOr/APJTfB3/AG+/+iRUT2+78zow3xv0l/6SzuaKKKZiFFFFABRRRQAViaj4fll1ldY0e++wah5YhlLxebDcIDkB0ypJGTghgRnuOK26KAMKbQb7VtPvbHxHqMF1b3Maoi2VqbcwsCTvBLuS2dpHYbRx1qIeHdRvZrL/AISDVor63sZVnjjgtDAZZF+60p3sGwecKFGcHoMV0VFAHOaj4c1HWZoodV1S3k0+G8W6SKKyKTfK+5FMnmEYHAJCAkDqM5qzBoVxaeIL++tLyNbXUdrXFvJAWfeqbAyuHGBgLkFT06jNbVFK2lg63M7w9pP9g+HbLS/O8/7JEI/N2bd+O+MnH50230byPFF7rHn7vtVtFB5OzG3YXOc55zv6Y7Vp0VV7u4dLHPSeELW6t9et76ZpYNZnEzBF2NCQiKMHJyQUDA469qnttJ1SWCS013VLfULKS3aB40svKeXcANztvIJxn7qqOfoK2qKXSwdbmBpGi6zpUNrZf23DPp9rhUDWWLho1+6jSb9p4wCdgJA7Hmt+iine4BRRRSAKKKKACiiigAooooAKKKKACiiigDj/AIrf8ky1X/tj/wCjkrdrC+K3/JMtV/7Y/wDo5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczq//ACU3wd/2+/8AokV01czq/wDyU3wd/wBvv/okVE9vu/M6MN8b9Jf+ks7miiimYhRRRQAUUUUAFFFFABRRRQAUUUjMFUsxAAGSSelAC1HcW0F5bvb3cMc8Mg2vFKgZWHoQeDRb3EN3bxz2ssc0MihkkjYMrg9CCOCKkoAAMDA6UVHcXENpbyXF1NHBDGpZ5JGCqgHUkngCnqyugdGDKwyCDkEUALRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/wAVv+SZar/2x/8ARyVu1hfFb/kmWq/9sf8A0clbtKPxP5fqbz/gR9ZflEKKKKs5gpGUMpVhkEYIPelooAx/CjH/AIRaxiY5a2Q2rE+sTGM/qtbFY2g/urrWLTtDfsyj2kRZM/8AfTt+VbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVjaf8A6T4o1a56rbrFZp7EL5jfn5qj/gNbPSsfwv8AvNCS8PW+lku8+qyOWT8kKj8KANiiiigAooooAK5nV/8Akpvg7/t9/wDRIrpq5nV/+Sm+Dv8At9/9Eiont935nRhvjfpL/wBJZ3NFFFMxCiiigAooooAKri/szqBsBdQG8Efmm38weYEzjdt64zxmrFeZ+I45bT4jX3iC1BMuj2VrNIo5L25aUTL9dvzD3UUdUHRs9GvL210+1e5v7mG1t48b5ZpAiLk4GSeByahbVrFNTttPM4NzdRNNCgUkOi4ydwGP4h371wXxKuxr2kXljZSiSysdOfUbp15V2KnyF/m//AVrorbUrqLxNoOnJLi0n0qSWSPaPmdTEFOcZ4DHv3oWrt/Wz/yB6LT+tv8AM6eorm5gs7aS4u5o4IIxueWVwqqPUk8CuEtr7xJeeEtW1sa2yTWM92ba3S2i8uRIpGwsmVychcZUrxjqeaswTx+JvHWmC/iBt7XSY9SggblfOkbG8juVAwD2JJoWv9eTf6A9P680v1Oq07WdL1cSHSdStL4R43/Zp1k2Z6Z2k46Grtcj44jSyuNF1q1UJfw6jBbCReGkikfa8Z9Qc5x2IzS6VdavqfijXVuNUkg0/TLxEhhhijy48pGZWJUnbz2w3J54FGj/AK9P8wen9ev+R095dwWFjPeXb+XBbxtLK+CdqqMk4HJ4HanwTR3NvHPC26OVA6NjGQRkGvPr+bWvEHw11TX/AO1Wto7qznki08QRmFYNrAKxK7y5UZ3BgMkcYGDPL4imjudI0S3ubyyiTS47q4uLGwe6mOQFRFARwo4Yksp7Ad6OrT8v1/yB/wCf6f5ne1FNdQW7xJPPHE0z7IldwDI2Cdoz1OATgelcHd+I/EEPhK6kgaZbuHVIbW1ur2yMBuYndAC0bKuPvFSQB0yMVoa9az2l/wCF0ur+a/kOsBjLMkakfuZOAEVRj65PuaF09Uvvt/mHf0f6/wCR2NV9Qv7bS9Pnvr6Tyra3QySvtLbVHU4GSfwrH8Wale6Mmm6hby7bKO9SO/TYCDC/ybs4yNrFTxjvWD4t1C81DS/GSJKRpthp32dUCDDzlS7tuxn5VKDGccmlfS40tUjvUdZI1dDlWGQfUUtQ2f8Ax4wf9c1/lU1U1Z2Ji7pMKKKKQwooooAKKKKACiiigDj/AIrf8ky1X/tj/wCjkrdrC+K3/JMtV/7Y/wDo5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczq//ACU3wd/2+/8AokV01czq/wDyU3wd/wBvv/okVE9vu/M6MN8b9Jf+ks7miiimYhRRRQAUUUUAFeaa+8umfFS48QQlvL06ztVu1B4NvI8iu2P9khW/4Ca9LrCGgNL4o1W9vBDLY39jFamIkljtMm7IxjBDjv60a3TQdGv63RzHxVun1HQ7nR7RgYo7KTULxlPSNB+6X/gT4P0Q1o3/AItTTJ9M0aLUNM06V7FbiW61J8JGuAqqqbl3sxB/iGAM85AqK18C3lt4J1vS3vI7nUNRia3juJGO1YlTZCpIGeFGTweSetaM+g6lZ6zY6xoxtJbmOyFjdW1zI0aSoPmVldVYhg2eqnIJ6YoWnpp+T/4CB6/j+n/BE0PxJda9ouoGzudNa7sbkwNeIS9q6gK3mABs42t93d1BG7vVbR/FT6rqGq6TJf6ZqghsxcR3mmHCEHKlGXe+GBGfvcgjgVa13RNV1/R7UXcWn/aLe9S5Ni0jPbzIvHlu5TJ6ls7MAgDBxmm2eiazL4nvtX1RrGNbnThaR29u7P5RDseWKjcDnOcDrjHGTMldNf1t/mVHRp/1v/kctL/Z/wDwgPgD+2vsv2HzofO+17fKx5D/AHt3GM461r6O1jH4wup/BqwDRY7Bvtf2MAWrXGQU27flL7c7ivbGecVftvCt9DovhOzaW3MmiyxvcEM2HCxMh2cc8sOuK6i4jM1rLGuAXQqM+4q5v4mu7/ImH2U+y/M4hNf1m++GN14h1W30x4pdNaZbBrZnVjjq5L4Knn5cdCPmNbF1rF8+qWWiaHHaxXD2gup550LR28WdoARSpYk5AG4AAE89KhPhm8/4Vb/wjXmwfbP7O+y+ZuPl79uM5xnH4VNeaJqNvrlprWjNbSXKWgs7m2uXZI5UB3KVdVYqwbP8JyDjjrTlbmfb/h/1sJX5V3/4b/glfXPE1z4ei06x1G90qPUb6SQfa58wW0Ua8lyrPknBUbd/JPUAVL4X8Uf2zqWoac97p+oPZrHIt5px/dSI+eMbm2sCpyNx4IPtRqOj6xfTaVrEf2GHWNPkk/cea5glifhkMm3cDtCndt6jpg1s6c+qSLI2rQWlucgRx20zTcdyXZU6+m3jHU54leY35F2iiigAooooAKKKKACiiigAooooAKKKKAOP+K3/ACTLVf8Atj/6OSt2sL4rf8ky1X/tj/6OSt2lH4n8v1N5/wACPrL8ohRRRVnMFFFFAGNb/uPGd6nRbmzhlX3ZGdW/QpWzWPqH7jxVo8/aZJ7Q+5ZRIP8A0Sf1rYoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMvxLO9v4bvTCcTSR+TEf+mkhCJ/48wq/bwJa2sVvCMRxIEUegAwKzNa/wBI1PR7Ech7o3Eg/wBiJSwP/fZjrYoAKKKKACiiigArmdX/AOSm+Dv+33/0SK6auZ1f/kpvg7/t9/8ARIqJ7fd+Z0Yb436S/wDSWdzRRRTMQooooAKKKKACsqLQ1TxLfarJKJFvLWK2aAx8AIXOc55zv6Y7Vq0UAcla+Abey8H6toUF4+7Ug6G5dNzRoV2xrgnkIgCjkdO2a1U0Dbrum6l9pz9hsntPL8v7+4od2c8fc6YPWtiijrf+u36h/X9fcYVn4a+yeF7/AEf7Xv8AtjXLed5eNnnMzdM843evOO1Mk8LbbfS5LG9NrqemW628d2IgyypgBkkTPzIdoOMgg8giugooD+vvMFPD93eapbXviHUIr37G2+2t7e2MEKSYI8xgXcswBwOQB6Z5q1pWjf2ZqGrXPn+b/aNyLjbsx5eI1TGc8/dznjrWpRQBx03gi/Ph678PWmu+RpE6yLEn2TdPCrc7BJvwUyTxtztONw61fm8Lzx3en3+k6glrf2doLOR5bfzYriL0ZAykEMMghuORzmuioo/r+vvDcwtQ0G81bRo7TUtSjedbuK5MsVtsQCORWCKm4kA7epYnJJ6YAk8Q6Jcau2nS2V5FaT2F0LlGmgMysdjLgqHU/wAXr2rZoo/4f+vuAx5tIvdU0S/03Xru0uUu4miDWto0OwEEZIaR8noR06VSj8IbfAt34fe/aSa8jlE960fLySEkvtz79M9AOa6WijuHbyGQx+VBHHnOxQucdcCn0UUbiSsrBRRRQMKKKKACiiigAooooA4/4rf8ky1X/tj/AOjkrdrC+K3/ACTLVf8Atj/6OSt2lH4n8v1N5/wI+svyiFFFFWcwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXM6v/AMlN8Hf9vv8A6JFdNXM6v/yU3wd/2+/+iRUT2+78zow3xv0l/wCks7miiimYhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcf8Vv+SZar/2x/wDRyVu1hfFb/kmWq/8AbH/0clbtKPxP5fqbz/gR9ZflEKKKKs5gooooAxvEn7q1srv/AJ9b+Bs+gZ/LY/8AfMhrZrL8SwPc+F9Sii/1v2Z2j9nAyv6gVftp0urSG4j+5Miuv0IzQBLRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjQ/6V4yuZOq2NmkKn0aRizj8kj/Otmsbw5++t76/PW8vZXB9VQ+Up+hWMH8a2aACiiigAooooAK5nV/+Sm+Dv+33/wBEiumrmdX/AOSm+Dv+33/0SKie33fmdGG+N+kv/SWdzRRRTMQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOP+K3/JMtV/7Y/+jkrdrC+K3/JMtV/7Y/8Ao5K3aUfify/U3n/Aj6y/KIUUUVZzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczq//JTfB3/b7/6JFdNXM6v/AMlN8Hf9vv8A6JFRPb7vzOjDfG/SX/pLO5ooopmIUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHH/Fb/AJJlqv8A2x/9HJW7WF8Vv+SZar/2x/8ARyVu0o/E/l+pvP8AgR9ZflEKKKKs5gooooAQgMpBGQRgg1keFCR4XsoWOWtVa1OfWJjGf/QK2KxtB/dXWsWn/PG/ZlHtIiyZ/wC+nb8qANmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACqWsXp03RL29Ay0EDyKP7zAcD8TgVdrG8Rfv49PsB/y+XsSsPVUzKw+hEZH40AXdJsv7N0azsgc/Z4EjJ9SFAJq5RRQAUUUUAFFFFABXM6v/yU3wd/2+/+iRXTVzOr/wDJTfB3/b7/AOiRUT2+78zow3xv0l/6SzuaKKKZiFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBx/xW/wCSZar/ANsf/RyVu1hfFb/kmWq/9sf/AEclbtKPxP5fqbz/AIEfWX5RCiiirOYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArmdX/5Kb4O/wC33/0SK6auW8T2+rx+JtA1fRtL/tL+z/tPmRfaEh/1iBRy34noelRPY6MNb2mvZ/kzvKK4v/hLPF3/AEIv/lXi/wAKP+Es8Xf9CL/5V4v8KXMv6TH9Xn3X/gUf8ztKK4v/AISzxd/0Iv8A5V4v8KP+Es8Xf9CL/wCVeL/CjmX9Jh9Xn3X/AIFH/M7SiuL/AOEs8Xf9CL/5V4v8KP8AhLPF3/Qi/wDlXi/wo5l/SYfV591/4FH/ADO0ori/+Es8Xf8AQi/+VeL/AAo/4Szxd/0Iv/lXi/wo5l/SYfV591/4FH/M7SiuL/4Szxd/0Iv/AJV4v8KP+Es8Xf8AQi/+VeL/AAo5l/SYfV591/4FH/M7SiuL/wCEs8Xf9CL/AOVeL/Cj/hLPF3/Qi/8AlXi/wo5l/SYfV591/wCBR/zO0ori/wDhLPF3/Qi/+VeL/Cj/AISzxd/0Iv8A5V4v8KOZf0mH1efdf+BR/wAztKK4v/hLPF3/AEIv/lXi/wAKP+Es8Xf9CL/5V4v8KOZf0mH1efdf+BR/zO0ori/+Es8Xf9CL/wCVeL/Cj/hLPF3/AEIv/lXi/wAKOZf0mH1efdf+BR/zO0ori/8AhLPF3/Qi/wDlXi/wo/4Szxd/0Iv/AJV4v8KOZf0mH1efdf8AgUf8ztKK4v8A4Szxd/0Iv/lXi/wo/wCEs8Xf9CL/AOVeL/CjmX9Jh9Xn3X/gUf8AM7SiuL/4Szxd/wBCL/5V4v8ACj/hLPF3/Qi/+VeL/CjmX9Jh9Xn3X/gUf8ztKK4v/hLPF3/Qi/8AlXi/wo/4Szxd/wBCL/5V4v8ACjmX9Jh9Xn3X/gUf8ztKK4v/AISzxd/0Iv8A5V4v8KP+Es8Xf9CL/wCVeL/CjmX9Jh9Xn3X/AIFH/M7SiuL/AOEs8Xf9CL/5V4v8KP8AhLPF3/Qi/wDlXi/wo5l/SYfV591/4FH/ADJPit/yTLVf+2P/AKOSt2uG8V33i7xP4Zu9I/4Q77L9o2fvf7UifbtdW6cZ+7jr3ruaI6ybHVjyUoxbV7vZp9F2CiiitDlCiiigArGg/ceNLxOi3VlFKB6sjurH8mT8q2axtR/ceKdGuOglWe0PvuUSD/0SaANmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACseX/AErxnAnVbGyaRh/tSsFU/lHJ+dbFY2if6RqGr33US3fkRn/ZiUIR/wB9+ZQBs0UUUAFFFFABRRRQAVzfiTSdZutd0bVfD72An03z8rel9reYoXooyeM9x2rpKKTV1YunUdOXMv6voc39r+Ivp4X/ACuKPtfxF9PC/wCVxXSUVPJ5mvt/7q+45v7X8RfTwv8AlcUfa/iL6eF/yuK6Sijk8w9v/dX3HN/a/iL6eF/yuKPtfxF9PC/5XFdJRRyeYe3/ALq+45v7X8RfTwv+VxR9r+Ivp4X/ACuK6Sijk8w9v/dX3HN/a/iL6eF/yuKPtfxF9PC/5XFdJRRyeYe3/ur7jm/tfxF9PC/5XFH2v4i+nhf8riukoo5PMPb/AN1fcc39r+Ivp4X/ACuKPtfxF9PC/wCVxXSUUcnmHt/7q+45v7X8RfTwv+VxR9r+Ivp4X/K4rpKKOTzD2/8AdX3HN/a/iL6eF/yuKPtfxF9PC/5XFdJRRyeYe3/ur7jm/tfxF9PC/wCVxR9r+Ivp4X/K4rpKKOTzD2/91fcc39r+Ivp4X/K4o+1/EX08L/lcV0lFHJ5h7f8Aur7jm/tfxF9PC/5XFH2v4i+nhf8AK4rpKKOTzD2/91fcc39r+Ivp4X/K4o+1/EX08L/lcV0lFHJ5h7f+6vuOb+1/EX08L/lcUfa/iL6eF/yuK6Sijk8w9v8A3V9xzf2v4i+nhf8AK4o+1/EX08L/AJXFdJRRyeYe3/ur7jifEGnePPEmhXGlXzeHY4LjbvaEzhhtYMMEgjqo7V21FFNRSIqVXNKNkku3n/wwUUUVRkFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFY3iT93Z2d33tb+B8+gZxGx/75c1s1meJLdrrwxqUMX+sa2kMfs4Ulf1AoA06KhtbhbuzhuY/uTRrIv0IzU1ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQ3VzHZ2c1zMcRwxtI59ABk/wAqpeHLeS18OWKTjE7RCWYf9NH+d/8Ax5jUXij97opsh1v5o7XHqrsA/wD45uP4VsUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFIQCCCMg9QaWigDH8KEjwvZwE5NqGtTn1iYx/wDslbFY2hfurzWbTtDfs6j2kRJM/wDfTN+VbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAY1/wD6T4q0q26rbJNeN7MAI1/MSOf+A1s1j6b/AKT4k1e76iIxWaH/AHV8w4/GXH/AfatigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAxoP3HjS8T+G6sopF/3kd1Y/kyVs1j6j+48UaNcdPNE9offcgkH/omtigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKQkKCScAckntS1leJpXi8N3iwnbLOgtoiOzykRqfzYUAM8LAv4fhumHzXzPeHPXErF1H4KwH4VsUyGJIII4YhtSNQqj0AGBT6ACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDG8S/u7K0u+9rfQPn0VnCMf++XatmszxHbNd+GdSgj/1jW0nln0cKSp/PFXbO5W8sYLmP7k0ayL9CM/1oAmooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKxtY/wBI1jR7IcgztdSD/YiXj/x9462ax7f/AErxheS9VsrWO3X2dyXf9BFQBsUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFACYzwayPCh2+GLSA9bTfaH/tk5j/9krYrG0L91e6zadBFfF1H+zIiPn/vpm/KgDZooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKx/Df76zur49b68llB9VB8tD/wB8Ipqzrd62naDe3cYzJDA7Rj+8+PlH4nAqXTLJdN0m0sUOVtoUiB9dqgf0oAtUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABWPB+48aXafw3VjFIB/tI7qx/J0/KtisbUv3HifRbj/AJ6ie0P/AAJBJ/7RoA2aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMfxB+/Om2H/P1ex7h/sx5lP4fuwPxrYrGb/SvGiDqthYlj/vTPgfiBE3/fVbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFY3iX93Y2t2OtrfQSZ9FMgRj/3y7Vs1m+IrZrzwzqVvH/rJLaQIfRtp2n88UAaVFQWVyt7YW91H92eJZF+hGf61PQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRUF7dR2On3F3NxHbxNK/0UZP8qAM7Qv8ASLrVr/qJ7xokP+zEBHj/AL7Vz+NbFZvh61ksvDtjDP8A64Qq03vI3zOf++ia0qACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKOtFFAGP4U+Xwza2562he0/wC/TtH/AOy1sVjaH+5v9ate0d95ij/ZkjR8/wDfRf8AKtmgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACsfxP+90lLIdb64itiPVGYF//ABwPWxWPef6T4s02DqtrDLdt7MQI0/MPJ+VAGxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjQ/uPGt0vQXVjE4H+1G7hj+UiflWzWNqX7nxNotx/z0M9oT/vJ5n/ALRrZoAKKKKACiiigAorz/4f/D/wzrfgXT9Q1TTPPupvM8yT7RKucSso4VgOgFdH/wAKp8Gf9Ab/AMmpv/i6zUpNXt/X3HXUo0YTcHJ6abL/AOSN2isL/hVPgz/oDf8Ak1N/8XR/wqnwZ/0Bv/Jqb/4unzS7fj/wCeSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4uj/hVPgz/AKA3/k1N/wDF0c0u34/8AOSh/M//AAFf/JG7RWF/wqnwZ/0Bv/Jqb/4usrwlpVlonjrxXp+lw+Raw/Y/Lj3s2MxMx5Yk9SaOZ3SaD2VNwlKEnprqrdUu77nZUUUVZzBRRRQAVj6X/pPiDWLzqEeKzQ+oRd5/8elYfhWszKiFnICqMknsKyvC6t/wjttcOCHvC12wPUGVjJg/QMB+FAGvRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBjeJf3dha3Q62t9byE+imQIx/wC+Xatms7xBate+G9Rt4/8AWSW0gQjs207T+eKtWN0t9p9vdp92eJZF+jAH+tAE9FFFABRRRQBhfCn/AJJlpX/bb/0c9dhXH/Cn/kmWlf8Abb/0c9dhWcPhR1Yn+PP1f5hRRXI+FfH0HiXxPruhvZmzudJuGjTMu/7SiuULj5RjBHI5xkc1W7sc+yuddRXLfEDxxD4E0FL97X7dcSy+XDaiXyy+AWY5wcAKCenoO9HifxXqOjaro+m6Noiard6oszKj3gtwgjVSeSrZ4b26UAdTRXMx+ItdtfDur6p4g8OxacdPtnniij1AT+ftVmIyEG3oB0PX2p/gjxhb+M9B+2pbtZXUbbLmzkbc0LYyOcDIKkEHHINHcDo6K5OXxx5Wg+KdS/s/P/CPXE0Pl+f/AMfHlor5zt+XO7HQ9K6NLzfpK3uzG6AS7M/7OcZpXSV/R/eO2tv60LNFcjF468zwp4c1r+zsf25cwQeT5/8AqPNzzu2/NjHTAz7UzXfiHb+H/H1j4evrFha3UCSvqIk+WBndkRWXHALKBuz1YcVVne3nb5k3Vr+V/kdjRWN4l8Qf8I9bWEv2b7R9s1CCyx5mzZ5jbd3Q5x6d/UVB4g8VrpF9BpWnWE2r6zcxmWKxgZUxGDgySO3CJnjJ6ngA0v6/UZ0FFc3omu+IrvVBa694UbS4mQsl1Dfpcx5H8LYClT+GKx4vHPifULnU/wCxPBkd/a6fezWZl/tZI3kaNsEhGTv6Z/GgDvKK5K+8fW8Xw3Pi3TbKS6T5ALSZ/JYOZRGyMcHBVic8HpVjSNY8WXepxw6x4Tt9OtGB33KassxXjj5Agzk8dadtbB0udLRXCQeN/E+pXepDRPBsV9a2F9NZmY6ukTSNGcEhWj4zx3/Gt7w54qg8QNd2z2lxp2pWLBbuwugBJFkcMCCQynswODS3VwejsbtFcZJ8QGX4Yt4uj0vewfYtmbnGT5/lff2/j93296vaRrHiy71OOHWPCdvp1owO+5TVlmK8cfIEGcnjrQB0tFcd4Z+Idvr/AIs1bQLixaxuLKeWK3dpNy3axttdl4GCOCV54PWtW58RPD4yj8PxWfmvJpsl8spl25KuqBMY77uuePSjt5/8OHVrsblFefTeO/GFvr1ro8nga3F5dwyTRKNbXBVCA3PlcfeFal74w1O0+wafH4da58Q3kLztp0d4gjgjVgpZpiAMcjGFJzxijpcDraKydAv9av4Jm1/RE0iVHAjRL1bkSLjrkKMc8YIrWoAKKKKACuG0j/kpvjH/ALcv/RJrua4bSP8AkpvjH/ty/wDRJqXuv66G9L4Knp/7dE6aiiitDmCiiigDI8USMvhu7iiYrJdBbWMjqGlYRgj6bs/hWrGixRrHGoVEAVQOwFZOrf6RrmjWQ5Alku3HqsabR/4/Ih/CtigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKx/CnyeGra3/wCfRpLTHp5TtGP0WtisbQ/3Ooa1a9o73zEH+zJGjZ/76L0AbNFFFABRRRQBhfCn/kmWlf8Abb/0c9dhXH/Cn/kmWlf9tv8A0c9dhWcPhR1Yn+PP1f5hXjdojaQdQ8X24OdJ8T3sd4AfvWkrqsnHfadrj/dNeyVj6f4X0/T9P1SyXzJ4NVuZ7i4ScggmX76jAHy+xz9aeqd12/VGGjVn3/R/5nmnjp18TaV4w10HfY6Ra/2ZYns0hZWncevO1M/7JroPGo1ZvHvgweH3skvvKvdhvkdoseXHnIUg9Pet5fAmkx/D/wD4Q+FriLTTD5RdHXzTltxbJGMk89O9P8R+DbfxHd6ddNqmpabc6cJFhm0+VEbDgBsllbso6Y709mrdP+D+rFvq/P8AS35GfrK+IF+G3iX/AISiTTZJ/wCz7jyzp0ciKF8puu9ic5rCMEnhPT9C8a6ejNavp1tb65Ag+/CEUJOB/ejzz/sk9MV1Vp4LWHStT0+98Qa1qcOo27W7/brhHMSsCCUwgwcHvnoK27TToLTR4NMA822hgW3AlAbegXb83GDkdeKNrteX6/5hvZPz/G3+R5gxF14A+KDWxEqyXt2yMhyGH2eM5B+lehW9xC3gqK5WRTCdPDhweNvl5z+VVPCngjSvB+kXmmaaZ5rS7naZ47plfG5QuwcD5cKBg5+tZSfCvSkhNkusa8NIJI/sn7efs2z+5jG/b7bqmSvFxXVJfcrDTtLm7Nv72c5CjR/Cv4cq42n+0dPOD6HJH6Vs6vo1n4g+K+oaVqcXm2t14ZVJF7j/AElsEehBwQexArq9U8OWOq22nW8m+CLTbqK6gSDCgNH91SCD8vsMU8aFbDxU3iAPN9rayFkUyPL2By+cYznJ9cY7Vbacm/Nv742JV1FLyX4SueZ6lq19/ZmmeGvEMm/WNH8QaejTHj7ZAZf3c4+oGG9GBrq9AZI/i34siuiBcy29nJbBvvNAEIO32D5zjua0vEfgjSvE+r6Tqd+Z4rvSZ1mhkgZV34YNsfIOVyoOOPY81N4h8Jad4ke3nuWubS+td32a+spjDPDnqAw6g+hBHtSTdtd7v8l/l+Y7K+nb9WzIOueKNP8AiHpukar/AGRJpuqfaWgNtFKJkWJQw3Fm25+YZwPXpXJ6Tf8AjPTbPxZdeHLPTbixi1y8Z95ke6HzjcUj+VXwOQu4Eniu40jwJZ6ZrEWq3eq6xrF7bqy28upXfmCAMMNtVQq8jqSD0FamiaFbaCl8tm8zi9vZb2TzSDh5DkgYA4446n3ot+T/ADX6Dv8An+jPPPEFnp1n+zxs8PXf2y2kEE0d1IP9a73KMzMO3zE5Xt07V2miJ4zXUM+I59Bez2HiwhmWTd25dyMde1V5/h7pc3h3VNDS7vobDUrn7T5ccif6M28ORFlTtBZc4Oepxin6X4Mm0zU4bx/FniK+ERJNvd3UbRPxj5gIwT1z1p31/rsien9f1qVPhx/x7+Jf+xivf/QhUdsY7r42X8tmdy2miRwXjL0WRpSyKf8Aa25P0NK/w0hF5fTWfijxJYR31zJdSwWl4kaB3OWxiPI/PPvXRaH4e03w5YNaaTb+SjuZJXZi8kznq7uclmPqTU9F5K34WG+vm/1ueWy7/wDhmc+VtD+eNu4cZ+3cZr0PRE8ZrqGfEc+gvZ7DxYQzLJu7cu5GOvaopPAWmSeA28J/abxLJm3ecrr5wPm+bwduPve3SjS/Bk2manDeP4s8RXwiJJt7u6jaJ+MfMBGCeuetUt3/AF0B7HH6b4duNZ0bXr3RnWHXNL8TXtzp0zdN+4bo2/2HHykfQ9qv6Br8Hib4n6VqUCNCz+H50mgf70Eq3CB429wQRXa6JoVtoKXy2bzOL29lvZPNIOHkOSBgDjjjqfeqFn4I0qw8dXfiu0M8d9eQGGaIMvknJUl9uM7jsGTnn0zSjo4+S/HlsEtb+v4c1zP1X/ksfh3/ALBl7/6FFV/xL4Wm1e+tdV0fUpNK1myVkiuVQOkkbEExyIfvKSAfUHkVoXGhW1z4lstcd5hc2cEsEaAjYVkKkkjGc/KMc+tUtf8ACMWu30V7HrGr6Tcxx+WZNNuvL8xckgMpDKcEnnGeaWyXz/Nj6v5fkiPwp4h1DUrrUdJ8QWsFtq+lsgn+zOWhmRwSkiZ5AOD8p5GOtdJWP4d8MWPhuK4+ySXVzcXcgkubu8mMs07AYG5j6DgAYArYqmSFFFFIYVw2kf8AJTfGP/bl/wCiTXc1w2kf8lN8Y/8Abl/6JNS91/XQ3pfBU9P/AG6J01FFFaHMFFFFAGNaf6T4u1CfqtpbxWy+ztmR/wBDF+VbNY3hn99ps98et9dSzg+q7tqf+OKlbNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVjRfuPGtyva7sI3A943YMfykT8q2ax9T/c+JdFuP+ejTWhP+8nmfzhFAGxRRRQAUUUUAYXwp/5JlpX/AG2/9HPXYVw3w3S+k+FekDTbi3t5d0xLXEDSqV86TjAdec45z+FdD9m8Sf8AQW0r/wAFcn/yRWUX7qOzERTrz16vv39DYorH+zeJP+gtpX/grk/+SKPs3iT/AKC2lf8Agrk/+SKq77GPJH+Zfj/kbFFYN5JrWnWj3Woa/olrbx43zT6e6IuTgZJuMDkgfjWdD4kNzcRwW/jbwtLNKwSOOODczsTgAAXOSSe1F32Dkj/Mvx/yOvorH+zeJP8AoLaV/wCCuT/5Io+zeJP+gtpX/grk/wDkii77ByR/mX4/5GxRWP8AZvEn/QW0r/wVyf8AyRR9m8Sf9BbSv/BXJ/8AJFF32Dkj/Mvx/wAjYorH+zeJP+gtpX/grk/+SKPs3iT/AKC2lf8Agrk/+SKLvsHJH+Zfj/kbFFY/2bxJ/wBBbSv/AAVyf/JFH2bxJ/0FtK/8Fcn/AMkUXfYOSP8AMvx/yNiisf7N4k/6C2lf+CuT/wCSKPs3iT/oLaV/4K5P/kii77ByR/mX4/5GxRWP9m8Sf9BbSv8AwVyf/JFH2bxJ/wBBbSv/AAVyf/JFF32Dkj/Mvx/yNiisf7N4k/6C2lf+CuT/AOSKPs3iT/oLaV/4K5P/AJIou+wckf5l+P8AkbFFY/2bxJ/0FtK/8Fcn/wAkUfZvEn/QW0r/AMFcn/yRRd9g5I/zL8f8jYorH+zeJP8AoLaV/wCCuT/5Io+zeJP+gtpX/grk/wDkii77ByR/mX4/5GxRWP8AZvEn/QW0r/wVyf8AyRR9m8Sf9BbSv/BXJ/8AJFF32Dkj/Mvx/wAjYorH+zeJP+gtpX/grk/+SKPs3iT/AKC2lf8Agrk/+SKLvsHJH+Zfj/kbFFY/2bxJ/wBBbSv/AAVyf/JFH2bxJ/0FtK/8Fcn/AMkUXfYOSP8AMvx/yNiuG0j/AJKb4x/7cv8A0Sa6SG38QLPGbjU9NeIMC6pp0isy55AJnODjvg/Q1zekf8lN8Y/9uX/ok0uq/roawSUKlnfT/wBuidNRRRWhyBWfr149h4fvrmHmZIG8oerkYUfixArQrG1/9/NpdgOftF6juP8AZiBlz/30ij8aANHT7NNP022sovuW8KRL9FAA/lViiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKxvE37vTra6HW1vbeQn0UyBWP8A3yzVs1neILVr3w3qVtH9+S1kVCOzbTg/nigDRoqvY3S32nW12n3Z4lkGPRgD/WrFABRRRQBhfCn/AJJlpX/bb/0c9dhXH/Cn/kmWlf8Abb/0c9dhWcPhR1Yn+PP1f5hRRRVHOcb8WoZbj4WaykETysFicqikkKsqMx47AAknsATXgnguGbVPHugf2dpePstzbPP9lWR8qsq7pnyW29RkjCjjgV9JeNP+RC1//sGXP/opq8phFqfHvgT7ZYQ3n/FPaf5Xm3otvJk83iRckeYw7RjJbPTigD3Gq+oX9tpenT31/J5VtbxmSV9pbao6nAyT+FWK57x9/wAk913/AK8Zf/QaT0VxxV2kbOn6ha6rp0F9p8wntrhBJFIoI3A+x5H0NV7XXtNvdavNJtbkSX1iFNxEEb92GGRzjB/A1yOnatH4Ig1XTrgM0EcA1HTowPvLIQrQr9JSMe0grM0dLjwlrWv3NyPtN9BoSXtxzxJMXldvwzx9BTdk/LX8n/lqTG7Xnp+n+Z6jRXnenL4gltdJv7Oz12W+kkhlurm6voTbTRNjzMQiYqo2kldqAjA983tO06bXde8UJe6rqSwwXixW0VveSQiA+ShLAoQTyfunK98cmh6X8v0t/mO6ev8AXX/I7amu6xxs7nCqCSfQVwWk61qV/a+B5ru6kMt1PPHdbG2rPsikGWAwDyoP1rWubuc+PtQtPPkNuuiLJ5O87A5kcbtvTOABmlP3b28/wVxxV3r5fi7HQadqFrq2mwX+ny+dbXCB4pNpXcp74IBH41ZrzHS2un8GfD+1tL+4sluZAkzW77S6eS5Kn64/DqOQK1Lm/ufCmua1Baz3N3ZxaKdRihu7h5ykqMVIDOS204HGfXGKqVot/P8ABXJjdpfL8XY7qqf9r2X9p3Gn+f8A6VbQrPLHsb5UJIBzjB+6eBzXC63p95p3gK21ddd1T+05HtpLiX7a5SXzJU3KI87EX5uNgBwMZwTnUH/JS9f/AOwLB/6FLUyvH8fwVyo2l+H4uxox+OdBktY7oz3UVpLtK3c+n3EUGGOATKyBADkck4roFZXQOjBlYZBByCK53wOkcnw30VJlVo20+MOrDII2cg+1Q/Dgv/wg1opJaFJJkt2JzmESsE59NuMe2KuStJrsQndJnU0UUVJQUUUUAFFFFABRRRQAUUUUAFcNpH/JTfGP/bl/6JNdzXDaR/yU3xj/ANuX/ok1L3X9dDel8FT0/wDbonTUUUVocwVj/wDH1409VsLH/wAemf8AmBD/AOPe9bFY2gfv5dUv+v2m9dUP+zEBF+WUY/jQBs0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBj+FPk8N29t3s2ktMf9cnaMfoorYrG0T9zqWtWvZLwSoP8AZkjRv/Qt9bNABRRRQBhfCn/kmWlf9tv/AEc9dhXH/Cn/AJJlpX/bb/0c9dhWcPhR1Yn+PP1f5hRRRVHOMmhiubeSC4iSWGVSkkcihldSMEEHggjtWXD4U0GC4jnGlW0k0LBoJJ081oMHKrGWyY1U/dVcKvYCteigAqnq+mQ61o13pt00iQ3cTRO0ZAYAjBxkEZ/CrlFAbGTqPhrTtVuNLnvEZ5NLlEsByOSBjDccjIU9uVFPGg2n9uXeqP5kkt5apayxOQYyilj0xnJ3HPNadFAbGFpvhj+yhBDa6zqf2C3bMVi7xmNFByqb9nmFRxgFzwADkcVe0/SINOvNQuYHkZ9QnE8ocghWCKmFwOmFHXNX6KAOf/4Q2xXQrDTILm7hOnzefa3Ubr50b5Yk5K7SCGYEFSCDT7DwnbWWrXWpyXt7d3l5bC3nluHT5lBJBCqoCkA44AHHTOSd2igDDtfCdjaWOiWsctwU0Vt1uWZcudjJ8/HPDHpirF1olpLqs2qTJJPI9ibN4OCjx7txGPU9OuK1KKHqC0PMrvRU1fTbDR9JfxFLClxC6QajbPBDYxq4Yne0amQhcqql36j0yO6/sG2/ty81XfN593arayLkbQqliCBjOfmPetOih6r+uugLQ5m18FJb6VDpUmuatcaZFGIvsjvCisgGApeONXx/wLnocjIro4YY7eBIbeNYoo1CoiLhVA4AA7Cn0U7gFFFFIAooooAKKKKACiiigAooooAK4bSP+Sm+Mf8Aty/9Emu5rhtI/wCSm+Mf+3L/ANEmpe6/rob0vgqen/t0TpqKKK0OYrajeJp2mXV7L9y3heVvooJ/pUGhWb2Hh+xtpv8AWxwL5p9XIyx/76Jqt4m/fadBYjrfXUUBHqm7c4/74R62aACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMaP9x42nXtd2EbAe8UjAn8pV/IVs1jap+58SaJcf8APR5rQn2ePzP5witmgAooooAwvhT/AMky0r/tt/6Oeuwrj/hT/wAky0r/ALbf+jnrsKzh8KOrE/x5+r/MKKKKo5wooooAKr6hf22l6dPfX8nlW1vGZJX2ltqjqcDJP4VYrnvH/wDyTvXv+vGX/wBBNTJ2i2VFXaRvxSpNCksZ3I6hlOOoPSnVxNzHcXvinw5p41C7trOTSpZJ4reZo/Nx5QAJBBHXqMHtnk1Ru9Y1Pw/Z+KrGyupZjZz2yWMt1IZmg+0bQcs+SwUkkbiew6VclZ2X9a2Ii7r+u1z0SiuG1jS5NA1Dw0bHWNUYTalHBcpcX0kouBsc5IYnHI6LhTnkcDE+iQT6h4s8RT3t/fSQWN8gtbZLmREQ+SjHIUjcDkfKcr145paa+X6W/wAxv+vx/wAjsqK8vsbzxBr3hr+2rCx12TVbhmmtZY76FLRMMdsflecAUwNpLJuPJ9MdFbpP4l8VatBqF1eW1rpgihitrS6eDLugdnZo2DN1AAJ28HjJo1A1Y/E0E1hcXVtZX1wLe+axeOGIO+4PsLAA/cGc59B0rZrzCylutO8F3YhvpmlXxR5TXCttaQG6UNnbgcjOR0Oa1L2/vNX8ZatZPZazc2emrFFHFpl4lt87pvMjt5sbE8gAcrwT1NC2T/rZP9Q6tf1u1+h3dBIVSScAck159OfEDWPhWy1S7vLC6m1KSGdklTzJYQkhUOUJXcVC5I6Hkc1ejtXtfFuo6D9svpdNuNMW7VZL2VpIpA5U7ZS28AjHG719TSlonbz/AAVw/r8bHQN4j0pdEt9XN1/oNyyLFN5bfMXYKvGMjJI6imXvibTbDUn0+U3cl0kayvHbWM8+1WJAJMaMBnB/KvPksYrX4M6NNE9wzTz2LOJbiSRQfPX7qsxCD2UCuwsP+Soaz/2DbX/0OWqa1t5tfhcXS/kvzsbWl6xYazA8um3KzCNzHKhBV4mHVXRgGU+xANXa5Qr5fxcQ2mB52kMbwL3xKBGW9+XAPpmurpdE/wCt7D62/rYKKKKACiiigAooooAKKKKACuG0j/kpvjH/ALcv/RJrua4bSP8AkpvjH/ty/wDRJqXuv66G9L4Knp/7dEf428QXfhvQ4ryxhiZpLqOB5p1ZorZGPMjhecDAHUcsPoeUHxG1M6pPZNdaLDaQmXy9ZkhlNtdFVBEaAN97nn5m9geM9B4r1HxxaarHH4S0eyvrIwhnkuHAYSbmyOZF4xt7d+tYn9t/Fn/oWNK/7+r/APH60OY2PDusXPiiTQb2+tfssqWct3LFg4DM3lRsM84ZfMYexHJ612Nc74Ug1uSO41PxTb29rqV0FjMFu2VSNC23ueSXY8E9R9B0VABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGN4m/d6bBdDra3lvKT6L5iq3/jrNWzWdr9q194b1G2T78ttIqEdm2nB/PFWbC6W+022u0+7cQpKPowB/rQBYooooAwvhT/AMky0r/tt/6Oeuwrj/hT/wAky0r/ALbf+jnrsKzh8KOrE/x5+r/MKKKKo5wooooAKp6xpcOtaNd6ZdNIkN3E0TtGQGAIwcZBGfwq5RSaurAtNTNXQrZdWstQDy+bZWr2sa5G0qxUknjr8g9O9UtU0DTVt9cu7y2ur6PUo0+020Q3MRGuB5YGDnv1zkcVv0U3qC0PPY9JfV/EGhG3uddvotMn8+S51W3NusShGARVMcfmMx25YhiADyM89np+kQabeahcwPIz6hOJ5Q5BCsEVMLgdMKOuav0U/wCv6+4DAt/CcdlJKmm6tqVlZSzGVrGF4/LBY5YKWQugJycKwwScYqW88NRT6vJqdlqF7pl1PGIrhrRkxOo+7uV0YZHIDDBwetbVFIDnLfwRplroZ0qCW6Ft9uF8C0oZw4kDgbiCSMgdcnHfPNWr3w3Hcaw2qWOoXmmXkkQimktDGRMoPy7lkRlyOcEAHkjNbNFH9foBlSaBBO2lvcXN1NLpsxnjkdwWlcqykvx/tk4GAOMYAxUx0e3PiA6uWkM5tfshQkbNm7dnGM5z71fooA5weCrP+xRpLX981jHPHNBCWj/cbH3hFOzJXOB8xJwOCKsXfhrz9cm1a01e/wBPuJ4UhkFuIWVlQsRxJG2D8x6Vt0UAZ+laLa6T5zwmSa5uGDXF1O++WZgMAk9AB2UAKOwFaFFFABRRRQAUUUUAFFFFABRRRQAVw2kf8lN8Y/8Abl/6JNdzXDaR/wAlN8Y/9uX/AKJNS91/XQ3pfBU9P/bonTUUUVocwUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeUReOdS0WO80mM6fZpo0Hl20d7HI0moFJGjxEQQMfLt6NyPTO31evP9Um8f6RruoReGNEsbzTZp/PiluJFDEsqlhjzV43bu1AFbTviJq13qelxSQWIlvdQa1n0hY5ftdog/5aOxOMD7x+QDH47fSa82/tv4s/9CxpX/f1f/j9d7pUl/NpFrJq8EVvfNEpniibcqPjkA//AFz9T1IBnfCn/kmWlf8Abb/0c9dhXH/Cn/kmWlf9tv8A0c9dhWcPhR1Yn+PP1f5hRRRVHOFFFFABVfUL+20vTp76/k8q2t4zJK+0ttUdTgZJ/CrFc94+/wCSe67/ANeMv/oNJ6K44q7SNnT9QtdV06C+0+YT21wgkikUEbgfY8j6Gq9rr2m3utXmk2tyJL6xCm4iCN+7DDI5xg/ga5HTtWj8EQarp1wGaCOAajp0YH3lkIVoV+kpGPaQVmaOlx4S1rX7m5H2m+g0JL2454kmLyu34Z4+gpuyflr+T/y1Jjdrz0/T/M9RorzvTl8QS2uk39nZ67LfSSQy3VzdX0Jtpomx5mIRMVUbSSu1ARge+b2nadNruveKEvdV1JYYLxYraK3vJIRAfJQlgUIJ5P3Tle+OTQ9L+X6W/wAx3T1/rr/kdtTXdY42dzhVBJPoK4LSda1K/tfA813dSGW6nnjutjbVn2RSDLAYB5UH61rXN3OfH2oWnnyG3XRFk8nedgcyON23pnAAzSn7t7ef4K44q718vxdjQk8WaPHplhf/AGiWS31E4tDDayyNKcFuEVS3QE8jtUth4j0zUr5rKCaSO7CeZ9mureS3kZf7ypIqlhx1AIrh9E/5Fv4b/wDXb/23lrofHi4j0KaDH26PV7cW5H3juOHA9im7I9BVtWlbzt+RCd438r/n/kdXRRRUlBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXDaR/yU3xj/wBuX/ok13NcNpH/ACU3xj/25f8Aok1L3X9dDel8FT0/9uidNRRRWhzBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGF8Kf+SZaV/22/wDRz12Fcf8ACn/kmWlf9tv/AEc9dhWcPhR1Yn+PP1f5hRRRVHOFFFFABVPV9Mh1rRrvTbppEhu4midoyAwBGDjIIz+FXKKA2MnUfDWnarcaXPeIzyaXKJYDkckDGG45GQp7cqKeNBtP7cu9UfzJJby1S1licgxlFLHpjOTuOea06KA2MLTfDH9lCCG11nU/sFu2YrF3jMaKDlU37PMKjjALngAHI4q9p+kQadeahcwPIz6hOJ5Q5BCsEVMLgdMKOuav0UAc/wD8IbYroVhpkFzdwnT5vPtbqN186N8sScldpBDMCCpBBp9h4TtrLVrrU5L29u7y8thbzy3Dp8ygkghVUBSAccADjpnJO7RQBzp8GWiaPo9ha399a/2O262njMZkztK/NuQqeGPardn4ehg1BL++vLrU7yJSsM12U/cqeoREVUBPdtu4jjOOK16Kd9bh0sFFFFIAooooAKKKKACiiigAooooAKKKKACiiigArhtI/wCSm+Mf+3L/ANEmu5rhtI/5Kb4x/wC3L/0Sal7r+uhvS+Cp6f8At0TpqKKK0OYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvhT/AMky0r/tt/6Oeuwrj/hT/wAky0r/ALbf+jnrsKzh8KOrE/x5+r/MKKKKo5wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4bSP+Sm+Mf+3L/0Sa7muG0j/kpvjH/ty/8ARJqXuv66G9L4Knp/7dE6aiiitDmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAML4U/8ky0r/tt/wCjnrsK4/4U/wDJMtK/7bf+jnrsKzh8KOrE/wAefq/zCiiiqOcKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuG0j/kpvjH/ALcv/RJrua4bSP8AkpvjH/ty/wDRJqXuv66G9L4Knp/7dE6aiiitDmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAML4U/8ky0r/tt/6Oeuwrj/AIU/8ky0r/tt/wCjnrsKzh8KOrE/x5+r/MKKKKo5wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4bSP+Sm+Mf+3L/wBEmu5rhtI/5Kb4x/7cv/RJqXuv66G9L4Knp/7dE6aiiitDmCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAML4U/8AJMtK/wC23/o567CuP+FP/JMtK/7bf+jnrsKzh8KOrE/x5+r/ADCiiiqOcKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuG0j/AJKb4x/7cv8A0Sa7muG0j/kpvjH/ALcv/RJqXuv66G9L4Knp/wC3ROmooorQ5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDC+FP8AyTLSv+23/o567CuP+FP/ACTLSv8Att/6OeuwrOHwo6sT/Hn6v8woooqjnCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArhtI/5Kb4x/7cv/RJrua4bSP+Sm+Mf+3L/wBEmpe6/rob0vgqen/t0TpqKKK0OYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvhT/yTLSv+23/AKOeuwrj/hT/AMky0r/tt/6OeuwrOHwo6sT/AB5+r/MKKKKo5wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4bSP+Sm+Mf8Aty/9Emu5rhtI/wCSm+Mf+3L/ANEmpe6/rob0vgqen/t0TpqKKK0OYKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA43SvCXibRNMi0/S/GfkWsOfLj/suJsZYseWJPUmrn9keM/+h5/8pEP+NdNRUezj/TZ0vFVG7u3/AIDH/I5n+yPGf/Q8/wDlIh/xo/sjxn/0PP8A5SIf8a6aijkX9Ni+sT7L/wABj/kcz/ZHjP8A6Hn/AMpEP+NH9keM/wDoef8AykQ/4101FHIv6bD6xPsv/AY/5HM/2R4z/wCh5/8AKRD/AI0f2R4z/wCh5/8AKRD/AI101FHIv6bD6xPsv/AY/wCRzP8AZHjP/oef/KRD/jR/ZHjP/oef/KRD/jXTUUci/psPrE+y/wDAY/5HM/2R4z/6Hn/ykQ/40f2R4z/6Hn/ykQ/4101FHIv6bD6xPsv/AAGP+RzP9keM/wDoef8AykQ/40f2R4z/AOh5/wDKRD/jXTUUci/psPrE+y/8Bj/kcz/ZHjP/AKHn/wApEP8AjR/ZHjP/AKHn/wApEP8AjXTUUci/psPrE+y/8Bj/AJHM/wBkeM/+h5/8pEP+NH9keM/+h5/8pEP+NdNRRyL+mw+sT7L/AMBj/kcz/ZHjP/oef/KRD/jR/ZHjP/oef/KRD/jXTUUci/psPrE+y/8AAY/5HM/2R4z/AOh5/wDKRD/jR/ZHjP8A6Hn/AMpEP+NdNRRyL+mw+sT7L/wGP+RzP9keM/8Aoef/ACkQ/wCNH9keM/8Aoef/ACkQ/wCNdNRRyL+mw+sT7L/wGP8Akcz/AGR4z/6Hn/ykQ/40f2R4z/6Hn/ykQ/4101FHIv6bD6xPsv8AwGP+RzP9keM/+h5/8pEP+NH9keM/+h5/8pEP+NdNRRyL+mw+sT7L/wABj/kcz/ZHjP8A6Hn/AMpEP+NH9keM/wDoef8AykQ/4101FHIv6bD6xPsv/AY/5HM/2R4z/wCh5/8AKRD/AI1P4e8PXuk6nqeoapq39p3Wo+V5kn2ZYceWpUcKSOhHp0rfopqCTuKWInKLjpr2SX5IKKKKowCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAP/2Q==)

Many meals in the menu will have many ingredients assigned to them. Therefore, my solution was to use a link table which takes the MenuID (Primary Key) and the IngredientID (Primary Key) and references them in the link table as foreign keys with MenuIngID as the primary key for the table. This solution allows me to access the multiple ingredients that are assigned to an individual meal item, which is important for the development of the stock replenishment aspect of the program.

The stock replenishment area will reference the many ingredients assigned to the meals in the customer’s order and compare the quantity in stock to the threshold level. A re-order will be processed accordingly if the quantity in stock is less than the threshold level.

## SQL Commands

‘CREATE TABLE INGREDIENTS (IngredientID Integer PRIMARY KEY,

Name tinytext,

Quantity varchar(20),

CostPrice decimal (2,2),

MinOrderLvl varchar(20))’

The above statement creates the ingredients table with its respective fields.

‘SELECT name FROM sqlite\_master WHERE type=’table’ AND name=’MENU’’

The above SQL command will check for the existence of a table in the database, which is useful for making the system versatile and not dependant on an existing database file.

‘UPDATE MENU SET Type=?, Name=?, Description=?, MealPrice=? WHERE MenuID=?’

This SQL command will update the Menu table with new values, passed as parameters in the code to the command in the space denoted by a ‘?’, for a given Menu ID.

‘INSERT INTO MENUINGREDIENTS (IngredientID, MenuID) VALUES (?,?)’

The above SQL command will add a new record into the link table passing the foreign keys as parameters into the command from the code.

‘INSERT INTO SUPPLIERS (Name, Address, Telephone) VALUES (?,?,?)’

This SQL command will add a supplies contact information to the suppliers table in the database.

## Classes

Diagram

Description automatically generated

Blue Diamond = Composition

White Diamond = Association

The above inheritance diagram shows all the links between each of the classes. The program has been coded in such a way where the main GUI package being used has a ‘QMainWindow’ class (which allows the app to display) is the main class used in the program. I have created a class for each of the individual forms in the program, to keep the code in a clean and organised manner. Each of the individual classes are linked to the ‘QMainWindow’ class by composition. This means that all the classes cannot exist without this class. Each of the classes are then linked to each other by association.

Each of the classes must inherit the properties of the ‘QMainWindow’ class especially its constructor method. The ‘super()’ function avoids using the base class name explicitly; it’s necessary for things to work to define subclasses based on other QtGUI objects in the main GUI package.

In my case, I am inheriting from ‘QMainWindow’. If I don’t define a constructor method in each of the individual classes, then the constructor defined in the ‘QMainWindow’ class (parent) is used. I have defined my own constructor to initialise variables, update input boxes etc so this overrides the default parent constructor however I still need to access the parent constructor to initialise everything it does by default, so this is where ‘super().\_\_init\_\_’ is called passing any parameters the parent method requires.

|  |
| --- |
| **QMainWindow**() – Predefined class in the GUI package |
| **Properties** |
| * animated * dockNestingEnabled * dockOptions * documentMode * iconSize * tabShape * toolButtonStyle * unifiedTitleAndToolBarOnMac |
| **Methods / Functions** |
| * isAnimated() * isDockNestingEnabled() * menuBar() * takeCentralWdiget() |

|  |
| --- |
| **MenuWin**() |
| **Properties** |
| * item\_to\_edit * menu\_item * meal\_record * ingredient\_ids |
| **Methods / Functions** |
| * ingredientsView() * refresh\_listbox() * edit() * save() |

+ Denotes Public

* Denotes Private

## User Interface Design (HCI)

I decided to use a GUI system instead of a command line as I felt it would be appropriate for the use of the program and would be more time efficient and user friendly for the client. The layout of each window within the program is systematic and data input flows smoothly, complimenting ease of use for the client. I made sure to match the colour scheme of the system to the colours used within the business which the client showed appreciation for.

A screenshot of a computer

Description automatically generated with medium confidence

Basic data input the manager \ head chef enters.

User chooses the ingredients related to the meal, so a link is established.

The client will select an item from the list box (virtual representation of menu) and all its properties are shown in the input boxed, allowing for a change to be made.

After a meal item is added to the menu, it is shown in the list box. The list box is a virtual representation of the menu.

# Technical Solution

*"""  
############################################  
# Restaurant POS with Stock Control System #  
# Wasim Ahmed - Kings School Grantham #  
############################################  
"""*# PYTHON 3.9  
# IMPORT PACKAGE / PYTHON FILES  
import sys  
import csv  
import os.path  
import hashlib  
import sqlite3 as sql  
import matplotlib.pyplot as plt  
from PyQt5 import QtWidgets, QtCore  
from PyQt5.QtWidgets import QApplication, QMainWindow, QMessageBox  
from datetime import datetime  
from Splash import Ui\_SplashScreen  
from Login import Ui\_frmLogin  
from Home import Ui\_frmHome  
from Menu import Ui\_frmMenu  
from Ingredients import Ui\_frmIngredients  
from TakeOrder import Ui\_frmTakeOrder  
from KitchenView import Ui\_frmKitchenView  
from Suppliers import Ui\_frmSuppliers  
from StockReplenishment import Ui\_frmStockReplenishment  
  
# DB INITIALIZATION  
conn = sql.connect('ACCOUNTS.db')  
command = conn.cursor() # Execute commands on Authentication DB  
  
conn\_2 = sql.connect('POS.db')  
command\_2 = conn\_2.cursor() # Execute commands on System DB  
  
# GLOBAL VARIABLES  
COUNTER = 0 # Progress Bar - Splash Screen  
USER\_ROLE = "" # Stores Users Role in business - Whole program  
USERNAME = "" # Stores Username of worker - Whole Program  
NUM\_PRESENT\_M = False # Boolean for num validation - Menu Form  
CHAR\_PRESENT\_M = False # Boolean for char validation - Menu Form  
EDITED\_RECORD = False # Boolean for updating record - Menu Form  
NUM\_PRESENT\_I = False # Boolean for num validation - Ingred Form  
CHAR\_PRESENT\_I = False # Boolean for char validation - Ingred Form  
EDITED\_MEAL\_ITEM\_ID = 0 # For storing updated meal record ID from SQL output - Menu Form  
ORDER\_ID = 0 # Stores id which increments with orders - Take Order Form  
FOUND = False # Boolean for finding matching order num in txt file in loop - Kitchen View Form  
  
  
class SplashScreen(QMainWindow, Ui\_SplashScreen):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
  
 # Remove Title Bar  
 self.setWindowFlag(QtCore.Qt.FramelessWindowHint)  
 self.setAttribute(QtCore.Qt.WA\_TranslucentBackground)  
  
 # Set timer for progress bar  
 self.timer = QtCore.QTimer()  
 self.timer.timeout.connect(self.progress)  
 self.timer.start(35)  
  
 def progress(self): # PROGRESS BAR  
 global COUNTER  
 self.prgbar.setValue(COUNTER)  
 if COUNTER > 100:  
 self.timer.stop()  
 login = LoginWin()  
 widget.addWidget(login)  
 widget.show()  
 self.close()  
 else:  
 COUNTER += 1  
  
  
class LoginWin(QMainWindow, Ui\_frmLogin):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(345, 396)  
 self.btnLogin.clicked.connect(self.login)  
 self.txtPassword.setEchoMode(QtWidgets.QLineEdit.Password)  
  
 def login(self): # LOGIN METHOD  
 # Get username from textbox then validate against db  
 command.execute(f"SELECT \* FROM USERS WHERE USERNAME = '{self.txtUsername.text()}'")  
 user\_exists = command.fetchall()  
 if not user\_exists:  
 msg = QMessageBox()  
 msg.setWindowTitle("LOGIN UNSUCCESSFUL")  
 msg.setText("Login not found!")  
 msg.setIcon(QMessageBox.Warning)  
 msg.exec\_()  
 self.txtUsername.clear()  
 self.txtPassword.clear()  
 self.txtUsername.setFocus()  
 else: # Compare hashed password from db to hashed password of user input  
 actual\_password = [col[3] for col in user\_exists]  
 sh = hashlib.sha1()  
 sh.update(self.txtPassword.text().encode('utf-8'))  
 hashed\_password = sh.hexdigest()  
 if hashed\_password == actual\_password[0]:  
 global USERNAME  
 USERNAME = self.txtUsername.text()  
 command.execute(f"SELECT ROLE FROM USERS WHERE USERNAME = '{USERNAME}'")  
 global USER\_ROLE  
 USER\_ROLE = str(command.fetchone()).strip("(',')")  
 home = Home()  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("LOGIN UNSUCCESSFUL")  
 msg.setText("Login not found!")  
 msg.setIcon(QMessageBox.Warning)  
 msg.exec\_()  
 self.txtUsername.clear()  
 self.txtPassword.clear()  
 self.txtUsername.setFocus()  
  
  
class Home(QMainWindow, Ui\_frmHome):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnMenu.clicked.connect(self.menu)  
 self.btnIngredients.clicked.connect(self.ingredients)  
 self.btnOrder.clicked.connect(self.takeorder)  
 self.btnKitchen.clicked.connect(self.kitchen)  
 self.btnSuppliers.clicked.connect(self.suppliers)  
 self.btnStockReplenishment.clicked.connect(self.stockreplenishment)  
 self.btnPopularProd.clicked.connect(self.popularprod)  
  
 # Set all buttons to 'not enabled' by default  
 self.btnMenu.setEnabled(False)  
 self.btnIngredients.setEnabled(False)  
 self.btnOrder.setEnabled(False)  
 self.btnKitchen.setEnabled(False)  
 self.btnSuppliers.setEnabled(False)  
 self.btnStockReplenishment.setEnabled(False)  
 self.btnPopularProd.setEnabled(False)  
  
 # Assign access control  
 if USER\_ROLE == "WT": # Waiter  
 self.btnOrder.setEnabled(True)  
 elif USER\_ROLE == "CR": # Cashier  
 self.btnOrder.setEnabled(True)  
 elif USER\_ROLE == "KT": # Kitchen Staff  
 self.btnKitchen.setEnabled(True)  
 elif USER\_ROLE == "HC": # Head Chef  
 self.btnMenu.setEnabled(True)  
 self.btnIngredients.setEnabled(True)  
 self.btnKitchen.setEnabled(True)  
 elif USER\_ROLE == "MA": # Manager  
 self.btnMenu.setEnabled(True)  
 self.btnIngredients.setEnabled(True)  
 self.btnOrder.setEnabled(True)  
 self.btnKitchen.setEnabled(True)  
 self.btnSuppliers.setEnabled(True)  
 self.btnStockReplenishment.setEnabled(True)  
 self.btnPopularProd.setEnabled(True)  
  
 def menu(self): # GO TO MENU WINDOW  
 menuwin = MenuWin()  
 widget.addWidget(menuwin)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def ingredients(self): # GO TO INGREDIENTS WINDOW  
 ingredwin = IngredientsWin()  
 widget.addWidget(ingredwin)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def takeorder(self): # GO TO TAKE ORDER WINDOW  
 orderwin = TakeOrderWin()  
 widget.addWidget(orderwin)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def kitchen(self): # GO TO KITCHEN WINDOW  
 kitchen = KitchenViewWin()  
 widget.addWidget(kitchen)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def suppliers(self): # GO TO SUPPLIERS WINDOW  
 suppliers = SuppliersWin()  
 widget.addWidget(suppliers)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def stockreplenishment(self): # GO TO STOCK REPLENISHMENT WINDOW  
 stockreplenishment = StockReplenishmentWin()  
 widget.addWidget(stockreplenishment)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 @staticmethod  
 def popularprod(): # GET DATA FOR POPULAR PRODUCTS GRAPH  
 # Populate the Orders array from text file  
 orders = []  
 order\_file = open('orders.txt', 'r')  
 for line in order\_file:  
 orders.append(line)  
  
 # Create an array of each element inside this array - so a double array e.g --> [['a', 'b'], ['c', ''d]]  
 orders = list(csv.reader(orders))  
  
 # Loop through each order (a line in the text file) + remove last 2 elements and first element  
 for order in orders:  
 del order[len(order) - 2:]  
 del order[0]  
  
 # Remove white spaces in each element of an order  
 orders = [element.strip(' ') for order in orders for element in order]  
  
 # Tally the food items and store names, quantities separately  
 Food\_items = []  
 Quantities = []  
  
 for i in range(0, len(orders) - 1):  
 string = orders[i].split("x", 1)  
 quantity = int(string[0].strip())  
 food\_item = string[1].strip()  
 exists = food\_item in Food\_items  
 if exists:  
 continue  
 else:  
 for j in range(i + 1, len(orders) - 1):  
 string\_2 = orders[j].split("x", 1)  
 quantity\_2 = int(string\_2[0].strip())  
 food\_item\_2 = string\_2[1].strip()  
 if food\_item == food\_item\_2:  
 quantity += quantity\_2  
 Food\_items.append(food\_item)  
 Quantities.append(quantity)  
  
 # Graph the data  
 plt.figure(num="ALATURKA")  
 plt.bar(Food\_items, Quantities)  
 plt.title('Popular Foods Sold')  
 plt.xlabel('Food Items')  
 plt.xticks(fontsize=6)  
 plt.ylabel('Quantities')  
 plt.show()  
  
  
class MenuWin(QMainWindow, Ui\_frmMenu):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnBack.clicked.connect(self.back)  
 self.btnEdit.clicked.connect(self.edit)  
 self.btnSave.clicked.connect(self.save)  
 self.btnDelete.clicked.connect(self.delete)  
 self.cmbMealType.addItem('Starter')  
 self.cmbMealType.addItem('Main Course')  
 self.cmbMealType.addItem('Dessert')  
 self.cmbMealType.addItem('Beverage')  
 self.ingredientsview()  
 self.refresh\_listbox()  
  
 def ingredientsview(self): # PULLS INGREDIENT NAMES FROM INGREDIENTS TABLE + POPULATES LIST BOX  
 self.lstIngredientsView.clear()  
 command\_2.execute("SELECT NAME FROM INGREDIENTS")  
 record = command\_2.fetchall()  
 for rec in record:  
 self.lstIngredientsView.addItem(str(rec).strip("('',)"))  
  
 def refresh\_listbox(self): # PULLS CERTAIN DATA FROM MENU TABLE + POPULATES LIST BOX  
 self.lstMenuView.clear()  
 command\_2.execute("SELECT \* FROM MENU")  
 records = command\_2.fetchall()  
 for record in records:  
 self.lstMenuView.addItem(record[1][0] + " - " + record[2] + " - " + str(record[4]))  
  
 def back(self): # GOES BACK TO HOME WINDOW  
 self.lstIngredientsView.clearSelection()  
 home = Home()  
 self.setFixedSize(980, 500)  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def edit(self): # EDIT A MEAL  
 self.txtName.clear()  
 self.txtDesc.clear()  
 self.txtPrice.clear()  
 self.lstIngredientsView.clearSelection()  
  
 item\_to\_edit = self.lstMenuView.currentItem().text()  
 meal\_name = (item\_to\_edit[4:])[:-7]  
  
 command\_2.execute(f"SELECT \* FROM MENU WHERE NAME='{meal\_name}'")  
 meal\_record = command\_2.fetchall()[0]  
  
 global EDITED\_MEAL\_ITEM\_ID  
  
 EDITED\_MEAL\_ITEM\_ID, meal\_type, meal\_name, meal\_desc, meal\_price = [meal\_record[i] for i in range(  
 0, len(meal\_record))]  
  
 self.cmbMealType.setCurrentText(meal\_type)  
 self.txtName.setText(meal\_name)  
 self.txtDesc.setText(meal\_desc)  
 self.txtPrice.setText(str(meal\_price))  
  
 # Get all current ingredient names for that meal item  
 command\_2.execute(f"SELECT IngredientID FROM MENUINGREDIENTS WHERE MenuID='{EDITED\_MEAL\_ITEM\_ID}'")  
 ingredient\_ids = command\_2.fetchall()  
 preselect\_ingred\_names = []  
 for ingredient\_id in ingredient\_ids:  
 command\_2.execute("SELECT Name FROM INGREDIENTS WHERE IngredientID='%s'" % ingredient\_id)  
 ingred\_name = str(command\_2.fetchone()).strip("('',)")  
 preselect\_ingred\_names.append(ingred\_name)  
  
 # Highlights the current ingredients in the list for the user  
 for x in range(0, self.lstIngredientsView.count()):  
 for y in range(0, len(preselect\_ingred\_names)):  
 if self.lstIngredientsView.item(x).text() == preselect\_ingred\_names[y]:  
 self.lstIngredientsView.item(x).setSelected(True)  
 break  
  
 global EDITED\_RECORD  
 EDITED\_RECORD = True  
  
 def save(self): # SAVES TO DB + DISPLAYS / UPDATES LIST BOX  
 # Check if 'MENU' table exists  
 command\_2.execute("SELECT name FROM sqlite\_master WHERE type='table' AND name='MENU'")  
 if command\_2.fetchone()[0] != 'MENU':  
 query = ('''  
 CREATE TABLE MENU   
 (MenuID integer PRIMARY KEY,   
 Type tinytext,   
 Name tinytext,   
 Description varchar(35),   
 MealPrice decimal(2, 2))  
 ''')  
 command\_2.execute(query)  
 conn\_2.commit()  
  
 menu\_type = self.cmbMealType.currentText()  
 meal\_name = self.txtName.text()  
 meal\_desc = self.txtDesc.text()  
 meal\_price = self.txtPrice.text()  
  
 # Validation for Name, Price Fields + Updates table based on any edits  
 global NUM\_PRESENT\_M  
 global CHAR\_PRESENT\_M  
 NUM\_PRESENT\_M = any(map(str.isdigit, meal\_name)) # Boolean value  
 CHAR\_PRESENT\_M = any(map(str.isalpha, meal\_price)) # Boolean value  
  
 global EDITED\_RECORD  
  
 if not [x for x in (menu\_type, meal\_name, meal\_desc, meal\_price) if x == ""]: # If not empty  
 if NUM\_PRESENT\_M: # If num present in 'name' field  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter only LETTERS in NAME field")  
 msg.exec\_()  
 else:  
 if CHAR\_PRESENT\_M: # If letter present in 'price' field  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter only NUMBERS in PRICE field")  
 msg.exec\_()  
 else:  
 if menu\_type == "Beverage":  
 if EDITED\_RECORD: # Record been edited  
 command\_2.execute(f"SELECT MenuID FROM MENU WHERE Name='{meal\_name}'")  
 updated\_bev\_id = str(command\_2.fetchone()).strip("(,)")  
  
 query = ('''  
 UPDATE MENU SET Type=?, Name=?, Description=?, MealPrice=? WHERE MenuID=?  
 ''')  
  
 command\_2.execute(query, (menu\_type, meal\_name, meal\_desc, meal\_price, updated\_bev\_id))  
 conn\_2.commit()  
 EDITED\_RECORD = False  
  
 self.txtName.clear()  
 self.txtDesc.clear()  
 self.txtPrice.clear()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Drink successfully UPDATED in menu")  
 msg.exec\_()  
  
 self.ingredientsview()  
 self.refresh\_listbox()  
  
 elif not EDITED\_RECORD: # Record not edited  
 query = ('''  
 INSERT INTO MENU (Type, Name, Description, MealPrice) VALUES (?, ?, ?, ?)  
 ''')  
 command\_2.execute(query, (menu\_type, meal\_name, meal\_desc, meal\_price))  
 conn\_2.commit()  
  
 self.txtName.clear()  
 self.txtDesc.clear()  
 self.txtPrice.clear()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Drink ADDED successfully to menu")  
 msg.exec\_()  
  
 self.ingredientsview()  
 self.refresh\_listbox()  
  
 elif self.lstIngredientsView.currentItem() is not None: # All validation passed  
 if EDITED\_RECORD: # Record been edited  
 query = ('''  
 UPDATE MENU SET Type=?, Name=?, Description=?, MealPrice=?  
 WHERE MenuID=?  
 ''')  
 command\_2.execute(query, (menu\_type, meal\_name, meal\_desc, meal\_price, EDITED\_MEAL\_ITEM\_ID))  
 conn\_2.commit()  
  
 ingreds = [] # Populate with newly selected ingreds from listbox  
 ingred\_ids = [] # SQL in For-Loop populates with ID's for ingredient in 'ingreds'  
 [ingreds.append(str(item.text())) for item in self.lstIngredientsView.selectedItems()]  
  
 # Gets ingredient id from their names in 'ingreds' + stores in array 'ingred\_ids'  
 for x in range(0, len(ingreds)):  
 command\_2.execute(f"SELECT IngredientID FROM INGREDIENTS WHERE Name='{ingreds[x]}'")  
 ingred\_ids.append(str(command\_2.fetchone()).strip("(,)"))  
  
 # Gets primary key value from Link-Table + Store in array  
 query = f"SELECT MenuIngID FROM MENUINGREDIENTS WHERE MenuID='{EDITED\_MEAL\_ITEM\_ID}'"  
 command\_2.execute(query)  
 p\_keys\_sql = []  
 for p\_key in command\_2.fetchall():  
 p\_keys\_sql.append(str(p\_key).strip("('',)"))  
  
 # Loop through primary keys + For each primary key, update ingredient id in link table  
 # With new ingredient id from 'ingred\_ids'  
 for x, y in zip(ingred\_ids, p\_keys\_sql):  
 query = ("UPDATE MENUINGREDIENTS SET IngredientID={0} where MenuIngID={1}".format(x, y))  
 command\_2.execute(query)  
 conn\_2.commit()  
 EDITED\_RECORD = False  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SAVED")  
 msg.setText("Menu item SUCCESSFULLY updated!")  
 msg.exec\_()  
  
 self.txtName.clear()  
 self.txtDesc.clear()  
 self.txtPrice.clear()  
 self.lstIngredientsView.clearSelection()  
 self.refresh\_listbox()  
  
 elif not EDITED\_RECORD: # Record not edited  
 query = ('''  
 INSERT INTO MENU (Type, Name, Description, MealPrice) VALUES (?, ?, ?, ?)  
 ''')  
 command\_2.execute(query, (menu\_type, meal\_name, meal\_desc, meal\_price))  
 conn\_2.commit()  
  
 # New record for new menu option in link table + stores 'MenuID' + 'IngredientID' with it  
 ingredients = []  
 [ingredients.append(str(item.text())) for item in self.lstIngredientsView.selectedItems()]  
  
 command\_2.execute(f"SELECT MenuID FROM MENU WHERE Name='{meal\_name}'")  
 menu\_id = str(command\_2.fetchone()).strip("(,)")  
  
 for i in range(0, len(ingredients)):  
 command\_2.execute(f"SELECT IngredientID FROM INGREDIENTS WHERE Name='{ingredients[i]}'")  
 ingredient\_id = str(command\_2.fetchone()).strip("(,)")  
 query = "INSERT INTO MENUINGREDIENTS (IngredientID, MenuID) VALUES (?, ?)"  
 command\_2.execute(query, (ingredient\_id, menu\_id))  
 conn\_2.commit()  
  
 self.txtName.clear()  
 self.txtDesc.clear()  
 self.txtPrice.clear()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Meal ADDED successfully to menu")  
 msg.exec\_()  
  
 self.ingredientsview()  
 self.refresh\_listbox()  
  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Select INGREDIENTS needed for the meal")  
 msg.exec\_()  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter ALL values")  
 msg.exec\_()  
  
 def delete(self): # DELETES RECORD FROM 'MENU' TABLE + ADJUSTS LINK-TABLE ACCORDINGLY  
 selected\_item = self.lstMenuView.currentItem().text()[4:]  
 delimiter = " -"  
 meal\_name = selected\_item.split(delimiter, 1)[0]  
 command\_2.execute(f"SELECT MenuID FROM MENU WHERE Name='{meal\_name}'")  
 meal\_id = str(command\_2.fetchone()).strip("(,)")  
 try:  
 command\_2.execute(f"DELETE FROM MENU WHERE NAME='{meal\_name}'")  
 conn\_2.commit()  
 command\_2.execute(f"DELETE FROM MENUINGREDIENTS WHERE MenuID='{meal\_id}'")  
 conn\_2.commit()  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Item DELETED successfully from menu")  
 msg.exec\_()  
 self.refresh\_listbox()  
 except ValueError:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Item HASN'T been deleted. Please try again")  
 msg.exec\_()  
 self.refresh\_listbox()  
  
  
class IngredientsWin(QMainWindow, Ui\_frmIngredients):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnBack.clicked.connect(self.back)  
 self.btnSave.clicked.connect(self.save)  
 self.btnDelete.clicked.connect(self.delete)  
 self.refresh\_listbox()  
  
 def refresh\_listbox(self): # PULLS DATA FROM INGREDIENTS TABLE + POPULATES LIST BOX  
 self.lstIngredientView.clear()  
 command\_2.execute("SELECT \* FROM INGREDIENTS")  
 records = command\_2.fetchall()  
 for record in records:  
 self.lstIngredientView.addItem(record[1] + " - " + str(record[2]))  
  
 def back(self): # GOES BACK TO HOME WINDOW  
 home = Home()  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def save(self): # SAVES TO DB + DISPLAYS / UPDATES LIST BOX  
 # Check if 'INGREDIENTS' table exists  
 command\_2.execute("SELECT name FROM sqlite\_master WHERE type='table' AND name='INGREDIENTS'")  
 if command\_2.fetchone()[0] != 'INGREDIENTS':  
 query = ('''  
 CREATE TABLE INGREDIENTS   
 (IngredientID integer PRIMARY KEY,   
 Name tinytext,   
 Quantity varchar(20),   
 CostPrice decimal(2, 2))  
 MinOrderLvl varchar(20)  
 ''')  
 command\_2.execute(query)  
 conn\_2.commit()  
  
 ingred\_name = self.txtName.text()  
 ingred\_quantity = self.txtQuantity.text()  
 ingred\_cprice = self.txtCostPrice.text()  
 ingred\_minlvl = self.txtMinOrderLvl.text()  
  
 # Validation for Name, Quantity, Cost Price fields  
 global NUM\_PRESENT\_I  
 NUM\_PRESENT\_I = any(map(str.isdigit, ingred\_name))  
  
 if not [x for x in (ingred\_name, ingred\_quantity, ingred\_cprice, ingred\_minlvl) if x == ""]: # If not empty  
 if NUM\_PRESENT\_I: # If num present in 'name' field  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter only LETTERS in NAME field")  
 msg.exec\_()  
 else: # if num not present in field  
 query = ('''  
 INSERT INTO INGREDIENTS (Name, Quantity, CostPrice, MinOrderLvl) VALUES (?, ?, ?, ?)  
 ''')  
 command\_2.execute(query, (ingred\_name, ingred\_quantity, ingred\_cprice, ingred\_minlvl))  
 conn\_2.commit()  
 self.txtName.clear()  
 self.txtQuantity.clear()  
 self.txtCostPrice.clear()  
 self.txtMinOrderLvl.clear()  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Ingredient ADDED successfully to pantry")  
 msg.exec\_()  
 self.refresh\_listbox()  
  
 else: # Fields are empty  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter ALL values")  
 msg.exec\_()  
  
 def delete(self): # DELETES RECORD FROM 'INGREDIENTS' TABLE  
 if self.lstIngredientView.currentItem() is not None:  
 selected\_item = self.lstIngredientView.currentItem().text().strip(" -")  
 delimiter = " -"  
 ingredient = selected\_item.split(delimiter, 1)[0]  
 query = "DELETE FROM INGREDIENTS WHERE NAME='" + ingredient + "'"  
 try:  
 command\_2.execute(query)  
 conn\_2.commit()  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Item DELETED successfully from menu")  
 msg.exec\_()  
 self.refresh\_listbox()  
 except ValueError:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Item HASN'T been deleted. Please try again")  
 msg.exec\_()  
 self.refresh\_listbox()  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("SELECT item from list to delete.")  
 msg.exec\_()  
  
  
class TakeOrderWin(QMainWindow, Ui\_frmTakeOrder):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnBack.clicked.connect(self.back)  
 self.btnAdd.clicked.connect(self.add)  
 self.btnDelete.clicked.connect(self.delete)  
 self.btnSave.clicked.connect(self.save)  
 self.btnClear.clicked.connect(self.clear)  
 self.populate\_cmb()  
  
 def populate\_cmb(self): # POPULATES COMBO BOX ON FORM LOAD  
 command\_2.execute("SELECT NAME FROM MENU")  
 meal\_names = command\_2.fetchall()  
 for item in range(0, len(meal\_names)):  
 self.cmbOrderName.addItem(str(meal\_names[item]).strip("(',)')"))  
  
 def back(self): # GOES BACK TO HOME WINDOW  
 home = Home()  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def add(self): # ADD ORDER TO LIST BOX  
 Meal\_Name = self.cmbOrderName.currentText()  
 Quantity = self.spnQuantity.text()  
 if Quantity == "0": # Validation Check  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("No Quantity was added! Please choose a quantity.")  
 msg.exec\_()  
 else: # Adds record to list box  
 quantity\_mealname = str(Quantity + " x " + Meal\_Name)  
 self.lstOrderView.addItem(quantity\_mealname)  
 self.cmbOrderName.setCurrentText("")  
 self.spnQuantity.setValue(0)  
  
 def delete(self): # REMOVE ORDER FROM LIST BOX  
 entries\_from\_list = []  
 if self.lstOrderView.currentItem() is not None:  
 for item in range(0, self.lstOrderView.count()): # Copy current list box contents to new array  
 entries\_from\_list.append(self.lstOrderView.item(item).text())  
  
 for item in range(0, len(entries\_from\_list)): # Remove selected item from new array  
 if entries\_from\_list[item] == self.lstOrderView.currentItem().text():  
 entries\_from\_list.pop(item)  
 break  
  
 self.lstOrderView.clear() # Clear the list box  
  
 for item in range(0, len(entries\_from\_list)): # Copy new array contents to list box  
 self.lstOrderView.addItem(entries\_from\_list[item])  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Entry REMOVED from order.")  
 msg.exec\_()  
  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("SELECT entry from order to delete.")  
 msg.exec\_()  
  
 def clear(self): # CLEAR THE LIST BOX  
 self.lstOrderView.clear()  
  
 def save(self): # SAVE ORDER TO FILE + SHOW 'RECEIPT'  
 global ORDER\_ID  
 # Store the order as a string for receipt + get food items to validate stock levels  
 orders = ''''''  
 Food\_Items = []  
 for entry in range(0, self.lstOrderView.count()):  
 string = self.lstOrderView.item(entry).text().split("x", 1)  
 Food\_Items.append(string[1].strip())  
 orders += self.lstOrderView.item(entry).text() + ", "  
  
 isStock = False  
  
 # Get Food Item ID, Type from Menu table  
 for item in Food\_Items:  
 command\_2.execute("SELECT MenuID,Type FROM MENU WHERE Name='%s'" % item)  
 result = command\_2.fetchall()  
 food\_id, food\_type = result[0]  
 if food\_type == "Beverage": # Ignore items where type = beverage  
 continue  
 else: # Get all ingred id's related to that food item(meal)  
 ingred\_ids = []  
 command\_2.execute("SELECT IngredientID FROM MENUINGREDIENTS WHERE MenuID=%s" % food\_id)  
 results = command\_2.fetchall()  
 for x in results:  
 ingred\_ids.append(str(x).strip("(,)"))  
 for i\_id in ingred\_ids: # Loop through each of the ingredient ID's  
 command\_2.execute("SELECT Quantity,MinOrderLvl FROM INGREDIENTS WHERE IngredientID='%s'" % i\_id)  
 results = command\_2.fetchall()  
  
 raw\_quantity, raw\_minorderlvl = results[0]  
 quantity\_filter = filter(str.isdigit, raw\_quantity)  
 quantity = int("".join(quantity\_filter).strip())  
  
 minorderlvl\_filter = filter(str.isdigit, raw\_minorderlvl)  
 minorderlvl = int("".join(minorderlvl\_filter).strip())  
  
 isStock = quantity > minorderlvl  
  
 command\_2.execute("SELECT Name FROM INGREDIENTS WHERE IngredientID='%s'" % i\_id)  
 name = str(command\_2.fetchone()).strip("(',)")  
  
 if not isStock: # Not in stock then write to file and break loop  
 if os.path.isfile("stock\_replenishment.txt"): # If file exists  
 file = open('stock\_replenishment.txt', 'r')  
 contents = file.read()  
 if name in contents: # If name already in the file then skip  
 pass  
 else:  
 file.close()  
 file = open('stock\_replenishment.txt', 'a')  
 file.write(name + "\n")  
 break  
 else: # Else file doesn't exist  
 with open('stock\_replenishment.txt', 'a') as f:  
 f.write(name + "\n")  
 break  
  
 if not isStock: # If not in stock, show meal that can't be made, then break loop  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText(f"{item} meal not available")  
 self.lstOrderView.clear()  
 msg.exec\_()  
 break  
  
 if isStock: # If every meal is in stock then save and show receipt  
 if os.path.isfile("orders.txt"): # If file exist  
 # Store New Order Num -> Get current order num from eof file and + 1  
 orders\_file = open("orders.txt", "rt")  
 lineList = orders\_file.readlines()  
 orders\_file.close()  
 ORDER\_ID = int((lineList[-1:][0]).split(",")[0]) + 1  
  
 # Get the full name of the user  
 command.execute("SELECT FNAME,LNAME FROM USERS WHERE USERNAME='%s'" % USERNAME)  
 result = command.fetchall()  
 first\_name = result[0][0]  
 last\_name = result[0][1]  
 full\_name = first\_name + " " + last\_name  
  
 # Get system date + time  
 now = datetime.now()  
 date\_time = now.strftime("%d/%m/%Y %H:%M")  
  
 # Store order in file -> Format with order num + order  
 orders\_file = open("orders.txt", "at")  
 orders\_file.write("\n" + str(ORDER\_ID) + ", " + orders + full\_name + ", " + date\_time)  
 orders\_file.close()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Order SAVED + SENT to kitchen")  
 msg.exec\_()  
  
 # Display the order in the format of a receipt  
 msg = QMessageBox()  
 orders = ''''''  
 for entry in range(0, self.lstOrderView.count()):  
 orders += self.lstOrderView.item(entry).text() + "\n"  
  
 msg.setWindowTitle("View Receipt")  
 msg.setText("Order ID - " + str(ORDER\_ID) + "\n" + orders + "\n" + full\_name + "\n" + date\_time)  
 msg.exec\_()  
  
 self.lstOrderView.clear()  
  
 else: # Else file doesn't exist -> Same code as above except 'Order\_id' = 1 for first order  
 ORDER\_ID += 1  
  
 command.execute("SELECT FNAME,LNAME FROM USERS WHERE USERNAME='%s'" % USERNAME)  
 result = command.fetchall()  
 first\_name = result[0][0]  
 last\_name = result[0][1]  
 full\_name = first\_name + " " + last\_name  
  
 now = datetime.now()  
 date\_time = now.strftime("%d/%m/%Y %H:%M")  
  
 orders\_file = open("orders.txt", "at")  
 orders = ''''''  
 for entry in range(0, self.lstOrderView.count()):  
 orders += self.lstOrderView.item(entry).text() + ", "  
  
 orders\_file.write(str(ORDER\_ID) + ", " + orders + full\_name + ", " + date\_time)  
 orders\_file.close()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Order SAVED + SENT to kitchen")  
 msg.exec\_()  
  
 msg = QMessageBox()  
 orders = ''''''  
 for entry in range(0, self.lstOrderView.count()):  
 orders += self.lstOrderView.item(entry).text() + "\n"  
  
 msg.setWindowTitle("View Receipt")  
 msg.setText("Order ID - " + str(ORDER\_ID) + "\n" + orders + "\n" + full\_name + "\n" + date\_time)  
 msg.exec\_()  
  
 self.lstOrderView.clear()  
  
  
class KitchenViewWin(QMainWindow, Ui\_frmKitchenView):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnBack.clicked.connect(self.back)  
 self.btnComplete.clicked.connect(self.completed)  
 self.btnRefresh.clicked.connect(self.refresh)  
 self.refresh()  
  
 def refresh(self): # UPDATES LIST BOX WITH ORDERS MADE ON SYSTEM DAY  
 self.lstOrderView.clear()  
 orders\_file = open("orders.txt", "rt")  
  
 # Add dates, order num from each entry in txt file  
 Dates = []  
 for entry in orders\_file:  
 order = entry.split(", ")  
 Dates.append(order[-1] + "," + order[0])  
 orders\_file.close()  
  
 # Bubble sort Dates array  
 for x in range(len(Dates)):  
 swapped = False  
 i = 0  
 while i < len(Dates) - 1:  
 if Dates[i + 1] > Dates[i]:  
 # Swap Values  
 Dates[i], Dates[i + 1] = Dates[i + 1], Dates[i]  
 swapped = True  
 i = i + 1  
  
 if not swapped:  
 break  
  
 # Compares sorted dates to the system date - if match store order num in another array  
 today\_order\_nums = []  
 now = datetime.now()  
 system\_date = now.strftime("%d/%m/%Y")  
 for date in Dates:  
 if date[:10] == system\_date:  
 split = date.split(",")  
 today\_order\_nums.append(split[1])  
  
 # Compare order num of each entry in file, to order num in array  
 orders\_file = open("orders.txt", "rt")  
 today\_order\_nums.reverse()  
 self.lstOrderView.setSpacing(5)  
 for entry in orders\_file:  
 order = entry.split(", ")  
 for x in range(0, len(today\_order\_nums)):  
 if order[0] == today\_order\_nums[x]:  
 string = ""  
 for y in range(1, len(order) - 2):  
 string += order[y] + ","  
 print("Order Number - " + today\_order\_nums[x] + " , " + string[:-1])  
 self.lstOrderView.addItem("Order Number - " + today\_order\_nums[x] + " ," + string[:-1])  
 elif order[0] < today\_order\_nums[x]:  
 break  
 orders\_file.close()  
  
 def back(self): # GOES BACK TO HOME WINDOW  
 home = Home()  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def completed(self): # REMOVES ORDER FROM LIST BOX  
 orders\_from\_list = []  
 if self.lstOrderView.currentItem() is not None:  
 for item in range(0, self.lstOrderView.count()):  
 orders\_from\_list.append(self.lstOrderView.item(item).text())  
  
 for order in range(0, len(orders\_from\_list)):  
 if orders\_from\_list[order] == self.lstOrderView.currentItem().text():  
 orders\_from\_list.pop(order)  
 break  
  
 self.lstOrderView.clear()  
  
 for order in range(0, len(orders\_from\_list)):  
 self.lstOrderView.addItem(orders\_from\_list[order])  
  
  
class SuppliersWin(QMainWindow, Ui\_frmSuppliers):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnFind.clicked.connect(self.find\_name)  
 self.btnEdit.clicked.connect(self.edit)  
 self.btnSave.clicked.connect(self.save)  
 self.btnDelete.clicked.connect(self.delete)  
 self.btnBack.clicked.connect(self.back)  
 self.btnEdit.setVisible(False)  
 self.btnDelete.setVisible(False)  
  
 def find\_name(self): # SEARCH SUPPLIER CONTACTS  
 supplier\_name = self.txtFindName.text()  
 if supplier\_name != "": # Validation check for empty string  
 command\_2.execute(f"SELECT \* FROM SUPPLIERS WHERE Name='{supplier\_name}'")  
 try: # If record exists - show info in respective fields  
 result = command\_2.fetchall()[0]  
 supplier\_id, supplier\_name, supplier\_address, supplier\_telephone = [result[i] for i in range(  
 0, len(result))]  
 self.txtName.setText(supplier\_name)  
 self.txtAddress.setText(supplier\_address)  
 self.txtPhone.setText(supplier\_telephone)  
  
 self.txtName.setEnabled(False)  
 self.txtAddress.setEnabled(False)  
 self.txtPhone.setEnabled(False)  
  
 self.txtFindName.clear()  
 self.btnEdit.setVisible(True)  
 self.btnDelete.setVisible(True)  
 self.btnSave.setVisible(False)  
  
 except IndexError: # Else record doesn't exist  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Record DOES NOT exist.")  
 msg.exec\_()  
 self.txtFindName.clear()  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter the NAME of the supplier.")  
 msg.exec\_()  
  
 def edit(self): # EDIT SUPPLIER CONTACT INFO  
 self.txtName.setEnabled(True)  
 self.txtAddress.setEnabled(True)  
 self.txtPhone.setEnabled(True)  
  
 self.btnEdit.setVisible(False)  
 self.btnDelete.setVisible(False)  
 self.btnSave.setVisible(True)  
  
 def save(self): # SAVE SUPPLIER INFO TO DB  
 # Check if 'SUPPLIERS' table exists  
 command\_2.execute("SELECT name FROM sqlite\_master WHERE type='table' AND name='SUPPLIERS'")  
 if command\_2.fetchone()[0] != 'SUPPLIERS':  
 query = ('''  
 CREATE TABLE SUPPLIERS   
 (SupplierID integer PRIMARY KEY,   
 Name tinytext,   
 Address varchar(40),   
 Telephone tinytext)  
 ''')  
 command\_2.execute(query)  
 conn\_2.commit()  
  
 Name = self.txtName.text()  
 Address = self.txtAddress.text()  
 Phone\_Num = self.txtPhone.text()  
  
 # Validation for Name, Phone\_Num Fields + Updates table  
 global NUM\_PRESENT\_M  
 global CHAR\_PRESENT\_M  
 NUM\_PRESENT\_M = any(map(str.isdigit, Name))  
 CHAR\_PRESENT\_M = any(map(str.isalpha, Phone\_Num))  
  
 if not [x for x in (Name, Address, Phone\_Num) if x == ""]: # If not empty  
 if NUM\_PRESENT\_M: # If num present in 'Name' field  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter only LETTERS in NAME field")  
 msg.exec\_()  
 else:  
 if CHAR\_PRESENT\_M: # If letter present in 'Phone\_Num' field  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter only NUMBERS in PRICE field")  
 msg.exec\_()  
 else:  
 command\_2.execute(f"SELECT Telephone FROM SUPPLIERS WHERE Name = '{Name}'") # Test Case  
 result = command\_2.fetchone()  
 if result is not None: # If record (Test Case) exists then update  
 command\_2.execute(f"SELECT SupplierID FROM SUPPLIERS WHERE Name='{Name}'")  
 Supplier\_ID = str(command\_2.fetchone()).strip("(,)")  
  
 query = '''  
 UPDATE SUPPLIERS SET Name=?, Address=?, Telephone=? WHERE SupplierID=?  
 '''  
 command\_2.execute(query, (Name, Address, Phone\_Num, Supplier\_ID))  
 conn\_2.commit()  
  
 self.txtName.clear()  
 self.txtAddress.clear()  
 self.txtPhone.clear()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Record SUCCESSFULLY updated.")  
 msg.exec\_()  
 else: # Else record doesnt exist  
 query = ('''  
 INSERT INTO SUPPLIERS (Name, Address, Telephone) VALUES (?, ?, ?)  
 ''')  
 command\_2.execute(query, (Name, Address, Phone\_Num))  
 conn\_2.commit()  
  
 self.txtName.clear()  
 self.txtAddress.clear()  
 self.txtPhone.clear()  
  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Supplier ADDED successfully to contacts")  
 msg.exec\_()  
  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("ERROR")  
 msg.setText("Enter ALL values")  
 msg.exec\_()  
  
 def delete(self): # DELETE SUPPLIER CONTACT FROM DB  
 Name = self.txtName.text()  
 command\_2.execute(f"DELETE FROM SUPPLIERS WHERE Name='{Name}'")  
 conn\_2.commit()  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText("Supplier REMOVED from contacts.")  
 msg.exec\_()  
  
 self.txtName.clear()  
 self.txtAddress.clear()  
 self.txtPhone.clear()  
  
 self.btnEdit.setVisible(False)  
 self.btnDelete.setVisible(False)  
 self.btnSave.setVisible(True)  
  
 def back(self): # GOES BACK TO HOME WINDOW  
 home = Home()  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
  
class StockReplenishmentWin(QMainWindow, Ui\_frmStockReplenishment):  
 def \_\_init\_\_(self): # CONSTRUCTOR METHOD  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFixedSize(980, 500)  
 self.lblUsername.setText(USERNAME)  
 self.btnBack.clicked.connect(self.back)  
 self.btnReorder.clicked.connect(self.reorder)  
  
 # SHOW LOW LEVEL STOCK ITEMS IN LIST BOX  
 ingred\_ids = []  
 Quantity = []  
 MinOrderLvl = []  
 self.reorder\_names = []  
 command\_2.execute("SELECT Name, Quantity, MinOrderLvl FROM INGREDIENTS")  
 results = command\_2.fetchall()  
 # Get names for ingreds that are lower than the min req amount needed for restaurant  
 for x in results:  
 ingred\_ids.append(str(x[0]))  
  
 quantity\_filter = filter(str.isdigit, x[1])  
 quantity = int("".join(quantity\_filter).strip())  
 Quantity.append(quantity)  
  
 minorderlvl\_filter = filter(str.isdigit, x[2])  
 minorderlvl = int("".join(minorderlvl\_filter).strip())  
 MinOrderLvl.append(minorderlvl)  
 # If amount in stock < req amount then add those names to array  
 for i, q, m in zip(ingred\_ids, Quantity, MinOrderLvl):  
 if m > q:  
 self.reorder\_names.append(i)  
 # Print those names in the list box  
 for name in self.reorder\_names:  
 self.lstStock.addItem(name)  
  
 def back(self): # GOES BACK TO HOME WINDOW  
 home = Home()  
 widget.addWidget(home)  
 widget.setCurrentIndex(widget.currentIndex() + 1)  
 self.close()  
  
 def reorder(self): # REORDER COMPLETED  
 string = ""  
 for name in self.reorder\_names:  
 string += name + ", "  
 msg = QMessageBox()  
 msg.setWindowTitle("SUCCESS")  
 msg.setText(f"Re-ordered 100 units more than required amount for {string[:-2]}")  
 msg.exec\_()  
  
  
'''  
3 ingredients below stock levels:  
 INGRED ID | MEAL NAME  
INGRED ID (4) | Cheese Balls, Grilled Halloumi  
INGRED ID (8) | Cheese Balls, Falafel  
INGRED ID (11) | Chocolate Cake, Vanilla CheeseCake, Waffle  
'''  
  
app = QApplication(sys.argv)  
widget = QtWidgets.QStackedWidget()  
w = SplashScreen()  
w.show()  
app.exec\_()

# Testing

## Introduction

The purpose of testing the system is to locate and fix any errors which cause the system to function incorrectly. In the sections below, I will show my testing and the method used with its results.

To thoroughly test the system, I will test the main program requirements individually and test functions that are reliant on other areas of the program as a whole. For most of the testing, I will use the back box testing method. This type of testing method involves calling a function and checking its output at the end. s

For other aspects of the program where necessary, I will use white box testing to test more complex processes. This is where you test each path by “seeing into” the function and tracing it line-by-line, with the use of the IDE. For each test, I will record a description, the input data, the test method, the expected results, and the actual results.

## System Testing – Normal Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Description / Purpose of test | Input Data | Test Method | Expected Results | Actual Results |
| 1 | System Login + User details stored | WAhmed007, password | Black Box | The Home window loads to navigate program + Each user should have details stored. | See “Test 1 Evidence” below |
| 2 | User Access Control | Login as “DaOne17, daniscool” - waiter | Black Box | All buttons except ‘Take Order’ button should be disabled. | See “Test 2 Evidence” below |
| 3 | Ingredients for customer’s meal is not in stock | Login as “DaOne17, daniscool” – waiter, select the ‘take order’ section, take customer’s order | Black Box | Message box should show informing waiter to tell customer, the meal can’t be prepared. | See “Test 3 Evidence” below |
| 4 | Ingredients for customer’s meal is in stock + receipt is shown | Login as “DaOne17, daniscool” – waiter, select the ‘take order’ section, take customer’s order | Black Box | Message box informing waiter meal ticket sent to kitchen staff, receipt is displayed. | See “Test 4 Evidence” below |
| 5 | Ingredient added to re-ordering txt file if low in stock and displayed in another form only accessible by manager | Login as “WAhmed007, password” – manager, select the ‘Take order’ window, take customer’s order, go back and select ‘Stock update’ | Black Box | Message box should show informing waiter to tell customer, the meal can’t be prepared. On new form, ingredients that are low in stock should be displayed. | See “Test 5 Evidence” below |
| 6 | Order Ticket is created after customer’s order is processed | Login as “WAhmed007, password” – manager, select the ‘Kitchen View’ window | Black Box | Orders made today should show in the list box on the form. | See “Test 6 Evidence” below |
| 7 | View + manipulate supplier contacts | Login as “WAhmed007, password” – manager, select the ‘Suppliers’ window, type ‘Jooma LTD’ in search field, click ‘find’ | Black Box | All the details related to the supplier should be shown in the input fields, ready to be altered if needed. | See “Test 7 Evidence” below |
| 8 | Graph showing popular meals on the menu | Login as “WAhmed007, password” – manager, select the ‘Popular Foods’ window | Black Box | A graph should show in the form of a bar chart, plotting meals and the quantities sold. | See “Test 8 Evidence” below |
| 9 | Commit completed order ticket in the kitchen | Login as “WAhmed007, password” – manager, select the ‘Take order’ window, take customer’s order, go back and select ‘Kitchen’ and click the order from list box and click ‘completed’ | Black Box | The order shouldn’t show in the list box anymore. | See “Test 9 Evidence” below |
| 10 | Add meals to the menu | Login as “WAhmed007, password” – manager, select the ‘Menu’ window, fill in input fields with dummy data | Black Box | The meal should be added to the list box in the same format / layout as the rest of the meals in the menu / list box. | See “Test 10 Evidence” below |

### Test 1 Evidence

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated*Login Screen*

*Home Screen*

Graphical user interface, text

Description automatically generated*User Details stored in Database – First + Last Name, Username, Password Hash and User Role*

### Test 2 Evidence

Video File In Folder ‘Images’ >> ‘Testing’ >> ‘Normal’ >>‘Test 2’ >> ‘Take Order.mp4’

### Test 3 Evidence

Video File In Folder ‘Images’ >> ‘Testing’ >> ‘Normal’ >> ‘Test 3’ >> ‘Take Order – not in stock.mp4’

### Test 4 Evidence

Video File In Folder ‘Images’ >> ‘Testing’ >> ‘Normal’ >> ‘Test 4’ >> ‘Take Order – in stock.mp4’

### Test 5 Evidence

Video File In Folder ‘Images’ >> ‘Testing’ >> ‘Normal’ >> ‘Test 5’ >> ‘Display low stock.mp4’

### Graphical user interface, text, website Description automatically generatedTest 6 Evidence

*Kitchen View Window*

### Graphical user interface Description automatically generatedTest 7 Evidence

*Supplier’s Window – Empty Fields*

Graphical user interface

Description automatically generated

*Supplier’s Window – Begin the search*

A screenshot of a computer

Description automatically generated with medium confidence

*Supplier’s Window – Search Completed*

### Graphical user interface Description automatically generatedTest 8 Evidence

*Home Screen*

Chart, bar chart

Description automatically generated

*Graph – Frequently bought items off the menu*

Chart, bar chart

Description automatically generated

*Graph - Zoomed In screen shot from graph above*

### Test 9 Evidence

Video File In Folder ‘Images’ >> ‘Testing’ >> ‘Normal’ >> ‘Test 9’ >> ‘Complete Order Ticket.mp4’

### Test 10 Evidence

Video File In Folder ‘Images’ >> ‘Testing’ >> ‘Normal’ >> ‘Test 10’ >> ‘Adding Meals to Menu.mp4’

## System Testing - Erroneous Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Description / Purpose of test | Input Data | Test Method | Expected Results | Actual Results |
| 1 | Log-in with incorrect username and password | Username - RJ007  Password – cool1 | Black Box | Message box alerting user that the account entered doesn’t exist. | See “Test 1 Evidence” below |
| 2 | Adding meal to menu with wrong data types in wrong fields | Login as “WAhmed007, password” – manager, select Menu, For meal name enter “tacos1”, remaining fields enter “test”, enter “1.99” for the price field and select “banana” for the ingredients used, click save. | Black Box | Message Box alerting user that only letters may be entered in that field | See “Test 2 Evidence” below |
| 3 | Adding an ingredient to virtual pantry without a name | Login as “WAhmed007, password” – manager, select Ingredients, Enter “1” for Quantity field, “1.99” for Cost price field and “12” for Cost price field, click save. | Black Box | Message Box alerting user that they should enter data in all the required fields | See “Test 3 Evidence” below |
| 4 | Add menu item to customer’s order with no quantity | Login as “WAhmed007, password” – manager, select Take Order, click add. | Black Box | Message Box alerting user that they should select a quantity before adding to the order | See “Test 4 Evidence” below |
| 5 | Finding a supplier’s contact info which doesn’t exist | Login as “WAhmed007, password” – manager, select Suppliers, enter “Jeffery LTD” and click find. | Black Box | Message Box alerting user the record doesn’t exist | See “Test 5 Evidence” below |

### Test 1 Evidence

Graphical user interface, application

Description automatically generated

*Login Screen – Incorrect Login*

### A screenshot of a computer Description automatically generated with medium confidenceTest 2 Evidence

*Menu Window – Incorrect data input*

### Graphical user interface Description automatically generatedTest 3 Evidence

*Ingredients Window – Data field left blank*

### Graphical user interface, application Description automatically generatedTest 4 Evidence

*Take Order Window – No quantity selected*

### Graphical user interface, application Description automatically generatedTest 5 Evidence

*Supplier Window - Searching for a supplier record which doesn’t exist*

## System Testing – Extreme Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Description / Purpose of test | Input Data | Test Method | Expected Results | Actual Results |
| 1 | Adding meal to menu with max data for the data types | Login as “WAhmed007, password” – manager, Select Menu, For meal name enter 255 characters, For description enter 35 characters, For price enter 999.99, Select ‘Banana’ for ingredient | Black Box | The meal should still be added to the menu as it doesn’t exceed the max data range for that data type. | See “Test 1 Evidence” below |
| 2 | Edit the meal added with extreme data from above | Login as “WAhmed007, password” – manager, Select Menu, Select the meal item and click edit | White Box | The meal’s properties/details should be shown in their relevant data fields on the form. | See “Test 2 Evidence” below |
| 3 | Adding data that exceeds the limit for the data range in the ingredients ‘virtual pantry’ | Login as “WAhmed007, password” – manager, Select Ingredients, enter data in following order, “test”, 21 characters, “11.990”, 21 characters. Click save | Black Box | The program should crash due to an invalid entry to the table in the database. | See “Test 3 Evidence” below |
| 4 | Adding 15 meals to a customer’s order | Login as “WAhmed007, password” – manager, Select Take Order, Add 15 meals to customer’s order (list box) | Black Box | Program should save this order without any issues and a digital receipt should be displayed. | See “Test 4 Evidence” below |
| 5 | Testing if quantity on graph can be a large value | Change quantity of a meal item in the ‘orders.txt’ file to a large number e.g. 75 | Black Box | The graph should still display everything like normal. | See “Test 5 Evidence” below |

### A screenshot of a computer Description automatically generated with medium confidenceTest 1 Evidence

*Menu Form – Data input*

A screenshot of a computer

Description automatically generated with medium confidence

*Menu Form - Data being accepted and meal added to menu*

### Graphical user interface, application Description automatically generated with medium confidenceTest 2 Evidence

Area of Focus

Text

Description automatically generated*IDE – Debugging Code*

*IDE – Code and Variables for normal data*

From the above screen shot you can see for normal data the SQL query at ‘command\_2.execute…’ is executed with no errors and all the fields on the form are updated with this information pulled from the database.

Text

Description automatically generated

*IDE – Max data entry, error identified*

From this screen shot above, the program crashes when the user attempts to edit the meal item that has all of its data at the upper boundary limit for entry. This is because of this “-” at the end of the text for the ‘meal\_name’ variable highlighted in the 2nd screen shot.

This leading “-” hasn’t been removed because when the chosen meal item to be edited is highlighted in the list box, the whole string is recorded into a variable. An example string would look like – “M – Chicken Wrap – 7.99 ”. The data is then supposed to be trimmed such that only the name of the meal is stored in the following variable. This trimming method used only works (i.e. removes the leading “-”) when the price is 3 digits e.g. “7.99” as shown above. When the max data meal item was initially entered in Test 1, the price was set to “999.99” which exceeded the normal format expected to be used.

Due to this, the trimming method doesn’t remove the “-” at the end of the meal name and therefore causing the program to crash. To resolve this issue, I would instead split the string into its own array of the relevant fields in the string using “-” as a delimiter. This way I could simply access the name of the meal without having to trim the data and account for varied prices. As a result, this allows the manager of the restaurant more flexibility deciding their own prices rather than being limited by the program.

### Graphical user interface Description automatically generatedTest 3 Evidence

*Ingredients Window – Data input*

Graphical user interface

Description automatically generated

*Ingredients Window – Ingredient item saved*

I expected this test to cause the program to crash due to the data entered being larger than the field size specified in the query when creating the table in the database. However, when the save button was clicked after the data was entered, the ingredient item was added to the list box and to the table in the database. This meant that the program didn’t crash and continued to work as normal.

I created a dummy table like this table used in my project and attempted the same test. I found that the data was still recorded in the table in the database. I then researched and found that the issue was due to a bug in the version of SQLite3 I was using within my project, rather than the code.

### A screenshot of a computer Description automatically generated with medium confidenceTest 4 Evidence

*Take Order Window – 15 meal items in customer’s order*

*Take Order Window - Receipt*

Graphical user interface

Description automatically generated

The above screenshot shows 15 meal items in a customer’s order. This would be rather extreme as typically customers have up to 7 meal items in their order.

### Test 5 Evidence

Chart

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

*Graph – Scale is large Graph – Scale is normal*

These screenshots show that the ‘Quantities’ scale can grow large, and this doesn’t affect the quality and usability of the graph for the user.

# Evaluation

## Overview

Overall, the new system works very well. The system has streamlined the ordering process from the customer’s table through to the kitchen / bar with the ability to track and report stock levels. In comparison to the outdated pen and pad method, the new system has a clear, user-friendly interface which makes it simple for the manager to have access to everything in one place. I am pleased with the whole program and the use of python in conjunction with SQL and in particular, my use of more complex algorithms which use various SQL queries to manipulate link tables and the general flow of logic to provide an elegant solution to this problem.

Based on my extensive design and analysis, I was able to outline several key requirements that have all been implemented, along with a few additional features that have been thoroughly tested as shown in the Testing section above.

## Review of Objectives

Below are the system requirements from the analysis section, with a review of how they have been achieved:

1. Each user must have a secure login which the program uses to identify the user’s role throughout the program respective of their job title.

* Each of the users have a hashed password which is saved to further provide security to the login process within the program. A global variable is stored of the users job title.

1. The program must store details of the manager and staff, each with a unique ID, position in the business, access rights and other relevant fields depending on the user.

* Details of the staff are stored in a database including their job titles

1. The program must apply adequate access control to different type of users of the system. For example, admin users should be able to access all areas, waiters should only be able to take orders, kitchen staff should only be able to update level of ingredients available and head chef should only have access to editing the menu.

* The job titles are taken from the database and used to apply adequate access control within the program.

1. Waiter must be alerted if an order cannot be fulfilled before committing it to the kitchen. The program will display a message if stock is below a threshold, in which the customer will be informed to choose another meal item.

* When the waiter takes an order from the customer, if an item is low in ingredients a message box shows and alerts the waiter that there aren’t enough ingredients to cook the meal. This allows the customer to reconsider their order.

1. The program should set a stock re-ordering threshold for each ingredient, if an item falls below this the system should automatically add the item to a re-ordering text file.

* The threshold is entered in the program when an ingredient is added which sets the level in the table in the database for individual ingredients. Therefore, the program cross checks this level with the current level of stock and will add the ingredients that are low in stock to a text file.

1. The program must display the stock re-ordering text file for the admin to make any changes before committing the order to replenish stock.

* The contents of the text file are displayed in another form in the program for the manager to view.

1. The program should display information about the individual suppliers that the business deals with and should only be accessed by the manager.

* Information related to the supplier is pulled from the database and displayed in the suppliers form which is only accessed by the manager.

1. The program must create a ticket of each order taken which is sent through to the kitchen / bar.

* After the waiter has taken the customer’s order, the order is saved to a text file but also sent to another form which is what the kitchen staff view when cooking. This digital method was used to replace the old style of handwritten food tickets.

1. The program must provide a receipt for each order taken and save a similar copy for the restaurant.

* After the waiter has taken the customer’s order, a receipt is shown digitally on the screen and the receipt is saved in a text file along with the name of the waiter, time and day.

1. The program must track ordered meals and display this on a graph for the manager to see the most selling meal item on the menu.

* The data for the ordered meals is collected and plotted on a graph which is on a separate form for the manager to see.

1. The ‘Kitchen View’ window should have a function to click each order that has been completed and remove it from the view of the Chef, simulating the use of a written ticket.

* There is a button on the ‘kitchen view’ window which allows the chef to select the order and click complete to remove it from the list.

1. In the ‘Supplier’ window the admin user should have the ability to search through each supplier by name.

* There is an input field for the user to enter and search through the supplier contacts. Their information is then shown in the relevant input fields below.

## User Feedback

This meeting took place on the 18th March 2021 with the owner to discuss the completed program. The owner seemed very pleased with the design of the program and the functionality overall and the requirements we discussed were implemented fully with a few additional features. The ease of use of the system seemed to be compatible with the workflow of the restaurant.

The staff found the program to suit their needs when tending the customer and managing the restaurant. They particularly mentioned the way the data entry form for the menu and ingredients were presented and the ease of use of taking a customer’s order. They appreciated the helpful hints which were on the input fields as ‘pre-text’ which told the user on that form what type of data was to be entered in that box.

The manager was very content with how the system managed the stock levels and updated the manager weekly when an ingredient was low in stock. He also appreciated how the program alerted the user (waiter) when taking the customer’s order if an ingredient was low in stock. This saved the waiter having difficult conversations with the customer had they gone back and forth from the kitchen. This also would improve the overall reputation of the business.

The owner had suggested a few improvements to the system:

* In the stock replenishment form, an extra option should be added which allows the manager to manually search for an ingredient and add it to the re-ordering list.
* An email could be sent automatically after all the items to re-order are listed.
* When an ingredient is out of stock, the meal can’t be prepared, and the waiter is alerted but all the remaining orders in the list of the customer’s order are cleared from the list box. It should only delete the meal that can’t be prepared so that the waiter doesn’t have to ask the customer to repeat their order again.

## Improvements

Based on the user feedback above, I have identified the number of changes and extra features that I would add to the program if I had the opportunity to re-code the system again. One of these would be adding a textbox which allows the user to enter an extra ingredient which is added to the re-ordering list. To do this I would code a search feature which gets the name of the ingredient from the table in the database. This would then be simple to add to the list box by adding an extra button to the form which puts the ingredient in the list box. I would then add another feature to this by generating an email format with the ingredients in the body of the email and send it off to the manager. This would be simple to code as I would use SMTP to connect my code to an email account and verify it. I would then take the text from the list box and insert it into the body of the email and with the manager’s email address, send the email.

Another improvement I would make is not clearing the list box of the customer’s order after a meal that can’t be prepared has been identified. This would be a very simple improvement which would be to remove a couple lines of code from the program.