

TripleV

John Lawless

Bachelor of Software & Electronic Engineering (Hons)

Galway-Mayo Institute of Technology

2019/2020

**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Hons) in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

John Lawless

**Acknowledgements**

I would like to thank my mentor on this project, Mr. Paul Lennon, his advice, and support have been invaluable through all stages of the project. Through our weekly meetings, in person or virtually, I was able to update Paul on my progress or lack of and he in turn would offer support, guidance, and on a lot of occasions, direction. Paul kept my head in the game and recognised when the pressure was getting to much. His advice was always heeded and without his support, in my opinion, this project would not have been completed.

I would like to thank Brain O’Shea for his suggestions and support with NodeJS and express. This change of direction helped me to kick start the webapp development in the right direction.

The acknowledgements cannot go by without mentioning Des O’Reilly and Mairtin O’Conghaile for supervising the project engineering labs each week without fail.

I would like to include a huge thank you to one of my best friends Noelie Byrne. Noel is a highly experienced network engineer and network designer. His advice on how the system should be designed was instrumental in completing this project. It led to the complete redesign of the form and function of the whole system.

Last, but definitely not least, I would like to thank my wife, Leigh, and my children, Paddy, Eilidh, Mickey-Joe and Shea. The last four years have been a sacrifice for all of you, but you all have given me such support and I could not have done this without ye. The last number of weeks of college have been the most difficult. Schools and colleges closed due to the Corona virus. We were all trapped inside our little house. With out your support and allowing me the time I needed and having patience with me this project could not have been completed. Thank you all and I love ye very much.

**Table of Contents**

[1 Summary 5](#_Toc39157762)

[2 Introduction 7](#_Toc39157763)

[3 Planning 9](#_Toc39157764)

[4 Face Recognition System 11](#_Toc39157765)

[4.1 Raspberry Pi 4 11](#_Toc39157766)

[4.2 Pi-Camera 11](#_Toc39157767)

[4.3 OpenCV 11](#_Toc39157768)

[4.4 Haarcascades 12](#_Toc39157769)

[4.5 TkInter 13](#_Toc39157770)

[4.6 Development of project for December demo 13](#_Toc39157771)

[4.7 Developing and testing the GUI 16](#_Toc39157772)

[4.8 Integration of Face Recognition into GUI 18](#_Toc39157773)

[5 Web Application 22](#_Toc39157774)

[5.1 Overview 22](#_Toc39157775)

[5.2 Changing Direction 22](#_Toc39157776)

[5.3 Design and Development 25](#_Toc39157777)

[5.4 Moving to the Cloud 31](#_Toc39157778)

[5.4.1 Setting Up Amazon EC2 Instance 31](#_Toc39157779)

[5.4.2 Accessing Instance through PuTTY 32](#_Toc39157780)

[5.4.3 Finishing the Connection 33](#_Toc39157781)

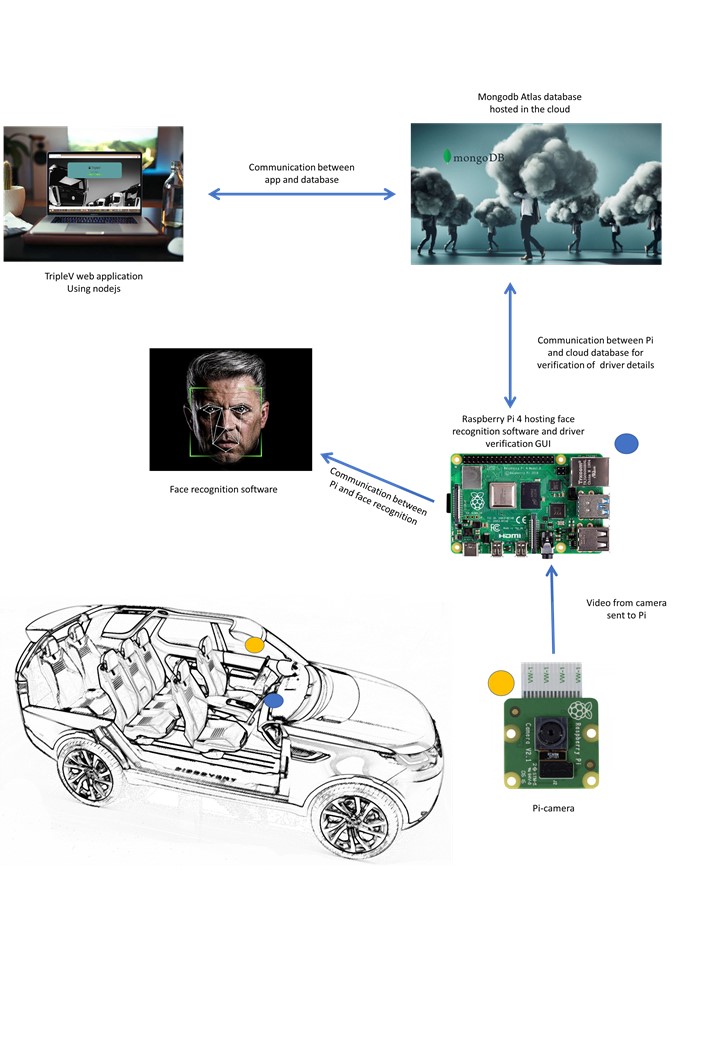
[6 Conclusion 34](#_Toc39157782)

[7 References 36](#_Toc39157783)

# Summary

Vehicle Visage Verification (TripleV) is a fully networked, cloud-based fleet management system, for the modern automotive industry. The inspiration for this project came while on placement with JLR, Shannon and observing how difficult it can be to monitor users and keep track of multiple vehicles, with multiple users, in a large fleet. From the outset this project has implemented the Agile project management style of project development. The project has been developed in small sections. Each section was tested and verified as working before the next section started its development phase. When all sections were complete the integration process started and continued until all parts of the project were integrated, tested and verified. As problems arose within the development of the project, they were investigated to find the cause and ideally to find a workable solution to address the issue. At times this meant a complete rethink and change of scope and/or functionality for the project.

What is the function of TripleV? It is a management system for fleet vehicles. The function of TripleV is to allow a fleet manager to have full control over who is registered on a vehicle and therefore who has access to a vehicle. This is achieved with the use of a cloud-based login/register application, a Mongodb Atlas database and face recognition software on a Raspberry Pi with a Pi-camera attached. The object is to allow a driver to login to the facial recognition software with a secure key provided by the administrator and allow them to register their profile in a dataset on the system of a particular vehicle.



**Project Layout**

# Introduction

At the inception of TripleV, the plan was to produce a cloud-based fleet management system for the automotive industry. The goal was to produce a system that could easily and securely manage the operation of small/large vehicle fleets. The scope of the project was not only to securely register drivers on vehicles, but in turn secure the vehicles themselves. Each driver’s profile is built through a dataset of images taken and stored within the vehicle. This dataset is used to provide positive images for that driver for the application of the face recognition software. Only through the recognition of a registered driver can the vehicle immobiliser be deactivated.

The only hardware in the system is a raspberry Pi 4 4GB (Pi) minicomputer with a Pi-camera attached. The Pi-camera is an add on board for the Pi consisting of an image sensor and a fixed focal length wide-angle lens allowing the capture of video or still images. The face recognition software is held on a Pi inside the vehicle. The Pi also holds a dataset of images of each driver registered to use the vehicle in which the Pi is fitted, along with a file containing the identity of the individual driver. This information is held locally for ease of access and to help speed up the face recognition process.

The driver of the vehicle must first have their details registered on the TripleV database and be provided with a secure login key and password. This registration can only be carried out by a fleet manager, registered on the TripleV web-application. The details stored on the database include the drivers name, a secure username, a secure password, registration number of the vehicle and a unique secure passcode. All sensitive information would be encrypted before being stored on the database. All details are stored in the cloud on a Mongodb Atlas Account, this allows for easy access from any vehicle, anywhere. Only upon a driver entering these details correctly can they proceed to register their face in a dataset held on the vehicle. Once the system learns the face of the new driver it runs the face recognition software. If the driver is authenticated the vehicle immobilizer is disengaged and the driver is cleared to use the vehicle.

A close up of a device

Description automatically generated

# Planning

With little to no experience in most of the modules that would be needed for the project to function the planning of this project was quite difficult. When needed advice was sought from lecturers and experts in my own personal network. At times this led to a complete change in scope of the project. This project has implemented the Agile project management system from the very start. Initially sprints of a period of two weeks were set for the completion of assigned work. This was the practice observed while on work placement. After several weeks were lost due to a serious illness, the period of the sprints was reduced to one week and more time was allocated to completing the weekly tasks. With the lack of knowledge on topics within the project it was very difficult to plan a concise timeline from inception. The project turned out to be very organic. The understanding of one topic led to a push to develop it further and a change to requirements of the project. Equally some ideas for the project were removed or reworked after seeking advice.

Initially, at the design phase of this project the face training module of the face recognition software were to be taken online and made part of the Web Application. After spending many frustrating weeks trying to implement control, remotely, over this module on the Pi and taking advice from lecturers and trying to use many different methods, .cgi files, Flask and NodeJS, in particular, a consultation was held with an experienced Network Engineer on how he might implement a system like this. He is a network systems expert with over 20 years’ experience in this field. He advised that implementing the system in the way it was planned was over complicating it and suggested not to build the face training into the web app. He recommended that all information and data concerning the face recognition software should be held locally on the Pi. The system should be secured with encrypted passwords and keys stored on a remote database accessible from the Pi. This would go to ensure only authorized users could register on the system. This led to a complete rethink of the scope of this project.

The start of the second semester was the planned start for developing the web app. As a module in Java networking, using Servlets, JSPs, MySQL and a Tomcat server had just been completed in semester 1, the development of the web application was started using these resources. This lead to a number of problems and again after seeking advice it was explained that a NoSQL database would be a better option, that networking in the form that was being developed was defunct and that developing a MEAN stack application utilizing NodeJS, Mongodb and Express should be assessed. This completely changed the scope and requirements of the project again. The timeline again was in disarray. Time with family, family events and weekends were all sacrificed until the project was back on track again.

The one of the biggest changes in scope and a positive that came from the changeover to node and mongo was the ability to push the database to the cloud. This allowed access to the database remotely from any IP address registered on the databases security settings. This meant the Pi could retrieve login information for authentication, “easily”.

# Face Recognition System

## Raspberry Pi 4

At the heart of this project is a Raspberry Pi 4 4Gb (Pi) minicomputer. The Pi is a low cost, credit card sized motherboard. It contains a Quad core, 64-bit, Arm CPU, 4GB of SDRAM, built in WI-FI module…., the list goes on. The hard drive for the minicomputer is a Micro-SD card, up to 32GB. The SD holds the Debian OS and any other libraries or applications loaded onto the Pi. The model of Pi was chosen to run the facial recognition software, essential to the success of the project, because of its upgraded CPU and its 4GB of RAM. The Pi was also chosen based on the amount of support and tutorials that could be accessed online. From inception, it was understood that this project was going to be a journey of self-learning and that it would be essential to use hardware that was well supported.

## Pi-Camera

The Pi-Camera is the only other piece of hardware in this project. The camera used for this project was the upgraded Pi-Camera V2. This model contains an 8-megapixel Sony IMX219 sensor and can capture high definition video with its 170-degree fixed focus lens. This camera is dedicated to the Raspberry Pi (Pi) and connects directly to the camera module port on the Pi board.

## OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. [1] It contains a vast library of more than 2500 state of the art computer vision and machine learning algorithms and is aimed at real-time computer vision systems.[2] It is open source, available to all for free, and cross-platform. For this project the OpenCV libraries are used for the face recognition algorithm. The algorithms of OpenCV are quite complex and relatively computationally expense. This makes them high cost in CPU performance and memory usage. The Pi is short on both and although the face recognition software can run relatively fast when running independently, when running concurrent with other algorithms the performance drops off quickly. Leading to the system struggling to recognize a registered user.

The installation of OpenCv on the Raspberry Pi should be straight forward and not cause any issues, are so every tutorial would have you believe. A considerable amount of time was lost on trying to follow these tutorials. Each time the step-by-step guides were followed, exactly, only to have OpenCv build, stall and crash at varying points, between 80%-99%. This was very frustrating as it was taking 3 hours or more for the process to complete. At times the process would be run over night, with the hope that the extra time might complete the process, but this had the same result. The conclusion was reached that downloading an OS image with OpenCv preinstalled was the only option. It was at this point that an article by Solarian Programmer was found.[3] Within his article he had a guide on installing the newest version of OpenCv on the latest Debian(OS for the Raspberry Pi) release. The guide on this site was excellent, easy to follow and installed OpenCv in under an hour.

## Haarcascades

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001.[4]

Haar cascades is a machine learning approach to object detection. It is used to identify specific objects in an image, in the case of this project it is used to detect a face. To develop a Haar cascade you must first have thousands of negative images. A negative image is an image that does not contain the object you are trying to detect, the more of these negative images you have the better. You also need thousands of positive images, images containing the object to be detected, for this project that would be a face. These positive images can be based on one image, but it is always better to have thousands of different images of the object you are trying to detect. It is good practice to have double the number of positive images to negative. You can create positive images manually, taking them yourself and building up a dataset or you can download datasets from various git repositories like “Flickr Faces”, which has a dataset comprising of 210,000 images of faces. These images are used to train the classifier. The classifier categorizes the image into the user determined class to which it belongs, e.g. face, eye, apple and so on.

This is only the first and easiest step in creating a Haar cascade from scratch, so fortunately there are many git repositories containing Haar cascades for faces and OpenCv contains a library of its own Haar cascades. The majority of the Haar cascades in the OpenCv libraries were developed by Intel.

## TkInter

TkInter is the most used GUI package for Python. It is the fastest and easiest way to develop a GUI in python.[6] TkInter is imported into Python code the same way as any other module is imported. This package was used to develop the log in GUI on the Pi. This was one of the final developments in the project and pulled a lot of the other elements together.

## Development of project for December demo

* **Face Detection**: it has the objective of finding the faces (location and size) in an image and probably extract them to be used by the face recognition algorithm.
* **Face Recognition**: with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image.[7]

Many tutorials were gone through during the research phase of the face recognition end of the project. Most were found lacking in functional requirements. The tutorial found most helpful and followed to get the base code operational, was on Hackster.io.[5] This tutorial was in-depth and relatively easy to follow. There were early successes in implementing the code written by following the tutorials directly, but each time the code was manipulated, adjusted or customized to the way it needed to preform, it broke. Initially, the objective was to change the code to allow for multiple user profiles to be registered on the system but the changes, again, broke the code. Exhaustion and frustration at making no progress were creeping in.

Due to a serious illness, a stay in hospital and a recuperation period at home, weeks were lost from the project. On returning to college, it was an uphill task to have anything to demo for the December deadline.

Although there was an allocation in the development timeline for unexpected delays it was impossible to have planned for the weeks that had been lost. Other modules were sacrificed to get working code for the December demo. The project code was taken back to grass roots with the Hackster.io code and it was went through line by line to get a better understanding of how the code worked. Slowly the code was changed to allow the functionality planned. For the December demo there was a very basic system on the Pi that would allow for the user to determine how many users they wanted to register on the system, save their names into a text document and fifty images of their profiles were saved into a dataset with a unique user id.

The face recognition software is divided into three modules, data gathering, face training and face recognition.

A screenshot of a cell phone

Description automatically generated

Diagram of the face recognition software

For the face recognition system to recognize a face it must first be able to detect a face. This is done by using the Haar cascades as classifiers, they classify the object that is required to be detected. The system takes in the user’s name and writes it to a text document, it also assigns an id to the user based on the index of the user’s name in the text file. When the system detects a face, it captures fifty images and saves them to a dataset folder. These images are captured in grayscale, for training and recognition it is computationally less expensive if the image is a grayscale image. The getImagesAndLabel() function in the face training code takes the dataset and divides it up into images and ids. The ids should correspond to the index of the usernames. These are stored in a .yml file, this file format is specifically designed for holding data in a plain text format. The final phase of the face recognition system is determining if a newly captured face has been registered and trained and if the system can recognize it. The software compares the input image to every image in the .yml file, the objective is to find the image that best matches the input and thereby find the user’s identification.

Face recognition algorithm is computationally expensive as it goes through every image in a dataset to find the one that most closely matches the input image. In Big 0 notation it is 1xN, the more images in the dataset the longer it takes for the system to do its comparisons. This uses a lot of CPU power in the Pi and leads to overheating problems. The December demo didn’t go so well, while setting up for the project and testing the system the Pi overheated and burned out the camera module. A camera was borrowed from another student, but even then, it was clear that the Pi had also been damaged.

At this point a new Pi and camera were ordered and the Pi end of the project was put aside so that work could begin on the web app. It would be early March before the project returned to developing a GUI on the Pi and integrating the face recognition, GUI and cloud database.

## Developing and testing the GUI

The plan for the Pi was to develop a Graphical User Interface(GUI) to allow a user to securely log into the system and register their profile with the face training algorithm. The user would be registered on a database through the web application. They would be given a username, password and a unique code to log into the system. The registration number of the vehicle would be hardcoded into the Pi. When the user logged in the system would first check that they were registering on the right vehicle, that the code was correct, username was correct, and the password was correct. All these checks had to pass before the user could register a profile.

Before returning to develop the code directly on the Pi, JupyterLab (JL) was used to write Python code and to test the GUI and Mongodb code. JL allows you to run Python in a browser. Here different pieces of code from tk tutorials could be experimented with, learned from and an understanding of how to best design the login GUI could be decided. It allowed the installation and import of the components needed to test the features proposed for the Pi GUI. The JL environment also allowed code to be tested for connecting the Pi to Mongodb Atlas for verification of usernames, passwords, etc.… Early tests involved unencrypted passwords. They were simply to show that data could be written to and retrieved from the database. The held belief was, as both the Web app and the Python code were using Bcrypt for encrypting and decrypting data, it would be an easy step to change to checking encrypted data. It was here that another problem arose, the two environments had different methods of using Bcrypt. Trying to find a solution for this was quite difficult. There was no information online and no easy solution. Again, this was a time where a lack of knowledge left the project in limbo. The solution was to look deeper into Bcrypt and how it hashes the passwords in both environments. The biggest difference was that in Python the password is converted to binary code before it is hashed and in JavaScript this did not seem to be the case. It was observed that the JavaScript code was encoded in UTF-8. So, the solution was to encode the Python password before checking it against the encrypted password registered on the database. This seems very simple now and the resulting code looks very simple, but it took a lot of learning to reach this solution.

![A screenshot of a cell phone

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDyRXhpZgAATU0AKgAAAAgABAE7AAIAAAANAAAISodpAAQAAAABAAAIWJydAAEAAAAaAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGpvaG4gbGF3bGVzcwAAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzE2AACSkgACAAAAAzE2AADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIwOjAzOjI0IDExOjQ0OjI0ADIwMjA6MDM6MjQgMTE6NDQ6MjQAAABqAG8AaABuACAAbABhAHcAbABlAHMAcwAAAP/hCx9odHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIwLTAzLTI0VDExOjQ0OjI0LjE2MTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5qb2huIGxhd2xlc3M8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgBBgJuAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKxPF8c7+GLmS0+2GeApOgsnkWVtjBiAEILZAPy87umDXL/wBsSXEfiqSyh8UWsE8MTW0slhdu4Zso7wxPhgASCY12NhSygZDFDPQ6K8n0qx8SL4V8Sf2dqOrzmQ24gnuba+jZUB/frFBcymfdsJO4SjcSAu0rmrmgLNb+GdeTUdX17ULUpGYok0jVbWeFzkZjeV3nfJ25CNhcdAGOXLRN9gSvY9Mory1YS3w3uLqLUfGkmptHAk7yQ3qTiYEHdHF5Y4AJB2JsbHz7jzV2d7jTbHxHpejp4gniEdu0Ul39vlaPf8srxyt+8fauG8uN92QQu0kkOWjaJi728z0WivGIJdTbRbnTru68WSRrqto1rc22n6nbv5DFRNgu0kpXAkyJG4OCAPkJtxReIodH1C2sJ/EAu7iKRWku47ubYkV0V/d7mXDG3xgo6tJ1DM3zUv6/C/8AwPUf9fjY9UvbGK/iWOd50VW3AwXEkJz9UYEj2qKy0m3sJjLBJeMxXaRPezTLj6OxGeOvWuV+HUd5D/aMdxrGoahBmNoor3Sb+18k/MGKveO7yZwOA2Fx0G7nqNbktotFuTfi9NuV2v8AYFnafBOPk8j95nnqvI61bqTjHlT0NI1anLyJu3YtW1zBeW6z2k0c8L/dkicMrc44I461LXmS6nLefCmW1/4qqDULVlXzVsb9LoZlO0hmTdLhRz94dN3WtG2FrZJ4j0tz4ml01IIZEaQahNKWZcMIpjmRv4MqjHHzcD5qzbsmzNatHVaZ4j0TWrie30fWNP1Ca3/10drdJK0XOPmCkkcg9fStKvNPDkItPhisWkJ4kt7uO3tI9QS6S986NRtEwt1uAeQu/AiHpjnbW/4PKf2hfjS21s6T5cRQa0t15gmy2/YboeYV2hM9Vz053VbVnb+v+GDpc6yisjxLb6zc6T5fh24+z3nmA7/PSH5ecjc8Ew9ONn4iszwvYeLbXUJX8S3/ANptjFhF+2xTYfI5wlnARxnncfp3FRp3i5XX6ibsdVRWJ4vjnfwxcyWn2wzwFJ0Fk8iytsYMQAhBbIB+Xnd0wa5f+2JLiPxVJZQ+KLWCeGJraWSwu3cM2Ud4YnwwAJBMa7GwpZQMhjkUeh0V5PpVj4kXwr4k/s7UdXnMhtxBPc219GyoD+/WKC5lM+7YSdwlG4kBdpXNXNAWa38M68mo6vr2oWpSMxRJpGq2s8LnIzG8rvO+TtyEbC46AMcuWib7Alex6ZRXlqwlvhvcXUWo+NJNTaOBJ3khvUnEwIO6OLyxwASDsTY2Pn3Hmrs73Gm2PiPS9HTxBPEI7dopLv7fK0e/5ZXjlb94+1cN5cb7sghdpJIctG0TF3t5notFeMQS6m2i3OnXd14skjXVbRrW5ttP1O3fyGKibBdpJSuBJkSNwcEAfITbii8RQ6PqFtYT+IBd3EUitJdx3c2xIror+73MuGNvjBR1aTqGZvmpf1+F/wDgeo/6/Gx65RXEfDqO8h/tGO41jUNQgzG0UV7pN/a+SfmDFXvHd5M4HAbC46Ddzoarqd5q/gd72ws9UsDJIomhaFo7tIBKFlKKuWDGMMV2/NyCvOKbTSuC1djp6K8phS8ubLxUvhW/8QjyltUgTUzdtLFETum8uOV0lJK7sMSHJ4VgAuJtJ0i91PwZ4n0yXVtX1JZ7X/RoJrDUdPeOTY3Cy3UrSSBiFyofaMcjDHKGldo9QqrLqunwJdvNfW0aWIzdM8ygW427vn5+X5SDzjg5rE8GSWRtZ49LtdfjgUoTNrTXWXcjlUFyxkAGBnACHPBJDYx/Bv2L/hD/ABN/b2zzP7S1D+1/N6Y3Njdn+HyfLx/s4rVQVm30/r+vkTF3t5s70HIyORRXNfDr7b/wrbQP7T3/AGn7DHu8zO7GPl3Z5ztxnNdLU1Ick3Dswi7pMKKKKgYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFcx8Q7++0rwZcahpeotp1xbzQnzhAsy7TKqsGQqSVwxOFw3HBFc6fFbW2jay+leLh4itYbeIrqayWn+hzSOUw8scflKgGHJZGKAEkMCooHY9Jory3w/47uJfC/iUX3ijSZrnT9n2fUBdpcRoJAFVmkSCNSokJG8RFV5zu2kUaB4q1QeGvE17qHiywu4rVIktNRW4hubWB3BHzXEUEYJ3FdwMWEBUncCRQ9L/eJa2+49SorybSNf1660DW72bxct1badPay/bdOltbuNYckzr54t0VmCfMV8sFQEwx3GtCLxXrd9Bq+qPPNpEVtFbQT2syxyxWDSPl592xWO2Fo3+YlRuORhadun9f10/wCAC1/r+v68z0mivP7XxFAdJ10w+PEv9NtreORdc82zzaysWHlF1QQnO1MApkb+c7lxW03xFeT+B31mPxxa6mZRbpe3FtDAYtL3Mu9wFztKo2T5pcAjccKCtL/gfiH9fl/n/TO71WwudRtlitNXvNKdX3GazSFmYY+6fNjcY78DPHWq2laNfaddNLd+JdU1VChUQ3kdqqKcj5h5UKNnjHXHPSuT8Bahplx468RR2HiO0195bW0kF3GIBJMFMoYs0IVJCuVXcFGAVU8jJ7TW7q2s9FuZ77Vl0aBVw1+0kaeRk4BzICmckD5gRzRtqPd2L9FeW2njGw/4VC8+k+L9Ntb2xZVe5ha1IjLTEL5kYAQBgCcAKTzgjrWtba7Bbp4j0q+8ciU2EEM/9oyy2iTWokXvtjEYGQuCyfx9TxSbsm+wlq0u53lFeReCV0XRfhzJq/hnXLC6v57e0GpXqQ2bNYbseY8hhjRmKhnY+aWOVyf4s9j4P1db/UL+3sfEy+KLCKOJ1vw0DGORiwaItAqoeFVsbdw3ckgri2rOwdLnWUVkeJbjWbbSfM8O2/2i88wDZ5CTfLzk7XnhHpzv/A1meF7/AMW3WoSp4lsPs1sIso32KKHL5HGUvJyeM8bR9exqNO8XK6/UTdjqqK5j4h399pXgy41DS9RbTri3mhPnCBZl2mVVYMhUkrhicLhuOCK50+K2ttG1l9K8XDxFaw28RXU1ktP9Dmkcph5Y4/KVAMOSyMUAJIYFRWZVj0mivLfD/ju4l8L+JRfeKNJmudP2fZ9QF2lxGgkAVWaRII1KiQkbxEVXnO7aRRoHirVB4a8TXuoeLLC7itUiS01FbiG5tYHcEfNcRQRgncV3AxYQFSdwJFD0v94lrb7j1KivJtI1/XrrQNbvZvFy3Vtp09rL9t06W1u41hyTOvni3RWYJ8xXywVATDHca0IvFet30Gr6o882kRW0VtBPazLHLFYNI+Xn3bFY7YWjf5iVG45GFp26f1/XT/gAtf6/r+vM9Jorz+18RQHSddMPjxL/AE22t45F1zzbPNrKxYeUXVBCc7UwCmRv5zuXFbTfEV5P4HfWY/HFrqZlFul7cW0MBi0vcy73AXO0qjZPmlwCNxwoK0v+B+If1+X+f9M9Jpk0MVzBJBcRpLFIpR43UMrqRggg9QR2rz/wFqGmXHjrxFHYeJLTX3ltbSQXcYgEkwUyhizQhUkK5VdwUYBVTyMnc1XxPDd+B31rw/ehbaSRYzfGIgW0fmiOWba4/gG9vmGPlycim07XGt7I2tL0bTNDtDa6Jp1pp1uXLmG0gWJCxwCdqgDPA59qu15aniPVm0vxNceH/FqeIobFbaO3uiIAkKu2ZZGkhhIZkXJyFKgKMoSGy7SdW1zWvBnieFPF2n6hNb2vmWtzompQ3l1E2xiVcpbxoASoC4Tdy3zZwQgir28z1Cqsul6fOl2k1jbSJfDF0rwqRcDbt+fj5vlAHOeBisTwZqtjqNrOun+KZ/EwjKF7p1hZIyR9wSQxohIxkrywyM4BFY3hKytdd8IeJpNc+aTVNQvotRDn/VqjNCqewWJEx+fetVGycm9iYu9n3O+AwMDgUVznw+v7zVPh3oV7qTM9zNZRs8jdZOOHP+8MH8a6OpqQcJuD6OwRd0mFFFFQMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAM/W9DsfEOnfYdUE7QeYsmILqWBtynKndGytwcHGeoB7VcghW3t44Yy5SNAimSRnYgDHLMSWPuSSe9YXjmGWfwrKLe7ubKYTweXPazNG6EyquflIyME/Kcg9xXKWGs3drr/AIgufMlbVIIrG0ltrm+ma1t7iaV0MgjLlUQgRuMAHaQMgsaFvYdup6ZRXE6lrfinSrHVLd59Hu9RtIoLmGZbOWKF0kdk8t081ipyhw4Y9fu/Lyl/4l8Q6Ho2si+Wx1LUbM2/kSWlubeMiZtg3JJMR8pySfMUEcZXrQLrY7eiuM0iLxH4nsbqw8ZWV7pKRSxSQ3VjOtm9yOd0ZENxKyAEDJ8wbg3QYOaXhnTdUs/ALavoWo3t3q1xaAiLVbue9iZkZj8qPICrMPl4YDheOuTbcN9j0CiuA17xhrcOm32s+H7vR5dLtEtVU3dvIokklZd4MglARQkiHO04JIOcGp9V8S+I/DthqUV+thql9FDBLby2dubdF82XysSJJMRgH5s+YoYAj5cbi2mgWup1WqxarNbKuh3tnZz78s95ZvcKVx0CrLGQc45yfpVbSrbxHDdM2t6rpd5b7CFjs9Mkt3DZHJZp5ARjPGPx9aXhDUfE96LyPxXpH2IxFDb3GyKLzgQdy+Wk823bgcl+d3QY5u+JdXutE0n7XY2P26XzAnlbZzwc8/uYZW7f3ce4pqLbsuobmvRXK+F/FWp67qEtvf6L9gjSLeJNl6NxyBj9/aQr37MT7dxpeLbjULPwfqt1o9zDa3lvaySxyzQeaq7VJ+7uXnA4ycA8kHoXOEqbtJBH3nZGxRXLm78RJoN/FBq+i6hrsMKXEUUNi6KqHOA8f2gsS21grblGR7GsjXvGGtw6bfaz4fu9Hl0u0S1VTd28iiSSVl3gyCUBFCSIc7Tgkg5wamzvYSd0md/RXD6r4l8R+HbDUor9bDVL6KGCW3ls7c26L5svlYkSSYjAPzZ8xQwBHy43HU8Iaj4nvReR+K9I+xGIobe42RRecCDuXy0nm27cDkvzu6DHKGamt6HY+IdO+w6oJ2g8xZMQXUsDblOVO6Nlbg4OM9QD2q5BCtvbxwxlykaBFMkjOxAGOWYksfckk968tvrzUtW+G+ta0urX1peT675dsYLh0+xol0tsqBQQCNoLFTwWYkjpiK28VXul+IPEl3cNO19ZwWFrNbXN7L9kt7iad42m2klVjIEb8DheOCTXrUssnVpOcJa3tbz91fdeVr/5oiU7b9P8z1uivP8AWfF/iDQ7fXbWR9LvL/S7WC8WeG3kSNkkcr5TxmQlH+UkNvIIOcccmteOtY8OR6zBf21peXdqto9u1upijQXMjxgPvfB2FCS25A2R9zrWcctrzty2d7W178uuv+JbhzpXv0PQKK4jStS8Qa9HrWg+Iohpdx9k3Q3VtLFBOgcMu7yo7iZkwRkPuAPTAxzs+BtZuPEHgTR9UvQBc3NqjSkDAZsYLAehIz+Nc1bDypNptO1tndaq6/Ialc3qKKK5igpk0SzwSQuXCyKVJRyjAEY4YEEH3BBFPooAztI0Kx0OOVbAXDNMwaSW6upbmVyBgAySszEAdBnA5x1NaNFFABWXL4a0ma21aB7XEWskm+VJXXziUEZPB+UlVAJXGcetalFNSa2AbFFHBCkUKLHHGoVEUYCgcACnUUUg2CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAqanpGm63Zm01nT7XULYsGMN3CsqZHQ7WBGaRtI014Z4X0+1aK4hW3mQwKVliUEBGGOVAYgA8cn1qHXtTn0jSWvba2juTHIgdJJjH8pYKSCFbJGcgYGfUVSl8QX0Da6j6UJpNMiWW3jtZmka5VlYjK7AVbKn5V3+2TxS0sx63RLH4M8LxaXLpsXhvSEsJnEktqtjEIpHHRim3BI9SKsxeHtFhuLqeHR7COa8j8q5kS1QNOmMbXOMsMcYPGK5Kz+JN4+jaxeah4fkt5tPaCOK3jNwz3DzNtX929ukyruwN3lHPzbdxUir+i+MdV1jSdUk/4R2S0vrFVeNLr7Tb29wCCfllmt0fI2nI8sgZXk54b0TbBJu1jWtPB/hmws7q0sPDuk21teKEuYYbGNEnUZwHULhhyevrUlv4c07TNJn0/wAOQQ6Akx3b9MtoYyr8DcFKFCcADJU1X8Nazq2t2qXepaIumW1xCk1v/pizSEHqHVVAU9CNrNkHnaeKm8S+I7XwvpP9oXyb4vMEePtMEHJz/FNIi9um7PtVcsnLl6vQlNPVGZJ4Dt5PBv8Awjh1fUBA1wbh7gR23myMZTKQw8nYQXOT8mT3PWtPSPC2iaHpT6fp2l2MMEyBLkRWkUQufl2kuqKqkkZzxjnpjiszwv8AEDTPFmoS2dhFskji80n+0LKfjIH3YJ5GHXqQB71ua1qf9kaVJdrbvcybkiigRgpkkdwiLk8DLMOT0606kJU3yyQ077EekeG9D8P+b/YOjafpnnY837FapD5mM43bQM4yevqa0q5hvEWv2NvdNrXhyGJ08qO0NnqImjuZZH2KmWRGTBK5JXGDxnGKsWWoarqb32k6tAui6gkKyxz6ddC6XYxIDK0kS/MCpyGQjpgnnEagb9Iyq6lXAZWGCCMgiuf8NSX2r+B4DdancLeSLJGb5I4hKCsjKG2lNmcAfw49qoWvii50r4Zx6zqAn1W7RvJAWL555DMYkysSHGSVztQ98KTxSdgV21Y27fw5p2maTPp/hyCHQEmO7fpltDGVfgbgpQoTgAZKmsqTwHbyeDf+EcOr6gIGuDcPcCO282RjKZSGHk7CC5yfkye561W0rx5eXmk38994dvLW6tZIY4Y/KnjjumlbYio9xDCc7iA3yYUEHJzgTX+v+LLHQtZubjw7p0FxY2huIJBqjS28xAJYZ8pXBUDOCoByAGHJD1uOKbtY1tI8LaJoelPp+naXYwwTIEuRFaRRC5+XaS6oqqSRnPGOemOKl0jw3ofh/wA3+wdG0/TPOx5v2K1SHzMZxu2gZxk9fU1FZXXiGXSZpb3S9MhvgM28MOoySRScfxSGBSnPorVn6f4kuF8CW2q3FtJc3sriFbfzkJeZpfLVfMCINu7HzbRxzgnq3e5Ks0W4/B2ihdRiurOK+tdQu/tslpeRpLDHKVAZkUjjJG49eST3rRTSdOiWQRafaoJYVt5AsKjfEoIVDxyoBIA6DJrm73xbremafqH27QLU6jZpFMttb6kXjnidiuVkaJSHBB+VlA6fNyca2hatqN9c3tnrWn29jeWuxitrdm4jZHBIO5o0OcqwI29hyc8ae1qNW5nb1/rt+AaX/r1Hp4U8PR6XJpseg6YlhLJ5klotnGInfj5imME8DnHYVJH4c0OKaaWLRtPSSeD7PM62qAyRYC+WxxyuABtPGAKs6lLdwabcS6dDDPdIhaOOeUxIx92CsR+RrkfA8ni+48NnVtYuIr64vbWKeztpL1BDll3HJS0RoxyBgmX68ZJ7eq7+8/PV9f8Ahh8q0Zu/8Ijo0Gj3+n6PYwaMmoQmKaXTII4JMEEZBC4yMnBIOK07Gyt9N0+3sbKJYba2iWKKNeiKowB+QrmYvE2pW/w5j11LAX94v+stpLtVLfvChCyCJVJ9MqoPGSOtSDxVqlva3sWpaNbx6pBLBFDbW18ZYpjMcJmVo0K4Ibd8hwBkbicVMqk5t8zv/Vl6istDqaK5bw1qWuXXirW7PXoRbeRDayRW8U6Twpv80ExyCNHIOwZDjIYNj5cVta1qh0jS2uY7drqZnSKGBWC+ZI7BEBY8KMsMnsM8HpUFF+iuQvfFut6Zp+ofbtAtTqNmkUy21vqReOeJ2K5WRolIcEH5WUDp83Jwtz4y1DStI1SbWtFjS/sWhVLawuJLpZvOYLH8whDg7sggRsQBkbicUC62Ouorm/CHim78Ri8TUNFutMltimGeG4WKZWB+408MTEjByNmBleTnhvhDW7vWtP1XWLyUtaNfzx2cCRZMUMLGL+EbmZmRmxz94AdKtQdm+wHTUVyfhfUNY1n4etqRumm1G8E8tvlI1EXzsI0AwMYAHD5YHIY54GhBPq+n213PqJW5CyxiIMyp8m1Qx+Veu4ng+nalKLjJxfQV7q5uUVnvqM8bX6m03taqHjWJyxlBBxxt4PHQZ9s9Kz4fE0xs72a409o3tzGqRqZCZC5wPlaNXAzjnac84zipGdBRWPZazd3lpdN/ZzRTwAMol82OOQH0Z41bIwc/L6evDTrOoDSWv20pVQorxxm6G5gTgg4XAPII5IPcigDaorJOrzwWt417bQwz2wUhEnLo+77o3bAc54wFJ6Yzmqq+I7qSyeSPTf38dxHC0Ukjxg78BSC8YPUjOVHfBNAHQUVz6+JZVtJpp7DaVA8pIpGkLt5nlkEBMgbumASR2B4qSLVNQ1PStQW2s5bS+ijPkMyOEdiDjBlRcnI5yMcjmjpcOpuUVm6QmqI11/asvmDeog4UfLsG4/KP727r6VLpF1fXmlxz6tp39m3TFg9t56zbAGIB3LwcgA+2cU7aXEXaKzde1OfSNJa9traO5MciB0kmMfylgpIIVskZyBgZ9RVKXxBfQNrqPpQmk0yJZbeO1maRrlWViMrsBVsqflXf7ZPFTcfWxv0VwNn8Sbx9G1i81Dw/Jbzae0EcVvGbhnuHmbav7t7dJlXdgbvKOfm27ipFX9F8Y6rrGk6pJ/wjslpfWKq8aXX2m3t7gEE/LLNbo+RtOR5ZAyvJzw3om30Gk3bzOvorjT4v8Qr4Wk1x/CsUcRiimt7dtTXzZFYgENhCqtyCoDMCD8zIeKsN4sv7LS9Xk1vTbO0vtOWNlhhvnmim8zhBv8kPuLArtEbHONu4nFN6XuJa2t1OqorgYviHqtxoktzb+Gwb6C/gs5LWeee2VvOKhHQzW6OeWAO6NRwcFsDKx/Ee8j0m8vb/AEHy2jUC2htrh7hp5PPNuQQsW5R5g4KhyV52hvkpf1+Fw/r9DvaK5vwh4pu/EYvE1DRbrTJbYphnhuFimVgfuNPDExIwcjZgZXk543rxrpLOVrCGGe5C/uo55TEjN6FgrED32n6UPQNyaiuSi8Talb/DmPXUsBf3i/6y2ku1Ut+8KELIIlUn0yqg8ZI61btNc1ue11SC50ixg1axRHjhTUHkgkVgSMy+SGU/KwI2HoOTnhN2Tb6AtbeZ0VFef+FtV8TWnhMa74kmbUZdQitjY2cd1CymSbAA4tojGMuo5aTAzySPm6XStU1K9urvTNasodNvo4llRrK7+0xtG+5QwZ40IYMpyCmOnJyQKs07AbdFcj4Z1LVfEfw5kdL54tXC3Fst61uAPNRmVZACuxxwpJUbSc49Bq+EdcPiXwfpesPGIpLy3WSRB0V+jAe2Qa0lTcb+Tt+f+Qk72NmiiishhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBW1HT7bVdOnsb+MyW9whSRQ5UkH0ZSCD7ggisOL4f8Ah6GC9iWC9IvwguXfU7l5JSjbkYuZC28EDD53YAGcACr/AIk8QweGNJ/tG8tbm4tllSOQ2wVjEHYKGIZgSMkD5cnnpVL/AITGJYtQE2j6lDeWMIuWsZFiEssJOPMQ+ZsI4OQWDDHIGRnohhqs488Vo9Onl/mvvC/Qs23hLR7bT7yy8ie4ivQBcNeXk1zI4H3R5kjs4A6gAjaSSME5psXhDTIdPu7IT6u8N2FEvm61eSOADn5XaUsnvtIyODkVX0TxmviDT/tun6FrHkSWwuLVpoY4xdL0whL4B9N+3I5GRzUMHjSOD4ep4mv7a4mjQHz0hhSN1xIUJ2GVgACOzt7Z6VcsHWvyta3Ubeb2X4ff8yVNdPU2dF0Gy8P2ZtdNa7MORhbq+mudoAwApldiowOgwK0a5qXxjusNYNvpN8L7TIFn+xzCJXljbdskH7zAU7W4YhhtPy5wDa8Ka3ea/oNre3+lXGnyS28UmZWiKSllySgSRyF/3sHke9KeGqxg6k+llur6q69dA5lokbdV7+wttTsJrK/hE1vMu10ORkfUcgjqCOQeRXN6d4ymuNL1PWZrJpdNh1JrO0FsF8wxo4ieV97AEeYH4HO0DgmtGfxTb2ketPe2d3bro8YllLiMiZCCQ6bXOB8p+/txjJwOa5px5XaRS3shkHgvRILa7hMFzObxFSWa6vp55tqnKhZZHLqFb5htYYbkYPNPt/COl21jdWsT6li72iWZ9WunnIU5UCYyGRQCTwrAcn1OcSw+Kuh3vh/VdYaKe3s9LMaSSvPbSRyO/CokscrRFslc5cbdwzgVY0P4j6V4j0zU7nR7ee7uNNCtNZQXFrLIVbO0q6TNEc7W48wEbTkDIynoncavpYv2PgvSNN0u40+yfVIra4xuX+2LssmGLfIxl3R5JOdpGe+aWw8FaJpulXWmwRXclndLtkhutQuLhRyTlfMdthyc5XBzg9QKgs/E95r+hXdzoOkXtpMbQzafLqtvsiuGIOBtD71GcAhgpIOVyOa0vDmtw+I/DWn6xbIY472BZdjdUJHKn6HI/CrcHyt/L+vu/Am6/r5EFv4S0q3sbuzb7fd294oWVL7Urm64GcbTLIxQ85yuDkA9hS2HhPR9OS6WOCe4a8i8meS+u5rqR4+f3e+VmYJ8x+UHGSTjmtmioGYVr4N0eysbq1tlvk+1hBNcHUrk3BVPuqJzJ5iqOcKGAG5uPmOW2PgnRNP0q5023S9a0uQA8U+pXM20glgULyExtuOdyYOcHOQK36KAOePgfRW025sZBqMsV0UMzy6tdPK2wkqvmmQuFBJO0MBknjk1oWWg2Nhq11qVv9p+03aIkxlvJpFIUYXCMxVfqAOp9TWjRQBk2/hnS7Pw6dCsoprbTzuwkF1LG43MWOJFYOMkno3fHSoLLwfpWnaNLpVnJqcdpIqLt/te7Zo1X7ojcyboxxjCEAjg1u0U2222+oHNw+ANBg0efS4xqf2OdlZ421i7bBViw2ky5Tk5O0jPfNWZfCGjzzX0s8d1I9+kaTl7+c/6sgoUBf8AdspAIZMMDznPNbdFIDl00Gy8Izy6noejaxq17dBYZwmptNIyjJVm+1ThTjoDksM4HGasRXMviWKfTdZ8Larp1rJHky3U9sASCMBTBOzqwPIYAYK5yDiugooA54+B9FbTbmxkGoyxXRQzPLq108rbCSq+aZC4UEk7QwGSeOTVhvCekST6hLNDcTHUYliuVmvJpFZVHy4VnIQjqGUA55znmtmigDN0jQbPRPN+xzahL52N323Uri7xjP3fOdtvXtjPGegqHQNAGgSamkFwXtLy9e7hgKY+zl8GRQc8gvuboMbiK2KKpSaTXcDI8MaG/h3RRprXS3Mcc0rwsItjKjuXCt8x3EFj8wwDxwK0rm3iu7aSC4XdHINrDJHH1HIqWilKTk7sNjMXw9pyJOojmP2gL5rNcylnKnIOS2cjA568Y6VNHpFlHbzQeW8iT8SGaZ5GbHT5mJPHbng8irtFICiuj2yW8sO+7ZJgA2+9mY8ehLZX8MZpo0OyFg1l/pJgYg7TdykjHQBi2QPYHFaFFAFF9Gsnkmd1mZpwocm4k/h+6R83ykeowagHhvTRuO25y7rIxN5NlmX7pJ3ckf0HoK1aKAM8aFp4DDyXIYOCpmcjDNuPGcdeR6Hpip7Swhst/kvcNvxnzrmSXp6b2OPwqzRQAVS0jR7HQdLj07SYPs9rEWKR72bBZix5Yk9STV2ind2sBW1HT7bVdOnsb+MyW9whSRQ5UkH0ZSCD7ggisOL4f+HoYL2JYL0i/CC5d9TuXklKNuRi5kLbwQMPndgAZwAK6WikBi23hLR7bT7yy8ie4ivQBcNeXk1zI4H3R5kjs4A6gAjaSSME5psXhDTIdPu7IT6u8N2FEvm61eSOADn5XaUsnvtIyODkVuUUPXcFpsc+vgnRl0GTRs6k1jIysUfVrpmXbjaFcyblUYHygge1SS+D9HnnvZpku3kvljWZm1C4OPLOUZPn/dspGQyYOec5rcooA5Zfhz4cUyER6lulmjuJHOsXZaSSM5R2bzcsRxyc/dX+6MWl8EaAokX7JMySrKjo15MykSP5jfKXwPn+Yf3T93Fb9FAHPrYJ4Ttnl0fTtc1p52VXhOqtcsgAPzD7XOFUdjtOTxwccEV3c+I45tP1Tw7rmj27puNw97DESQR8qvbXDSKfpgYBBPOD0FFHqHoc3D4A0GDR59LjGp/Y52VnjbWLtsFWLDaTLlOTk7SM981abwnpbz38xOoCTUI0iuGXU7lSVT7u3EnyH3XBOTnqa2qKAMKy8G6NYaLNpMCXj2MqqnlXGo3E3lhfu+WzuTHjAI2EYIBHIFWNM8P2miRXP9mNcGedQDNfXc143GdoLSuW2gsTtBA5PTOa1aKd2BgWHhyfTPAx0Gw1Hybr7PIi33k52SvuJkCbugZiQu70GTWlo+lW2h6LZ6XYKVtrOFYYwTk4UYyT3NXaKpzlK9+uorbBRRRUDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDnPHejaj4g8Ky6ZpKW7SzTQs7XFy8AVUkVzhkRjk7QOMYznPGC2y8Js1ndy6rqN5Pqd/Zi1muWkjdrdMHKR4iRSAWJ3GPJ4z0AG9fXa2NjNcyLIyxIWIjjaRj9FUEn8KwvDni2LVdAn1LUQbRLfBkaS3kiUDaOhcfMd2RxnsPrvHGyhT9hFpW9777f5L7jOUoxkk3q/wBNTQ0nRW0jw+mlRaneTCKLyobmYRebEoXauNqBTtxxlT75rJTwHAvgyfw02tao9pM5bzW8jzUBbeVBEQGC2TyCeeCOKtXXjCySyuprMM8lmEkuIbuOW1aONmxvw6Z9SOMHHUVdg8RabcWNxdpLKkdvjzBLbSRuM/dwjKGOc8YByeBmlHGzTbUle6lst11/P8SVKn8KZWt/C0MOuX2p3GoXl49/Zx2c8M/lCNkTOD8qKQfncnnHznjgYdoHhz+wVCf2vqV/HHAlvDHdvGEhjXoFWNFBPT5my3A5qeHxFpk1nc3JnaGO1AM63MLwvGD0JRwGwexxzggZqrL4qtZLDUJNNR5rqxg897W6iltW28nnemRkA44OcUpYybi4uW6XbZbfh2K5oaalLTPB8ltout6Hc3O3Tru8lnspLdsTQLKfMYfMCMrKXK9RjGR1p0HgqSD+0H/4SjXXudQWISXLSQB0aNsqyBYgq8fKVxtIzlckk6una/Y6neS2lqZmngAE3+jyCNGwDt8wrtJww4BzV+4nS1t3nkEjJGMkRRtIx+iqCT9AKzlPmfO2VGUZK8WYMPg6AWt+t/qeoahdX3lb76YxRzR+Ud0W3yo0UbGJYHaTk85GBUqeHb/+zby0uPFesXLXKhVnkisw8AB52bYAvI4O4N7YPNXLXWYZ9BXVWVzEVLFYYpJG4JGAuwOTx025ql4Z8QXPiGF7iXT/ALJBsV4i3nBmDcjiSFAeO6sw9+hOfNFvl7iVSKcbPfYS10TU9G8LXVhpesT6jdrbmOxk1QRKsLBNqAmGJcqDgnIJ96ueHNEh8OeGtP0e2cyR2UCxb26uQOWP1OT+NW76+g0628+7LrHuVSyRs+CTgZCg4GT16DvUJ1mxX7b5krx/YcGfzIXXAPQjI+YHBwVznpWjq6OLfn93/DjvFWX9f1oXqKyYPE+lXEV3Iss8f2MKZ0mtJYnXdnb8jKGJOOAAc0+HxDp89nc3EbXOLXBmiazlWZAeh8orvwecEDBwfQ1nzx7iU4vZmnRWdpev6brWf7MuGnUKHD+U6o6njKsQA2DwdpODwcGrtxOlrbvPIJGSMZIijaRj9FUEn6AVV1a44yjJXTJKKzrTWbefQV1WZ1ig2M7EB+ACR0ZVbPHTaDnil0zXLHVpp4rNphLb7fNjntpIWTdnb8rqDzg0uZXtcFOLtruaFFQ3d3DY2klzdPsijGWIUsfQAAckk8ADkmsuTxdo8No9xNNcRLHIsTpJZTLIjN93KFNwB6AkYJ4BzQ5xjuwcox3ZtUVhHxpoS20tw11KkMRUSSNaTBVycckr0B4Y/wAJ4bB4qRvFOmjTbu8X7Wy2iCSWE2UqTbT0IjZQxBwfmxjg88Glzx7k+0g3a6NmisseI7D+y5tRdb2K3hxvMthOjDPcIU3Ec9QCKtR6lbS6k1ghl+0JCJmBhcKEJwPmI25yDxnPB44quZdxqcXsy1RRRTLCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAp6tY/2lpF1ZhmUzRlVKytGQccfMpBAz1x2rH0nwmNN8OyaelzKkzTeckpmeYRsr7oyA57YXIGM8/Wt+6u7extZLq9nit7eJS0kszhEQDuSeAKoW3iLTZ/Da67Ncw2tgULtNLcRMiDOOXRmQ8+jGmqDleaW9l/kiJQjKSb3RVutI1u+064gutZtTJLtCeXp5WJAGyTtMhYsemd+Mdu9NvfDU2pvqK6hdW0sF5HEEj+x52NGdwLbnIcZ6rgccZ71D4T8dad4yeU6RDJ5MUayGVri3fhvugpHKzoSATh1U8HOCMVvXt/Z6dCs2oXcFpEzrGrzyBFLscKoJ7k8Ad6dbCujN06qs+1/67kqEJa/q/66mHY+FDaWN1Av9k2kkpjeOXTNMFttdG3KWBdt4BA447+tWZtK1i80u/tr7VrZ5LqEwp5ViUjjByCxUyFmbB/vgdOOub8esaZL9t8rUbR/sBIvNs6n7MQMkSc/Jxzzjiq1l4p8P6jY3N7p+u6bd2tqu+4ngvI3SFeeXYHCjg8n0NY+zja3S34DjSjGyX6kPhzQZ/D0UtnHdwyaaGLW1stuVaDPJG8udwzk4xxn0wK1btbl7V1sZYoZz9x5ojIo57qGUn8xVGHxRoFzo0ur2+uabLpsDbZb1LuNoYzxwzg7QeR1PcVPY61pep3E9vpupWd5NbBTPHbzrI0W4ZXcAcjI5GetXy6co4wjCPLHYz9L0jV9O0WWxbU7KV8HyJBYuoQsxLbl807uvGCuPerWg6ff6XpsVnf3lvdpBGkULQ2zQkKox82ZGyeB0xV66u7extZLq9nit7eJS0kszhEQDuSeAKp2GuWV/oCayJoobJo2laV542RFGckujMmBg8hiKqNF254rTRCUIppLp5sm1WybUdLntEl8l5Fwsm3dtOcg4yM8jpkVkDQdXcai1zq9pLLfRohB0792m0nI2GQ7lKkggnPJOegGnpeu6RriyNouq2WorEQJDaXCShCemdpOOhqt4o8SW3hPQ31a/trme1ikRJfsyqzRhmChiCwyMkDAyeenWq+qznVVPlfM9Lbb/wDDjcYy94zrfwVCukX9lcGxVbsxsIrSwENujIcq3klmDEnG7J+YADjFT6d4buNMsb2Kz/sS0muFCrJZaQYVHXO9RLl+DxyMe9PsPF1nf61e2C21xDFZ2kd419K8XkPC+7Y6kOWwQjHlRgLzjjNuPxPoM2mNqMOt6bJYrIImulu4zEHOMLvzjPI4z3FEsDOLs4PZL79tu/3mcadJWa6f8OSaLZ32n6fHa6hd2915KqkbQWzQ4UDHILtk8dePpVq7W5e1dbGWKGc/ceaIyKOe6hlJ/MVXg1vSrq2uri11Ozmgs2ZbmWO4RlgKjLByDhSByc9KdpmsaZrVu1xo2o2moQo+xpLSdZVVsA4JUkZwRx70SpTs3JPTfc0jGMYqK2Mqz0HVbfw/Ppk2p2Uu8ERv/Zx2gFizh0aVg4IJGOOKn8O+H/7D+0ndZr9oKnybC0+zQJtB5Ee5vmOeWzyAvpWnZ3trqFqtzYXMN1A5IWWCQOrYODgjjggj8KnrJU1F7a7EqlBNPtsUda0qHWtJmsbjZtkwRvjEihlIYZU8MMgZB6jisKHwhc29qY7OTQ7JjcQzFrLRjDu8ttwDATc8gc9ufXjq6KTpxcubqOVOM3eSOI1jwxqttpMiafL9td3YLHDborqGuPODZklC/LyPfjgc1t2+kXt7b3Mmt3MbT3dp9nCwW3k+SrA5BHmSAtk9Qcccep3KKlUorb+t/wDNiVKKlzL+rGRFa6jbadc/2rMNXLRiNbW1t1gUrjBADucsc85cDgYA5yzwva3cGmtLqkM0V5K5DCdkZwi/LGCUZh90AnnqWPGa2qKvl1uNU0mrdAoooqjQKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBk5lW3kNsiSTBSY0kcorNjgFgCQM98HHoa5fw5pWv6N4Hk02e1017+NpfJVL6TynDuWJZzDlSNxGArZwOmeOnnOIxzj50HXH8Q9x/nsehkreFZwg4WTTafXpfz82HVM57wTp+saN4YsdI1q3sYzp9rFbxy2l083nbF2liGjTb0HGW6+1XvEVhcan4eu7SxMQuXQGLziQhYEMAxAJAJGMgHHoa06KitUdWbm1ZvsKMVFWOIk0bxVfR68+pWGhM2pQQxxWy385RQjEMjSCJWBKsSJFAKkgbfl3Gnb/D++m0PW4LsR2k+oPbyRxrq11eMWgbcN904SbDEBdo4QAleWNd/OcRjnHzoOuP4h7j/AD2PQyVmM4jRvDeq6TZazNBo1nHqV9FHGrT+Jby9EwXcPnkmhJQKGJG0NnODjrWt4N0m60fQbaxv9J0rTns4lgiGmzmVWXGWPMUe3LZJAB5Oc10NFPv/AFt/w7D+v6+4ZOZVt5DbIkkwUmNJHKKzY4BYAkDPfBx6GuJs/C+tr8N5NAv7LSprkXG9UF/OIpEM/mn94savGwyQCAcEA+1dzRXRRxEqStFdU+u626+YrXdzj/AvhfVvD0+pTaxeG4+1snlLLeG9lQLu4Nw0UbsvzABCDtwTn5jWj410nUNb8MyWGkLbNcPPBJ/pUzRoBHKsh5VGOTsx0757YO/RVSxdSWIWIduZNPy02/ISikrHm7fDbUHk1vF1aR296lk9navukjhaCZ5mgfgboi7kA9dpxt+UAza14Q8Qa7a6/czQ6Ta32qWsFkltFdSNFsjcsZJJfKBZvmIA2cADnnj0Kiuj+1MRzcztfT8OX8+VX66ebFyLXzPOtd8Cazrt3quPsFjbXNpYiCOG6lP722dpNjBUQhGL7dytuAUEDPA1/Cfhm90nXdQ1G+s7W2NzBFEpTVbvUJX2lj80k+AFGeAF7nntXXEhVJY4A5JPasrRvEmn6+X/ALMF6yKocSz2E8EcinoUeRFVweuVJ4we9RPMK06To9H6+Xnr8K3v5ByLcXw5ZXWn6DBbX9tptrOjOWi0uMpAuXJG0HnJBBPuTWpUN5dR2NnLczLM0cS7mEELyuR7IgLMfYAmotL1O11nTIb/AE92kt5wSjPG0bcHByrAMDkHggGuCT5m2WW6KKKQBRRRQAUVFbXCXVus0QkVW6CWJo2645VgCPxFOmmjt4HmuJEiijUu8jsFVVAySSegAoAfRQDkZHIooAKKZNNHbwPNcSJFFGpd5HYKqqBkkk9ABTwcjI5FABRUVtcJdW6zRCRVboJYmjbrjlWAI/EVLQAUUVly+JdJhttWne6zFoxIvmSJ28khBIRwPmIVgSFzjPrTSb2A1KKbFLHPCksLrJHIoZHU5DA8ginUg3CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCOc4jHOPnQdcfxD3H+ex6GSkZQwwc9QeDjpUd1d29jayXV7PFb28SlpJZnCIgHck8AU0m3ZAS0VBZX1pqVnHd6dcw3dtKMxzQSB0cZxww4PIpbW7tr61S5sbiK5t5BlJYXDqw9iODTcZK91sF7jpziMc4+dB1x/EPcf57Hocjxfpjar4YuYIbJL24QpNBC+3LSIwYbS3AbjgkjB7irNjr+iaxdS2mmaxY3txB80sNrdo7x4OPmCnI545+laVE4Sg7SVmF0eX6pot9q1n4l83wFcRJqZtpYoVmsxJLIjHdIxEoCydwwYnG0hs/KL50GTT7PxNpui+ELmz027ghEMFjLa26zMRtlMY3lQ4Uj76hW2YOR19Boqelg63PKdE8BS2Ph3xAsHh6SM3D281ra30dlHM8kJLcLbEQIORtPytuLFjjaRprperag3iV4/Cs2kJqQt5vLNzAjXmxv3qO0TnbI65XOSMFcsOQPQ6KHqC0PPIPDCHTdXOheCT4cFxHCr232uKFrxUk3PEqQO0cYZMrv3AktyAADUEvg8Xuha1DoHhB/C32uKFGiWe2RroJJueNYkMkChlyu5vvFsMu0ZPd2mvaRqEFxPYarZXUVqxS4khuEdYWHUMQflPsas295bXe77LcRTbMbvLcNtyMjOOnFNpp2f8AX9fcO55vonhabSfCviCzstD8Qot9HGgsZ5NMtg2cq7QrbYiV9py28APtUHIzUegeDrnRvDXiaO18P38aakkUSWjPY211IACrlRBi2AAfKggFjuD5BFepUUnrfzEtLHmfhXwcui6L4jEvha4mtb9YlXTbqKwWa4KhgR5cLLbBPmBGcMTv3Z+WnaP4USL4aPp0HgaPS9SggtmeNo7RTezxAHerRuw3ZBwzlTlu3NelUUd/O34B0scVor6lY6xr+qf8IVeWa3ghljghmtPNuXGVcviUKH5zyxBUD5s/KL/gQ6lDoK2OraJe6XJbsxDXMsDrLudm+XypH6AjOcdeM101UdV0i21m2WC8kvI0V94NnfTWrZxjlonUkc9CcUAQeJrdLvw1fW8ukHWVlj2/YQsR80k8cSsqcHnkjpxziuItPBZl+HV3Y2Hh+80PVJYYYrpp57eSbUFRlZ1Mm+VXyN6Dzc4DYwF4rtdK8M2OjXTXFpPqkjshQi81a6ukxkHhZZGUHjrjP51qSzRwQvLPIsUaDLO7ABR6knpRor+Y7vTy/wCB/keXab4Vk0nwnrkUOnaxZWsjQTz2WpPptvBPEjZmRFtSI0LxgqxfaGG0E4BweC4vD+oR+J9R0LwfDcadOILZbC2azmWZcEvGpWUwBctuKB8c5I3NivSW1bTktZ7l7+1W3tl3zymZQkS4zljnAGOeaksr601OyjvNOuobu2lGY54JA6OM44YcHkU7NpsXY4bSbTVE+GF1oU3hPUrR4omhS2S5tIzLHJI2REVkZAVQ52uAp6cgms7w/wDDxU8M+KrCTQvs8OqQKkEGpw2auZFRsMUtf3CqGZSpAD7txOflr1Gil3Gna1uhwtnpUCeENVs4fhxDp9pNGo/sxFsxJeSYwWZFbysD5SC0m44PyjA3ULDws0Hw9u7bT/Cc+n6nNBFbXgu3t55r1AR5jZMrpIwUuE844HAI28V6TRQ9b+Yloku3/A/yPLvD/gy7PhXxR4fvdBnhsb6PfYRamLHYH8vb/qrYCOMh1DcL6NuLZA3dI0+3tfDeo26fDiKwspFQPpcS2fmXrHhyyK/k7QMYLSZOG+UYG7tKKd7h28jz3SdO1Z/hu2hyeGr3Tjasv+iNPbqt3D5xdoYzFKwXKfLhto+bGcZIo33hYXPh3WYNN8CTadZ3K24j0iNrSIvMrktOoSby4yF2jcGDnb/sg16hRS/r+v6/EFocr4V0lNE17VrbTvDh0XSpFhkh8oQJDJIFIchI3JDfdBJUZ29TxWX4SvbXQvCHiaPXPlk0vUL6XUS4/wBYrs0yv7honTH5dq76itFP3XF9QWlvW5znw+sLzS/h3oVlqSslzDZRq8bdY+OEP+6MD8K6OiilUm5zc31dxJWVgooqK2uYLy3We0mjnhf7skThlbnHBHHWoGS0UVm6Z4j0TWrie30fWNP1Ca3/ANdHa3SStFzj5gpJHIPX0oA0qKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDnPHn9t/8ACH3X/CNGQXm5N5hTfIIdw83Yu5SW2bsAMG/ukHBrA8G3d5pVxq0l9e63fabHFE6ibRr1dshJU+WJ5JZ3OAMgDaOCOprv5nMaAggfMo59yB6j1/8A19KfXdTxfJh3QcdG/K/TyfbTtd7kuN2mcP4Hnhl8ByaffWur2r24nedWsrq2k2tLIw8ttqszYxwhLcjvTfDt01x8LZYtZTXfNh82OXzILxbpgZCU2nHmt8pXlc8ZBPBruqKqeMjOUnytXkpb7PW9tOt/y3CK5Wn2OJ+Flrb2fg+xgWHWIL2Kyt47uLUlu1VJAmCI1n+UAHP+r46e1bfi+Od/DFzJafbDPAUnQWTyLK2xgxACEFsgH5ed3TBrbormxNZ160qr6u+rv/XkEI8seU88/tiS4j8VSWUPii1gnhia2lksLt3DNlHeGJ8MACQTGuxsKWUDIY5WlWPiRfCviT+ztR1ecyG3EE9zbX0bKgP79YoLmUz7thJ3CUbiQF2lc16xRWGli76nmegLNb+GdeTUdX17ULUpGYok0jVbWeFzkZjeV3nfJ25CNhcdAGOd74fwodJW6e98QXN40MaXn9tR3EWZAM5WOVFUYzjMYAOOcnmuuop31b9P6/ryJ/r8v8jjvANpHf8AwwgtJywjnN1GxU4IBnkHFa91pMen6beyWH2jzJHSYiJju+UKuBtwSML05zmr2naPpmkLMNJ060sRO/mSi2gWPzH/ALzbQMn3NXKqpJSm5LqC2sc4bx5F1ZoE1WKOREMTNbysQTkMyK3IAJHyjBwCQOhqrawakNL1P7NcXkhYx+W8kc6kKD84RJX35x3Dc5AGCK62ioGczYB49M1AXF5f3ERVSiLaXcUiHn7rOzSNk44BwMe5qLyyfDUkqXOttdFY1kLJOJA+Ryq7R05HAwe+a6uigDm5GltodStbNdQkTbGUabz2K7uHKsfmbAwdqnPXGDWcj3Rs5baWXVmUXcJilit7qNvLJAflizYxu+8foBxXa0UdQOPRdSSzuIoH1ESyIwLSrK+AkuPlyRg+X0wQW65J5q7pVk15YahZXd7dT29wmzbJazwmPIIJVpixbPHGcDHTmujoo6WYdSnYaXb6a1w1tuzcOrtuPTChQB7YWsH4af8AJP7H/rrcf+j5K6W6tbe+tJbW9gjuLeZSkkMqB0dT1BB4I9qbZWNppllHZ6daw2ltEMRwQRhEQZzwo4HJrTmXI4+n4XFbUz/EtvrNzpPl+Hbj7PeeYDv89Ifl5yNzwTD042fiK5VNI+IwikD6vlyBsP8Aalvxzz/zDB/I16DRVQq8ityp+qB6nK+F7Dxba6hK/iW/+02xiwi/bYpsPkc4SzgI4zzuP07i/wCL4538MXMlp9sM8BSdBZPIsrbGDEAIQWyAfl53dMGtuioqS53eyXoCVjzz+2JLiPxVJZQ+KLWCeGJraWSwu3cM2Ud4YnwwAJBMa7GwpZQMhjlaVY+JF8K+JP7O1HV5zIbcQT3NtfRsqA/v1iguZTPu2EncJRuJAXaVzXrFFRpYq+p5noCzW/hnXk1HV9e1C1KRmKJNI1W1nhc5GY3ld53yduQjYXHQBjmFYS3w3uLqLUfGkmptHAk7yQ3qTiYEHdHF5Y4AJB2JsbHz7jzXqVFHf5fh/X5C/r+vuPOp3uNNsfEel6OniCeIR27RSXf2+Vo9/wAsrxyt+8fauG8uN92QQu0kkc7BLqbaLc6dd3XiySNdVtGtbm20/U7d/IYqJsF2klK4EmRI3BwQB8hPs9FHW7/q39f10NloeRxReIodH1C2sJ/EAu7iKRWku47ubYkV0V/d7mXDG3xgo6tJ1DM3zV0vw6jvIf7RjuNY1DUIMxtFFe6Tf2vkn5gxV7x3eTOBwGwuOg3c9vVHVbC51G2WK01e80p1fcZrNIWZhj7p82NxjvwM8daE7L+u4bia3JbRaLcm/F6bcrtf7As7T4Jx8nkfvM89V5HWuBXU5bz4Uy2v/FVQahasq+atjfpdDMp2kMybpcKOfvDpu6122laNfaddNLd+JdU1VChUQ3kdqqKcj5h5UKNnjHXHPSteiwHB2wtbJPEelufE0umpBDIjSDUJpSzLhhFMcyN/BlUY4+bgfNVDw5CLT4YrFpCeJLe7jt7SPUEukvfOjUbRMLdbgHkLvwIh6Y5216XRR387fgC0Vjk/B5T7ffjS21s6T5cZQayt15gmy2/YboeYV2hM9Vz053VpeEP+RYtv+Qv96T/kNf8AH1/rG+/7f3f9nbW1RVc3u2EFFFFSMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAjnOIxzj50HXH8Q9x/nsehkqOc4jHOPnQdcfxD3H+ex6GSgAooooAraiVGm3DPetp6LGzNdqUBhAGS+XBUYHPzAiuI8J65Y3PwjM48QWtmtqJYpL21a3iFuRKwTIC+WhIK8bRncCByK76RxGoJ7sF6gdTjv9adXVSrxhTcHG+qd9Ol9Nnvf08hfaTPOPg7p+hJ4bttR03UrXUNTnsoPt3lpa+bA7LuZXaJFkJLZz5jMSV65zWt8VJra28Byz3MttDJFd2r20tzH5gSUToQwXqxAycLyRnFdjRRjMQ8TXlWel3933JfkKnHkVjzq38S3Kadqx0XxUviO0gigZtW8mGVbNmk2ycwKqOEj/AHm3G5cfMSGFPttdtf7M1ueH4ljULGC2RlvxHaYs5Cx63Cx+SxY7RsK7gCD/ABA16FRXI9U0WmkzgPD1zqviXwZcXek+O4dS1Ga2Q7rGK18u0nA3bANr7cggMH3nuNvSqeveINWuPDuo+JdE1+70+xtvs0EMclpDInmeaqzs6+WXON2wqGB3I4GOK9EvbOO/spbWdpkjlG1mgneFx9HQhl+oIrIfwXosnhxNDZL42CTecF/tK5Em/dvyZfM3n5ucFsZwe1W3dvQUdEr/ANf1/mct/ad1qmgeIrXw/wCNZ9djh04zRalZG2eaCbD/ALoNFGEO4KDjbvXJOeVx0XgzVbDUbWddP8Uz+JhGUL3TrCyRkj7gkhjRCRjJXlhkZwCK6GCFbe3jhjLlI0CKZJGdiAMcsxJY+5JJ71JSuLt/XY5vUfE0V54Hl1vw5dhoWYKLtrdmEKCXZJLsOCdg3tjp8vpXGa1q2mXPhbWmfx5Z+ILOFbdre4u4Ld47SfzOGM0QSNm4BCDDADOfmBr1eik7Xdiuh5vdeI9S0/TvEenaX4li1rULBLaYT3DQLJbRyECR2EMRAVF+cExtjPO4YFRaF4p1p/DetSt4m0TV/sxgMd1Y30eoS2aO22WSURQwrtVQXUbc/K2SR09Noo7/ANf1/WojzDUNS0y+8GeLFn8aW/inS49MYgzw27RwyENgNPEqxMxbZhMBl4PO4VkR+KtYl8L6xqH/AAmLadq+mXEUcuiRw21xHYRmSNBuxH5kylWLbgy/e28Mpr2as/W9DsfEOnfYdUE7QeYsmILqWBtynKndGytwcHGeoB7U07D7f12/yOEbX7htO1GTRvHT6zp9gLe6uNTgFnI0K+YRNEWSPyyPLBcDbuXHUhgBt+ENb1PxDrl/dXhuLKG3gijfS38t0hmbL5DiNXJ8sxHBJH7wjtXWQQrb28cMZcpGgRTJIzsQBjlmJLH3JJPepKP6/r+vPcX9f1/X4HlEeo39n8OvEHiK11J9P1d9dkN1MYVmCBLkW6xshBJURBeFw3oeebx8VtbaNrL6V4uHiK1ht4iuprJaf6HNI5TDyxx+UqAYclkYoASQwKiutvvB2iajDexT286Jf3KXdz9mvJoDJKgUK2Y3UjhV6YyVBPIzWxBCtvbxwxlykaBFMkjOxAGOWYksfckk96upJTk2v60/Tp/SH/n+tzzHw/47uJfC/iUX3ijSZrnT9n2fUBdpcRoJAFVmkSCNSokJG8RFV5zu2kUaB4q1QeGvE17qHiywu4rVIktNRW4hubWB3BHzXEUEYJ3FdwMWEBUncCRXqVFZvW4lpY8m0jX9eutA1u9m8XLdW2nT2sv23TpbW7jWHJM6+eLdFZgnzFfLBUBMMdxrQi8V63fQavqjzzaRFbRW0E9rMscsVg0j5efdsVjthaN/mJUbjkYWvSaKf9f1/XnuC/r+v6+48/tfEUB0nXTD48S/022t45F1zzbPNrKxYeUXVBCc7UwCmRv5zuXFbTfEV5P4HfWY/HFrqZlFul7cW0MBi0vcy73AXO0qjZPmlwCNxwoK16TRS/4H9fMP6/L+v+AeeeAtQ0y48deIo7DxHaa+8traSC7jEAkmCmUMWaEKkhXKruCjAKqeRk9prd1bWei3M99qy6NAq4a/aSNPIycA5kBTOSB8wI5pdVi1Wa2VdDvbOzn35Z7yze4UrjoFWWMg5xzk/Sq2lW3iOG6Ztb1XS7y32ELHZ6ZJbuGyOSzTyAjGeMfj6m+g+tzhLTxjYf8ACoXn0nxfptre2LKr3MLWpEZaYhfMjACAMATgBSecEda1rbXYLdPEelX3jkSmwghn/tGWW0Sa1Ei99sYjAyFwWT+PqeK7yik1dNdxLRryPIvBK6Lovw5k1fwzrlhdX89vaDUr1IbNmsN2PMeQwxozFQzsfNLHK5P8Wex8H6ut/f39vZeJl8UWEMcbrfhoGMcjFg0RaBVQ8KrY27hu5JBXHWUVTd2HQ5TXfEz3nw9udb8ITpLuIVLiRWQRKJQkknMbEbAHOSjD5c4YcHj4PF2qJ4N1u5ufHegu1q8Btr601K2vXiZmwUmxBFHsOOAFDfe+bpj1uih2u7DvoefDxHcafD4g0638ULq9zYLbS/ap0gkltI5SBI7RwqgKov7wZXvySMVHF4haLTtVn0jx4mv2sSQBrxraCVdPDSbZJWlhVY2Kqd2wjKhdzZBr0Wil1/r+v18xLax5ynieSPTNU/srxhHrWnwm2363i2lNj5khWXJiVYztQB/mX5M7nyuBWLb68LW88YTeFvEljrt81pYyC/8A3Kv5YkdJZXaGMpJ5asPnEZCgKGDY59goo/r+v6uNOxxHw61vUNV/tGO/8RaLrkcJjaJ9P1KK8mj3btwkMUEKgcDb8mfvZJ4wzwXZ/wBueGfEMt25W/1PU72K6YjJiKOYY1x6LGiYH4967qqlnpVlp91e3NnAIpb+UTXJDHDuFCbsZwDtVRxjOOa0Ulyyj3/r+vQnX8f8zJbSx4f02/ubS5aMu8cpYIvyqqqpBznjAJzx1/GmPqX9ow61Fp2rPKY4RJA0BQsvByEIXkZAGeTnPOenSUVmNaGHbX9rJpN00evyToigteMYtsRPYMECZ9jkjI9RWTpVtpyeEbiayvIp5pbaP7Q8YiBiJHO8xqrEckncScA812VFHcForHMaZYW1/pWrWFrc2jw3C7BJYRBYFLJjhQx+YdTzzx0qdvC+nx6LrFlesHs9RiKTAjaFj8oIRn6AnPvXQVW1HT7bVtMudPv4zJa3UTQzIHZNyMMEZUgjIPY01vca0aMX4fX95qnw70K91Jme5mso2eRusnHDn/eGD+Na+larZ61YC902UzW7O6BzGyZKsVbAYAkZB56HtVhII4rdYIkEcSoEVU+UKoGABjpUdhY22mafb2NhEIba2jWKKNeiqBgDmtKkozlKSVrvQiKaSRmeL7ufT/DFzfWt69k1qUmeVFRhsVgXDblI27c5PBA6EVy//CdWUsfiqTTfF1nPDbQxSWlzM0ZhtjJlMqyJ80YfaN5DhTnJO0rXodFYlnk+leJvE0nhXxJdW+vWmsvbG3W3u4LmG4gt1Y/vW8+K3UMyqdxUwkIAp+fJFXNA8TXc3hnXpdX8caJJBapG8Oo2Gq211LalsjEjfZ44gCQoXMZJy3J4x6ZRTlqml1BO1jkfh/qUus6SuoyeLrbxC0sMfnxWqQiO1lxkqvl5ZTgjIdmORkbRxVL4f6JZ3HwxeC3ghtZNRa6E80cQDSMZZFDNjG4gY69hXd1m6LoGn+HrWS20pJooZJDIY5LmSVVJOTtDsdoyScLgZPStXNOEo97fqQk1ZmVeac3hzR7660+8MDyTQvu8oMFUBI8Ec5HU8YNOOrGK0vTa6v8A2jEkaYug0X7l2bbyyrsAxzkqcDJOcgVu31jBqNv5F0JDHuDYSVozkcjlSD1qaNBHGqKWIUAAsxY/iTyfqay3K0OXsNelbS9TE+q2jyW+3yrgSiRQGAAJZUUEbuMhcDvnBosNVuxpuqTXGrW8qxKghuQ6SxRscjmREUE5IzlflGDzk11VFAHI2moX8lhfTvq4litnibzrdoplCZ/eDeI1BOOSNvHy8nNWE1e+uI7y6LvZpEkSPE4V1tyzZZ87QThCrcnAyfSumooA56LUk+y3+zXluLaKNWF9uhzE5JGzIXYeg6jPPuKittRnfQjerrkV0XEazyRpGVtckbmAHTAPO4kAjPAyK6aigDk7OK01a61ywh1WK8N1axr9oVULHhwSdmA+MgZAHYVq2Xhuytbe/t544rq3viBJDLEChQIE2kHIYHB6+ta9RXVsl5aS28rSqkqlGMUrROAfR1IZT7gg01owOK8E6bY33wdTTb2NFsJoruGRANqrGZZAcemB+Va/w+v7zVPh3oV7qTM9zNZRs8jdZOOHP+8MH8avQ+GdJg8Lv4dit3GlyRPC8JnkLMj5LAuW387jznPNacUUcEKRQosccahURRgKBwAK3qVFLm83f8/zv+BKVkl2/wCAOrF8IX39peGLa6/tj+2t7SD7b9l+z+ZiRhjy+23G33xnvW1RWN/daKCiiipAKKKKACiiigCOc4jHOPnQdcfxD3H+ex6GSo5ziMc4+dB1x/EPcf57HoZKACiiigCOc4jHOPnQdcfxD3H+ex6GSo5ziMc4+dB1x/EPcf57HoZKACiiigChrl7Pp+h3dzZLC90kZ8hJ5AivIeFXJIGSxAAJGSQMjOa47TfF/ieGx1seIbSyt7yza1S2jli+z489ygkl2TTqEB/uuWwjZC5Wu4vrCz1Sxks9StILy1lGJILiMSI4znBU8HkVWsPD2i6Vp02n6XpFhZWU5Jmtre2SOOTIwdyqADkDBz2o7j7GMl14ngbU7S81bw9Ldw2i3Fs0NpKrjJbJkgMxOz5cBhJyc+mDQl8ReK7DwRJr2oW+lzy3EMDW1rZqw8h5HVcu8kirIuHDf8ssYK5/jHQw+EfDdtpVxplv4e0qLT7lg09pHZRrFKRjBZAuGIwOo7CptP8ADeh6TYz2WlaNp9la3OfPgtrVI45cjB3KoAPHHPajv/X9f8AXVf12/r5nJf8ACVeMdO8L6rd634fEd1A8KWcpWCJZTI4Q7oxdSABCQSTKoYHHy4JpU8QeOYfDWqXGo6Ta2V5atE1rNdwokdyGbDIY4rmUoRx8+8/e+7xz1OneFfD2kQXMOk6DpljFdrsuI7azjjWZcEYcKBuGCeD6mq8XgbwnDZT2cPhfRY7a4KmaBNPiCSlclSy7cHGTjPTNAEGnPrF//a2i65fwC4iSMpe6XA1sQkgPAV3kwwKn5s9xwCKs+Dy//CL26yTTTGOSaMPPK0rkLK6jLsSzHAHJJNTN4W8Pu12z6FprNfII7omzjJuEGMK/HzAYGAc9BUmkeHdF0BZV0HR7DTFmIMosrVIfMxnGdoGcZPX1oEVdX19U8PrfaFcWdybieO2guGk3wK7yCPcxU/MAT90EZI25GcjkZdd1/wAPXviuW5lttUvreLTxEts7pCrSvIm8wyzbYsDaSokAYKGLLuOO9Gj6YulPpY060GnyKyvaCBfKYMSWBTGCCSc8c5qpY+EvDmmCQab4f0uzEsTQSC3so498bHJQ4HKk9R0NDtd2/r+v1KTMLTde8XpoOry69o0NrdW6KbCWYw26XLtkBCi3MoUhtoBMgDFwPlxms6DxJ4ym8PeKl1H7Fpep6ZaLcWxksfmClHYsyLcSpzsIXEhIIJZcYB6u28GeF7OyurO08N6RBa3gUXMEVjEqThTkB1C4bBPGelW9M0LSNEs3tNG0uy0+2kYu8NpbpEjMQASVUAE4AH4Uwi7NXMuC71pdHv4Ztc0a61eGFZ4/JsHRYkIO0yReezENtbBDKOO+DVC21zxRD4XTVtUXR/MvktxawwCUJbPK6oDI7H51+cNwqEY2853VuQ+E/Dtvo82kwaBpcWm3D75rNLOMQyNxyyAbSflHJHYelOsvDGgabp1xp+naHptpZXQIuLaC0jSObIwdygYbjjntR1fy/r+uxK2S/rp/XzMaS/8AFkMGraes2lXerW1tHd2s8VjIkThiw8tojMTu/dnDeZg56Dbzf0XWLrWdYuZrW4tZtFjt4hG6QMsjzsN7fMXI2hCnG3OWxn5TVi28NaZpel3Fl4btoNAE53GTTLWGJg397aUKk4GPmU07w7oUfh3R10+G6nu1EskpluFjDszuXYny0VfvMT070f1/X9eY2c74YN34n8K+ILq61G7gk1K9vIIjHMymxSNmgQR4PykeXvJHVmJrY8DazceIPAmj6pegC5ubVGlIGAzYwWA9CRn8acPClpHY65aWt1d2setSPLL5MigwO6BXaLKnaTjdzn5iT3rVsbK303T7exsolhtraJYoo16IqjAH5Ct6k4OLS8reWmv6fcLr9/8AX9bE9FFFc4wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAjnOIxzj50HXH8Q9x/nsehkqOc4jHOPnQdcfxD3H+ex6GSgAooooAjnOIxzj50HXH8Q9x/nsehkqOc4jHOPnQdcfxD3H+ex6GSgAooooAqapcXVppVxcafbw3NxEhdIp5jEjY6guFYjjP8JrmD411W38Kya7qXhpraGSKFrKCO6NxNK8jBFWRY42K8spGzzDtPQN8tdRqWnwarp8tldmcQzDD/Z7iSB8Zzw8bBh+B5HHSqFn4T0my0SfSEjup7GdQrRXd9PcbQAAArSOxTGBjaRgjI5pa6j00/rt/wTD03x9eXek30154durW8glhht43juIobp5m2IFkngiYYb7x2EKCCN2cVe0zXvEV7e6rY3Og2Nrd2EUbJ/xM2eKd3yRhvJBCYHLFc5yNuACbI8IafFpN9Ywy30ovECk3+pXV2EYcqy75SykHByjKcgcggEZ/h7wINL0/UoNT1K4vJtSdGmmtrm6gcBBhcSNPJMD6nzMYwMAdWLsW9K128k8AjWJLOa7vI4ZXNsjiR5HRmG0FY13dOyZx/CTwcS18e63qHh3xFdQ6DbWt9pFuJo4rq4njWUFWYkrJBHKAAvBKAMcgNwSNy28CaFaaPc6VEmoNZ3JUvHLqt1JtKtvBRmkJjO7nKEEnrVzT/DGl6baXdvDFPOl6u24a9u5bp5VwRtLyszbcE4XOBk8cmjuONk1cz73xJq2i+FbnVdd0i2FwjRpBbaddS3Xms7Ki5IgVh8zDIVHOOmTxVbRfGOq6xpOqSf8ACOyWl9Yqrxpdfabe3uAQT8ss1uj5G05HlkDK8nPGnaeDdDs7C7tEtZZo71FSd7u6luJXVfur5kjM4C5JUAgKSSMEk0ReENMh0+7shPq7w3YUS+brV5I4AOfldpSye+0jI4ORRLZ2+Qo6WuJ4a1nVtbtUu9S0RdMtriFJrf8A0xZpCD1DqqgKehG1myDztPFT+JtWutD0CfUbGwTUJYSubdp/JLgsAcNtIzzwDgHuR1qtFYx+DtKEWhaVrGrIzhfs66h57xgLgEG6mAVRgDCnv061C5bxdby6XrvhnW9MtGAcyy3sMQYqQQoa2uGfPfsOOe1OVm/dFG6XvFa58ZahpWkapNrWixpf2LQqltYXEl0s3nMFj+YQhwd2QQI2IAyNxOKp2fj3VrvS7iZvDL29xb3NvE32prm3geOV9m9JJrZHYqfvL5eACDuNdC3hPSJJ9QlmhuJjqMSxXKzXk0isqj5cKzkIR1DKAc85zzTYPCWlwade2LNqNzb30flzpeapc3B28/dMkjFOvVSD09BSH0sQaD4sTxHrE8elxQzaWlskyXgldZGZnddhiaMbf9WxzuORtOMGtnUpbuDTbiXToYZ7pELRxzymJGPuwViPyNZklpb+GIp7jQvD91ey3cqedDYyRK3yoEDfvpEUAKoGAffHJNMhu7nxHHPp2q+HNa0i3kj+aeW8gjzyPlV7e4aRSfUYGAeexUldNRGt7s53w1q/iTT/AATN4l8U3kd0bi0hntbdrpGjLSAEKPLtEdSSyqF/fE5wMn72hpPjya/0PULq50WeC9s54rdbTEsYneUqI9pniiYAs4BLIAOcZrTtPBei2WiyaVCt81m4Tak2pXMpi2fc8tnkLRbcDBQrjA9KbZ+BtBsYb+OK2uZV1FQt39pvp5zLjoxMjsdw4Ab7wAXngYp2u7CKV74q13RdLv7vXfDccf2cRLbGwvXu0uZJH2BABEJBglcnyzwflDHirPhDxTd+IxeJqGi3WmS2xTDPDcLFMrA/caeGJiRg5GzAyvJzxPb+DNDt7a7hNvPc/bIxFPLe3k1zKyA5CiSV2dQCSQARgnI55q3pGg2eieb9jm1CXzsbvtupXF3jGfu+c7bevbGeM9BR6gzm9K8VapP4e1XxCY4rq2OqPBa2rv5Xk28cogJ3BGLMWVnwfUDI61ry+IL6BtdR9KE0mmRLLbx2szSNcqysRldgKtlT8q7/AGyeKbY+ELW1sda06eQ3GmapdSXC22Chg8wAyKrqc8vucEYKluOgqOL4f+HoYL2JYL0i/CC5d9TuXklKNuRi5kLbwQMPndgAZwAKqpZyfLt/wP6/EOvz/AxrP4k3j6NrF5qHh+S3m09oI4reM3DPcPM21f3b26TKu7A3eUc/Nt3FSKv6L4x1XWNJ1ST/AIR2S0vrFVeNLr7Tb29wCCfllmt0fI2nI8sgZXk541bbwlo9tp95ZeRPcRXoAuGvLya5kcD7o8yR2cAdQARtJJGCc02LwhpkOn3dkJ9XeG7CiXzdavJHABz8rtKWT32kZHByKiWzsNW0uZJ8X+IV8LSa4/hWKOIxRTW9u2pr5sisQCGwhVW5BUBmBB+ZkPFWG8WX9lperya3ptnaX2nLGyww3zzRTeZwg3+SH3FgV2iNjnG3cTirS+CdGXQZNGzqTWMjKxR9WumZduNoVzJuVRgfKCB7VJL4P0eee9mmS7eS+WNZmbULg48s5Rk+f92ykZDJg55zmnLd2JV9LnOxfEPVbjRJbm38Ng30F/BZyWs889srecVCOhmt0c8sAd0ajg4LYGVj+I95HpN5e3+g+W0agW0NtcPcNPJ55tyCFi3KPMHBUOSvO0N8lai/Dnw4pkIj1LdLNHcSOdYuy0kkZyjs3m5Yjjk5+6v90YtL4I0BRIv2SZklWVHRryZlIkfzG+UvgfP8w/un7uKX9fh/mP8Ar8f8iLwh4pu/EYvE1DRbrTJbYphnhuFimVgfuNPDExIwcjZgZXk543rxrpLOVrCGGe5C/uo55TEjN6FgrED32n6VirYJ4Ttnl0fTtc1p52VXhOqtcsgAPzD7XOFUdjtOTxwccEV3c+I45tP1Tw7rmj27puNw97DESQR8qvbXDSKfpgYBBPOCPXYFpuUovE2pW/w5j11LAX94v+stpLtVLfvChCyCJVJ9MqoPGSOtW7TXNbntdUgudIsYNWsUR44U1B5IJFYEjMvkhlPysCNh6Dk54bD4A0GDR59LjGp/Y52VnjbWLtsFWLDaTLlOTk7SM981abwnpbz38xOoCTUI0iuGXU7lSVT7u3EnyH3XBOTnqaTu00gW6ucz4W1XxNaeExrviSZtRl1CK2NjZx3ULKZJsADi2iMYy6jlpMDPJI+bcTUdc1G21XSbmC30bWUtfMtZ7W4+1xAOGCvl405DKcqVxjGCcnE9l4N0aw0WbSYEvHsZVVPKuNRuJvLC/d8tncmPGARsIwQCOQKF8LQ2Oj6lbaJd3VteX0JjF7eXM17JGcEKQZXJwpYkLkDJPqavS+n9f1/SGt9STwjrh8S+D9L1h4xFJeW6ySIOiv0YD2yDWzVLR9KttD0Wz0uwUrbWcKwxgnJwoxknuau06ji5tw2voRG9lcKKKKgoKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCOc4jHOPnQdcfxD3H+ex6GSo5ziMc4+dB1x/EPcf57HoZKACiiigCOc4jHOPnQdcfxD3H+ex6GSmTBig25zuXp6ZGe4/z2PSn0AFFFFAFXUrx9P02e7jtJ71oU3eRblA7jvjeyr78kdK5z/hYenx+HbjXb7TdT0/TY4Ulhnvoo4Rc7ztAXc42HcQP3mwc7gSvzV0WqWT6lpk9pFe3Fi0y7ftFsIzIg743qy8jjlT19eay7TwoIfDj6NeaxqWoQ4QQzT+SktvswUKGKNBlSoYFgeRzkcUu/9ev6D7f12/4Jm6P8T9B1vS9QurPzHmsGjSS0imguJHaU7YlVoZHjJdvlA38H720c1e07xdLqU2o26eG9Yhu9OjVpbeb7OrOzZKojCYqxIGc7tvON2QQFm8MXcmjXlrd6/qWqyyBJLZ7v7NEbeVDuRlaKAYO4KcsrjjoeQaHhzwz4ggtdVl1zW7iLUdQeMLcW00VxJCqDjDNbRx85Ix5XHXJJyGLsaVj4mjPgc+IdQSRY4oZJpkEQRl2EgjaHZRjHXeR3yB0yLH4m2mreGta1bR9HvbsaRGGeNLi1dZTtLELLHK8Z2gAsN2QCOCSBWhp/g2XTtBn0qLxPrTxyFWimJtkktiH3koUhUHcTyHDDHGMZqxZ+FIILfU01DUL3VZtUiEFzc3XlI5iCsAgESIoA3uc4z8x56YO442urkVx4xj0zwxd634g0q80aC22jy7yW23SliAu1klZBliBlmXHU4HNVND+I+leI9M1O50e3nu7jTQrTWUFxayyFWztKukzRHO1uPMBG05AyM3oPCSjTLq01DWtW1J7gJtuLmZFeDyzlDGsaIisrfNu25JxuJAAD08O3/wDZt5aXHivWLlrlQqzyRWYeAA87NsAXkcHcG9sHmiWzt8hR6c3zM/8A4WHp8fh2412+03U9P02OFJYZ76KOEXO87QF3ONh3ED95sHO4Er81RaP8T9C1zTb65sFmlnsnijezglguJHaU4jCtDI8Z3EEffG3BLbRzWnaeFBD4cfRrzWNS1CHCCGafyUlt9mChQxRoMqVDAsDyOcjik/4ReebT57XUfEmrX5kaOSGeZbZJLV0bcrp5cKgnIHDhgcYxgkFu13bb+vx/AFe2u5F4c8RX+seItYsr6wm09LOO3eO3uYAsqeZ5mdzpI8cg+QEMhGMlTyDW1q2pQaNo93qV2HMFrE0smwZOFGT14/EkAdSQOa56LTofB+oXGtalrHiDV5b5I7eRTYi5+5uKHZbQArgMwzgL83OTg1Z/tez8WW8+mWE2vaZMyiQXX9l3FmybWU/K88IQnttOcjPGM0t1oPS+pS0T4k6Trml6jf29peLDp7RxsUaG5E0j8LHG0Ekiu+So2hsjeucZq9D4ruZIb8zeFtctp7KJZTBMkH74EniOQSmNiACSN4IGPUUQ+DoBa363+p6hqF1feVvvpjFHNH5R3RbfKjRRsYlgdpOTzkYFOTwvciyvIp/E+tXFxdRiL7VIbfdCgOcJGsQiyckFihbB68DClfWwRtpcpf8ACw9Pj8O3Gu32m6np+mxwpLDPfRRwi53naAu5xsO4gfvNg53AlfmqrY/Fnw1e6DqmqeeIl0vy/tEJuraQ/vDiPEkcrRfM2Ry4xj5to5rYtPCgh8OPo15rGpahDhBDNP5KS2+zBQoYo0GVKhgWB5HORxTk8MzSaddWeq+INU1NZ9hjknFvG9u6ncrxmGJPmDBT827lR75p2u7f1/wfwFHZXMWw+Kuialouo6hYW11dNpjR/arW0lt7mREc4VwYpWRhwflVi3H3eRnotE106w1zFPpl7pd1bFfMtr3yi21hlWBjd1IOCOueDkDiqE/hO7vNIutPvvFmtXC3BQiZ47NXi2nOF224XnjO4Hpxjmr9hojWOuXuptqd5cteRxo0EwiEcewHBXagbPJzknr9MHURl6Xr+ra9oevX+lRWimG4nt9JEqsRKYvkLSYI4MquBjHy471reHNbh8R+GtP1i2Qxx3sCy7G6oSOVP0OR+FZOl6Bq2g6Hr1hpUtoxmuJ7jSTKzARGX5ysmAeBKzkYz8uO9a3hzRIfDnhrT9HtnMkdlAsW9urkDlj9Tk/jW9T2fK7eVvS2t/w+dw6/f/wDTooornGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUARznEY5x86Drj+Ie4/z2PQyVHOcRjnHzoOuP4h7j/PY9DJQAUUUUAVtQ1G00qwkvNQnWC3iHzO3PJOAABySSQABkkkAAk1Qs/Edtq8N2miZkv7ZQTaX8ctm4z90srx7wp5wwUjg4zg03xZ4at/Fnh6XS7p/LDOksbmNZFV0YMu5G4dcjlT1GRx1rK0PwbeaBJf3Gnz6DaXNzFHHEtjoK28Me1iSzqsu+QkHGPMAGM46130qeFdBynO076LW1tOy9eq2JbldWNDSfFcF34NTxBqsaafH84lRXMoUrIY8KQoLEkDAC5JIAGadD4z0Oa3vZmuZ7cWKo1wl3ZzW7rvyEwkiKzFiCAFBJPA5rJs/Bmr2/g+bQLjWNLuYmcPG0mj70IMxkdZEeZg6tkrxtI65zzWbZ/CO1g8N6npMt7EkV7NHcRQW1uwtYJY23BvJlkkzuOAy7gpUAADrXV7LLW5uVRr3tLJ25brur6K/wDW83nZfidlpPiHTdaknisZZVnt8GW3ubaS3lQHoxjkVW2nnDYwcEZ4NTaRrFjr2lx6jpM/2i1lLBJNjLkqxU8MAeoIrE8K+Dz4d1C7uj/Y8YuI0jWHStHSyRdpJLMdzsxOR1bAx05rb0i1vrPS44NW1H+0rpSxe68hYd4LEgbV4GAQPfGa4MRGhGbVGV1pb7tei6/0youXUu0UUVylBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAEc5xGOcfOg64/iHuP89j0MlRznEY5x86Drj+Ie4/z2PQyUAFFFFAEc5xGOcfOg64/iHuP89j0MlRznEY5x86Drj+Ie4/z2PQyUAFFFFADZZY4IXlmdY441LO7nAUDkknsKzbDxPoOq201zpet6bewW6F5pbe7jkWNR1LFSQBweT6UeI9Lk1nQZ7OHyTIWjkRLhd0UhR1cI4/utt2n2J4PSsK60HWtYt76fUdL0C0u5Ps5SOGR5/tJhk8wLNMY0Ow/dC7G25LZOdtHUfQ118aeF307+0E8SaQ1kZfI+0i/iMfmY3bN27G7HOOuOaa3jbwqsbO3ibRwi7NzG/iwN6lkz838SgsPUDIrntV8O+JNZke8udL0OO4kuLLzLYajI8bx28xlLM5twSxJ2hduAMncc4qhL4H8RJHbQwWWiTwpeGaVZb2RAyi++0g4EDAsQcEHoecmhatLz/pidkjt4PFGgXNvdT22uabNDZxLNcyR3cbLBGy7ldyD8qleQTwRzSx+KNAm0kapFrmmvp7SCIXa3cZiLk4C784znjGc1xyeDfEdvZT29p/Z0IkE4LxXskburXZnA3CLKF1ZlLjJQ8ruPSG38F+I4dF8Q2bWGlSjVnhdYbrWrq64UBWWSWaJmYFVGOMc7cDGSbq/wDW/wDkHX+ux3Fn4k0PUdLn1LT9a0+6sLbPn3UF0jxRYGTucHAwCCcnpUOieKdN8Q3t5DpEq3UNqkbrdwTRSwzq+4fIyMejRspDAHI7jmsbUPD+t32oapN9k0tY5Gs5rVGunYTPA+8pIPKG0HoGG4jAO09KseHdK1y38W6vq+safpFmmpQQKRYXTyvvi3gFi0Sbsq/3uCNoGD1p/wBf1+IdDqqKyPEvhy18UaT/AGffPsi8wSZ+zQT8jP8ADNG69+u3PvWZ4X+H+meE9QlvLCXfJJF5RH9n2UHGQfvQQRsenQkj2q4xg4tuWvawnfodFe39np0KzahdwWkTOsavPIEUuxwqgnuTwB3qrL4k0OBbhptZ0+MWsywXBe6QeTI33Ubn5WPYHk0niKwuNT8PXdpYmIXLoDF5xIQsCGAYgEgEjGQDj0Ncpf6D4t1W110X2naA51OOCNLf7fKY1VGO5XYwfMCDnO0dduONxy/r+v6/4LOtsPEOi6rp02oaXq9he2UBImube5SSOPAydzKSBgHJz2qC28X+Gr21a5s/EOlXECyJE0sV7Gyh3OEUkNjLEEAdz0rH1DRNdu5fEGNO0h4dQt4IoY5L6YeZt4cOViBj4Y7WUsQQpx2FHTvBeq2+j68DHa295qIi8mKTU574Fo+QXuZUEuDwNuCEAyOWIph0Oyi1fTZ2vBBqFrIbFit2EnU/ZyBkh8H5TjnnFQaf4m0LV7eafSta06+hgUvLJbXaSLGvqxUkAcHrXNv4f8S6t/bLazb6NF9tW2eCGK4knTdC+7ypd0a7kbuwA4YjacZae50DWdatdRk1TTdAsbqdYAiwM9z9oMUnmBZpWjjJjONuzYcAscnO2j+v6/yGbdn4q8PajZz3en69pl1bWyGSeaC8jdIlHVmYHCjg8mpbHX9H1TTZdR0zVrG8sYd3mXVvcpJEm0ZbLgkDA5PpXNaj4Z1fxHp+p/2tY6Ppt1cLB5ZsLmZ5JjFJ5iiS4VYnCZG0BVyuSwY5wK1l4Q1G30fXE/sWwS91GOKI/afEd9fLcIpOVaSSMPEQrNtZMkE57UdGGh1en+JtB1a0uLrS9b069t7Zd081tdxyJEOTlipIUcHr6Gorfxd4bu9NudQtPEGl3FlaKGuLiG8jeOEHoWYHAzg9fSuYsPBuq2ui6+rWVgbnVBCgtLnVrm7VlThi126CYEqSFABCFQw5Y1a8PeG9Y0S11q6S0sxe34jWK2m1ee7BKAjc91LEZeQ3CbSq7ePvNRK6Tt2BdLk/hz4laF4kt5rmCWO0tImjQXNxfWhV2kOEXEczsjHsrhT2xnIrebWbWWK+/sx01S5sTsmtLOaNpVfGdh3MArezEVz1joviJvh5/YWoW2l293a2kMFs8F7JNHM0YHLkwoUBKjoG6+3Mun2/i631DVtRu9P0RprqOEW9vFqMoVShIKvIYMnhi24L1+XbxuLdr2X9f1/XmvM3dE1aPXNGt9Rhgmt0mB/dT7d6EMVIO0kZyD0JFT31/Z6XYyXmpXUFnaxDMk9xII0QZxkseByaxfBdlrml6P8AYNetNPh8ksYpLK9efzNzsx3Bok24yBxnPPStXWFvX0e5TS4bee7dCscdzO0MZzwcuqORxk/dPpx1qZXSuhxs2kyKLxHocy3jQ6zp8i2CCS7KXSH7OpGQz8/KCATk44pbPxDouo6XNqen6vYXVhBu827guUeKPaMtucHAwOTk8CuXg8G3l18PBoF3pul6VPbJElv/AGbdy7JPLdZOXWOJ4wzrztyRnOSaqw+D7y10fVp5tMgtryRra4ydcvNU+1fZ5PMWN/Nj3KvGPkDE7jwcAGnZN/18/wDgCjqkdXb+L/DV1bS3Fr4h0qaCGLz5ZY72NljjyV3sQ2AuQRk8ZBqzYazZ6hoaatHNGlm8Zl80zRsgQZy29GZMYGcgkVyXhO41u8m8Q63DoNra3V5JAIoJnuLZJmRdrMzy26yZwR83lYOABnBNTWGieJV8CXeiahpuitKQREo1Kdo5g8hZwzCFGjIU/Ky7jnB4xSdwRtf8Jv4WOj3eqx+ItLmsLNQ1xcQ3aSJHnoCVJ5JBAHUngVLbeLfDl5pU+p2mv6XNYWxAnuo7yNooSccM4OF6jr61z2i+DL6DSvEUV4sNnLrFv5CqNQm1FlxGy72nmVZG+8MJ91duRyxrVjHiyXRL1byy0ZbpolitrWK8lKHjDO85iyM54URcbfvHd8pLROw1ZtXNBPEuhSpePHrWnOtggku2W6Qi3UjIaTn5QQM5OOKifxb4ej8PTa9/benvpMOfMvYrlXiBHGNykgnOBgckkCsGz8NaxF4AGjRadpOk3dvHHDD/AGddvtljDK0g8wQq0JchvmVWKltw5qDQfB+sW2n+JrTUEtbaPWYdsTLqlzfyRsYjH87zqGYYwc5HptGMluyb/r+v6+ajra509p4r8PX+l3GpWOu6bcWNr/r7qK8jaOHjPzsDhePWp7DWbPUNDTVo5o0s3jMvmmaNkCDOW3ozJjAzkEiqFlJ4sXTriS8sdG+1KqLbWkV7KUOPvM85iyM54URHG37x3fLi2Hh7xC/gt9H1Wz0hJbeZJ7cR3sk8VwRN5pSQNCu1TgLkBuuccYI9xRvZXOhg8WeHLnTW1G21/S5bFZRA11HextEJDjCbwcbjkcZzyKs6Vrela7atc6HqdnqUCPsaWzuFmVWwDtJUkZwRx71ymqeH/EOs2Gry3Wl6LHeX8EFr9kF/JJDJGjsxaSQ24JOHKhdhGM884G7pdjqkHijVLy8trKKzuYoFhaC5Z5CyA53KY1AHzYBDHoOmeENm5RRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUARXLiOJSSR+8Qce7Aeo9f/ANfSpaKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAo6romla7bLba5plnqUCPvWK8t0mVWxjIDAjOCefeq2leEvDmhXTXOiaBpem3DIUaWzso4nKkg7SVAOMgce1FFGwGvRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAf/9k=)

Above is the code from the GUI which shows the solution to the problem. In the “elif” we can see the username from the database is checked against the username entered by the user. If these match the password entered by the user is encoded in UTF-8, hashed and checked against the hashed password stored in the database. This solution worked exceptionally well and allowed the project to move forward to developing the GUI further and integrating the GUI with the face recognition software.

A screenshot of a cell phone

Description automatically generated

Image of the GUI pages

## Integration of Face Recognition into GUI

The integration of the face recognition code into the GUI code went well with very few hitches. The reason for the easy integration of the face recognition code with the GUI code is that the original face recognition system was designed using the Single Responsibility Principle(SRP). The system had been separated into modules that had only one responsibility, face detection, face training, face recognition. This meant that when it came to integration the system had all code that would change for the same reason grouped together. This resulted in very small changes to integrate the complete design even though they were created months apart.

The last piece of the jigsaw was to divide the face recognition code into two separate functions. These functions were imported into the GUI software.



The register new face takes in a parameter username, this parameter is passed to the function only if all security checks return true.

![A screenshot of a cell phone

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDyRXhpZgAATU0AKgAAAAgABAE7AAIAAAANAAAISodpAAQAAAABAAAIWJydAAEAAAAaAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGpvaG4gbGF3bGVzcwAAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzQzAACSkgACAAAAAzQzAADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIwOjAzOjI3IDE1OjA1OjE4ADIwMjA6MDM6MjcgMTU6MDU6MTgAAABqAG8AaABuACAAbABhAHcAbABlAHMAcwAAAP/hCx9odHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIwLTAzLTI3VDE1OjA1OjE4LjQzMzwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5qb2huIGxhd2xlc3M8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgBTgJ9AwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOIfx5PPr3h9bKymt9I1SaVPtN5bYE8awvIssbLJ8g+QfLIgZg2QODTtG+K/hrXNbttNs5mzeMy2spmgYTEc42JI0keQCR5iL0wcEgGcfD22V9NA1vVjb6XLvsrZmhaOFCjIY+YsspRiuWLMB0Yck2dM8GJpMtklrrusf2fYOTbaaZ0EKLghUJCCR0UHhWdhwOuK9ucsucHZa203XWXk7v4fLfyMnz/wBf1t+JmaT4o8S634xvbOHSjYaXYXYtpmnt4pW4iWQ5dbobSd4xtjcDIyeoXuK57RvCb6Lq11fJ4g1S6+2TGe5guFttkr7FQE7YVYYVV4BA47856GuHGTpSmvZJWSWye/W99d/wLV7u5k2mvfa7toBpl/F5chjmkkRNkRxkZIY5z6rnGRnFVdM8a6Tquox2dvJh5gxhYzRN5mBk/Krll4BPzBenrxV7TtIl0+4kkbVby6WQlmjnWEKWOOfljU9vXFFnpD2UqCLUrxrWMEJZuIjGq4wFzs34Hb5uwof1f3lbppq/8v8AgW69T0n9V95W6aavfXy728rdb6kNt4iS5vLaFNOvlS7y0E7KgSRAM7/v5Axg4IDHPTrizp+oSXl5eQywSW5t2UBJFXJBGc5V2BB/Aj0rN07RNQtdZiea5J0+0V0tYjOHwpGAMCJSMDuWc/mTVyy0WWz1B7ttXvrgyY8yOVYQr4GBnbGCMZ7EUVI0VdRa28+/+W/mOtHDq6i1t5vW/wDlv5lez8WWd/rbaZbwzNMrOCwaJguzgllDlkGcD5lHJFWNK1mfUtxl0m6s0V5FMsskTLlHK4+Vyc8emOvJ4JW00M291BNcale332cEQrcmM7CRjdlUDE4yOSep780+30gW1+bhb27MW95FtS6iJXckseAGbkscMSATkDgYVR0LNQXTz8/Ty30JqPDWagunn57bb6bq2/zoaX400rVtSjsrZ8PNu8lvNibzMDP3UcsvAJ+YL+fFW7PXftl4bddLv4ikhjlkkRNkTYyMkMd2f9nOMjOKks9IexmTydTvDax5CWjCIxquMBc7N+B2+bsKjtNOfSDPcy6pf3cZ3SPE8UbZOByBHGGJ46D8qc/YNvkXTTV7/d/T69B1PqzcvZrppq3r93bptfrbQ1GJVCQpYgZCjqfbmqFtrCXL2YFrcIt3GXV32YQjqrYYnP0yPemRa/ZzTJEkOoBnYKC+m3CqCfUlAAPc8VFb+Hhb3sFwupXzrbsxigZo/LRWzlcBMkc9SSRjg8nOUaaimqqs+l7+f62Mo0lBNVlZ9Lp9n+tiKz8WWV9rbaZBFMZVZwzh42CBOCWUOXQZ4+ZRyRTNL8aaVq2pR2Vs+Hm3eS3mxN5mBn7qOWXgE/MF/PirlpoZt7qCa41K9vvs4IhW5MZ2EjG7KoGJxkck9T35p9npD2MyeTqd4bWPIS0YRGNVxgLnZvwO3zdhW0vq1nZdNNXvr5enZbm83g7SUVrbTV76+Xp2W+vUjs9d+2Xht10u/iKSGOWSRE2RNjIyQx3Z/wBnOMjOKrLrtxcePn0K0jiNtZ2Aub2VgSweRysSLzgcJITnP8OO9XNO0iXT7iSRtVvLpZCWaOdYQpY45+WNT29cVTXQri38fPrtpJELa8sBbXsTEhi8blonXjB4eQHOP4cd6yl7LnfL289/n+HyOTEezv8AuvLv313/AKt5h4b1241S81nT9SjijvdKvmgbyQQrxMBJE4BJ5KMAf9oN2xWjbal9p1a9sPsV5F9jEZ+0SxbYZ94J/dtn5sYwfQkVneG9CuNLvNZ1DUpIpL3Vb5p28kkqkSgRxICQOQign/aLdsVo22m/ZtWvb/7beS/bBGPs8su6GDYCP3a4+XOcn1IFZz5Lu3Zffpf9Tn7+pdooorIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUhGVIyRkdR2oAybDxRpeo6o2n273UdyoYqtzZTQLKFOGMbSIqyAZHKE8EHoarQeOfD9zqsGnw3czTXMrQ28hs5hDO6glhHMU8t8bWztY9DWDoPw2n0PxLZ6rFe6Rm3WSOV4dG8u5vFcctNOZSzyZAO7ABJbK8jbt6ToOuabJa2h16B9Hsxsht49P2zvGBtRJJS7KQBjlY0J2jkc5P6/r+mD6j4PHPh+51WDT4buZprmVobeQ2cwhndQSwjmKeW+NrZ2sehqXT/ABfo2qaitlaTz+bJu8lprOaGO429fKkdAkvGT8hPAJ6c1W0rQNb05rWzbXoH0azGyK2TTws7xgbUjklLsrADGSsaE7RyOc5Xhz4bweGtat7mztvDf2e23CKRNAWO+2lSBm5WTBbBwW8v5hnPJzQvMH5G3beNNDu9bi0qC4uPtU7yJAXsp0imZAS+yUoI3wFPKselFt400O71uLSoLi4+1TvIkBeynSKZkBL7JSgjfAU8qx6Vk6F4DuNM8TR6ze3WkPcJ5hkn07RhZXF2zjB8+QSMJBzuxtGWCt2xSaF4Cn03xLHrF9daRJcL5hkn07RhZXF2zjB8+QSMJBzuxtGWCt2xR2/rUH1sasHjnw/c6rBp8N3M01zK0NvIbOYQzuoJYRzFPLfG1s7WPQ1LYeMNF1PUlsbO5maSQsIZXtJkhnK/eEUzKI5CACfkY8AnoDVbS/D+t6ebayfXoH0a0GyK3j0/ZO8YBVY5JS7KQBjlURjtHI5zDZeFdUjNjaanq9pd6Tp3/HtBHYNFOQEaNBJL5pDYVudqLkgHgZBOn9f1/XQel/6/r+uoRePtOvvEml6XpYM6X7yqJ5Y5oAwSNnDwl4wk6fLgsj4G5TyGq/p/i/RtU1FbK0nn82Td5LTWc0Mdxt6+VI6BJeMn5CeAT05rJg8Ja+l7o32jxDYzWOjyfuYP7KZZJY/LaLEj+dgtsY/MqqM87SOBT0XwDYeDtQj1Lb4Xt9PsFdvtTaGkN3HHtIy10JQuQDy2wZGc9c09L+QdNNzvax9L8RW+q3mpwwxyqNPZVcSW88chyCc7HjXI4OCpYN+lV7fx94OvLqK2tPFmhzzzOI4ootShZnYnAUANkknsKraXoPiay8QXeo3eu6TOl4qiWKLSJIyNisE2sblscnJyDnGBjrS1EWtF8b6F4guo7fS7i5d5FYxtNYTwpIVOHVXkRVLKeCoO4YORwcZmm+O7jWfGU2kado0wsreYwy3l1DdwsGEYc4BtvL/iAG6VSeoGCuZ9O8L6zZDQ1m1mxmXTJJXl2aa6GcOCAF/fHZgMeTuzx0qbRNB17TtWv5tQ1jTbuz1Cdp5oIdMkhkDGNYwFczsAMIuflOeenY6/1uHQtWHjDRdT1JbGyuJnkkLCGVrSZIZyv3hFMyiOQ4BOEY8AnoDWxNNHbwPNPIkUUal3d2AVVAySSegrmtN8M6tbTadBf6xaXWl6W2bWCOwaKY7UKJ5kvmsrYVucIuSB0GQelmiWeB4nLhZFKko5RgCMcMCCD7g5FPQOpi23jnwle3UVtZ+KNFuJ5mCRxRahEzOx6AANkn2rdrCtvB+mWl1FcRXOtM8TB1Euu3sikj1VpSrD2IIrcIypGSMjqO1XU9n/AMu7/MSv1Mmw8UaXqOqNp9u91HcqGKrc2U0CyhThjG0iKsgGRyhPBB6Gq0Hjnw/c6rBp8N3M01zK0NvIbOYQzuoJYRzFPLfG1s7WPQ1g6D8Np9D8S2eqxXukZt1kjleHRvLubxXHLTTmUs8mQDuwASWyvI27ek6DrmmyWtodegfR7MbIbePT9s7xgbUSSUuykAY5WNCdo5HOc/6/r+mN9R8Hjnw/c6rBp8N3M01zK0NvIbOYQzuoJYRzFPLfG1s7WPQ1Lp/i/RtU1FbK0nn82Td5LTWc0Mdxt6+VI6BJeMn5CeAT05qtpWga3pzWtm2vQPo1mNkVsmnhZ3jA2pHJKXZWAGMlY0J2jkc5yvDnw3g8Na1b3NnbeG/s9tuEUiaAsd9tKkDNysmC2Dgt5fzDOeTmheYPyNu28aaHd63FpUFxcfap3kSAvZTpFMyAl9kpQRvgKeVY9KLbxpod3rcWlQXFx9qneRIC9lOkUzICX2SlBG+Ap5Vj0rJ0LwHcaZ4mj1m9utIe4TzDJPp2jCyuLtnGD58gkYSDndjaMsFbtik0LwFPpviWPWL660iS4XzDJPp2jCyuLtnGD58gkYSDndjaMsFbtijt/WoPrY1YPHPh+51WDT4buZprmVobeQ2cwhndQSwjmKeW+NrZ2sehqWw8YaLqepLY2dzM0khYQyvaTJDOV+8IpmURyEAE/Ix4BPQGq2l+H9b0821k+vQPo1oNkVvHp+yd4wCqxySl2UgDHKojHaORzmGy8K6pGbG01PV7S70nTv8Aj2gjsGinICNGgkl80hsK3O1FyQDwMgnT+v6/roPS/wDX9f11CLx9p194k0vS9LBnS/eVRPLHNAGCRs4eEvGEnT5cFkfA3KeQ1X9P8X6NqmorZWk8/mybvJaazmhjuNvXypHQJLxk/ITwCenNZMHhLX0vdG+0eIbGax0eT9zB/ZTLJLH5bRYkfzsFtjH5lVRnnaRwK/hz4bweGtat7mztvDf2e23CKRNAWO+2lSBm5WTBbBwW8v5hnPJzT0uD28zua5228d+H7vVoNNtrm5e5uXZLf/QJwk+3O5kk2bXQY5cEqMjnkZ1ILO9XVL+a71D7RZXCotvaeQq/Z8Ah/nHLbiQeemOK43Q9F8SReKdOs76L/iRaJHLDazSWsUZmTZ5afMtxIznb6xxDqSM4FHUXQ6Sw8YaLqWoLZ2s9x5sgYwtLZTRR3G3k+VI6BJeMn5C2QCenNM0HxlpviK4eGxt9VidHkQm70q4gT5GKn53QJnI+7nd6gEEDF8OfDeDw1rVvc2dt4b+z224RSJoCx320qQM3KyYLYOC3l/MM55Oa2tO0LVtP1eRhrUf9km4muVs0sgJWaRixV5S5BQMzEBUQ/dyxAIYVgfkTtr4bxknh+3tzKyWRu7qffgQAttjXGOSxDnqMBPcUaBr41p9Sgltza3em3r2s8JffwMMjg4HDIyt04yR2qnFpF7ZfEq41aCIS2Gp6ckNw+8AwywuxTg8kMsrdOhTnrR4W0m9tdU8QatqkQgn1W/3Rwhg2yGJBFGSQSMsF3+24DqDW7jT5fl+N/wDK/wBy7ie/z/T/ADOkooornGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFNd0ijaSRlRFBZmY4AA6kmnUUAULfXtHu5rWG01WxnlvIjPbJFcozTxjq6AH5l9xxUcHibQbrWH0m21vTptSjYq9nHdxtMpAJIKA7gQAe3auc03RPFNrqmkRzwaP8A2bpk83zpdymaZHDgPtMQVSAwymWzn7wxhq2g+DNVg8T2epalZ2NktsZ3Y2ur3V0jtICCI4JkCWwJO7MZzxt6MaAOusfEWianqU+n6brFhd3ttnz7a3ukkkiwcHcoOVweOe9Qw+LPDlzLDHb6/pcsk/meUqXsbGTZy+0A87cHOOneuU8NeA7vQ9Z003NhFcwaZvW3vn8R30jKCjJuFo6mJSQcEB8DJI7CtPw/o+tWmsardapoOgxG/n+1G4t755ZWdVCxq263XIGD82eM8Cj/ACGza03xT4f1m8Npo+u6bf3Kx+aYbW8jlcJx821STj5hz05HrSQ+K/D1y96lvr2mStp6s14qXkbG2CnDGTB+QAg5zjGK5/R/D2vWEfh9H0/R4BYzXEl2YLyRseZu5QeSN5Jcsc7eR1Oc1k6f4N8VWGuWuotFYXUttbzwvJc69dyfaGkUfOsRiMcALKMoikAHg/Lhk720BHa2fi3w5qNzDb6fr+l3U87MkUUF7G7SMoDMFAOSQCCQOgINTWPiLRNT1KfT9N1iwu722z59tb3SSSRYODuUHK4PHPeuI0fwRrulrYkWmjo1vbafC4iu5FybcyB8HyehEm4f7WRx96pvDPgO70PWdNNzYxXMGmb1t75/EV9IygoybhaOpiUkHBAfAySOwqrK/wDX3k9D0KsvTvEWl6rLfjT722uI7B9k80VzFIitjJB2sSpGOQwX8a1K5LSLbxZB4n1C+v8AS9Gjt75YwfJ1WWR4zGjAcG2UHJIzyMDPXpUjNjTfFPh/Wbw2mj67pt/crH5phtbyOVwnHzbVJOPmHPTketY1l8SdD1TxefD2kOt/OrmN57e9tSqsF3NhDMJWAHBKoRnI7HFfRvD+u6evh5JNP0eFbGe4kuzBeuceZu5QeQNxJck528jqc5q74fs/EtrrGqHVLDS4bPUbprjzbXUpJJYv3SRgBGgUH/V5zuGM98cvr/W4dGbFr4h0W+1afS7LV7C41C3BM1nFco80WCAdyA5GCQOR3rRrjdI8P61bvothe2OkwWGiuWiu7W6dppsRvGP3RiURlt+5sO3cc5yOsu7ZLyxntZDhJo2jY7VbgjHRgVPXoQR6g09A6k1Nd0ijaSRlRFBZmY4AA6kmuG034S6LpeqW19BPmS2lWVB/ZOmpkg5+8lqrD6qQfQiu7q6kYR+CV/kJXvqULfXtHu5rWG01WxnlvIjPbJFcozTxjq6AH5l9xxUcHibQbrWH0m21vTptSjYq9nHdxtMpAJIKA7gQAe3auc03RPFNrqmkRzwaP/ZumTzfOl3KZpkcOA+0xBVIDDKZbOfvDGGraD4M1WDxPZ6lqVnY2S2xndja6vdXSO0gIIjgmQJbAk7sxnPG3oxrMZ11j4i0TU9Sn0/TdYsLu9ts+fbW90kkkWDg7lByuDxz3qGHxZ4cuZYY7fX9Llkn8zylS9jYybOX2gHnbg5x071ynhrwHd6HrOmm5sIrmDTN6298/iO+kZQUZNwtHUxKSDggPgZJHYVp+H9H1q01jVbrVNB0GI38/wBqNxb3zyys6qFjVt1uuQMH5s8Z4FH+Q2bWm+KfD+s3htNH13Tb+5WPzTDa3kcrhOPm2qScfMOenI9aSHxX4euXvUt9e0yVtPVmvFS8jY2wU4YyYPyAEHOcYxXP6P4e16wj8Po+n6PALGa4kuzBeSNjzN3KDyRvJLljnbyOpzmsnT/Bviqw1y11ForC6ltreeF5LnXruT7Q0ij51iMRjgBZRlEUgA8H5cMne2gI7Wz8W+HNRuYbfT9f0u6nnZkiigvY3aRlAZgoBySAQSB0BBqax8RaJqepT6fpusWF3e22fPtre6SSSLBwdyg5XB4571xGj+CNd0tbEi00dGt7bT4XEV3IuTbmQPg+T0Ik3D/ayOPvVN4Z8B3eh6zppubGK5g0zetvfP4ivpGUFGTcLR1MSkg4ID4GSR2FVZX/AK+8noehVkx+K/Ds2otp8WvaY96k32drZbyMyLLz8hXOd3ytx14PpViBdUbVL9bw2g04qgszCW84HB8zfnjrjbj8a4Hw9bal/wAJdoukzaPClv4egnt/7RjhuUMqlAgO6SFEG4gEiN5MnnOBkltbD6XO6s/Eeh6jqFxYafrOn3V5a5NxbQXSPJDg4O5Qcrg8HPeq2j+NPDHiG6a10PxBpt/cqXzBb3SPJhTgnaDnGe+MHgg4Ncx4Z8B3eh6zppubGK5g0zetvfP4ivpGUFGTcLR1MSkg4ID4GSR2Fb+kWPiHT9VmtmTTF0g3U9z54lke4lEjs4Ty9qqmGf7298hfujdlRWB+Roya5ap4mh0JVlku5bV7piigrFGGCguc8biSBwc7W9KNG1y11yO7NsssUlndyWk8UygOjoe4BPBBVge6sDWPBZXVh8V7y8a3lltNV0uJFuFQlYJIHfKMf4dwmBHrtb0o8IWV1/bfifWLm3ltY9R1ELBDKhVikMaxeZg9NxRiPVdp71u4Q5b+V/nfb8/uE9/n+n9fkdVRRRXOMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooqG8+y/YZ/7Q8n7J5bef5+PL2Y+bdnjbjOc8YoAmorxz4YWNhe6Xod34d8OxwXFq5F1rkF1byJMgRv3LMkhlOQ0Y8t0ATAx9xc3vCXhnWtH8c2upSaFfW3npMmpzINOhgaR8MGUQnzpEDA4MrM4yOCSxDtrYHoeq0V5b4Z8HXMfxAttel8P3+nSq9xLcyXz2LpmRWGI5YP30py2AZsgLu6Nio9A8DvL44i1a+8N3tnv+0tenUWsZI2EqspVJIP30p+bA8/I2bujYpa6D0Vz1aivNNC8G6xHqGjWuqafZrpOkys9mvkwZtGiQxo67ecys3m9TjYAcE1Y8PeHZbPWdMkXwnLpmpWzMdR1xpLdvt42MGy6yGWQO5VgJFGMZ4KgU9P6/r+ulxHodZ+pXWnNNb6RqkXnDVBJEkMls0kUuELMjHaUGVB4YjODjODXFeFvC9rFr99cyeCLrRp2uHnsb2V7Zo7UFAu2NI5mMZYliwVQG/iJqppHh25sPEXh25tvA1xY3NnMw1TU0ubbF2WheMynEu+b523bpFDgE4GWIpLW1/6/4bqPZs7C38A+DrO6iubTwnocE8Lh45YtNhVkYHIYELkEHvW3HcJJcSwqJA0WNxaJlU5GRtYjDe+Ccd6lrg/DdvcWHiTWfL8CXmmafqmxzg2IjLqjby6xzkkuSBnBznnHWi7Ed5WT/wAJLpx1xtIT7ZLdK4R2isJ3hjYqG2tMqGNTgg4LDqPUVxXhTwv/AGDdeH7ix8Fy6bdF7hNQuEa1DpG2diyMspLJnYVVdwUKBgYAqz4S8P29j4q1S7Pw/wD7Imubt5rXUzDY/uUMKKRmOVnG5g5wBj5uepo6h0PQqK888PeHZbTWdMkXwnLpmpWzt/aOuNJbt9vGxg2XWQyyB3KsBIoxjPBUCu9u4XuLGeGOTy3kjZFf5vlJGM/KVP5EH0I60w6k1FcJpvgPWrLVLa6n8SefHDKrvF5mpHeAc4+e/df++lYeoNdrefZfsM/9oeT9k8tvP8/Hl7MfNuzxtxnOeMVdSMI/DK/4CV2yaivHPhhY2F7peh3fh3w7HBcWrkXWuQXVvIkyBG/csySGU5DRjy3QBMDH3Fze8JeGda0fxza6lJoV9beekyanMg06GBpHwwZRCfOkQMDgyszjI4JLERbWw3oeq0V5b4Z8HXMfxAttel8P3+nSq9xLcyXz2LpmRWGI5YP30py2AZsgLu6Nio9A8DvL44i1a+8N3tnv+0tenUWsZI2EqspVJIP30p+bA8/I2bujYpa6D0Vz1aivNNC8G6xHqGjWuqafZrpOkys9mvkwZtGiQxo67ecys3m9TjYAcE1Y8PeHZbPWdMkXwnLpmpWzMdR1xpLdvt42MGy6yGWQO5VgJFGMZ4KgU9P6/r+ulxHodVLnU7W01CysrhpFmvmdYMQuysVUsQXA2qdoJAYjODjODXC+FvC9rFr99cyeCLrRp2uHnsb2V7Zo7UFAu2NI5mMZYliwVQG/iJqppHh25sPEXh25tvA1xY3NnMw1TU0ubbF2WheMynEu+b523bpFDgE4GWIpLW1/6/4bqPqz1CiqMFzqE2qX9tNYfZrWFU+y3nnK/wBoLA7vk6rtOBz1zXlfhq00afx/o+nR6JbLreliZtVvo7m2maacJt891jdpNxckh5VVl34H3iKdmnYXS57FTIp4p1LQSJIqsyEowIDKcMOO4III7EV5d4N8D3Oh+KrW5vLHXBdx+Yl1fqNMW1vCVOWkaNVuZFY4YBwSGwT0zWz4V8NQeHvEU0Vj4HsbI/abhzrUa28YMDuXRE2ZlJ5VSrKigKcE4AIkD0Owl1Wyh1i30qScC9uInmjhCkkohUMxIGAMso5xnPFGm6rZavBLNp83nJDPJbyfKVKSRsVZSCAQQR+PUcEVzVuDZ/GW9N4Cf7R0eEWLnpiGR/NQe/72NvfPtR4IBuNe8XalbjbYXWq7IPR3iiSKVx7F0Iz32Vu6a5b+V/xtb+uzE9/n+lzsKKKK5xhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFNlljgheWZ1jjjUs7ucBQOSSewp1NlkEMLyOGKopYhELEgegHJPsOaT2AzP+Eq8Pl7FRrumbtR/48h9sjzdc4/d8/PzxxmtWvLPDFnqENx4W+x6ffCwa4mnlFzaywmznFtKrBlkAYI5ZSCBjdv5ywq7ojSHUbA+Z4oOqneNZF5HeC2K+W2/y8jyQQ4XZ5GD0xkbqf8AX9f5/Mdv6/r8vkejU13SKNpJGVEUFmZjgADqSa828KiSwuNAnWbxVNcXzXEV1/aSX0saKM7N6SDbHghAGIUsMks2WNVfB9prkfie2XXda1OSeQSpqFlPouoNbzEqcjzpJpLZVDcho1UEYUYB20b7C8zubDxv4U1W+jstM8T6Ne3UpIjgt9Qikd8DPChiTwCa3K4vw7LY3njDW4Bpl7FGkkL2j3GlTwRL5cQjJjd0CgjJAwckZIyKgstKmuNWTw5Ndat5GkTPdSXf9pXAe4jl3eTG0gfc2MvkEnHkrwA4oWuwep3dRXNul3ay28pkVJUKMYpWjcAjsykMp9wQRXl3gSaae20KW2u/FE+rOT9uk1L7W9o8O1v4nzAeibSh3k43HJeneDLfXrfxXaHWNc1Q3TeYt9aS6NqBglbaSQJ5JntkAYAq0aqCAFGM7adr6A9DtLfwVpdrdRXEV1rjPE4dRLr99IpIOeVaYqw9iCD3roKhu1mexnW0bbO0bCNtwXDY4OSrAc99p+h6VxWm6V4/i1S2fUdV8y0WVTMn9p277kzyNo01CeOwdfqOtaQhzp6pW7iemp3dFBGVIPf0OK4DwjYWmja9caeln4le8N3dF5Li5vHtEhaRnR90r+U5IKj5Nz5JJAwxGS1G9rnf0VwllpU1xqyeHJrrVvI0iZ7qS7/tK4D3Ecu7yY2kD7mxl8gk48leAHFYngSaae20KW2u/FE+rOT9uk1L7W9o8O1v4nzAeibSh3k43HJenYH1PVqrXWo2ViwW9vLe3YxvKBNKqEogBduT0UEZPQZ5rzTwZb69b+K7Q6xrmqG6bzFvrSXRtQMErbSSBPJM9sgDAFWjVQQAoxnbXQRyLD8YtTl1EMSmhxNYBULExiV/P2gZJbd5WQOfu+1aU4cz17X/AOGB6X/rrY6D/hJ9B3Wa/wBt6buv+bQfa483HOPk5+bnjjPNaleL+D9H1ix/4QuK2029WyS6lnuFubWWI2VwLaZW3BwDscupBHG7dzlq29DOsi70YS/8JF/a5Z/+Eg+1ib7Js2Nv8vP7rO/Zs8jnHX+KvXxGWU6cmqc72v8AnL7vh+9ro7mam/6/rfy+R6bRXlGgTazZw+HLhW8RzXlxd3VtdpfQ3jxxw7ZDEXVxgfMYf3h+bBb5sAgM8JjxPH4g0l9W1DWjfySump2zaVdGDOGLZlknNuE3AbWhUE8AAAkVEsr5VJuotL9Hra/3LTfzXcXtNNj1W2u7e9hMtnPFcRh2QvE4YblJVhkdwQQR2IIqWuO8E/8AIxeMfsn/ACDv7WHk4+753kp5+P8AtpnP+1u75rsa8mpHldvJP71c17hRRRUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFNdS8bKrshIIDLjK+4zkfnQA6ivOtF8awT634f0+XxUtzfOZ7W/s28kZkUNtMm1AUlJXhQV3ANhTgkUfDXivVdX8f20dv4r0/VtOne4drKxuIZWjQKxj8yLyEkt1BwCWkkO7avfINwPU6K8r8GeKfEN94qtINY8R6E7z+YLvR31OM3MDBSdkcAto5FZSMEPI/y5OTw1aPhLXf7a8SXcFv8RLXVFtLphBYRC0aSeHaNzSFACwBbCsgTGPm301q7eVwen9f12PQ6yrPwt4f043Z0/QtNtTeqVujBZxp54Och8D5hyeuepq/dtMljO1ou6dY2Ma7Q2WxwMFlB57bh9R1ritN1Xx/Lqlsmo6V5do0qiZ/7Mt02pnk7hqTkcdwjfQ9KuFPnu7peoN2R12l6Npmh2htdE060063LlzDaQLEhY4BO1QBngc+1XaD04rzvRPEEt9q2n7/G4l1OWRje+HEtIWMB2tmLaq+bCFOAZJGYHHbcKzvqFtLnolFeeeHvE0l5rOmBfF8WqX92zLqGhKLcfYBsZmwqqJUKMoQ+YzZzjqRh/g/W/ByeOtesvDup6GrXrQyRwWNxCDPIEcyMFQ/M3dj19aA2PQKK8+8Ia34PTxxr1l4b1LQw96YZIrexuIR9okCOZCFQ/M3djgn1rM8GeKfEN94qtYNX8R6E7z+YLvRn1OM3MDBSdkcAto5FZSMEPI/y5OTw1AbK539n4W8P6cbs6foWm2pvVK3Rgs4088HOQ+B8w5PXPU1Y0vRtM0O0NromnWmnW5cuYbSBYkLHAJ2qAM8Dn2riND1+TUdY05ZfHQbUppG+3eH47SF2tztbMe1V82IIcAySFgcDpuFTeENRjt9fn0zUPGmoapqa3d1GdKnW3dokEjFZHEcIeNSm0gswT5lA6qKa2CWm/wDW/wDkd9UT2lvJdRXMkET3EKssUrIC8YbG4A9QDgZx1wK4bwn4p0Wfxjrlja+KtP1m4uGie1AntfNl+RiyKYlUuFAxzuIHU1X0TxO93qNh/wAVpFfX11vGo6Ootl/s0eWzMwUL5kZjZQpMrMDnBGSMK/8AX9f13GejUV5p4U8TyNdeH5r7xxFrA1l7iFbZ/sqIdmdrR+WoYv8AKA3JBLHCrwB31trFjeate6Zbz77ywEZuYtjDyw4JTkjByAehNOzauibl2ivKfCFt4X1X4lapqMniqz1nV4L5ltYpPsEkrRiCPlWWISgKS4+RgPlOQTuzreF/FOhTeLNesYvGOn6jLP5b28ontPNICMWCmNV3hAP4gxAHJqeiflcpqzsd3bWlvZQmKzt4reMuzlIkCjcxLMcDuSSSe5JNS15l4Z8UTiTQri58bRa22sG4iS0c2qI5jB2mPy1DF8qobkglzhV4Ao2OvWF74y8IXN74ws9T1CSeVZ7GSK3R7CV7dx5SbVEkXz4QxyFmYhecqc1uxHrdFIxIQlRuIHAzjNedaJ4ne71Gw/4rSK+vrreNR0dRbL/Zo8tmZgoXzIzGyhSZWYHOCMkYX9f1/XqM9GorzTwr4nlNxoE1544h1g6w1xCts5tURimdpTy0DF/lUNyQSxwq8AVvBninxDfeK7SDV/EehO8/mC70d9TjNzAwUnZHALaORWUjBDyP8ueTw1NauwujZ6pRXH25N58Zb0XhI/s7R4TYoemJpH81x7/uo19se9L4fTzfiX4quLqKMXMUdpDEwO8rCVZgAxAIy2SVHAIByc8a+ysr36X/ABE3Z2/ra519FYVj4XisdTF6k5Lg3DDEYBBlYHP4AY6c1BY62kl7p1u+qiSdi8VxCdn3hnG7C/K2R0yM88ViM6SiuV0zVru81+JY9Wt7u3kMjGGCRGKrg7dy7A0Y6clm5wO9Jouq6jPqkKXmpWJMm4TWbXKmWM4JwsYiVgQR3ZuM9etAHV0Vzel3/wBsvrhF8SxTLDI2yJfJLMm0ZZiBzgnggADHOaZpWqv5mnvca4t59sMieUfKA+XOCu0Ak8AHkjnoOKAeh09FcbaeIZWv9lxq8CQzxSNhp4mktyBlQQIwEYDPysz9D6E0um61dGS1e51pZ1ZIHZWWJVbzAwIJAzgMoAx3JzngAB6HY0Vyui6rqNxqsKXmpWJMm4TWbXKmWM4JwsYiVgQR3ZuM9etXtP8ADENhqq3yzbmVp227McysD1z2Ax757UAzcorE02+8/wAW63af2x9q+zLbn7B9l2fY9yk58z/lpvxn/Zxitl1LxsquyEggMuMr7jOR+dOScQHUV51ovjWCfW/D+ny+Klub5zPa39m3kjMihtpk2oCkpK8KCu4BsKcEij4a8V6rq/j+2jt/Fen6tp073DtZWNxDK0aBWMfmReQkluoOAS0kh3bV75C3A9ToryvwZ4p8Q33iq0g1jxHoTvP5gu9HfU4zcwMFJ2RwC2jkVlIwQ8j/AC5OTw1XvDGvDWNcv4YviTa30dncSeTaxfY2kkgCAs8hVckKW+VlCgbfm30f5XG9Py/r7j0aivNPCviiT7RoE9944j1gaw1xCLZvsqqdmdpj8tAxf5QDyQSxwq5AGTpPxBupdfeHUfFtjBZX1tcuFe/tpJ9PdRuRWUW6rC4G7KSPKflPXaxpN2QLU9horyDw34x1Np9Ln1LxnHexSQ2M0kbpaxpKbkSKykqmQqOihcEHcSGLZCi34M8U+Ib7xXaQav4j0J3n8wXejvqcZuYGCk7I4BbRyKykYIeR/lzyeGqra2/rQV9L/wBf1oeqUUV5/wCF/FOhTeLNesYvGOn6jLP5b28ontPNICMWCmNV3hAP4gxAHJqQPQKK808KeJ5GuvD81944i1gay9xCts/2VEOzO1o/LUMX+UBuSCWOFXgCn4StvDOqfEfVdSPimz1vWLe9cWkMn2CSV0FunKssQlAUlx8jADacgndl9fxDoer0V554e8SyXus6YF8XRapqF27LqGhKLcfYBsZmwqqJUKMoQ+YzZzjqRjU8Pobf4meK4baGNLV47SeXadp85lYE7QMHKgZYkHIAwetaRhdN9v8AMTdmdfRXH+CCbfXvF2m253WFrqu+D0R5YkllQewdycdt9dhSnHkdvJP71cYUUUVABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVHcLM1tItrJHHOVIjeRC6q2OCVBBI9sj6imtWBJRXn3h3xrrV/pnhzVdQOlTWutTfZDbWiussUmGJcMXYMB5bZTAK/3jgg3PDureMdV1fUEuTov2HTdRNnIyQypJcqACzKN7CMruHB3buR8nU+hUy+rT5uZpcvn52/PT8dtSOdL+vX/Jna0VzGg3eov4s8SWN7PBK0H2eSGSNJVUB1bAKNKwGNozs27jkkZNZOgeLteum0CbV1042+r3NxZ+XbQujq8YlYSbi7DaRERswSMg7u1R9Sm+Zxa0Sf3x5tPkmHOv68jvaK4fwlea4uqasde1ywe2XV5LVIntXjd28tCqxs0xCjuE2k9eeeLkd5eah8U76yS6eC30rSo3jhydk0s7v+8ZQRuCiEAem5vWs62GdKbje9le6v8ArYale/k/1sdZRXIaV4l1G5u/Dsd1daax1KKcTwRQssjPGDmRMyHCZABUhsFh81UdN8U+KZvGNnb39nZR6LfSTi3njjVvOVUZk8qVLh95IXdl4ohgEdcA8pR3tFcD4R8U+Mdc1GzuL/QAmi3qu3nKsKfZRglcsLqRpTkbCPKjIJycYK1p6JeeLb3V7pL2bRTY2N41u5it5UluF2hsjLkRldw/v7+fuUdbf12B6f1/XY6uiuV8IXt1/bfifR7m4luo9O1ENBNK5Zgk0ay+Xk9dpdgPRdo7V1VXOPK7ej+9XAKKKKgAooooAKKKKACiiigAqK5Wd7WVbOSOK4KERySxmRFbHBKhlJHtkfUVLRQBz9vZ+MVuomu9d0OWAODIkWizIzLnkBjdEA474OPQ10FFFABRRRQBUl0qym1i31WSAG9t4nhjmDEEI5UspAOCMqp5zjHFNi0iyh1qfVYomS8uIlhmdZGCyKpJXKZ2kjJG7GccZxV2iq5ntcAoooqQCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKiuVne1lWzkjiuChEcksZkRWxwSoZSR7ZH1FS0UAc/b2fjFbqJrvXdDlgDgyJFosyMy55AY3RAOO+Dj0NdBRRQAUUUUAFVLPS7Swury5to2E17IJZ3eRnLEAAAbicAAcKMAc4HJq3RTu0BU03SrLSIJYdPg8lJp5LiT5ixeSRizMSSSSSfw6DgCrdFFDbbuwCiiikAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFR3EbzW0kUc8lu7qVWWMKWjJH3huBGR7gj2qSii9tQOc8LeB9J8KRr9kX7XdKnlre3MEIuBHgAR+ZHGpK/KOuT78DGpZaFpGm3k95p2lWVpc3JJnngt0R5STk7mAyeeea5Hwz4nuNU8c39s17p01uC+yODUzKQAF5RNgDcrz0xuP49Daatq9+1vJHpCR6fd8pcLeKZY0IJV2jK454wAzHkZHWqeNqYhe0cm3L+tehhCpCV0ujt9xah8OaJbajPf2+jafFe3AYTXMdqiyShjltzAZOT1z1qKHwn4ctxbi30DS4hayGW32WUa+S5xllwPlPyjkc8D0qhoOt6m62cWq2qtFP5iR3ZmBlkZMnLRqgVQQp6H04GeGaN4wuNW1KGJ9GuobW5DNFcCGfCjGQXLwqgBHdXbkgdOaSxk3b33r5sftabSv1Nr+wtI/tn+1/7Ksv7Txj7b9nTzsY2/fxu6cdenFMk0O1fxNDrqtLHdxWr2rBGAWWMsGAcY52kEjkY3N61WstX1a9ltZk0aMaddHKT/bAZETGVd49uBnjhWY8jOOcZmieItSu9XudIt4l1FrK4kW7urmYQGJPMIQBVjw5Kg9MD5cEg0vbttXb2st9u34+g3Vgt+vl/V9jXi8J+HINQa/g0DS47xpvtDXCWUYkMvPz7gM7vmbnryfWnab4V8PaNfPeaPoOmWF1IpV57WzjidgTkgsoBIJAP4Vq1k6VfXlxq2pW19EsRtzGY1jm8xdrA4P8Aq1IJxyDu9jSvZpGjkk0n1CLwr4eh1o6xDoOmR6mzM5vUs4xMWYEE78bskEgnPOabpnhHw3ot6bzR/D2lafdFSpntbKOJ8HqNyqDg1naZL4i1DxPdvdTR29haXBhMEFyrA/u1YZDW+453DJEi49OOeoYsEOwAtjgE4BP1qYTUo8yFGXPcztG0O10OO7Fs0ssl5dyXc80xBd3c9yAOAAqgdlUCtKsWz1y5uX0zzbKKOO+DhytwWMbqCcAbBuHynnIPtVO18VXk3iVNMuNHe2hd5VE8jSKfkBPRogjk46I78ZPQGiVaLd2yPaw37v8AFnTUVy+i+MLjVtSghbRbqG1uQxiuBDPhRjILl4VQAjurtyQOnNW9P1jV7zUJoJdGhiht5zDNOt7uzwCCi7AWxkZztxnjdRGpGVrdf6/UFWg1dPyN2iisnSr68uNW1K2voliNuYzGsc3mLtYHB/1akE45B3exqm0nYtySaXc1qK5q08U3s/iaPS59Ha2hkkkRZ5HkU/ICRw0QRicdEd+MnoDXSk4BJ6ClGcZLmWwRmp3t0CiuS07xnealcKsWhTrFPG728zpcKvCll8xmgCqCB1Vn5I69aNP8Yalevbq+iwQedHE+TfFsCVWMfSP1U7vQYI3HIEKrB7fqZ+3p9zraK5vSvEmo31vZXd7pEVpaXrbI3W88x1Yg43LsACkg4IJPIyBzizomo63ftK2oWFhDCkssYeC8d2yjlQNpjAxx1z+HaqU09ilVi7W6+TNuisK31y+n+xxNYW63UtzJDcQi6Y+QiZy4Pl/Nxt4IH3155rdqk01dFRkpbBRRRTKCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKNlpEFjfXV3E0jSXP395GB8zNxx/tH8AKrxeGdKhuhPHDNlWZ0jN1KY4mOcskZbah5PKgEZOKh0XXZtT17WLC4tZrVrBotsc0aA7XBw25JXDA7SRwhAIBGeaw9J8UeJdb8Y3tnDpRsNLsLsW0zT28UrcRLIcut0NpO8Y2xuBkZPUL1RwMndOy5Enq+jta3fpsZtQte3X8To7fw1ptr9j8kXQ+xMzwbr2ZtpbrnL/N/wACz1NOtfD9hZXgubX7VEQxYRLezeSCc5xFv2AcnjbitF2KRsyozkAkKuMt7DOB+dYlj4pivpdKUabfQpqcbtHJL5W2J1BJjfa5O7CnlQV461xqMU9F/X9Ifs4LoixB4b0u2vVuYoZd0bl442uZGiiY55SIsUU8nkAYyfWmw+F9Kt72C7hjuFuLcuUf7ZMSd7bm3fN8wJwcNnoPSsbTPiVpOreLm8PWltdNcI8yvKJIGWMRZ3NIiymWIZGB5iLkkDvUPh/4seG/EfiCHSLGbEtzv+yubm3kE+0Fj8kcrSJ8oJ/eInTHXiiMI6WXmDpw1ujtqybPw1p9hqBvbdr7z2+8ZNRuJFbggZVnIOMnGRxVHTfGJ1PVXsU8Pa1A0E5guZpYYvKgbGVyyyHeCMcpu25G7bXSU+VO0hyim7NbGXY+HbDTr57u1a9EsjF3El/PIrMRjJRnKk4AHTjA9K1Kw9F12bU9e1iwuLWa1awaLbHNGgO1wcNuSVwwO0kcIQCARnmszQviRpXiPW007S7W6l3l/wB+skDqgXPzOiymSMHGBvReSAcE4rqWCqpS5Y6RSbtsk1dfgJcsVpoa8PhXSYNQjvYoZxPDI0kRN3MVjLZ3BVLYCnPKgYPHHAp9h4a0zTbiOa1S43RAiJZryaVIs8HajsVXjjIA4JHQ1pTiVreQWzpHMVIjeRC6q2OCVBBIz2yM+orzTQPiJrmsjw2J106zmvrjyr6Frd2/dmKSVJY28wbVZYiuCGwwbk4wTD4GVaEp00rR38tG/wBPyJcacdbHd23h+ws7wXNr9qiIYsIlvZvJGc5xFv2AcnjbRp/h+x0y7e4tGvPMkJLiW/nlVicclXcjPA5xmsyx8c2V/LZldP1GG01EldPvZo0WK8YKWAX59y7lUlfMVAQODyMx6V8QNP1RrImw1CygvmljhuLpYwhljDF4/lcnICMc428H5s8Uf2fWjrybenn/AJO68n2C1O+3n/wTqqybPw1p9hqBvbdr7z2+8ZNRuJFbggZVnIOMnGRxWBo3xX8Na5rdtptnM2bxmW1lM0DCYjnGxJGkjyASPMRemDgkA9Rbal9p1a9sPsV5F9jEZ+0SxbYZ94J/dtn5sYwfQkVNfCVaEkq0LPfX7ivcn52K9h4a0zTbiOa1S43RAiJZryaVIs8HajsVXjjIA4JHQ1q0UVzpJKyHGKjsjJXwzpscjtF9tiV92YotQnSMbs5xGHCr1PQDB5FRDwnpkMIW0SaOSOONIma6mbb5ZJjz84JwSe/TjOK26Kn2cOwvZw7I5zw54Ri0SCH7TcNdTwMTGUeaOJcjkiFpWQNy3IA6n3zpRaBp0WqHUEhczl2cBpnaNHYYLLGTtVjzkgA8n1NaNFNQjFJJbCVOCVkjH03R7u11y91K+urW4e5RUVYbVotgXPcyNkkYBOBnaPQVsUUU4pRVkVGKjsFFFFMoKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAxdP8N/2f4lv9ZGq30734US20oh8pQudm3bGG+UEgZY5zzk81Bo3hN9F1a6vk8QapdfbJjPcwXC22yV9ioCdsKsMKq8Agcd+c7tsNtrEMYwgGMYxx6YH8h9BUldDxNVppvdJbLZbdP607C5UFcvZ+B0s9WtL1Nf1h0spZJLa0d4fJjV87kwIgWHzcFiWGMBhk56iiufrcfQ53TvCBs9QtLq+1/VtXFluNrFfmAiJmUqW3JErsdrMvzM3U98ES6Z4Zm0m6iNr4g1Q6fDkRabILdoUXBAQN5Xm4XPGX7AdOK14mENijMrAJGCQqEngdlAB/AAfSs238U6VdNZCCS5b7fC01uTZTAOqgkgkp8r8H5DhuOlAEOieGbjRdQuLlvEeq6glw7SSW90lqELkAbsxwo2cKBjOPat6sDRfG+heILqO30u4uXeRWMbTWE8KSFTh1V5EVSyngqDuGDkcHBD430KfV002Ke6NxK0iwk2E4jnMYJcRyFNkmAp+6TRsg3bJLHw19h8SahrA1a/mfUFCy28gh8pQudm3bGGG0EgZY5zzk81nab8PrGx16z1i61K+1O9sk2QzXiQeYBtKfNKkSyP8AKSMOzDnOMgEa1p4n0u+msI7eS4J1GJpbYvaTIrBeoLMoCsMH5WIbjpVWDxz4fudVg0+G7maa5laG3kNnMIZ3UEsI5inlvja2drHoa6li66uk91bZbLS33aE8qaN2VGkhdEkaJmUgSIBuQ+oyCMj3BFcpp3w40rT5NDka6vbuXQzILSSdo8mN0KGN9qKGUbiRxnJ64JFXoPHPh+51WDT4buZprmVobeQ2cwhndQSwjmKeW+NrZ2sehqXT/F+japqK2VpPP5sm7yWms5oY7jb18qR0CS8ZPyE8AnpzUUsRVpxapuyf+TX5N/eNq+5TsfAtnYzWWzUtRls9NLNp9jLIjQ2jFSoK/JvbarMF3swAPTgYhtvh9a21tpNudY1OaLSrqW5iWQQfvDIGDK+IhlcPIOMH5zzwuL9t400O71uLSoLi4+1TvIkBeynSKZkBL7JSgjfAU8qx6UW3jTQ7vW4tKguLj7VO8iQF7KdIpmQEvslKCN8BTyrHpWv13EP7W/p5/wCbv6sXKiDTPBiaTLZJa67rH9n2Dk22mmdBCi4IVCQgkdFB4VnYcDriti2037Nq17f/AG28l+2CMfZ5Zd0MGwEfu1x8uc5PqQK5ix8fT6r4uuNLsNFn+w2srRTXdxb3kTgrGHOF+zFO4ADSKW6gcrnetPE+l301hHbyXBOoxNLbF7SZFYL1BZlAVhg/KxDcdKxqV6lZ80ne/wDw/wCO9+o+VR0NaisbRtYvr7VtU0/UrG3tZLFo9jW900wlRwSpOY02njkc/U1s1iMKK56x8daBqOqf2daXVw12JfJkiexnQwvjIWQsgEZYD5d+N38Oam0/xfo2qaitlaTz+bJu8lprOaGO429fKkdAkvGT8hPAJ6c0bhsbdFc7beO/D93q0Gm21zcvc3Lslv8A6BOEn253Mkmza6DHLglRkc8jNzSNfg1nUNRtrdJE+wuqMJoJoXJIPJWSNeOOCpYEc5oDbQ1qKwbfxrod1rMWlwz3JuZnkSEtYzrFMYwS/lylNj4Cn7rGk0HxlpviK4eGxt9VidHkQm70q4gT5GKn53QJnI+7nd6gEEA3DY36K56x8daBqOqf2daXVw12JfJkiexnQwvjIWQsgEZYD5d+N38Oas2PinS9Q1J7CB7pLhQxVbixngEoU4YxtIgEgGRyhPBB6GjzDyNiisDQfGWm+Irh4bG31WJ0eRCbvSriBPkYqfndAmcj7ud3qAQQLOka/BrOoajbW6SJ9hdUYTQTQuSQeSska8ccFSwI5zQBrUVhWfjPRL+++yWlxcO7B/KkNlMsVxsyWEUpTZKcAnCFiQCR0NN0XxvoXiC6jt9LuLl3kVjG01hPCkhU4dVeRFUsp4Kg7hg5HBwBsb9FY7a+G8ZJ4ft7cyslkbu6n34EALbY1xjksQ56jAT3FGga+NafUoJbc2t3pt69rPCX38DDI4OBwyMrdOMkdqv2crX+f6B/X6mxRRRUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBHbDbaxDGMIBjGMcemB/IfQVJWfa6/o99qc2m2WrWNxfQbvOtYrlHlj2nB3IDkYJAOR1qHSdfTVNV1LTmsLuyuNOKeYLjyyHV9xVlKO3BCnrg+1a+xqJNtba/J9RXRrUVm6Xr1hrF5fW+nXENx9hkEcjw3EUg3EcjCMWUggghgpyDjI5rEtfiRoeo+Kf7C0lhfzCXymlgu7baG27mwrSiRwo6lEYcEdQQNI4WvJySj8Ku/JeYnJJXOpthttYhjGEAxjGOPTA/kPoK5K08I69aajYSDxDZtZ6dLK1vbHSzkrIGGHfzuWG4AEBR1ypJBXr3dIo2kkZURQWZmOAAOpJqlb69o93Naw2mq2M8t5EZ7ZIrlGaeMdXQA/MvuOK5upRh6d4X1myGhrNrNjMumSSvLs010M4cEAL++OzAY8ndnjpVbQvAdxpniaPWb260h7hPMMk+naMLK4u2cYPnyCRhIOd2NoywVu2K6GDxNoN1rD6Tba3p02pRsVezju42mUgEkFAdwIAPbtTrHxFomp6lPp+m6xYXd7bZ8+2t7pJJIsHB3KDlcHjnvQvIDBsfCOt2mqadJJ4gtZLDTppWhtRppVmSTcCHk805cBgAwAHXKnIxc0nQdc02S1tDr0D6PZjZDbx6ftneMDaiSSl2UgDHKxoTtHI5zetfFPh+9ube3std024nuWdYIoryN2lKffCgHLFe+Onem2/iFLrxVeaHDY3bNZRxyT3f7sQrvBKr9/eSQD0XHHWnGLa0BspaVoGt6c1rZtr0D6NZjZFbJp4Wd4wNqRySl2VgBjJWNCdo5HOcrw58N4PDWtW9zZ23hv7PbbhFImgLHfbSpAzcrJgtg4LeX8wznk5robTxLaXd+LVYbiNszDfIFCjyiAxzu6ZPFaEeoWc0kSRXcDvMm+NVkBLr6j1HvSXcHrdM5PQvAdxpniaPWb260h7hPMMk+naMLK4u2cYPnyCRhIOd2NoywVu2KNC8B3GmeJo9ZvbrSHuE8wyT6dowsri7Zxg+fIJGEg53Y2jLBW7Yrqk1OwlvDaR31s9ypIMKzKXBHUbc5pYNRsrm5kt7a8t5p4874o5VZkwcHIByOaF0sD1vcxNC0PX9L1i+ub7WdNurW+uDcSww6ZJFIG8tUAVzOwAARScqc89O1Sx8Ja3Z6rp0r+ILWSw06aVobUaaVZkk3Ah5PNOXAYAMAB1ypyMdGmr6bIyLHqFq7Pu2hZ1Jbb97HPOO/pS22q6fezGKzv7a4kC7ikUys2PXAPTkfnR28gfW5z1jpet6LrV9rmueItHaxmRWuwumPb7EjVgCJGuGCgZySQenarlv4+8HXl1FbWnizQ555nEcUUWpQszsTgKAGyST2Faqatp0jTCO/tXNuCZgsynywOpbnj8aSHV9NuZEjt9QtZXkJCKkysWIGSAAecDmgGZOg6Lr2nX94+qavpt7a3crTPDBpkkDhyFXhzO424Xpt/Gsfw/8Nbbw9q8E1rbeG/s1uGWORNBWO+2FSozcrLgtg4LbPmGe5zXYwajZXNzJb215bzTx53xRyqzJg4OQDkc1RsvEdtfaitnFb3CuzSqGdV2/uyAx4b1PFC7eX4A+py+haP4lg8VadaXcZGh6HHLFazSW0UZlTZ5afMtxIznGOscQ4JIzgVsaRoXiOx8SXOpahrel3MN4EE8EGlSQthFIXa5uGA65OVOcdq0NJ8Qpqur6nppsLuzuNNZBILjyyJFfJVlKO3BC55wfUVr1Uk4vUN7nF6F4DuNM8TR6ze3WkPcJ5hkn07RhZXF2zjB8+QSMJBzuxtGWCt2xWrp2hatp+ryMutRf2Sbia5WzSyAlZpGLFXlLkFAzMQFRD93LEAht+ip8gepzug6Lr+m395JqWs6dd213K0zR22mPA6uQqjDtO4wAvTb+NYeg/DafQvEtnq0V7pGbZXjleHRvLubxXHLTTmUs8mQDuwASWyvI2947pFG0kjKiKCzMxwAB1JNUrfXtHu5rWG01WxnlvIjPbJFcozTxjq6AH5l9xxQtHoD6+Zn6doWrafq8jLrUX9km4muVs0sgJWaRixV5S5BQMzEBUQ/dyxAIatpXh/xFa+IbvUNS1rS7q3vVVbiCDSpIWIRSF2ubhsdecqc+1asHibQbrWH0m21vTptSjYq9nHdxtMpAJIKA7gQAe3anWPiLRNT1KfT9N1iwu722z59tb3SSSRYODuUHK4PHPejcO7Mew8MazbNYW91rNlcadphzZxLpzJLlUZI/Mk84h8BudqJkgdBkFuneF9ZshoazazYzLpkkry7NNdDOHBAC/vjswGPJ3Z46VqQ+LPDlzLDHb6/pcsk/meUqXsbGTZy+0A87cHOOnenab4p8P6zeG00fXdNv7lY/NMNreRyuE4+bapJx8w56cj1o3dwfmUItIvbL4lXGrQRCWw1PTkhuH3gGGWF2KcHkhllbp0Kc9aPC2k3trqniDVtUiEE+q3+6OEMG2QxIIoySCRlgu/23AdQavQ+K/D1y96lvr2mStp6s14qXkbG2CnDGTB+QAg5zjGKSz8W+HNRuYbfT9f0u6nnZkiigvY3aRlAZgoBySAQSB0BBrX2kmreVvle/6fgDXX+trGvRWdY+ItE1PUp9P03WLC7vbbPn21vdJJJFg4O5Qcrg8c960ayAKKy9O8RaXqst+NPvba4jsH2TzRXMUiK2MkHaxKkY5DBfxpNN8U+H9ZvDaaPrum39ysfmmG1vI5XCcfNtUk4+Yc9OR60bgatFcjZfEnQ9U8Xnw9pDrfzq5jee3vbUqrBdzYQzCVgBwSqEZyOxxu2viHRb7Vp9LstXsLjULcEzWcVyjzRYIB3IDkYJA5Hehaq6B6GjRUEV7azXc9rDcwyXFuFM0KSAvFuGV3L1GQOM9ano2AKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACmTxedbyRbmTepXcp5GR1FPopp2dw2PNfDnw71LRr3Q4riGzng0eUlb19XvZGZQrKGS1Y+VE5yM/MyjLYHTG9o2n+JbbxhqOpX9hpMdrqKwq/kajJI8QjVgCAYFDZ3DjIx711MBY28ZfJYqM565x9B/IfSn16FbMKtZydRJ3Vuve/fvr28iFBLY5vSdM1m18ba1qN5b2K2F+sSxNFdu8o8oMAWQxAfMGycMcYxz1qv4a07xHp2vatPqVjpa2uqXv2pmt9QkkeHEKRhQpgUNzHnO4Yz3xz1lFYvFSaknFapLrsrW6+S+4fL+dwridN0TxTa6ppEc8Gj/2bpk83zpdymaZHDgPtMQVSAwymWzn7wxhuwiuUmuJ4UWUPAQHLwsqnIyNrEYb32k4PB5rPPibTP7ZbS42uprqORYpDBYzyxxOQCFeVUKIcEH5mGAQT1rCNKpJ+7Fu2vy7jbVtTltB8GarB4ns9S1KzsbJbYzuxtdXurpHaQEERwTIEtgSd2Yznjb0Y0zw14Du9D1nTTc2EVzBpm9be+fxHfSMoKMm4WjqYlJBwQHwMkjsK9AlkEMLyOGKopYhELEgegHJPsOag03UbbVtMt9QsJDJbXMYkicoyFlIyPlYAj6EA1mvIb1+ZzXhvR9XtNc1O61jQtDgXULn7S9za3rzTFlVVRSGt0zjBwd3GeBTtIMlt8TvESTW12q3cFq0E32WTyX2IwYebt2AgsOCc11lFXGXKnHyt+X+Qmru5ElrbxyeZHBGj/N8yoAfmOW59yMn1rItbDVYbqzWRLP7Nau/zCZt7hsgHG3AIB6ZOfUY53KKgZz9hot3HqcFzcw28Ai3kmG7llDFgchY3GI+TnKnPGOhNN0zQJbG8tfNgSVLXcI5zqM7EDBGREwKgkHkZx6elb8U8U6loJEkVWZCUYEBlOGHHcEEEdiKfQBj6fZXsN5dy3VhYKbiTzTJHOzMSAAoOYx055zxnpUVnpuoW66cpgs0+zvI0pSZj97PI+QZPOTnFbbuI42dgxCgk7VLH8AOT9BUGnajbatpsF9Yuz2867kLxtG2PdWAYH2IBo8gMG30XVoL6G4K28rxRyIzSX8zeYWH3gpTbGMj7qjv145Wz0K/tRBiKzBjjt0bZMw/1e4HB2dw2fr+ddNRQBzul6BLY3lr5sCSpa5Ec51GdiBgjIiYFQSDyM49PSt1LW3jk8yOCNH+b5lQA/Mctz7kZPrUGlataazZm6095GiEjxN5sLxMroxVgVcBhggjkVcoA5Pw0z3HjzxVeC1u4reT7LFHJcWskIkZFcNt3qNwB7jIOeDXTXdsl5Yz2shwk0bRsdqtwRjowKnr0II9QamqpLqtlDrFvpUk4F7cRPNHCFJJRCoZiQMAZZRzjOeK0k3OV0v6SBaHEwfBzQoJllS4yVOR/xJ9LH6i0BqbTfhLoul6pbX0E+ZLaVZUH9k6amSDn7yWqsPqpB9CK7uitPrVbv+RPKgridN0TxTa6ppEc8Gj/ANm6ZPN86XcpmmRw4D7TEFUgMMpls5+8MYbtqK5+tyuhwGg+DNVg8T2epalZ2NktsZ3Y2ur3V0jtICCI4JkCWwJO7MZzxt6MaZ4a8B3eh6zppubCK5g0zetvfP4jvpGUFGTcLR1MSkg4ID4GSR2FehUUbA9bnIeH9H1q01jVbrVNB0GI38/2o3FvfPLKzqoWNW3W65AwfmzxngVBo/h7XrCPw+j6fo8AsZriS7MF5I2PM3coPJG8kuWOdvI6nOa7aqlzqdraahZWVw0izXzOsGIXZWKqWILgbVO0EgMRnBxnBo6WDdnn2n+DfFVhrlrqLRWF1LbW88LyXOvXcn2hpFHzrEYjHACyjKIpAB4Py4aTR/BGu6WtiRaaOjW9tp8LiK7kXJtzIHwfJ6ESbh/tZHH3q9IqKO4SS4lhUSBosbi0TKpyMjaxGG98E470f8D8Aet/63OC8M+A7vQ9Z003NjFcwaZvW3vn8RX0jKCjJuFo6mJSQcEB8DJI7CvQqiubaC9tZba8gjuLeZCkkUqBkdSMEEHgg+lcdb6N8PbTxMljY+E9PXUbeVcT2vh4lIJAA4zOsWxGAIPLAjI9RRvoD7lnSLbxZB4n1C+v9L0aO3vljB8nVZZHjMaMBwbZQckjPIwM9elVtG8P67p6+Hkk0/R4VsZ7iS7MF65x5m7lB5A3ElyTnbyOpzmu2ooB6nLeH7PxLa6xqh1Sw0uGz1G6a48211KSSWL90kYARoFB/wBXnO4Yz3xzW0jw/rVu+i2F7Y6TBYaK5aK7tbp2mmxG8Y/dGJRGW37mw7dxznI7KihaW/rbYHqZdjZXUPiLVLqa202O3uFhEM1vGRcS7VIbzm6EAn5cdBmtSiim3cAoqpc6na2moWVlcNIs18zrBiF2ViqliC4G1TtBIDEZwcZwat0gCiqkuq2UOsW+lSTgXtxE80cIUklEKhmJAwBllHOM54qK213Tru8Frbzlpi0ihfKccocNyRjgkU7MDQoopqSJICY2VgCVJU5wQcEfnSAdRVT+07Y3xtF855VO1ilvIyKcZwXA2g4I6nvSW+rWd1dG3gkZn+bBMTBXwcHaxG1sexNAFyis+213Tru8Frbzlpi0ihfKccocNyRjgkVLpuq2WrwSzafN5yQzyW8nylSkkbFWUggEEEfj1HBFOzauBbooopAFFFFABRRRQAUVmx6/p83iCbRInme+t41klVbaQxxhhlcybdgJ7Ddng8U6213Tru8Frbzlpi0ihfKccocNyRjgkU2mtwNCiiikAUUUUAFFFFABRRUMt1FDcQQSFg85IjwhIJAyRnGBwD164oAmorPttd067vBa285aYtIoXynHKHDckY4JFN03X9P1a+vrOyeb7RYOqXEc1tLCVLZ2kb1G4HBwVyD600m1dAaVFVNN1Wy1eCWbT5vOSGeS3k+UqUkjYqykEAggj8eo4Iq3Q007MAooopAFFFFABRRRQAUUUUAFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlR2w22sQxjCAYxjHHpgfyH0FSUAFFFFAHHeGtMn0vx14hkh8Oyabp995TRXCfZ1jkdNwZiqOWyxbIJXJ5zg1m+D/AA7Fp/i3VL248Cf2e1zfGezvjHZk20fkIhXKSF1yyvwoI+fJ6mvQICxt4y+SxUZz1zj6D+Q+lPr0HmFR8+i95KL36WXffTr9xHIrW87/ANfeNldo4XdI2lZVJEaEZY+gyQMn3IFeX+C/Ct3p50UP4Ql0fVLVi99rEtzBIZo9rDytySNI4OUGxgEXbwTsXPqVFef1L6WPPNC8MJbazp0l14JKaraSM91r73cQFy21g0m5XMspYnISRAFz1G0ZPCfhiyh8R3lzdeBLnS3N09xaXU0lsY7dSoBRFjmYxljuLBVAbPzE16HRQtP69P8AIHr/AF6/5nl+i2vhLU9XuNEg/sG70Lw3JJfqI2gkSMy7vlYDhRH+9znH/LPnINZnwv0/Tr/S9DudA8OxwTWzlbzXILm2lWZNjZhZkkMrZDIPLdAqYGPuqT7HRTT7/wBf1/w4PW/9f1/VjhPCvhqDw94imisfA9jZH7TcOdajW3jBgdy6ImzMpPKqVZUUBTgnAB622ur+XVr63uNO+z2cIj+zXfnq32gkHd8g5XacDnrmr1FF9LMOrZ51ovhpbPVrCW48DP8A2tZuz3Oum7hX7U21gZAyuZZSxPCSqqrnqNozH4b8OTaVf6HfQ+Ebmxv5ZLkaneGS2aVUbO0SOJSXXOwhV3BQo4GAK9AtdSsb6e5gsr23uJbR/LuI4ZVdoX67XAPyn2NWaQPU8r8GeCLrQvFVrdXVlrq3kfmLc6gn9mLbXZKnLSNGFuZFJwwDgkNgnpmtHQfDCW2tac914JKaraSM914ge7iAuW2sGk3I5llLE5CSIAueo2jPdzX9nb3tvZz3cEV1dbvIgeQB5toy21Ty2BycdKsULoxvW/meSWHhOYeMJ9V8aeFl1HS53upoXvzaMmjK07OSwMhDh1wxb5mTbgcMQtT4a6TpOr6Joz6R4ahQxFo7/WYri2mjuIijBoGKSGVgQyDy3QKmBj7q59mopra39f1+fW4N3bfc810Dwdq0Wp6Jbatptn/Y+kys9pH5MGbUxIY0ZdvOZWbzepxsAOCa05Rb2fxW1Q64IjaajocYge4x5flwvJ56HPGMSoxz1B9jXb0VUZ2TT6r+n/XTQX9fc7njnwwsbC90vQ7vw74djguLVyLrXILq3kSZAjfuWZJDKchox5boAmBj7i5veEvDOtaP45tdSk0K+tvPSZNTmQadDA0j4YMohPnSIGBwZWZxkcEliPVaie7t47qK2kniS4mVmiiZwHkC43EDqQMjOOmRUrfQHqeZeGfB1zH8QLbXpfD9/p0qvcS3Ml89i6ZkVhiOWD99KctgGbIC7ujYqPQPA7y+OItWvvDd7Z7/ALS16dRaxkjYSqylUkg/fSn5sDz8jZu6NivUba7t72Ey2c8VxGHZC8ThhuUlWGR3BBBHYgipaVraNeQ7vU800LwbrEeoaNa6pp9muk6TKz2a+TBm0aJDGjrt5zKzeb1ONgBwTVjw94dls9Z0yRfCcumalbMx1HXGkt2+3jYwbLrIZZA7lWAkUYxngqBXodFO7/r+v+H6iPOvC3he1i1++uZPBF1o07XDz2N7K9s0dqCgXbGkczGMsSxYKoDfxE1U0jw7c2HiLw7c23ga4sbmzmYapqaXNti7LQvGZTiXfN87bt0ihwCcDLEV6hUVzbpd2stvKZFSVCjGKVo3AI7MpDKfcEEUlotP6/rqPdt9yWuD8N29xYeJNZ8vwJeaZp+qbHODYiMuqNvLrHOSS5IGcHOecda3LfwVpdrdRXEV1rjPE4dRLr99IpIOeVaYqw9iCD3roKLCPNPCnhf+wbrw/cWPguXTbovcJqFwjWodI2zsWRllJZM7CqruChQMDAFWfCXh+3sfFWqXZ+H/APZE1zdvNa6mYbH9yhhRSMxys43MHOAMfNz1NehUUf5WA888PeHZbTWdMkXwnLpmpWzt/aOuNJbt9vGxg2XWQyyB3KsBIoxjPBUCu1t7m/m1W+tp9P8As9pCsf2a8E6t9oLA7vk6rtOBz1zV6infSzDrc8o8J+BrjQ/EcFze6Zrsl0iypd3aDS1tr7crZMjIEuJVY4IEgJDYJ6Zq54V8LnQbnw/cWPguXTLotcJqFwjWodI2zsWRllJdM7CqruChQMDAFel0Ugep53onhdbfVtPa48FmLVrWRmufEL3UQFw21gZcq5llLE5CSIFXPUbRlNC8Nz22qae0PhaXSdUgL/b9fZ7d/tuY2DZZZDLIHcqwEijGM8FQK9Foo6WHfW55dpPhy5sfEHh64tfA9xYXNnK66nqcdzbf6UWheMynEu+b523bpFDgE4GWIpngzwRdaD4rtLq7sddW8j8xbrUE/sxba7JU5aRowtzIrHDAOCQ2Cema9Uop31uJ6qxx9uDZ/GW9N4Cf7R0eEWLnpiGR/NQe/wC9jb3z7V0cWkWEE/nR2yh/3nJJP+sYM/B9SKu0VU5c1vJWDq2c1ZWVzDeacjaMUS0MkTXAaL7hzjaA2dvfHB6cHtPpOmJp2oOkOhwQ/vZD9tURqNjMSqrj5ieQMEAcHnpneoqAOe0fT47fVrmZvD32VpJi8VyUg+RdijGVYsMkN0Hf60umWE8OpxM9lcRmPzPMllnDxNk/8s03nZ/3yvGRXQUUAU4NJsra5FxBAElBkIbJ/jYM3fuQK5vwQDca94u1K3G2wutV2QejvFEkUrj2LoRnvsrsKKuMuWLXdW/FP9AeplafrEuoa9qtkloFtdPaOL7V5mfNlZd7IFxxtDJzk8tjtWm6JLG0ciq6MCrKwyCD1BFOoqZWeyA860XSdSstb8Pxf8Ic1tFpJntn1FZbbiFgwTywH3+V0JBCkHbhTyRR8NeDblPH9trsmgahpswe4luZb57F1BkVhiOSD99K2W4M+QF3fxYr1Oil/X3geV+DPBF1oPiq0urux11byLzFutQT+zFtrslTlpGjC3MiscMA4JDYJ6ZrR8JeGbODxJd3Vz4FutKk+1NcWd3LJbGO2XaAVjWKZjGWO4sFUBs/MTXodFNaO/lYHr/X9dzjtFvrSP4r+KLKS6hW7mgs3jgaQCR1VH3MF6kDIye2RXRDSdMti9w0EaALKXZz8u1zufOTjBIq/RVTlzO/lb8LCWhxnhaCCezsJdO05Y5ImPnXySxsHGD8hIbceq/KRgcegqfSNMvbPXIblrCePeri6dRbpGWPOQE+dlyDy5J5HHJx1lFSM5XS9FkXxBDfvp1xbuGkeVp2gYZYH7rR/Oxyf488Z74pun6Ez64l3Ppk0O7zTP8AaTAyneCCA0fzv1x8+RjPfFdZRSA5ew0W9W4soru3hFnaMWhGxP3RRdqkY5+cnf7bR0qXTtNeG8tGGkvbXMRP2m+LRn7QNpB5DF23Ng/MBj2IFdHRQBzelaVEl/PI2hy2bmVpIJ2aIrEMAYUK525OcgDB71FZ6bLb6jpskehSQSQORdXKyx/vSUK7zh8vyc5YbsZ45IrqaKNgKcGk2Vtci4ggCSgyENk/xsGbv3IFcz4XvrS/+Ivi+SxuoblIxZxO0MgcK6pIGUkdCDwR2rsqKuMuWLXdfrcTV3c4/wAEA3GveLtStxtsLrVdkHo7xRJFK49i6EZ77K7Ciiicud38kvuVhmJptj5Hi3W7v+x/sv2lbcfb/tW/7ZtUjHl/8s9mcf7Wc1t0UUpScmAUUUVIBRRRQAUUUUAFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlR2w22sQxjCAYxjHHpgfyH0FSUAFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlR2w22sQxjCAYxjHHpgfyH0FSUAFFFFAHlvhmPVbr4gW18t/4glt3e4kuLW+sr60SFSrbA5ldoJNpIULCqckN0Uir+hR79Z043cnjA62kjNqKv562TNtYNnfiAxg42iH5j8vB+avQ6KFpbyB638zzzwnFBd+I7xhe+LleG6eSxi1GK+jt1i2jIJlQK+WJ4kZiMfJtrO8G2/iC38U2p1bXdUN43mLfWk2jagYJX2k4E8kz2yKGAYNGEBACjGdteqUULS3pYHrc8o8J2+uW/iOA65rmrm4ZZV1C0OjakYpTtYkLO0zwJhgCrRKucBRjO2rnhQPYXXh+483xVPNfPcRXQ1FL6REQZ2b0kG2LBCAOQpIySWyxPpdFCB6nHeHdTsB4x1K1sNN1G0hmSNYS+jXNvCWXzC/ztGqDls5z8xPGc10MepPc6jqFhDZXUUlmiFbieLbBOXUkbGz82MYb0zWhRRo1ZhszyvTWZPEnhmUL4tkvlnf+1ft1tdyWyTNBIm75l8tQJGIBhIjwTngKQeDLfXrfxXaHWNc1Q3TeYt9aS6NqBglbaSQJ5JntkAYAq0aqCAFGM7a9Uop31uD1Vjyzw1Fql54/trwX/iGW2ke4kuLW9sr6zWFWVtgcyu0Em0kKFhVOSG6KRWr4QsbTRdfnsY7PxK959rui8lxcXklokLSM6PulfynJBUfJufJORwxHfUULRA9fvued+FIobrxFeMl74uV4bl5LCHUY76O2WLaMgtKgDhmJ4kZiMfJtpfD5mbWNN2SeJjq7Mw1gX6XYtCNjb9m8eQMSbdhhwcdMrur0OiktEl/X9Pr3B6tv+v6XQ4jS9PuDr0Ggtc6mYtEuGvXunv52NykmTDG7F8uMlgVYkDyRxhxU1t/yWi/+3ff/sWH+z89Nnmv5+333eVn22+1b+meHdE0W4nuNH0fT9PmuP8AXSWtqkTS85+YqATyT19auvaW8l1FcyQRPcQqyxSsgLxhsbgD1AOBnHXArSE+Xfqv6/rsJ63/AK63OT8E/wDIxeMfsn/IO/tYeTj7vneSnn4/7aZz/tbu+a7Gora0t7KExWdvFbxl2cpEgUbmJZjgdySST3JJqWlUlzO/kl9ysPuFFFFQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlR2w22sQxjCAYxjHHpgfyH0FSUAFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlR2w22sQxjCAYxjHHpgfyH0FSUAFFFFAHnnhPXTrXiO8gtviHa6qLS6cQafH9kZ7iHaCTIUGSAWwrIExj5t9Zvg7xV4ivfFFtDqviLQpZJxKLrRX1SM3MLhSdkcAto5FZWGDvkf5cnnhq9VooWy9P6YPqeZaD4ku9T8TaHAPGouri6Ekup6PHFbbrN1QnyeE3xqGJUhyXO0YYYbOh4Q8UaHceNNasLfxZp+rz3DxPbYntfNk+RiyL5SqXCAd9xA6mupg8O6fb60+qr9qlu33YNxezTRx7uvlxu5SP0+UDjjpWpQD3Z5Z4a8VarrHj+2ig8W2Gp6dcPcM1pYTwzPEgVvLMkXkJJbgcAlpJDu2r/FkanhLUVt9duNMvvGeoarqi3V1GdJuFt2aNRIxSRxHCHjUrtIZiE+dQByorv6KOlgev3nlfgzxT4hvvFVpBrHiPQnefzBd6O+pxm5gYKTsjgFtHIrKRgh5H+XJyeGrR8Ja7/bXiS7gt/iJa6otpdMILCIWjSTw7RuaQoAWALYVkCYx82+vQ6iuVne1lWzkjiuChEcksZkRWxwSoZSR7ZH1FG2vkD1HTkLbyM8vkqFJMuQNgx1544688V5v4V8Tym40Ca88cQ6wdYa4hW2c2qIxTO0p5aBi/yqG5IJY4VeAOrt7Pxit1E13ruhywBwZEi0WZGZc8gMbogHHfBx6GugoDyPKPCfiXxJqniKGz1LxTosc10sqXOmf2lE93aPtYgRW/2VHVkYciR5BtB+9wau6B4m1rVtT0TSJby8ilhlb7VeYgYX3lIftCsvlAIElKR/IQTknPFelVFcrO9rKtnJHFcFCI5JYzIitjglQykj2yPqKP6/r+tR7v+v6/yJa8o8I2/hfU/iRqupSeKrPWtXt75haxOLCWZ4xbpyjLEJQFJcfIwHynIJ3Z7e3s/GK3UTXeu6HLAHBkSLRZkZlzyAxuiAcd8HHoa6CjZ3F0seU+EvHd7eeOLW0vNfsbm11JJilk99DLPauMMkbJHAnlPjcCjSSN8pxnaxrS0PxBLf6xpwk8cLLqc0jfbfDq2kLGA7WzFtVfNhCHAMkhYHHbcK9EooB9TyvwZ4p8Q33iu0g1fxHoTvP5gu9HfU4zcwMFJ2RwC2jkVlIwQ8j/AC55PDV0UYa/+MN/HdSNGdP0aL7ABjjzpH82QA8E5ijXoenvXZVUl0qym1i31WSAG9t4nhjmDEEI5UspAOCMqp5zjHFaQko79mv6/L0B63/rqcLovjWCfW/D+ny+Klub5zPa39m3kjMihtpk2oCkpK8KCu4BsKcEij4a8V6rq/j+2jt/Fen6tp073DtZWNxDK0aBWMfmReQkluoOAS0kh3bV75HqdFZgeV+DPFPiG+8VWkGseI9Cd5/MF3o76nGbmBgpOyOAW0cispGCHkf5cnJ4ar3hjXhrGuX8MXxJtb6OzuJPJtYvsbSSQBAWeQquSFLfKyhQNvzb69Goo/yG9TzTwr4ok+0aBPfeOI9YGsNcQi2b7KqnZnaY/LQMX+UA8kEscKuQBk6T8QbqXX3h1HxbYwWV9bXLhXv7aSfT3UbkVlFuqwuBuykjyn5T12sa9hopPVAtDyDw34x1Np9Ln1LxnHexSQ2M0kbpaxpKbkSKykqmQqOihcEHcSGLZCi34M8U+Ib7xXaQav4j0J3n8wXejvqcZuYGCk7I4BbRyKykYIeR/lzyeGr1SorlZ3tZVs5I4rgoRHJLGZEVscEqGUke2R9RVX1b9REtef8AhfxToU3izXrGLxjp+oyz+W9vKJ7TzSAjFgpjVd4QD+IMQBya6C3s/GK3UTXeu6HLAHBkSLRZkZlzyAxuiAcd8HHoa6ClYDzTwp4nka68PzX3jiLWBrL3EK2z/ZUQ7M7Wj8tQxf5QG5IJY4VeAKfhK28M6p8R9V1I+KbPW9Yt71xaQyfYJJXQW6cqyxCUBSXHyMANpyCd2fV6KOtw6WPPPD3iWS91nTAvi6LVNQu3ZdQ0JRbj7ANjM2FVRKhRlCHzGbOcdSMafggm317xdptud1ha6rvg9EeWJJZUHsHcnHbfXYVU03SrLSIJYdPg8lJp5LiT5ixeSRizMSSSSSfw6DgCtIzSjJPr/mn/AF67g9f69S3RRRWYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBHbDbaxDGMIBjGMcemB/IfQVJUdsNtrEMYwgGMYxx6YH8h9BUlABRRRQBHbDbaxDGMIBjGMcemB/IfQVJTIEMdvGhABVQCB06ewH8hT6ACiiigDk9GvfFt5q12t5Loz2NjdtA/k28qS3K7A2RmQiMruH9/fz9yszwn4r8Y61fWt1f8Ah/bot4jv5qLChthglfmF1I0pyNhHlRkE5OMFa6fTPCPhvRb03mj+HtK0+6KlTPa2UcT4PUblUHBp8XhXw9BrR1iHQdMj1Mszm+SzjExZgQT5mN2SCQTnvQtl6A+pzel+IvE15qfh6Se40UafrivcLAlvJ58EYTf5e4y4duVBfaoUjG07hjX8P3GonxJrtrqlxBcGFoWieCOSNQjKcKUeRwCMclduepFQ6b4A0jS/F1x4hg5upmdgi2ttEqF+p3RxK7nkj94zdSevNXbDwX4W0q/S+0zw1o9ldxklLi3sIo5FyMHDBQRkEj8aAe7Od0zxR4quPGVna31rYxaNfyTrb3CxK4lCozJ5cqXDl8hdxLxRDAI6kA3fDGpa60rN4i17SLhJLy6tYLaGwa3mleORwArGdgcKhO3aTgck4JO3pvhXw9o1895o+g6ZYXUilXntbOOJ2BOSCygEgkA/hU0OhaRbaxNq1vpdlFqVwuya9jt0WaReOGcDcR8o6nsPSjoD1+85Dwj4p8Y65qNncX+gBNFvVdvOVYU+yjBK5YXUjSnI2EeVGQTk4wVrS0a98W3mrXa3kujPY2N20D+TbypLcrsDZGZCIyu4f39/P3K1ovCvh6HWjrEOg6ZHqbMzm9SzjExZgQTvxuyQSCc85pumeEfDei3pvNH8PaVp90VKme1so4nweo3KoODR/kD12OX8K+LPGOsXlteX/h7Gi3cbyeZGkKG2ABKjcLqRpTkbCPKjOTk4wVpLXUtdvPEXhC/1C/082mqNLJFbWLzRlENs7hXzIUuAPl+cqm0qMD5jjrIvCvh6DWjrEOg6ZHqZZnN8lnGJizAgnzMbskEgnPeq0vg/w/bTTahpvhfRDqe83Ecr2scReYEkM0gQsDk53YJGc80bf1/Vx9dDergtM8UeKZ/GVlbahaWUWi30s6286Rq3mqqMyeXKlw+8kLuJeKIYBHXAO7b3njFrqJbvQtDigLgSPFrUzsq55IU2oBOO2Rn1FW9N8K+HtGvnvNH0HTLC6kUq89rZxxOwJyQWUAkEgH8KPMXSxh6VrfiG+bSdTuL7w/FpeqSYW0ZZFuIwVYqiyFyssny/MuxcYbrjmXRr3xbeatdreS6M9jY3bQP5NvKktyuwNkZkIjK7h/f38/crat/Deh2mry6raaNp8GozbvNvIrVFmkz1y4G4575NQ6Z4R8N6Lem80fw9pWn3RUqZ7WyjifB6jcqg4NC/r8P6+YPXb+t/6+RzHhLxV4x1u/tLm/8AD4XRrxHfzUWBDbDBK/MLqRpTkbCPLjIJycYK0nhTxT40129tbi70JItIvkdln2QqLTglMst07S8jaR5cZyc8YIrq4vCvh6DWjrEOg6ZHqZZnN8lnGJizAgnzMbskEgnPemjwl4cXVn1RfD+ljUJCxe7FlH5rFgQxL4ycgkHnnJoH3MLw5rfia4uNLk16bSZbfUjNGsVlayxvGyAkOXaRgQwQ/LtGNwG5sZNqC9ur/wCK95ZtcSxWmlaXE626uQs8k7vl2H8W0QgD03N61p2/hLw5aNZta+H9LgNizNaGOyjX7OzcsUwPlJ74xmpZNDtX8TQ66rSx3cVq9qwRgFljLBgHGOdpBI5GNzetaQkle/Z/1+hPR/11Mfwhe3X9t+J9HubiW6j07UQ0E0rlmCTRrL5eT12l2A9F2jtXVVm6Nodrocd2LZpZZLy7ku55piC7u57kAcABVA7KoFaVFSSlK68vy1+9j6sKKKKzAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAI7YbbWIYxhAMYxjj0wP5D6CpKjthttYhjGEAxjGOPTA/kPoKkoAKZNKkEEk0hwkalmOOgAyafQRkYPIprfUDhfD3xGudf1GxVPDd/Hp2oAmC7WC4JjUjKNLuhWNVI7pI/JHUHI1fD91f8A/CX+IdOvNRmvre1FvJb+dHGpi8wOWUFEXI4GM5PHWpLLwR4etLy1vLSzmQ20jT20DXUxgt3YHJSBm8tD8zcBRjPGKfY+C9H07VH1G2Oo/apMeY8uq3UgkwCAGVpCGABOAQcdq9WtVwL51Ri0mtLpPW9929NNNO+3fNKd9TndJ+KB1DxYNFu9LjsH2SSvFcXEsd1FGqkhjFJCqtnHSN3xnPIBNTeHviNc6/qNiqeG7+PTtQBMF2sFwTGpGUaXdCsaqR3SR+SOoORuaZ4K0PSJraSzgum+yLtto7m/nuI4Bt2/IkjsqnaSuQAcEjoTTbTwN4fsb+3urWzlQ2krTW0BvJmggdgQSkJfy0+82MKMZ4xWlStlj5uSnJaaeuuvxenf59S0+5p211fS6te29xp3kWcIjNtd+erfaSQS42DlNpwOeuau1SttHsbPVr3U7eDZeX4jFzLvY+YEBCcE4GAT0Aq7XjytfT+v6/qxoFFFFSAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAR2w22sQxjCAYxjHHpgfyH0FSVHbDbaxDGMIBjGMcemB/IfQVJQAUUUUAR2w22sQxjCAYxjHHpgfyH0FSVHbDbaxDGMIBjGMcemB/IfQVJQAUUUUAc3Z+MRfXtqkWg6yLG8Yi31IwR+Q6gE7zhy6IQOGdFByMZyKZF43SbZIug6uLW4VzZXTpCqXhClgqAybkLKpKmRUBx1GRmaw8JCwuodmuatLYWxJt9NeWMQRDBAXKoJHUAkBXdh04OBiGLwT5Sxxf8JFq72turCztnFsUtCUKKVPk7mKqxC72cdCQSAaTvbT+v6+4elzPsfida6gsTQeHdcXzoklXzI4E+WRGaI8y/wAexgB2Iy20EEzaL8R7TWrqzhj0TWLVboRsstzHEqosqkxM2JCcPtYAAEjGWCggkj+HNvDaRwwa/q8bR29vAkw+zFlEG7Y3MJGdrlTxgjHGeafafD22s47dYta1Um3itolYmDJFuW2E/uuu1yp9vQ81Wl36/h/mS9vl+IukfEbSNZ8S3OhW8Ui38AlKxfabaQyeWcMNscrNGeRxIE/MECbRPHEWsT26yaJqumwXKyeTc3ywqjtH99cJIzAjDckBSFJBIIzFp/gJdNvLCa38Rax5WnRtFaWp+zCKGNl27MCEFsALhmJb5evLZms/BYtE0pDruqXCaY8jKkq22J94IIkxCOACQNu3rzmp1toN7lSHxpd33iPQ7e20m8s9O1J5Qs97bLi5QQtIrxskpMf3R8sqBmDcAbTXXTSrBA8rhysaliEQuxAGeFAJJ9gMmuYt/A32e/024/4STWpYtKk32drI1uY412lNhPk7nGwlcsxbHIbOTXTzQx3EDwzxpLFIpR0dQVZSMEEHqKrToMxbbxhpl3dRW8VtrSvKwRTLoV7GoJ9WaIKo9yQK23YpGzKjOQCQq4y3sM4H51iW3gbwlZXUVzZ+F9Ft54WDxyxafErIw6EELkH3rdq6ns/sX+ZKvfUwbTxWl4dM2aVfoupRO8bOYcI6AkxNiQ/NhTyMr/tVX0LxsmuXVtGdD1bTorsSC3uL1IQkjp95AEkZgRhuSoU7SQSCMx2/gVbXUra7i8Razts5JHtrYtAYoRJu3IB5WWHzcFiWGMAgFszWPg77D/ZAGvapMNKeR0Ei2/77fkEPiEcAEgbdvXnNZobKWnfEvS9V8WSeH7GyvJrpGmUsklu23ys7i6CUyRAkBR5iJksB3rUsfFMV9JpQGm30KanG7RyS+VtiZQSY32uTuwp5UFeOtRab4RNlqNpdX2v6tq4sdxtYr8wEQsylS25IldjtLL8zN1PfBEFp4GSz1a0vU1/WJI7KWSS2tHeHyY1fO5MCIFh83ViWGMBhk5OwPqTWXjEX99axwaBrP2K8crb6kYEMDgAnecOXRSBwzooORjORSWfjEX17apFoOsixvGIt9SMEfkOoBO84cuiEDhnRQcjGcipNP8J/2fdwGPXdWksbVibfTWljWGIYIC5VBI6gEgK7sOnBwMGn+E/7Pu4DHrurSWNqxNvprSxrDEMEBcqgkdQCQFd2HTg4GAb6mT4f+LHhvxH4gh0ixmxLc7/srm5t5BPtBY/JHK0ifKCf3iJ0x14qbSviTpOseMG8O2VrdNcq8qNIJIHEfl5DGRFlMsQyAB5iLkkDvWnpfhmbSbuI2viDVDp8ORFpkgt2hRcEBA3lebtXt8/YDpxUWm+ETZajaXV9r+rauLHcbWK/MBELMpUtuSJXY7Sy/MzdT3wQ9NBPrYo6V8SdJ1jxg3h2ytbprlXlRpBJA4j8vIYyIspliGQAPMRckgd6r6Z4m8U6t4wvbdNGks9IsbgwSedbwySOREr/AOsW7+UsXGB5TAAjJHO3W03wibLUbS6vtf1bVxY7jaxX5gIhZlKltyRK7HaWX5mbqe+CHaN4Wn0fVrm9/wCEj1a8W6mM01tcJa+W7lFQHKQqwwFXADAcc55pLp6fiN21H2PimK+l0pRpt9Cmpxu0ckvlbYnUEmN9rk7sKeVBXjrRod/qk2u6xp+rTWc4s2iaB7a3aH5HBO1g0j5Ix1GM+gqnaeBks9WtL1Nf1h0spZJLa0d4fJjV87kwIgWHzcFiWGMBhk5W08Ny6Dqd1rlx4m1y/DIGubd7e2cTBFIUbYrcOSM8BDknHXpR6i1OormtO8Z/2jqklkPD+s25t5zBdTzRReVbtjK5ZZDuBGOU3bcjdtqW38a6XdXUVvFa64ryuEUy6BfRqCTjlmhCqPckAd6XRPDFxo1/cXDeI9V1BLh2kkt7pLUIXIA3fu4UbOFAxnHtRrf+t/6/QOhj6B8WfDXiLX4tKspwslxv+zSG6tpBNsBY/JHK0ifKC37xU6Y68Ves/HUV7qmn2kWhausWpFjZ3jpCIZowCxl/1m5VxggMochhhTg4t6b4Yl0q6j+z6/qjadFlY9MlW3aBEwQE3eV5u0Z4y/YDpxWLovhHW7HxZbTXN+x0TTFlTT7ZrxZSqsu1V2C3jK4Xu0kpGMdyaAfWxvaLrc+p6tqtpc2U9kbJ41WKdI9xVgSGDJK4YHGRkKR3FZOmfErStX8XN4esrS7kuleZWdZLdwnl53F0WUyRDICjzEXJIHeruleE7jTNdl1R/E+sXrz7RPBcR2gjlCqQudkCsMZzwR75pdN8Imy1G0ur7X9W1cWO42sV+YCIWZSpbckSux2ll+Zm6nvggAPDnii916WRZ/DOpabFHLNF9ouJrZ48xyFMfJKzZ4/u468kYJbpvjE6nqr2KeHtagaCcwXM0sMXlQNjK5ZZDvBGOU3bcjdtqxZ+GFs9ae9j1fU2tzNJOmnmVFt45JMlm+VA7ZLMdrsygtkAELhmi+F59Hvrid/Eeq6hHcu0ktvdJbBC5AG7McKMDhQPvY9qOvy/H+r/AIA+tu/4a/8AAKWjfETSNa8T3GgQI8d9CJCENzbSl/LbDfLFK7RnkcSBD+IIFrw54ovdelkWfwzqWmxRyzRfaLia2ePMchTHySs2eP7uOvJGCa+meBE0vUNOuIdf1eSHTFaK0s3+ziGOIjGzCxAsMBfmJLcfe5bN6z8MLZ6097Hq+ptbmaSdNPMqLbxySZLN8qB2yWY7XZlBbIAIXDVgfl3HaNrk2parq1rdWc9j9hdAsdwkYJVgTv3JK4YHGRkKR3FVrLxjHe3UCjRtUhtbzcLK9mSJYrohSwCjzN67lBKmRUBx1yQCaZ4Sn07XJtSk8TavfG4AE8FzHaeXKFUhQdkCsMZ7Ee+aLLwf9iuLcnXdUuLSzybOymEBjtjtKqVIiDttViBvZvU5IBpf1/X9WAZoXjZNcuraM6Hq2nRXYkFvcXqQhJHT7yAJIzAjDclQp2kgkEZ6aubsfB32H+yANe1SYaU8joJFt/32/IIfEI4AJA27evOa6SgQUUUUDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAjthttYhjGEAxjGOPTA/kPoKkqO2G21iGMYQDGMY49MD+Q+gqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCO2G21iGMYQDGMY49MD+Q+gqSmQIY7eNCACqgEDp09gP5Cn0AFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlMgQx28aEAFVAIHTp7AfyFPoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD/9k=)

The above excerpt of code shows the username entered by the user is passed to the variable username\_info. Further into the snippet it is possible to see if the username and password correspond with the username and encrypted password stored in the cloud database then register\_new\_face is called and the username\_info is passed to it as a parameter. This removed the place holder code that had been written for the Christmas demo. It no longer allowed single or multiple users to register freely on the system.



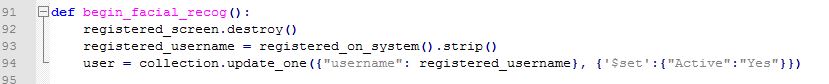
![A screenshot of a cell phone

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDyRXhpZgAATU0AKgAAAAgABAE7AAIAAAANAAAISodpAAQAAAABAAAIWJydAAEAAAAaAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGpvaG4gbGF3bGVzcwAAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzI4AACSkgACAAAAAzI4AADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIwOjAzOjI3IDE1OjE2OjU2ADIwMjA6MDM6MjcgMTU6MTY6NTYAAABqAG8AaABuACAAbABhAHcAbABlAHMAcwAAAP/hCx9odHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIwLTAzLTI3VDE1OjE2OjU2LjI3NTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5qb2huIGxhd2xlc3M8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgA0gJpAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A96qO4uEtbWW4m3+XEhdtiM7YAycKoJJ9gM1JRXcrX1JOe/4TjRJbOzms7pp2vyqWifZ5suzAlN4CFowwBIZhggEjIFR+G/G1jr0NtHMjWl9OZFEOyRoiyMwZVmKKjnC52jkDPHBrP0b4cQaLrkOpQX5keFpWCNAADkFIQcHpHGzKPXd2qWx8GajZw6DG2sWsi6ReS3RxYMpm8wOCv+tO3Alfnnnb6EH2J08uUXGEn5N37S022vy/f01Rl75t6VrsOq6nqNpCpQ2LopDxyxudwPJV0UYyDgqWBAzkdKj0nWrq91rVdOv7OG2fT/LZXiuDKJFcMQTlF2nC8jnr1pmnaJqFn4q1PVZ9Qtpre/VF+zpaMjxhMhPnMhB4Y5+UZOMY6VUsfDmsw67fX19q1hcQ6giR3MEWnPGSqKwG1jO2D83JIP4VzuOF9+0vsq2976X6eu+m1ive69/wL1n4t0a/vo7W2unLzFhBI9vIkVwR1EcrKEk4yflJyASOBTzrgbxcuhW9uZWSzN1czb8CEFtsa4xyWIc9RgJ71zmh/DePQr/T3tpNJ+z2DkpIuixLeSrghVefcckZGWVFJx15NbEelXln8RJ9Ugi8yx1GwSK4fcAYZYWJTg8kMsjdOhX3qcRTwsalqEuZWe/fXyXTX1BOVnf+tS1Lr6r4q/sWG3aUxWRvLqZSf3ILbY1CgEsWw/A6bO+RT7XxJpt7LYx28k5OoRtLbl7SVFYDqCzKArcH5Ww3HSqUelXln8RJ9Ugi8yx1GwSK4fcAYZYWJTg8kMsjdOhX3qrY+FdZtNU0+V9dtpLHT5ZGhthpxVmR9wIZ/NOWAOAwAHXKnIxxzsuW39f1+XqV3/rt/wAE0Lfxhot1rEWlwz3H2md3SEtZzLFMUBLbJSgR8AH7rGnWHizSNS1BbO1nn82Td5LS2ksUc+3r5UjqEk45+QngE9Oax9F8DTaf4hTVb640mSZRIZZrDSBZz3ZcYPnyCRhIOd2No+YKe2KZ4f8Ah7D4d1e3uLO38PeRbbhFImhrHehSCBm4WTBbBwW2fMM9zms1fS43bWxrWXjXQtQ1L+z7W5na6EnlSRvZTIYXxkLIWQCMsB8u7G7+HNb1c5pVjq2gzX97rmuaXJYSO9zMUsHtvLO0DJkadgFAXnI/GpYPHPhK8uI7a18UaLPNMwSOKPUImZ2JwAAGyST2oT0XcHu+wsHjTQrnVIbCG6lM1xK0UDm0mEM7qCWEcpXY+Np+6x6Gn2ni/Rr69+y2s87uwfypPscwin25LCKQpslOAThCSQCR0NQaboOs2BtrN9chfSLUbIreOx2zvGBtVJJS5BAGOVRScDkc5Zp3hzVLabT4b7VrW503TWzbQx2LRzHahRN8nmlWwG5wi5IHQZBFf+v6/ryG7dP6/r+rl6DxNply1mIXuG+3QtNbk2cwDhQSRkpw3B+Q4bjpUGkeNNE1y4SHTZ7l2kVmRpbGeJH28Mqu6BWZTwVB3DByODjPtfCuuWuoWMg1+0a00+SVoLc6aclXDD5383lhuABAUdcqcgiXT/DWr2Y0UTavZSrpskrS7dPdDMHBAC/vjswGPJ3Z46UK4mXNE8Xadr87w2Vvqcbo8ik3WmXEKfIxU/O6Bc5HTOfUAggXrXV4LvWb/TI0kE1gsTSswG0+YCRjnPY5yBWfY6HqdnqkudXibSWnmnFmtmBIzSEsVeUsQUDMxAVFP3csQCDW0n4f6Bo/iS71e00vTY2lEX2dIrGNDalVIYow6Fs84A6d61jytPmE730NjTr+4u7m7hurQWzQOAo80OWUjIJwMA+wJ+tQaFrg1h9Rhltza3WnXj200Jfd0wyODgcMjK3tkjtS2On6pb6pNc3V/ZzRTY3Rx2bRsMDAwxlb8ePyqn4Z0q8ttS13VNSi8ifU77ckIYNthjURxkkEjLBd3tuA6is6OsHz7/8AB8vI6MQqan+7atptftrvrudDWD/wmekS6s2n2Mj3zQFvtlxbANBZBQc+bKSEUjB+XJb2xzW1cW8N3bSW91DHPBKpSSKRQyup4IIPBHtXO2fg0aReKNB1O5sdLfcJ9LLF4cNu5hOQ8By2fkbbxwoPNaw5NeY5y3b+NPC15cx29p4l0eeeVgkcUV/EzOx6AANkn2rbrEt/CenWtzHPFc6wzxsGUS61eSKSPVWlIYexBFbdKfJ9i4K/UydN8QW+p3eoxQpKosGVWElvNHIcgnOx41yOOCpYGoNI8aaJrlwkOmz3LtIrMjS2M8SPt4ZVd0CsyngqDuGDkcHFfTdD8R2evXeoXWtaXOl2qiWKPSpIyNikLhjcNjk5OQc47daZp/hrV7MaKJtXspV02SVpdunuhmDggBf3x2YDHk7s8dKyVxsuaJ4u07X53hsrfU43R5FJutMuIU+Rip+d0C5yOmc+oBBAll19V8Vf2LDbtKYrI3l1MpP7kFtsahQCWLYfgdNnfIqHT9F1Sx1aRhrEf9lm4luFtEswJWaQliryliCoZmICqp+7liAQYRpN9afEC51O0RWtNS09YZpGIPkTRMxQlcgsGEjdOmz3rSNmte34/wBfiD6/11/yJNK8baHrUwj06a6csjsjPYTxrJs++qs6AMw6FASwIIxwamHizSTDbSq90yXUD3EJWwnO5VBLDGzh8A/Ifm46VQ0/w1q9mNFE2r2Uq6bJK0u3T3QzBwQAv747MBjyd2eOlVV8I6/FNbeR4ishb2TTfZoX0tm+WQMP3jCYFmG4YI2jrkHIIyblbQFvqXbDx9oGsSRQ6RdSTz3C5gElpPEjZUlCzmPCq2CFY8MQwXJBFJ4b8d6X4hgsgC8F3dJ9zypTD5gGWjScoEdlwcgHPytwMHGfa+BtWsoLf7Prln59vBaRI76a5Utb7wGKiYHBV8EZ6jOe1WND8Iarp9rpthqet217p+mnfHFFp/kySMAQN7+YwKjcTgKDwuWPO6tb/wBf1/XQDVsPFmkalqC2drPP5sm7yWltJYo59vXypHUJJxz8hPAJ6c1FZeNdC1DUv7PtbmdroSeVJG9lMhhfGQshZAIywHy7sbv4c1k+H/h7D4d1e3uLO38PeRbbhFImhrHehSCBm4WTBbBwW2fMM9zmtHSrHVtBmv73XNc0uSwkd7mYpYPbeWdoGTI07AKAvOR+NJN9Qfl/X9aHRO4jjZ23EKCTtUk/gByazLXxJpt7LYx28k5OoRtLbl7SVFYDqCzKArcH5Ww3HSq1v458JXdzHb2vijRZ55XCRxR6hEzOxOAAA2SSe1UbHwrrNpqmnyvrttJY6fLI0NsNOKsyPuBDP5pywBwGAA65U5GHrfyA0Lfxhot1rEWlwz3H2md3SEtZzLFMUBLbJSgR8AH7rGnWHizSNS1BbO1nn82Td5LS2ksUc+3r5UjqEk45+QngE9Oax9F8DTaf4hTVb640mSZRIZZrDSBZz3ZcYPnyCRhIOd2No+YKe2KZ4f8Ah7D4d1e3uLO38PeRbbhFImhrHehSCBm4WTBbBwW2fMM9zmkr6XB21sa1l410LUNS/s+1uZ2uhJ5Ukb2UyGF8ZCyFkAjLAfLuxu/hzT7fxhot1rEWlwz3H2md3SEtZzLFMUBLbJSgR8AH7rGo9C0jXdNv7qTUdW0+7t7qVpmjg054XDkKOGM7jGF6bfxrO0TwRPp3iKPV7y50p7hPMMk2n6QLOe6Zxg+fIJGEg53Y2j5gp7YoV9L/ADB9bG1oWuDWH1GGW3NrdadePbTQl93TDI4OBwyMre2SO1Gha4NYfUYZbc2t1p149tNCX3dMMjg4HDIyt7ZI7VU8M6VeW2pa7qmpReRPqd9uSEMG2wxqI4ySCRlgu723AdRR4Z0q8ttS13VNSi8ifU77ckIYNthjURxkkEjLBd3tuA6iuhqGvovv0v8AqL/P/M1bfVrK71W8023m33diIzcR7GGwOCV5IwcgHoTVyqdvbXsWq3k9xqHn2kwj+z2vkqv2cgHd845bccHnpirlZO3QAooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKANeiiiuEojthttYhjGEAxjGOPTA/kPoKkqO2G21iGMYQDGMY49MD+Q+gqSgAprukUbSSMqIoLMzHAAHUk06igChb69o93Naw2mq2M8t5EZ7ZIrlGaeMdXQA/MvuOKjg8TaDdaw+k22t6dNqUbFXs47uNplIBJBQHcCAD27Vzmm6J4ptdU0iOeDR/wCzdMnm+dLuUzTI4cB9piCqQGGUy2c/eGMNW0HwZqsHiez1LUrOxsltjO7G11e6ukdpAQRHBMgS2BJ3ZjOeNvRjQB11j4i0TU9Sn0/TdYsLu9ts+fbW90kkkWDg7lByuDxz3qO18U+H725t7ey13Tbie5Z1giivI3aUp98KAcsV746d65Lw14Du9D1nTTc2EVzBpm9be+fxHfSMoKMm4WjqYlJBwQHwMkjsK1PDej6vaa5qd1rGhaHAuoXP2l7m1vXmmLKqqikNbpnGDg7uM8Cmt/kD02Osd0ijaSRlRFBZmY4AA6kmszTfFPh/Wbw2mj67pt/crH5phtbyOVwnHzbVJOPmHPTketablxGxjVWfB2hjgE9snBx+VcXo/h7XrCPw+j6fo8AsZriS7MF5I2PM3coPJG8kuWOdvI6nOaXUHsdBD4r8PXL3qW+vaZK2nqzXipeRsbYKcMZMH5ACDnOMYpLPxb4c1G5ht9P1/S7qedmSKKC9jdpGUBmCgHJIBBIHQEGuK0/wb4qsNctdRaKwupba3nheS5167k+0NIo+dYjEY4AWUZRFIAPB+XDSaP4I13S1sSLTR0a3ttPhcRXci5NuZA+D5PQiTcP9rI4+9Qv8v6+QPS9jt7HxFomp6lPp+m6xYXd7bZ8+2t7pJJIsHB3KDlcHjnvWjXnvhnwHd6HrOmm5sYrmDTN6298/iK+kZQUZNwtHUxKSDggPgZJHYV3l3bJeWM9rIcJNG0bHarcEY6MCp69CCPUGn0QdbE1Nd0ijaSRlRFBZmY4AA6kmuG034S6LpeqW19BPmS2lWVB/ZOmpkg5+8lqrD6qQfQiu7q6kYR+CV/kJXvqULfXtHu5rWG01WxnlvIjPbJFcozTxjq6AH5l9xxUcHibQbrWH0m21vTptSjYq9nHdxtMpAJIKA7gQAe3auc03RPFNrqmkRzwaP/ZumTzfOl3KZpkcOA+0xBVIDDKZbOfvDGGraD4M1WDxPZ6lqVnY2S2xndja6vdXSO0gIIjgmQJbAk7sxnPG3oxrMZ11j4i0TU9Sn0/TdYsLu9ts+fbW90kkkWDg7lByuDxz3qGHxZ4cuZYY7fX9Llkn8zylS9jYybOX2gHnbg5x071ynhrwHd6HrOmm5sIrmDTN6298/iO+kZQUZNwtHUxKSDggPgZJHYVp+H9H1q01jVbrVNB0GI38/wBqNxb3zyys6qFjVt1uuQMH5s8Z4FH+Q2bWm+KfD+s3htNH13Tb+5WPzTDa3kcrhOPm2qScfMOenI9aSHxX4euXvUt9e0yVtPVmvFS8jY2wU4YyYPyAEHOcYxXP6P4e16wj8Po+n6PALGa4kuzBeSNjzN3KDyRvJLljnbyOpzmsnT/Bviqw1y11ForC6ltreeF5LnXruT7Q0ij51iMRjgBZRlEUgA8H5cMne2gI7Wz8W+HNRuYbfT9f0u6nnZkiigvY3aRlAZgoBySAQSB0BBqax8RaJqepT6fpusWF3e22fPtre6SSSLBwdyg5XB4571xGj+CNd0tbEi00dGt7bT4XEV3IuTbmQPg+T0Ik3D/ayOPvVN4Z8B3eh6zppubGK5g0zetvfP4ivpGUFGTcLR1MSkg4ID4GSR2FVZX/AK+8noehUUUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAR2w22sQxjCAYxjHHpgfyH0FSVHbDbaxDGMIBjGMcemB/IfQVJQAUUUycBreRXi85SpBiwDvGOnPHPTnih7APorzTwr4XOgXGgXOn+CpNNuy1wl/cK1qGSNs7FkZZSWTOwhV3BQoAAwBWTpPhPV9P197y48KX063dtcxapEiaZFBcu4yNqoweVNwIBncsNwyOWITeg1qew1Fc20F7ay215BHcW8yFJIpUDI6kYIIPBB9K8l8N+B7vRJ9LuofCMltPaw2JLpJbF45AJEuWB837zAoWI+8qgclQot+DPBF1oPiu0urux11byPzFutQT+zFtrslTlpGjC3MiscMA4JDYJ6ZqrK7XqK/VHaW/gHwdZ3UVzaeE9DgnhcPHLFpsKsjA5DAhcgg966CvLfDPg65j+IFtr0vh+/06VXuJbmS+exdMyKwxHLB++lOWwDNkBd3RsV6lS6IOrM7S9dsdYuLyCyNwJrJwk6XFpLAVJzggSKu4HBwwyD61cjuEkuJYVEgaLG4tEyqcjI2sRhvfBOO9croV3q7eMtSnvPC2qWVrfLEEuJ5rRlj8tGzuCTs3JIAwD15xVbwpBc6d4s1VoPBN5otjqTxv5oNksauqtud1imZiWJHIBJ74oA7imRTxTqWgkSRVZkJRgQGU4YcdwQQR2Iry/wANeDbhPH9trk/h6/0+QPcS3Ul+1i6ZkVhiOSD99KctgGfIC7ujYrY8K+GoPD3iKaKx8D2NkftNw51qNbeMGB3LoibMyk8qpVlRQFOCcAFrUHp952Euq2UOsW+lSTgXtxE80cIUklEKhmJAwBllHOM54q3XGxxCD4w366ggdNU0aJbMuMqVikfzo/8AyLG2O4PtWZouk6lZa34fi/4Q5raLSTPbPqKy23ELBgnlgPv8roSCFIO3Cnki5xUeW3Vf1/l94nu/67HotFeWeGvBtynj+212TQNQ02YPcS3Mt89i6gyKwxHJB++lbLcGfIC7v4sUngzwRdaD4qtLq7sddW8i8xbrUE/sxba7JU5aRowtzIrHDAOCQ2CemazQ3pc9Uorzzwl4Zs4PEl3dXPgW60qT7U1xZ3cslsY7ZdoBWNYpmMZY7iwVQGz8xNegTgNbyK8XnKVIMWAd4x05456c8UfZTD7TQ+ivNPCvhc6BcaBc6f4Kk027LXCX9wrWoZI2zsWRllJZM7CFXcFCgADAFZOk+E9X0/X3vLjwpfTrd21zFqkSJpkUFy7jI2qjB5U3AgGdyw3DI5YhN6DWp7DRXkHhvwPd6JPpd1D4Rktp7WGxJdJLYvHIBIlywPm/eYFCxH3lUDkqFFvwZ4IutB8V2l1d2OureR+Yt1qCf2YttdkqctI0YW5kVjhgHBIbBPTNVZXsLpc9J03VbLV4JZtPm85IZ5LeT5SpSSNirKQQCCCPx6jgirdcf4IBuNe8XalbjbYXWq7IPR3iiSKVx7F0Iz32V2FXUioysvL8Vf8AAOrCiiiswCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCO2G21iGMYQDGMY49MD+Q+gqSo7YbbWIYxhAMYxjj0wP5D6CpKACiigjKkHv6HFABRXA+ELG00XX57GOz8Svefa7ovJcXF5JaJC0jOj7pX8pyQVHybnyTkcMRF4Tjt7nxHeN9s8X7obp5LGC/jvo7cRbRuBaZAr7mLcSMxGBs20dflf+v67g9L+tvz/wAj0Ooo7mCW4lgimjeaHHmxq4LR5GRuHUZHIzXl/hK81uHxxayXjaz5F6kwvLWXT9QaG1lOGRDNPI0ZxhhviREPqAVB2fDer2q+JNZtEj8TfY7vY9u19ZaiVUhGMm15U/djpgAgZwFHSjoHWx29zbpd2stvKZFSVCjGKVo3AI7MpDKfcEEVz9n4U0SLUc22o6xLc2jo7xP4ivZdh6rvRpiCD6MMEeormfCgfT7nw/ceb4qnmvnuIrsail9IiIM7N6SDbFghAHIUkZJLZYm/4b1e1XxJrNokfib7Hd7Ht2vrLUSqkIxk2vKn7sdMAEDOAo6UbaoN9DvKK8z8LCWwn0C4SXxTPcXzXEVyNSS+kjRRnZvSQbYsEIA5CsRkktliYtELP4s8PQ3V34oTUX82TVIrt7uO1knWMn5N+I2UMWwsWUIwSOFph0PTHu7eO6itpJ4kuJlZoomcB5AuNxA6kDIzjpkVLXHW3/JaNQ+2/f8A7Fh/s7d02ea/n4993k59tvtWZoaF9Y05rp/GDa4kjNqCOZ1smba27Jkxb+WDjaITuPy9fmqpx5beav8A1+od/L/K56JRXj2k3erw6+8mpN4i+yX1tci+totP1RhayEblRZnkYEjDAPAiKexGUBf4Zj1SxudLubqbxZMUhsZJFuxeyLvkEiXO9CMHDCP5SMIPmAXJYwtfw/H+vuCWl/n+B69RWdaazDe65qOmQQzb9PEfmzEDyyzruCA5zuC4J46MvrXGaJG0mraebiTxeddSRjfrJ9oWxL7WBzvxAYg2NohO4/L1+am007MOh39zbpd2stvKZFSVCjGKVo3AI7MpDKfcEEViW/grS7W6iuIrrXGeJw6iXX76RSQc8q0xVh7EEHvXNaCJZdV05RJ4m/taQuusfblvFtdpRt+wsBAuJNuww4OMYJG6o9M1GVNT0OzuLXxR5lqLi0vppIb0w7cPs3MeJCSARKN2MDLAkArZ6Aek0V5p4VV7G40CYzeK5pr5riG7GoLfSIiDOzesg2xYIQByFYjJLNljV3whY2mi6/PYR2fiV737XdF5Li4vJLRImkZ0fdK/lOSCo+Tc+ScjhiGlcOlzuba7t72Ey2c8VxGHZC8ThhuUlWGR3BBBHYgipa47wT/yMXjH7J/yDv7WHk4+753kp5+P+2mc/wC1u75rsaqpHldvJP71cO4UUUVABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAR2w22sQxjCAYxjHHpgfyH0FSVHbDbaxDGMIBjGMcemB/IfQVJQAUUUHpxQAUV53oniCW+1bT9/jcS6nLIxvfDiWkLGA7WzFtVfNhCnAMkjMDjtuFJoPiSW/1XTo4/GMeoajel0v8ARY1twdP+RmJVAvmoY2UL+9Zgc4IyRg3/AK/r+txvT+v6/rY9ForzbTPG1tLqeh2Fx4vV7xhcWuoQMbcbZED7Wkwg8uXK8LlQQGIUgHDfCviaV7jQJb3xzFq39stcQLbt9kVfkztaPy0DFxtUHkqSxwq8AC12E9Nz0uiuA8Jaitvrtxpl74z1DVdVW6uo/wCybgW7GNRIxSRxHCHjUrtIZiE+dQByopukar4wl8beJIJrCw2ww25ESanJMICY3I8tGiQOWIGQWj/3u41jTck2uiuJ6O3meg1n2mgaPYapcalY6TY21/dZ8+6htkSWXJydzgZbkA8muU03VfH8uqWyajpXl2jSqJn/ALMt02pnk7hqTkcdwjfQ9K0vDt/oCeItTs9J1CxkkmZJPLiuVkeV8Hex5JY+p7VFVeykldO/Zm9Kk6sJySfupPbTf8DpHtLeS6iuZIInuIVZYpWQF4w2NwB6gHAzjrgVLRRUmIUUUUARxW8MDSNDFHG0z+ZIUUDe2ANx9TgAZ9hUlFFABRRRQAUUUUARW1pb2UJis7eK3jLs5SJAo3MSzHA7kkknuSTUtFFG4BRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAEdsNtrEMYwgGMYxx6YH8h9BUlR2w22sQxjCAYxjHHpgfyH0FSUAFFFNcOY2EZVXwdpZcgHtkZGfzoAdRXIaV4l1G5u/Dsd1daax1KKcTwRQssjPGDmRMyHCZABUhsFh81UdN8U+KZvGNnb39nZR6LfSTi3njjVvOVUZk8qVLh95IXdl4ohgEdcAgHe0VwPhHxT4x1zUbO4v9ACaLeq7ecqwp9lGCVywupGlORsI8qMgnJxgrWnol54tvdXukvZtFNjY3jW7mK3lSW4XaGyMuRGV3D+/v5+5R1t/XYHp/X9djq6KDwK43TNc8Q3Y0rVLi90CPS9Sc7bPZItwgKsyxpIXKyy/LyuxMYbrjkDodlRXH6brniG4/sm/uptKex1olYLa3t5PNtS0bSIWkMmJcBMMAidSQeME07xJqd1P4fimvNM8zUoZxLCluwcvGD+8T96cICACpBPzD5hQ9AOwori/Dut+J7ibS5Ncm0mSDUvOjSOztJY2iZASHLNKwYEIcrtGNwG5sZMvhbUddecnxFr+jzxyXt1awW8Vg1vLI8cjgBWM7A4VCdu0nA5JwSXYOlzr6KRiQpIBYgdB3rhNL8Y65ceLNetJ/Duprb2sMDxRStbEWxMbsd5ikd2DEDGwSEenY3Cm5ptdAO8orhNN8ea1e6pbWs/hvyI5pVR5fL1IbATjPz2CL/30yj1Iru6KlKdP4kJNMKKx9Ihe21bU4Wurm4G6N8zyl8FgScDoo9gAK2Kxi7q5rUgoSsnfb8VcKKKKozCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiio7hpltpGtY45JwpMaSOUVmxwCwBIHvg/Q00ruwCwIY7eNCACqgEDp09gP5Cn1xmi+O7zUbPRdQv9DWz0/WGWGGVLzzZI5SCQGTYvyHawDAk9MqAch+ieLde1fVru1bwxFHBYXv2O6uU1EMM4B3Rq0alwAQWztxn5dxyK7ZYDEQ5uZJcu/vR7279+hPMl/X9djsKK53RNW1G78R6/YX0KI9kYWhRbkSR7XVsYPlIyk7ckHfgng44rO0Px1eanJpDX2jR2dtqs01rFIl4ZWE0Ycn5fLX5CImw2Qc/w45qfqVbWyWiTeq6rm766J7BzI3IvCfhyDUGv4NA0uO8ab7Q1wllGJDLz8+4DO75m568n1p2m+FfD2jXz3mj6DplhdSKVee1s44nYE5ILKASCQD+FYvhHWfEuqalqyatZ6eLW21GWDzIr1meIKikIq+SocZP3iwPJ44FdfWVajLDz5JWv5O+40+a/wDWxkxeFfD0OtHWIdB0yPU2ZnN6lnGJizAgnfjdkgkE55zUFt4T0LQ3lvvDfhrRrXUfLZY3itkty5P8LSIhYAnGTg/Q1BFq97e/Eq40mCURWGmack1wmwEzSzOwTk8gKsTdOpfnpVjT4teGqqb+fdabpyy4j6bgIhwM/dyf51lKLjZeQ+5Hb3njFrqJbvQtDigLgSPFrUzsq55IU2oBOO2Rn1FXrfw3odnrEurWmjafBqU2fNvIrVFmfPXLgbjnvk0umySxC4sZWMk1qfkaRiTJG2ShJ5PqpPqpNQ6dq95dSW/2uwjt4rneI2S48xty+o2jg4ODnPTIHaQH2nhjQNP1WXVLDQ9NtdQmLGS7htI0lfcctlwMnJ5PPNRr4P8ADSXkl2nh3SluZJfOeYWMYd5OfnLbclvmbnr8x9a2KzreSeXXrxWuH8mFIwsIVduWBJOcbs8euKAIbfwl4ctGs2tfD+lwGxZmtDHZRr9nZuWKYHyk98YzViHQtIttYm1a30uyi1K4XZNex26LNIvHDOBuI+UdT2HpVazjvG1i9ik1S6kihVdiMkQxuB9EB47frms+w1C/nn0uG5vpVkdyZAqR4nj2MwJ+Xg5XaduPwyKAsdRUUdrbxXM1xFBEk84USyqgDSbfu7j1OMnGelc9ZTapJfT+Zd6gYbd5GDvHAY5gkhGwbU3ZwOpIOexFWree+jjsb6a+89LxlD2+xQiBxkbCBu49ycjPTs0DNyis+xkn/tbUIJrh5o4yjRh1UbAwJIGAMj65PvV6QM0bBG2MQQrYzg+uKQFC18PaLY3n2ux0ewtrnn99Daoj89fmAzzWjWHp8Oo/2rPHNrM8yWroGR4YgJAUyfuqCOT69u9T2epahc3ckUmmxxxQymOWVbnd2zlRtBPUZzj2zSSS0Rc6k6j5ptv1NWisXTvEDXupC2a12xyBminTzCrhfdkVTkf3S39as2V1dS319DcRqrwldirLuUgg452Ajpz19qZBo0Vkadq95dSW/wBrsI7eK53iNkuPMbcvqNo4ODg5z0yB2msrq6lvr6G4jVXhK7FWXcpBBxzsBHTnr7UAaNFY9hrN3cPAbyxjt4bjeEZLgyMCvXI2jg4ODnPTIHZLfWb2e4sgdOjW3vctFL9pyQmN3zLt4YjHAJHXnjkA2aKKwbGTWLvW7hriRIbS3m8sxRTKwPyKRwYsnO7rvH045OoG9RWLp/iBr3UhbPa7Y5AzRTp5hVgvuyKpyOflLf1qa21K/uWgmXTUFlP8yyi5BdExkMybe/HAJPNAGpRWFpniObUbyKM6bPHBOCY5vLl4GMgvujVRkejNzj6063165m1tLGXTmhjd3USuzqflB/vIFbOOis3HPQUAbdFFFABRWRYpdjW7xJdSuJ4oAhSJ0iCncD1KoDx25/OpNKvNSuy5vLW0iiWSRN0VwzNlWK42lB6dc/h2oA06KbIXEbGJQzgHapOAT2Ge1Y1r4hN21nFDDEbiZlE8Xmt+5DKXyCUwx2jkcYOBQBt0Vn2F1cTahew3SKhhKbQku9cEHn7ikHjkHP1punvcLqd/bz3UlwkZRo/MVAV3Akj5QOPrzQBpUVm6e9wup39vPdSXCRlGj8xUBXcCSPlA4+vNaR6cdaACiszSr3Ursu17a2kUSySJuiuGZsqxX7pQenXP4dqg07xA17qQtmtdscgZop08wq4X3ZFU5H90t/WgDaorndL1m8nv57CJFvDazOLi4mk8sxrvIUBVT5jgHpgcckGpdM8RzajeRRnTZ44JwTHN5cvAxkF90aqMj0ZucfWgDdoorF07xA17qQtmtdscgZop08wq4X3ZFU5H90t/WgDaoorBspdYutauXuJEhtLeYxmKKZWB+RSODFuOd394fTjkA3qKwLDxJcX9wiJpcyxzIzQyFJQOmV3lowqg+oLckdetSWviE3bWcUMMRuJmUTxea37kMpfIJTDHaORxg4FAG3RRWJBr1zLrS2U2nNDGzSASuzqflBPRkCsTjorNxz0FAG3RWFpniOXULyJG02eOCcFo5vLl+UYyC+6NVGR6M3OPrVm21K/uGgmGmoLKf5hL9pG9ExkMybe/HAJPNAGpRWVBq13K1vJLYxx211xA/wBoy+SCy7124UEDsWxxUdjrV3OYGvbGO3hnD7GS4MjArknI2jggHBznpwOxsBs0VhpqWoz6lpjPbi2tbpnwFmV96+WzDeCoKngH5SR1z2pbfXrmbW0sZdOaGN3dRK7Op+UH+8gVs46Kzcc9BQBt0UUydXe3kWJzG5UhXUAlTjg88fnSeiAfRXJ6ZrFze3WnwC+n3blMjMIiLgGNnYcIMbWG3j1rS07xA17qQtmtdscgZop08wq4X3ZFU5H90t/WqsBtUVn2N1czX99DcoqNCU2Ksu9cEHH8CkHjkHNM0q81K7Lm8tbSKJZJE3RXDM2VYrjaUHp1z+HakBp0UUUAFFFFABRWHp8evDVVN/PutN05ZcR9NwEQ4Gfu5P8AOq/hbVr261TxBpOqSiefSr/bHMFC74ZUEsYIAAyobZ77QepNVGLlFtdNfxsD0OkoooqQCiiigAqO4WZraRbWSOOcqRG8iF1VscEqCCR7ZH1FSUU07O4HHeD/AIe2vhq0tVvpk1C4sifs0iLLFFHkYLCFpXQSHLZdQpO4+pJ3dL8O6do19eXenrcpLeyNLOJLyaRGcnlgjsVU8DkAcADpXO6R4k8Uaz40u7VdIbT9Hs7k28rXFvDIxIiV+ZEu8qTvGAImGCMnJO23ovxE0fW/FE+gQq0N7D5hCtc28u7y2w2Vildo+o4kVD+IIHRVxlevJzqTb5t/Tfba3WwuVI1LXwxplnrV5qtuLtby9G2dmv52Vh2+QuVGMnGANueMVVtvA2g2aaetvBdqum3D3NqDqNw3lyN948yc5yeDkfM3HzHLLPxiL69tUi0HWRY3jEW+pGCPyHUAnecOXRCBwzooORjORWb4f+LHhvxH4gh0ixmxLc7/ALK5ubeQT7QWPyRytInygn94idMdeKSxNfpN/e+it+Ta9AcV1RvHwpo7a3/arW8pufOE+37VL5Pmhdok8nd5e7A+9tz361sVx+lfEnSdY8YN4dsrW6a5V5UaQSQOI/LyGMiLKZYhkADzEXJIHeuwrKdSdRJzbeml+w7WbMdtAC+Mk8QW9wYmeyNpdQbMicBt0bZzwVJcdDkP7CtisTR9dl1DVNWt7y0msRYMgCXCRg7SpO/ckrhgcZGQpHcVkw+NLu+8R6Hb22k3lnp2pPKFnvbZcXKCFpFeNklJj+6PllQMwbgDaanmcrJ9P+H/AK/4AW6nVS2cct2lwS6yIjRna2NynHB/EZHpVeHRLKD7N5YnH2Vi0WbmQ4J65y3P45rntK+JOk6x4wbw7ZWt01yryo0gkgcR+XkMZEWUyxDIAHmIuSQO9dbNKsEDyuHKxqWIRC7EAZ4UAkn2AyaXRMOth9U4rKaLV7i6E6GGdFDRGM7gV4B3bsY56Y/Gs628YaZd3UVvFba0rysEUy6FexqCfVmiCqPckCt0kKpLHAHJJ7VUoyh8SsG5Qh0WzgupLiM3XmyAhy15K2eMdC2OM8enalt9Gsbb7MYomJtSxhaSV3ZNwwQCxJxjt0rDtPHcN5qdhapoerpDqRZrO9kSEQzRKCxlz5m5VxggMoYhhhTg4paB8WPDfiLX4tJsJf31zv8AsrfabaT7RtBY4SOVpEyoJ/eKnTHB4qQOsstNt9PMhtvO/eEswkuJJBkkkkBmOMknpTbfR7G1uhcQxMHUsUUyMyRk9SqE7Vz7AdT61leHPFF7r0siz+GdS02KOWaL7RcTWzx5jkKY+SVmzx/dx15IwTTsdT8TR+INXiv7nTr6z02JZBb2WnNFPPvVmVQ73BQEYxyMH/Zoem4dbHRWlnLBfXdzNMkhuGXaqRldirkAHk5PPXj6VcrjtF+I9prV1Zwx6JrFqt0I2WW5jiVUWVSYmbEhOH2sAACRjLBQQTLo3xF0bWvE9xoMQaC9hEhw9zbSbhG2GysUrtHjI4kVD+IID8v60A6C10q1s7uW5g8/zZvvl7mRwfwZiP04ptpo9pZTNLAbnc+dwkupZFYnuVZiM8dcVh2njuG81OwtU0PV0h1Is1neyJCIZolBYy58zcq4wQGUMQwwpwcU/D/xY8N+I/EEOkWM2Jbnf9lc3NvIJ9oLH5I5WkT5QT+8ROmOvFJAzpYdB0+C6iuIkmEkORHm5lYID/CAWwF9sY4HoKINDs7a5a4ia78xhhi17M27jHILEHrx6VgaX8SdK1jxe3h2ytbprlXlRpBJA6x+XkMZEWUyxDIAHmIuSQO9L4b8Q6zPql1ba5JYXcUhm/s6awt2gM3ksElBDyuM7jxyOFJ+gM6CLRLKD7N5YnH2Ylos3UpwT1zlufxzTYNDs7a5a4ia78xhhi17M27jHILEHrx6Vk6J44i1ie3WTRNV02C5WTybm+WFUdo/vrhJGYEYbkgKQpIJBGakfjO9v/EGiW1ppV5ZWGpNKEuL61UrcKIWkR0ZJSY/ug7ZUVmDcY2mgSOjh0Syg+zeWJx9lYtFm5kOCeuctz+Oaq2nh/7NrAvPPBij3+TCvmYQN2+aRlAx2VV7emKg8KXWuXkV6+vXmn3HlXMlvELKye3xscqS26WTOcDgYxz1qjpXxD0vXfEd14fsv9H1KJZfL82a3nBKHaSUhmZ0wSDiQIT065wegd7nXVQtdGtLO5aeA3O9yWcPdyurEjGSrMQeAO3auc8K+PJNY03S5NV0m8tftqBBqHkqlrLMFJZVBkMij5WwzKFOOGOVzH4f+LHhvxH4gh0ixmxLc7/srm5t5BPtBY/JHK0ifKCf3iJ0x14p210Dbc6WHQdPt7qK4iSYSQ5EebmRgg/uhS2AvtjHA9BTotFsYblZo4nyjF0RpnaNGPdUJ2qeT0A6mqOh3+qTa7rGn6tNZzizaJoHtrdofkcE7WDSPkjHUYz6CszSviTpOseMG8O2VrdNcq8qNIJIHEfl5DGRFlMsQyAB5iLkkDvSXYPM6ODSLS2uBNB56EEkRi5k8sZ6/u923v0xTbTRbKxlSS3WbMYIjWS4kkVM/wB1WYgenA6Vmy+Ibp/G82jWNss8Fjp32u7x/rGkdiIY0ywUEhJCS3H3eRyaraF48g1wWMw0TV7Gy1A7La8u44hG74J2YWRnU/KwDFQpI4Y5XNOLVgf9fmdDDp9rb39zeQxbbi62+c+4ndtGBxnA49Ks1yejfETSNa8T3GgQI8d9CJCENzbSl/LbDfLFK7RnkcSBD+IIFjw/4n1DXzOj+GtR0tUkmiW5uZbaSING5TBEcxfOR0xjg/MeCZ6XQeprW2j2lreNdQm585/vF7uVw31VmI+nHFLHpNnHfG7WJvN3FhulYqrHqyoTtUnnkAHk+prn/C/jK41YWdtqml3cE04kRb7yVjtriSM/MI1MjSKMAkFhg4OGPGb/AIb1241S81nT9SjijvdKvmgbyQQrxMBJE4BJ5KMAf9oN2xVKLabXT8gN6qcGk2NtcLPDAFkUuQ24n75Bbv7D6dsVcoqQM+20SztLs3MJuvNP3i95M4bjHIZiD179KLbRLO0uzcwm680/eL3kzhuMchmIPXv0rQooAz7bRLO0uzcwm680/eL3krhuMchmIPXvWhRRQBSTSLJL03YiYyliwDSMyKx6sEJ2qTzyADyfU1HDoOn291FcRJMJIciPNzIwQf3QpbAX2xjgegrRooAzotCsIbmK4iSZZYt21vtEhzubc2fm+bJ55z2qSDSLS2uBNB56EEkRi5k8sZ6/u923v0xV2igAIyMHkVnQ6Dp9vdRXESTCSHIjzcyMEH90KWwF9sY4HoK0aKACqNro9pZXTXEBuRI5LNvu5XViRjJVmIJwB27Cr1FAFKDSLS2uBNB56EEkRi5k8sZ6/u923v0xSwaTY21ws8MAWVS5DbifvkFu/sPp2q5RQAVQtNFsrGVJLdZsxgiNZLiSRUz/AHVZiB6cDpV+igClBpFpbXAmg89CCSIxcyeWM9f3e7b36YpsWi2MNys0cT5Ri6I0ztGjHuqE7VPJ6AdTV+igChDotlbzCSJZgVzsU3MhRMgj5VLYXgkcAYzxRFollB9m8sTj7MS0QNzIcE9c5bn8c1fooAzY/D+nxXEMyJOHgbdEPtUpWP2Vd2AMcYAxjjFPtNFsrGVJLdZsxgiNZLiSRUz/AHVZiB6cDpV+igCtDp9rb39zeQxbbi62+c+4ndtGBxnA49Ks0UUAUoNGsLY25gtlU2wcRfMTtD8t35z71HDoOnwXUVxEkwkhyI83MrBAf4QC2AvtjHA9BWjRQBn22iWdpdm5hN15p+8XvJXDcY5DMQevenx6TZx3xu1ibzdxYbpWKqx6sqE7VJ55AB5Pqau0UAFVodPtbe/ubyGLbcXW3zn3E7towOM4HHpVmigCtDp9rb39zeQxbbi62+c+4ndtGBxnA49KIdPtbe/ubyGLbcXW3zn3E7towOM4HHpVmigArH0DQBor6lPLcG6u9SvXup5imzg4VEAyeFRVXrzgnvWxRVKTSaXUAoooqQCiiigAooooA57SPCs2k6pdXh8R6teJdytNPbXCWwjdyipnKQq4wqrjDDpznmq2meBE0vUNOuItf1aSDTEaK0sn+ziGOIjBjwsQLDAX5iS3H3uWzZsfHWgajqn9nWl1cNdiXyZInsZ0ML4yFkLIBGWA+Xfjd/DmptP8X6NqmorZWk8/mybvJaazmhjuNvXypHQJLxk/ITwCenNC2Vgd9bkWn+E/7Pu4DHrurSWNqxNvprSxrDEMEBcqgkdQCQFd2HTg4GH6X4Zm0m7iNr4g1Q6fDkRaZILdoUXBAQN5Xm7V7fP2A6cUyDxz4fudVg0+G7maa5laG3kNnMIZ3UEsI5inlvja2drHoalsPGGi6nqS2NlcTPJIWEMrWkyQzlfvCKZlEchwCcIx4BPQGheQPzINN8Imy1G0ur7X9W1cWO42sV+YCIWZSpbckSux2ll+Zm6nvgi7p+nXkOvarqF7ePJHctGlrbrKxjhjROTtPAdmLEkDoF54qhD4+8PT6lFYwXN1JPOzLAF064Kzhc7mjfy9roMcupK8rz8wzLovjfQvEF1Hb6XcXLvIrGNprCeFJCpw6q8iKpZTwVB3DByODhpu1kD8yLTfCM+n61PqM3ibV783IC3Fvcx2nlzAKVUHZArDGexGe+aht/A32e/024/4STWpYtKk32drI1uY412lNhPk7nGwlcsxbHIbOTWppGvwazqGo21ukifYXVGE0E0LkkHkrJGvHHBUsCOc1UsPHGhanrUWk2kt6buYSNEJdNuYo5FjOHZZHjCEAkDIOMkeopLfQBNN8Imy1G0ur7X9W1cWO42sV+YCIWZSpbckSux2ll+Zm6nvgjfmhjuIHhnjSWKRSjo6gqykYIIPUVj6f4v0bVNRWytJ5/Nk3eS01nNDHcbevlSOgSXjJ+QngE9OahsfHWgajqn9nWl1cNdiXyZInsZ0ML4yFkLIBGWA+Xfjd/DmjeyDbUktvA3hKyuormz8L6LbzwsHjli0+JWRh0IIXIPvW6QGUhhkHgg96CcKSe3oM1gaD4y03xFcPDY2+qxOjyITd6VcQJ8jFT87oEzkfdzu9QCCBUpSn8TuG2pjaL4R1qx8WW01zfsdD0xZUsLZrtZdqsu1V2C3jK4Xu0kpGMdya29L8MzaTdxG18QaodPhyItMkFu0KLggIG8rzdq9vn7AdOKjsfHWgajqn9nWl1cNdiXyZInsZ0ML4yFkLIBGWA+Xfjd/DmrVh4o0vUdUbT7d7qO5UMVW5spoFlCnDGNpEVZAMjlCeCD0NSujB9SKz8MLZa099Hq+ptbmaSdNOMqLbpJJks3yqHbJZjtdmUFsgAhcVbfwfcW+p397/wAJTrMj38eyRHjtNq4UhSMQA5XPGSQe+avaX4it9VvNThhjlUaeyq4kt545DkE52PGuRwcFSwb9Kg0XxvoXiC6jt9LuLl3kVjG01hPCkhU4dVeRFUsp4Kg7hg5HBwb6B5lC0+HttZx26xa1qpNvFbRKxMGSLctsJ/dddrlT7eh5p+m+A49Mv9Pnj17VpYNNRorWyk+ziGOIjaYyFiBYYC/MSW4+9y2dNtfDeMk8P29uZWSyN3dT78CAFtsa4xyWIc9RgJ7isXTfHdxrPjKbSNO0aYWVvMYZby6hu4WDCMOcA23l/wAQA3SqT1AwVzTTT166/wBf1t5B09CHRfCOtWPiy2mub9joemLKlhbNdrLtVl2quwW8ZXC92klIxjuTW3pfhmbSbuI2viDVDp8ORFpkgt2hRcEBA3lebtXt8/YDpxU1h4o0vUdUbT7d7qO5UMVW5spoFlCnDGNpEVZAMjlCeCD0NVoPHPh+51WDT4buZprmVobeQ2cwhndQSwjmKeW+NrZ2sehqV0B9SC38FyRTQy3PifWr17VHFm1z9mY2zspTzFYQgswVmA3lxzyCeazrH4VaPpb6bPpV3cWV/YOzHULezsluLrcMYlfyPmGCRxgnOTk4I3dP8X6NqmorZWk8/mybvJaazmhjuNvXypHQJLxk/ITwCenNMtvGmh3etxaVBcXH2qd5EgL2U6RTMgJfZKUEb4CnlWPSgZXs/BYtE0pDruqXCaY8jKkq22J94IIkxCOACQNu3rzmoofAhhvdOmHibW3i0uTdZ2ztbmOJdhTZnydzjYxXLszd85ya6umu4jjZ2DEKCTtUsfwA5P0FGwjlrLwM9rcxPceKNavYY7prs2062io8pySSY4FbqxOAwptj4CXT7qxlt/EesiLTo2hs7b/RhFDGy7doAhG7GFwzEt8vXls2tL8eaBrL7dOnvJMxySIW025RZNn31QtGA7ryCi5YEEYyDS6P488Pa9cW8Wl3c8n2rPkSyWU8UUzAElFkdAjOAGygO4bWyPlOC3RB5kGieBYtGFjE2u6vqFnp/wA1taXjwmNHwRuJSNWbG5sBmKjPAGFxc0zwzNpN1EbXxBqh0+HIi02QW7QouCAgbyvNwueMv2A6cVu1kWHijS9R1RtPt3uo7lQxVbmymgWUKcMY2kRVkAyOUJ4IPQ0AU9K8J3Gma7Lqj+J9YvXn2ieC4jtBHKFUhc7IFYYzngj3zS6b4RNlqNpdX2v6tq4sdxtYr8wEQsylS25IldjtLL8zN1PfBC2PjrQNR1T+zrS6uGuxL5MkT2M6GF8ZCyFkAjLAfLvxu/hzRY+OtA1HVP7OtLq4a7EvkyRPYzoYXxkLIWQCMsB8u/G7+HNNa2sD63CXQ7218bza9pht3ju9O+zXNvM7JuljYtCwIU4HzyK3HA2kA9Kx/BngrVdM0yxi8SapPL9hYmGwhuUmtgdpAcOYI5eNzYUswXjHQAdDY+KdL1DUnsIHukuFDFVuLGeAShThjG0iASAZHKE8EHoabB4t0i4tdPuInuzHqNybW23WE6s0g3ZDKUygG1uWAHBOabk5Wv8A1/Wwd/66f5GfpngRNL1DTriHX9Xkh0xWitLN/s4hjiIxswsQLDAX5iS3H3uWzctvCwtdWlu4tZ1QQNLLNHYiWNYIpZM7nGEDPyzHa7MoJzt4XEza+G8ZJ4ft7cyslkbu6n34EALbY1xjksQ56jAT3FO0jxBBrGoajb26SJ9hdUYTQTQuSQTkrJGvHHDKWBHOaGmt+v8AX9feH9f18ijY+DvsP9kAa9qkw0p5HQSLb/vt+QQ+IRwASBt29ec1N4b0K40u81nUNSkikvdVvmnbySSqRKBHEgJA5CKCf9ot2xVOLx9p194k0vS9LBnS/eVRPLHNAGCRs4eEvGEnT5cFkfA3KeQ1XLbxpod3rcWlQXFx9qneRIC9lOkUzICX2SlBG+Ap5Vj0oUmlZdf6/NA0b1FY+ga+NafUoJbc2t3pt69rPCX38DDI4OBwyMrdOMkdqNA18a0+pQS25tbvTb17WeEvv4GGRwcDhkZW6cZI7U3CSv5B/wAMbFFFFQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAc7oOi69p1/ePqmr6be2t3K0zwwaZJA4chV4czuNuF6bfxrI8O/De38N6zb3FnbeG/s9tuWKRNAWO+2lSBm5WTBbBwW8v5hnPJzXc0Uf8MG9/M5vS/D+t6ebayfXoH0a0GyK3j0/ZO8YBVY5JS7KQBjlURjtHI5zFp/hfVYZNPttR1e0utK00/wCjQR2DRTEBGRBJL5pVsK3OEXJAPAyD1NFA7nn2i6N4nh8T6fZzqE0bRIpYrO4mtIk8xdnlx5Zbh2cgY6pEDgk84A1tO8L6zZjQ1m1mxlXTJJXl2aa6GcOCAF/fHZgMeTuzx0rq6KBbnMaRoXiOx8SXOpahrel3MN4EE8EGlSQthFIXa5uGA65OVOcdqh0bwtrun+IrjUdQ16xvo7qVmuAulsk0keGEcPmGZlWNNwICoMkEnlmJ62igDhvDnw3g8N6zb3NlbeGxb224RSpoCx3wUqQM3Ky4LYOC2z5hnPJzWvoGja/pmoXcup6xpt5bXUrTvFb6ZJA4chQMOZ3GML025966KigHruUra1votWvbi41Hz7OYRi2tPIVfsxAIc7xy+44PPTFZunaFq2n6vIy61F/ZJuJrlbNLICVmkYsVeUuQUDMxAVEP3csQCG36KbbYdLHO6Douv6bf3kmpazp13bXcrTNHbaY8Dq5CqMO07jAC9Nv41h6D8Np9C8S2erQ3ukbrZXjkeHRvLubxXHLTTmUs8mQDuwASWyvI299RS2De/mctpeg+JrLxBd6jd67pM6XiqJYotIkjI2KwTaxuWxycnIOcYGOtN07wvrNkNDWbWbGZdMkleXZproZw4IAX98dmAx5O7PHSurooB6nNxaRe2XxKuNWgiEthqenJDcPvAMMsLsU4PJDLK3ToU560mh6F4g0zVr+4vda065tb+4a4lig0ySKRW8tYxtczsAAEU8qcnPTt0tFVKTla/a39f10A4HQfhtPofiWz1WK90jNuskcrw6N5dzeK45aacylnkyAd2ACS2V5G3a0rQNb05rWzbXoH0azGyK2TTws7xgbUjklLsrADGSsaE7RyOc9JRUgcN4c+G8HhrWre5s7bw39nttwikTQFjvtpUgZuVkwWwcFvL+YZzyc1LoXgO40zxNHrN7daQ9wnmGSfTtGFlcXbOMHz5BIwkHO7G0ZYK3bFdpRTuweplaemrNr2qz38pXT90cdjb4Q8BcvISBn5mYjBPATOOa03DmNhGVV8HaWXIB7ZGRn86dRRJ3A5G08Na3p1vozPqlnef2S00kkUOnMjXW4MAqFp8IcMRliQTjoKzfA/h/xKNF0q38Rulna6bJvSxks4xOzAHaTNFO6MgLHHyKx2jPct6BRSWgdLFG3tL+PVb6e41IzWc6xi2tRAq/ZiAQx39X3HB56Yrj9B+G0+heJbPVob3SN1srxyPDo3l3N4rjlppzKWeTIB3YAJLZXkbe+op31uHSxzugaNr+mahdy6lrGm3ltdStO0VvpkkDhyFAw5ncYwvTbn3o0HRdf02/vJNS1nTru2u5WmaO20x4HVyFUYdp3GAF6bfxroqKX+VvkG/wB9/mcDoPw2n0LxLZ6tFe6Rm2V45Xh0by7m8Vxy005lLPJkA7sAElsryNt/QLK/m8Yahd3ukXunWEWZLKO5eBl82Q4mdfLlcjO1SMgEb5P72B19FO9gepzcekXtl8SrjVreISWGp6ckNw+4AwywuxTgnJDLK3ToU561Fpnh7xDb6/eX2p61pd1bXyqlxbwaVJCxCqyrtc3Dbfvc5U59q6minKXPv2sHW5x9t4T8QRX2itP4ispbLRpcwwjSmWSVPLaLEj+dgtsY/MqqM87SOBHoXgO40zxNHrN7daQ9wnmGSfTtGFlcXbOMHz5BIwkHO7G0ZYK3bFdpRSu73DpY5vwtpN7a6p4g1bVIhBPqt/ujhDBtkMSCKMkgkZYLv9twHUGjwtpN7a6p4g1bVIhBPqt/ujhDBtkMSCKMkgkZYLv9twHUGukorR1JO/mkvkrf5IAooorIAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKoW2s293q1zp8SyebbZEjEDb0U+v+2PyNX6SaauhJp3t0CisqDxJpdxqSWMU0vnys6RFraRY5SoJbZIV2tjB6E1q0KSaugUk9mFFFFMYUUUwTxtG8kbiRUJDbPmwRwRx3BGMUAPoqCyvIdQs4rq1ZmhlGVLIyH8VYAj6EVPQJNNXQUUUUDCiiq817Bb3dtbSlxLdFlixGxUlVLEFgMA4BPJGcHFF7BsWKKKp2Oq2uozXENsZhJbMFlSW3kiIz0I3qMg46jIpXV7Culoy5RRTEmV5pIwHDR43EowBz6EjB/DOKYx9FFNjljlUmJ1cBipKnOCDgj6g8UAOooooAKKbHLHKpMTq4DFSVOcEHBH1B4p1ABRRTEmV5pIwHDR43EowBz6EjB/DOKAH0UUUAFFFVpL6OPUobHa7TSo0nygYRVwMt6ZJAH/1jRcTdtyzRTY5Y5VJidXAYqSpzgg4I+oPFOoGFFFMSZXmkjAcNHjcSjAHPoSMH8M4oAfRRRQAUUUUAFFFFABRVOx1W11Ga4htjMJLZgsqS28kRGehG9RkHHUZFXKSaauhJp6oKKKKYwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOE8K+GLix8ZahqFzaacigsUki0swsSWZfkYuccDJ65De9d2enFFFZwgqcFCPQzp040726u5yek2t+NWt5oNI1HSAS7Xcc98k1swIPyogkbad5U5CLwD64OZomg39n4liup9HuES4WVL9ttmkLl+cgR/vJEyDzIS3I45OO/oqfYxun2IdBO2r3v0/yOS07wfb22sRK2m2kdjpryTWMixJuLyHPpkbMEc9flPVao+G/DE+l6/BcXFrqguULia7X7EILjIOS7IFmcE4IDAkHGema7uimqUU010B4eHTvcCAQQRkHqK5TRdGj0bVpltPCttARNM/9oRiBN0bEsqJj5yeQuGCqMHngZ6aO5gluJYIpo3mhx5sauC0eRkbh1GRyM1LVuKbuaSgpWv0OK0bRpbC60u6i8Pz2t28kwvbnfCXCtnbvYSEsudpAGcbeg4qHw34Yn0vX4Li4tdUFyhcTXa/YhBcZByXZAszgnBAYEg4z0zXd0VnGjGLTXQy+rx018yvqEay6bcxvbfa1eJgYMKfM4+78xA59ziuC0fwvc6bLZTxeH3glgitfmV4SyON6zkHf1IKkn+JQB2Ar0WinKmpS5vT8C6lKNS1+l/xOO0vRZYb2zMehSWF/EW+16sXib7T8rA5IcyPuYhsOBjHYgVXsNGntdX0iWDwxNaz28jC+v1mg/flo2QyHEm6T5m3ZcbgM4GSRXc1FHcwS3EsEU0bzQ482NXBaPIyNw6jI5GaXsY/1b+l8rE+wjp/wP8vx3OJ8OeGJ9K16C4uLXVBcpvE12v2IQ3GQcl2QLM4JwQGBIOM9M1s6XPqB8S3s1xoV9bQXQjVZpJLcqmxWzkLKTzkYwD+FdFRTjTUUknsEaKjs3vcK5TRoJrTXNTKeFriys77axx9lClgrbiyrKSSxI5wc98V0z3dvHdRW0k8SXEys0UTOA8gXG4gdSBkZx0yKVLiGSaSGOWN5YseYisCyZGRkds1UoXabNJR5mn2OM0LQf7Lm0iW38NPZzF5lvJVMG5YznaHIkJZfu4UZwF6DAFXNE0SPSdWmjtPDFrasZpm/tNFhQeWxLKq7cuTyBtIUcHngA9XRUxpRVjONCMbW6O/T/I4/SNGe31GxYaDJZXsLE3mqF4m+1jaQ2WDmR9zEHDqMYzwQK7AgEEEZB6iimrLG8jxo6s6Y3qDkrnpkdquMVFWNIQUNjl9C0WLSNYlS18L2tsfOmb+0kWFAYmYsqrty5PIGCFAweeADDoGh28OtXM8vhaaxcztNbTyPDshG0DaoSRim47s7Vw3fNdhRUKkk0+3p/XQj2Mfxv0/yOC0DRNS07xTBfNpV1D5okW9kUWccTM3OVEZ8x1yDzISwyOMkkaWjQTWmuamU8LXFlZ321jj7KFLBW3FlWUkliRzg574rq6KUaKikk9hRoqLum97nEaFoTaVNpM1r4bksZ98y3cyGDciHO0MRISy524UZwF6DAFR+HPDE+la9BcXFrqguU3ia7X7EIbjIOS7IFmcE4IDAkHGema7uihUYpp9hLDwVvLU4/wAP6Jbw61c3E/haexY3DTW1w7wlIRtAwqpIxQsd2dqgN3zW1osMk5udTu4njmvGwscikNHCuQikHoTksR6uR2q/bXtreeZ9kuYZ/Lba/lSBth9DjoaelxDJNJDHLG8sWPMRWBZMjIyO2aqNNRSS6FxpqL073OZ0LRYtI1iVLXwva2x86Zv7SRYUBiZiyqu3Lk8gYIUDB54AMOgaHbw61czy+FprFzO01tPI8OyEbQNqhJGKbjuztXDd811dtd297CZbOeK4jDsheJww3KSrDI7gggjsQRUtHskmvL0/roL2MdvO/T/I4LQNE1LTvFMF82lXUPmiRb2RRZxxMzc5URnzHXIPMhLDI4ySRpaNBNaa5qZTwtcWVnfbWOPsoUsFbcWVZSSWJHODnviuroqY0VFJJ7CjRUXdN73OI0LQm0qbSZrXw3JYz75lu5kMG5EOdoYiQllztwozgL0GAKj8OeGJ9K16C4uLXVBcpvE12v2IQ3GQcl2QLM4JwQGBIOM9M13dFCoxTT7CWHgreWpx/h/Q7eLW7mebwtPYsbhpra4d4dkI2gYUJKShY7idoAb+LNdHb3F7Nqd5b3On+TaRBPIufOVvtGR8w2DlcHjnrV2irjBRiorZGkaajt3v0OCtbTRdUurnR7K203U7HSWe6it7cwvueTOEwTgFTv64/g54NQaL4YudLmsbiPw68UlvFanKvCWRxvWcg7+rAqSe6gDsBXolFZqik1Lqrfht/XkZPDxe/n+P+XQ53S59QPiW9muNCvraC6EarNJJblU2K2chZSecjGAfwroqKK1jHlVjeMeVbhRRRVFBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAeeeE4oLvxHeML3xcrw3TyWMWoxX0dusW0ZBMqBXyxPEjMRj5NtZ3gy31+38VWp1fXNUN23mLfWkujagYJW2kkCeSZ7ZAGAKtGqggBRjO2vVKKFpb0sD1v5nneiRtJq2nm4k8XnXUkY36yfaFsS+1gc78QGINjaITuPy9fmo0Brg6tpphk8SHWizDVhqMd2LMjY2/bvHkACQJt8nBx6rur0SijoN6nlWlHf4k0KG9u/FSX7pPJq0U73kdvJMsZJ8vOEYBi21YcoRgkcLVrwoH0+58P3Hm+Kp5r57iK7GopfSIiDOzekg2xYIQByFJGSS2WJ7u00DR7DVLjUrHSbG2v7rPn3UNsiSy5OTucDLcgHk1oUCepw/hTV7dfFmq2kEfiP7NdvG9s2pWV+Y1YKxkw86YjGcYGQOgArJ8LHzfHjTf8AFXxKsk0EEGoC/MEhwxM8xlxCF42xonTcCeThPTqKAPK/Blvr1t4qtG1jXNUN0fMW/tJdG1AwSttJIE8kz2yAMAVaNVBACjGdtaPhKK3uvEl2xvfF6mG6aSyh1CO+jt1i2jcC0yBX3EtxIzEY+TbXodFNaO/kD1M5b1dRvtS0p7S/t1gjRTdFTHHKJFP+qkBySvcjBBxXDaBD4iudS0XTtTg1KGysJWYXX2m6V5jChWTz2Z/3ivKysgYYKK2MivSqKOt1/XmHQ888JRW914ku2N74vUw3TyWUOoRX0dusO0bgTMgV9xLcSFiMDZtq14V1a3HivVrOGPxJ9lvGje2bUbLUPLVtrGTEk64jGcYGVHTFdzRS6JA92zgPCVha6LrtxYQ2fiR777VdFpbm4vJLNImkZ0ctK/lOSCg+Tc+ScgYYjK8GW+vW/iu0Osa5qhum8xb60l0bUDBK20kgTyTPbIAwBVo1UEAKMZ216pRTvrcHrfzOOtv+S0X/ANu+/wD2LD/Z+emzzX8/b77vKz7bfaunh0+GC+u7tN3m3ezzMnj5RgYqV7S3kuormSCJ7iFWWKVkBeMNjcAeoBwM464FS1c5cyXkv6/ruHVv+trHP6PBFZX8kAh1MzedKS0kkzQqhYkNl22MSCOmWyee5pmkKkmpTHztYykpaBLhZ1jCYGRlxg5JPDEkY4xXR0VmByekTXya5C0xvfLnV/Oia3uCkTHkAu7FTjkZRVX9BV3TbuManexBdT8ibaYzPBckA4O7DOPl+mR7Vv0UAcxpW+3k0+QPqryTmRJftKzsoHO3IYYXnbg8E+p5qPRY7+PVITe3135vzC4hezuCjnB/jZ2jAzyCoGenfFdXRQBzukJHLqUxM+sApKWgS4WdYwmBnlxhsknhiSMcYrZhvPPvrm1+zXEfkBf3rphJNwz8jd8d/SrNFHSwdbnNtbuWewklvvLsS87yx3EpeVWB2LlTubqeP9gcc1S0wXcElpLK+rvtSBmEwnYbm3CXII55C8Hp1GMk11Ftp1jZSPJZ2dvbvJ99oolUt9SBzVmgH2MTTr6OXXrzEF4qziMRtJZyop2qc8soA/GtKCwht7+6vI93m3WzzMnj5RgYqzRQBx3gn/kYvGP2T/kHf2sPJx93zvJTz8f9tM5/2t3fNdjUVtaW9lCYrO3it4y7OUiQKNzEsxwO5JJJ7kk1LV1Jczv5JfcrB3CiiioAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA/9k=)

These snippets of code show the register\_new\_face function taking the username parameter and the users name to the names\_list function to be written to the text file containing registered usernames. The next line of code calls the face training module. The GUI window closes, and the Pi-camera starts, opening a new window on the screen. The user can now see their own image on the screen. The system captures 50 images of the users face and stores them into the dataset of profiles. The camera window closes. The count parameter is passed and is written as the unique id for the dataset images. The system next takes the path to the dataset and passes it to the getImageAndLabels function, this function is part of the training module. The function takes the images in the dataset and splits them into ids and faces and stores them in two arrays. These two arrays are used to create a model for the face recognition software using the recognizer included in the OpenCV package, the LBPH(Local Binary Patterns Histogram) Face Recognizer. The model is then stored in the trainer.yml file.

The GUI opens on the “Begin Facial Recognition” screen.





When the button is clicked the begin\_facial\_recog() function is called and the register\_on\_system() function is then called inside of this. Register\_on\_system() returns the username of the recognized driver which is passed to registered\_username, which in turn is used to update the user in the database. The returned username came with a blank space in front and after it. These blank spaces had to be removed using the .strip() Python function for the database to recognize the username.

The in-vehicle part of the system was now complete, tested and validated.

# Web Application

## Overview

The back end of the web application is a NodeJS server with a NoSQL Mongodb database. Both are hosted in the cloud. The NodeJS server is running on an EC2 instance of Ubuntu running on Amazon AWS. The database is hosted in the cloud by Mongodb Atlas. The front-end design and functionality use JavaScript, html, CSS, Express, Bootstrap and Ajax. All passwords on the system are hashed using Bcrypt before being stored, securely, on the database.

The web application has a landing page, a login and register for admin users, a register new driver details page and a page that displays all drivers, active or inactive, registered on the system.

## Changing Direction

The planning, design and development of the web application was the most challenging part of this whole project. As we had covered networking with Java in semester one of this year using MySQL, Servlets and JSPs, and this was my only exposure to networking up to this point on the course, the decision was made to use a Tomcat server to host the web application using Java, html and CSS to design, build and control the system and a MySQL database to store the information. Several weeks were spent writing code and researching further how-to set-up MySQL on the Pi and how it would be possible to have the two systems work seamlessly together. It seemed that MariaDB was a version of MySQL for the Pi, which was fine as it had the same syntax and basic language. Heidi had been used to build and host the MySQL databases, when we were covering the module in our labs, but Heidi is not available for the Pi. The plan, at this time, was to look at hosting the database on a local machine and trying to get the Pi to access the information there.

There was a direction change in the functionality requirements of the app in the middle of January, instead of the Pi contacting web app, the setup was changed to allow the login of the web app redirect to a landing page on the Pi. After extensive research, the conclusion was that an Apache server would have to be installed and run on the Pi. SSH access had to be enabled on the Pi to allow the webapp gain access. The functionality of the project at this stage was to have the face training run remotely on the web app and the dataset to be saved to the MySQL database. Researching the topic of storing large datasets on databases led to the conclusion that this was not the standard method of storing dataset information. The Standard would be to save a link to the dataset in the database and access the dataset in this way. Which would allow the dataset to be stored on the Pi. There was some success in contacting the Pi from the web app. With a sever now running on the Pi on top of the other applications the Pi was struggling, simply because of the limitations of the CPU and memory on the board.

Investigations were still ongoing with trying to run the face training end of the recognition software remotely. The problem statement was trying run a Python Script from a web application. A possible method was investigated and experimented with using .cgi files. Common Gateway Interface (CGI) is an interface specification for [web servers](https://en.wikipedia.org/wiki/Web_server) to execute programs like [console applications](https://en.wikipedia.org/wiki/Console_application) (also called [command-line interface programs](https://en.wikipedia.org/wiki/Command-line_interface)) running on a [server](https://en.wikipedia.org/wiki/Server_(computing)) that [generates web pages dynamically](https://en.wikipedia.org/wiki/Dynamic_web_page).[8] Following a number of tutorials on using CGI to run a Python Script from the web. Simple text-based scripts ran and displayed on the web page, but there was no information on using these files to run more complex scripts. The project mentor had no idea and was not able to give any support with CGI but advised that there should be research carried out on Flask for Python and investigate if it is possible to install on the Pi.

Flask is a lightweight web server gateway interface, understanding what this actually meant was very difficult. Flask was installed on the Pi and following tutorials available on YouTube, once again, simple sample scripts could be got to run and display text in the browser. This seemed like it was progress, but when running any scripts that had more complexity or required external dependencies errors occurred, usually based around permissions. The understanding was that even though the scripts that were being run are on the pi and have all the add-ons need to run them on the pi, running from the server those dependencies needed to run the scripts do not exist on the server. Which caused the system to crash.

At this point a consultation was had with highly skilled Networking Engineer with over 20 years’ experience. The discussion revolved around the best approach to take when dealing with communication between the Pi and the Web App. His advice at that time was invaluable, it again changed the functional requirements and scope of the project. He advised that everything local on each part of the system should be kept local to that part. He maintained that there was no requirement to have the face training held remotely, but to host it locally on the Pi as it had been done initially. The system should have at least two levels of verification, a user access code and a password. These should be set by an administrator or fleet manager and should be held in encrypted form on the database. Access to the face training and registering a profile on the vehicle should only be allowed if a user is registered on the database and provides the right information to the system. He advice was also to keep the dataset and driver info for the face recognition system local on the Pi and to remove Flask and the Apache server as they were causing a strain on the Pi’s resources.

His advice was taken onboard and an email was sent out to the project mentor and to one of the course lecturers so a discussion could be held on were the project was now and the new changes in functionality that needed to be made. After meeting with both and taking their advice onboard, a decision to try and go in the NodeJS and Mongodb direction was made. It was felt that more technical support could be gleaned from the course lecturer if any major problems arose. The work that had taken the last five weeks of the project was set aside. The project was changing from Tomcat to NodeJS, from MySQL to NoSQL , Heidi to MongoDB and from Java to JavaScript. Any work that was transferable was saved everything else was set aside. There was very little that could be transferred, and this was akin to starting the web application all over again. After weeks of feeling that there was no focus or direction, this new start was what was needed.

## Design and Development

Once the decision was made that NodeJS was the best approach to take for this project it was time to plan the work. An idea of how the new web app should look had already been formulated. A timeline for how to proceed had to be developed, and the functionality of the app had to planned with a timeline of deliverables . The scope of the app changed several times through its development. Research about the languages need to design the app was conducted online. The layout of the initial pages for the app were quickly designed without any functionality.

Through discussions with the project mentor, and heeding his advice, it was decided that the Mongodb database should be moved to the cloud . This was another functional change in the project as it had always been planned to have the database hosted locally. Time was spent researching how this might be done. The conclusion was to sign up for a Mongodb Atlas account and host the database there. An account was setup and clusters(groups of databases) and collections(individual databases) were created for the project.

A lot of time was spent looking at tutorials on NodeJS. The biggest problem was the lack of understanding of html and JavaScript, which was making the navigation through the tutorials so much slower. The landing page had been designed for the webapp but there were problems getting it to display properly. The page would not display with the CSS stylesheet info. It was the lack of understanding of the languages and features being used that caused the problems at this stage. It took some time to figure out the URL format in JavaScript that Node was looking for to access the stylesheets. This piece of knowledge alone allowed for the code to be rewritten with the proper URL format and for the landing page to display. A mock login and registration page could now be used to write to, and retrieve data from, a local instance of MongoDB.

The next major problem was trying to change from the local database to MongoDB Atlas. A consultation was held with one of the course lecturers again. He explained that the way the code for the application was being written was the most difficult way that could have been chosen. He advised not to proceed any further in that direction and instead to look at his mini project code as an example of how the application should be written using express in NodeJS. The major problem with his application was it required the code to be written in the Atom IDE. The code in Atom was examined and broken down to where there was a level of understanding. A new project was started from scratch again in IntelliJ , it was initialized as a NodeJS project with express, and all dependencies were installed (express, express-handlebars, Mongodb, mongoose and HBS). The server was displaying the landing page in good time. The web app was now taking shape. Writing to the cloud instance of Mongodb was still causing an issue.

This was the next problem that had to be conquered. Research was again carried out through the internet. A lot of the tutorials and forums had inaccurate information which was incredibly frustrating. The first solution to connecting to Atlas came simply in the naming of the collection. A hyphen had been used to separate two words in the name of the collection. Node read it as two names and therefore could not find the database. A new database with a single name was created.

After looking again at setting up a cluster and collection on Atlas the realization was made that no credentials were setup for the user and no IP address had been added to the whitelist. Only if an IP is contained in the whitelist can it access the cloud server. New credentials were setup on Atlas for accessing the collection. The connection URI string, copied from Atlas, replaced the localhost URL contained in the index.js file, in the routes folder of the app. The app was now connecting to Atlas and reading and writing from the collection. Solutions that took so much time to find, caused so much frustration, always looked so easy when discovered. The app was also able to take the data returned from the database and write it to a json file. This json would be used, later, to dynamically populate a table of all drivers registered on the system.

The next step in the development of the web app was to introduce encryption for any sensitive information that was held on the database. It is never good policy to store passwords or access codes in plain text on a database. Research showed that Bcrypt was the most used hashing algorithm available as a plugin on NodeJS. Bcrypt has a built in Salt which makes it more secure than just hashing. In cryptography, a Salt is random data that is used as an additional input to a one-way function that hashes data, a password or passphrase.[9] The Salt adds to the complexity of the cypher, counteracting rainbow table attacks. This is an example of the word password hashed with Bcrypt, with a Salt value of 12 in the web app and stored on the database, "$2b$12$tFr7lF4upQE0sTwuBJd0/uTk.uOC5saBPPh3qZfXbWWlA69Ml4yKm". There are three fields stored in the hash delimited by “$”. [10] The 2b identifies the version and memory allocation of the hash. 2 refers to the Blowfish-based crypt, b indicates that the crypt will allocate 184 bits to store the resulting hash value. 12 refers to the Salt, and a cost value of 212 , this cost value slows down the computation of the hash when encrypting and decrypting. tFr7lF4upQE0sTwuBJd0/u is the Salt text added to cipher text. Tk.uOC5saBPPh3qZfXbWWlA69Ml4yKm is the cypher text itself. For testing a Salt value of 10 or below will do, but for production 12 or more is recommended. The main challenge when using Bcrypt was when it came to encrypting multiple entries for the driver info. All tutorials online only encrypted one password. The functionality of the app required a driver’s username, password and access code to be hashed before sending to the database. The solution for this was to get a good understanding of JavaScript functions, of how to use nested functions in JavaScript and how to make this work in the app.

var driver = {  
 driversname: req.body.driversname,  
 username : req.body.username,  
 password : req.body.password,  
 reg : req.body.registration,  
 code : req.body.code,  
 active : "No"  
 }  
const dbName = 'facerecog';  
MongoClient.connect(uri, { useUnifiedTopology: true } , (error, client) => {  
 if (error) {  
 throw error;  
 }  
 database = client.db(dbName);  
 collection = database.collection("users");  
 ***console***.log("Connected to `" + dbName + "`!");  
 collection.insertOne(driver);  
 database.collection('users', function (err, collection) {  
 bcrypt.hash(driver.password, BCRYPT\_SALT\_ROUNDS)  
 .then(function(hashedPassword) {  
 return collection.updateOne({"driversname": driver.driversname }, {$set: {"password" : hashedPassword}});  
 });bcrypt.hash(driver.reg, BCRYPT\_SALT\_ROUNDS)  
 .then(function(hashedCode) {  
 return collection.updateOne({"driversname": driver.driversname }, {$set: {"code" : hashedCode}});  
 });bcrypt.hash(driver.reg, BCRYPT\_SALT\_ROUNDS)  
 .then(function(hashedUsername) {  
 return collection.updateOne({"driversname": driver.driversname }, {$set: {"username" : hashedUsername}});  
 })  
 .then(function() {  
 writeToJSON();  
 });

The code snippet shows the solution to the problem of hashing multiple pieces of data. The decision was to take all the drivers details into a driver variable and write this to the database. To immediately then take each of the sensitive pieces of information and to hash them separately and update the information in the database with the newly hashed values, overwriting the plain text initially stored. This happens so quickly that the original information never appears on the database. This solution worked extremely well.

Moving on from that success it was time to start looking at developing the page in the app for displaying the table of driver data. As always, it was back to the web and to find tutorials that could give some direction in on how to code this dynamic table. The data from the database was already stored in a json file on the system, what was need was a way to extract the data and display it in the table. The knowledge online pointed towards using a JavaScript with Ajax to read in the file and convert it to an object, retaining the json key/value attributes. The initial setup was to have a button that would call the function to read from the json. The code could not get the data to display in the table. To try and find a solution console.log(), print to console methods in JavaScript, was dropped into the code to try and debug and understand why there was no output. As a result of printing to the console the problem started to become clearer. In creating the json file the code was using stringify() and JSON.parse() to write to the json file and Json.parse() again in Ajax. What this meant was, the object returned from the database was been taken, converted to a JSON string, then converted to a JSON object and finally converting that JSON object to another JSON object. The code could not extract any information from this last object. Again, the solution was a simple one, but took an understanding of what the code was doing to figure out. The solution meant that as the information came from the database it was to be converted to a JSON string and then saved to a file, allowing the Ajax function to convert that json file into an object. The table would now display from the button and show all the data retrieved from the database. The code had to be rewritten to use the power of the JSON object and only display the relevant information in the table. The final step for this page was to have it display the populated table, automatically, when the page launched. This was achieved by changing the code, including the Ajax function, from waiting for a reference to a button push to now loading when the window loaded.

$(***window***).on('load', function () {  
 $.ajax({  
 url:"tripleVData.json",  
 dataType:"text",  
 success: function(data)  
 {  
 ***window***.obj = ***JSON***.parse(data); //setting the variable to window makes in global to all JavaScript functions  
 var table\_data = '<table class="table table-bordered table-striped ">';  
 ***console***.log(obj.length);  
 for(var count = 1; count<obj.length; count++) {  
 if (obj[count].active == "Yes"){  
 table\_data += '<tr class="table-success">' +  
 '<td>'+obj[count].driversname+'</td>'+  
 '<td>'+obj[count].reg+'</td>'+  
 '<td>'+obj[count].active+'</td>'  
 + '</tr>';  
 }else{  
 table\_data += '<tr class="table-danger">' +  
 '<td>'+obj[count].driversname+'</td>'+  
 '<td>'+obj[count].reg+'</td>'+  
 '<td>'+obj[count].active+'</td>'  
 + '</tr>';  
 }  
 }  
 table\_data += '</table>';  
 $('#employee\_table').html(table\_data);  
 }  
 });  
 });  
</script>  
<script>  
 setInterval(function () {***document***.getElementById("dataButton").click();}, 10000);  
</script>

The top line in this code snippet tells the system to run this function when the window loads. Every time a new window loads it creates an a new JSON object and setting it to window makes it global to all JavaScript functions. This piece of code successfully populates the table with the data held in the .json file, but it had no effect on updating the .json, so each time the table was loaded it showed the same data. So, the logical step was to automate the update and display of data in the .json. The solution for this was to create an invisible button on the webpage. Assign the button an id and set a timer that functioned to click that button at a particular interval. The last line of code in the above snippet satisfies that functionality. Every 10000msecs(10seconds) it grabs the button, from its id, and clicks it.

<form method="post" action="/updatedata">  
 <button id="dataButton" type="submit" class="btn btn-info"style=" margin:40px; float:right; visibility:hidden;">Update Data</button>  
</form>

The above code is for that invisible button.

***router***.post('/updatedata', function(req, res, next) {  
 const dbName = 'facerecog';  
 MongoClient.connect(uri, { useUnifiedTopology: true } , (error, client) => {  
 if (error) {  
 throw error;  
 }  
 ***console***.log(dbName);  
 database = client.db(dbName);  
 collection = database.collection("users");  
 ***console***.log("Connected to `" + dbName + "`!");  
 database.collection('users', function (err, collection) {  
 writeToJSON();  
 setTimeout(function () {  
 res.redirect('/pages/fleettable.html')  
 }, 2000);  
 });  
 });  
  
});

function writeToJSON(){  
 collection = database.collection("users");  
 collection.find().toArray((err, resultArray) => {  
 if (err) throw err;  
 ***console***.log(resultArray);  
 var json = ***JSON***.stringify(resultArray);  
 fs.writeFile("C:\\Users\\john\\OneDrive - GMIT\\FinalYearProject2020\\Final\_Year\_Project\_Logs\\WorkingFYP\\public\\pages\\tripleVData.json", json, function (error) {  
 if (error) throw error;  
 ***console***.log("Write to tripleVData.json successfully!");  
 });  
 });  
}

This snippet of code is from the index.js file in the routes folder. Every time the button is automatically pressed it preforms the action of calling the top function. This function connects to the cloud database and runs the second function, writeToJSON(), which extras the data from the database and overwrites the .json file. It then redirects to the page displaying the table and updating the table. This all takes place in the background, so the user only gets to see the updated table. The redirect for the fleettable page is held within a delay function to give time for the new data to be received from the database and written to the .json, before trying to display it. The last detail to be added to the table was to use Bootstrap to color code the entries. If a driver is active on the system the table displays them in green, otherwise their details are displayed in red.

This was the final detail of the webapp. It was now running and functioning how it should. The database was held in the cloud, but at this stage the NodeJS server was a local instance held on a PC. It was time to move it to the cloud.

## Moving to the Cloud

### Setting Up Amazon EC2 Instance

The webapp was moved to an Amazon EC2 instance[11]. EC2 or Elastic Computing, is a platform where you rent a virtual machine. It gives the ability to manage the resources of the machine, CPU, RAM, and storage as needs require. Creating an EC2 instance is creating a virtual machine in one of Amazons massive data centers.

To move the system to the cloud an Amazon Educate account, PuTTY, PuTTYgen and WinSCP are needed. An Amazon Educate account was setup first. This is a special account that Amazon have for students where no credit card information is need and the account initializes with $100 of credit. Once the account was setup and logged into the AWS Management Console was selected. A list of services is displayed on the screen, EC2 is found under the Compute tag and selecting it brings up the EC2 Dashboard. It is here that an instance can be created and managed. When a new instance is launched an image must be selected for the virtual machine. This image is the operating system that will run on the machine. For this project, the latest version of Ubuntu was selected. When setting up the configuration of the system, only free tier options were selected. These do not have any cost involved unless they are running and even then, the cost of hosting the virtual machine is greatly reduced. On the Security Group screen a had new rule needed to be added to allow PuTTY SSH into the instance through a certain port. This is similar to the white list setup on Atlas. The port chosen for this project was port 3000, an initially the code had been written to allow access locally through port 8000, but AWS has those higher ports reserved and will not allow access through them. To make the instance publicly accessible “Anywhere” must be selected from the Sources menu.

The next step in the process was to create a key pair to allow the virtual instance to SSH into the system. The key pair consists of a public key that AWS stores and a private key file that must be stored locally and kept safe. Without the private key it is impossible to connect to the EC2 instance. With the instance launched Amazon goes to work to set it all up. When the instance is shown as running, the information on the instance appears at the bottom of the page. The public DNS was copied from this information and pasted into notepad, to be used when setting up PuTTY.

### Accessing Instance through PuTTY

The private key file was not in a format that PuTTY supported for SSH keys. Using PuTTYgen the file was converted to a supported format. The private key file is stored in a .pem file and was converted to a .ppk file. PuTTY now need to be setup to allow it to connect to the EC2 instance. In the Session window of PuTTY, the host name had to be set in the format of “<username>@<public\_DNS\_of\_Instance>”. As an Ubuntu OS was chosen the username was ubuntu, as list of the required usernames is available on AWS. The second parameter in the host name was the public DNS that was stored in notepad. The last step to setting up PuTTY was to open the Connection tab, click on SSH and Auth, here is where the converted private key file is loaded. Returning to the Session screen, the session was given a name, the configuration was saved, and the session was started.

### Finishing the Connection

At this stage PuTTY was connected to the EC2 instance and it was now time to move the project files on to a directory in Ubuntu. The new TripleV directory was created and WinSCP was used to move the files over, this made the transition very easy and trouble free as it was simply drag and drop. Some of the larger files to time to transfer. All the dependencies required for running the project then had to be installed on Ubuntu. This was the only part of the migration that caused any issue, again this was down to a lack of knowledge. NodeJS was first to be installed on the virtual machine, followed by Mongodb. Then it was a case of running the instance and looking at the error messages. The error messages pointed to each dependency in turn that was required to be installed. When this was completed the migration to the cloud was complete. Entering the public DNS of the instance followed by the access port number opened the webapp in the browser. The application could now be accessed remotely from anywhere and the all the planned cloud component parts of the system were now hosted in the cloud.

# Conclusion

None of the modules we had covered on this degree course at the inception of this project had covered machine learning, Python, NoSQL, cloud computing, html, JavaScript or many of the other skills I would need to complete this project. Therefore, choosing elements that were supported in detail online was essential. This project was a journey of self-learning and self-discovery as much as creating a proof of concept for demonstration at the end of a year. I faced many challenges during the development of this project. Many within the project, but the greatest challenge was the deterioration of my health, the diagnosis of an ongoing illness and the amount of time missed in the research and early development stages. There were many times my head dropped, and I felt like calling it a day, but I persevered.

The scope of this project changed many times and planning became a nightmare, but with each new functional change the timeline was revisited and revised. The project has many differences from the one I had initially envisioned, but I am very proud of the finished project.

The Raspberry Pi has turned out not to be the powerhouse I thought it would be. I found that it struggled with the face recognition software and unless lighting was very similar at the time of face training and face recognition it could not discern the individual in the image, leading to a number of false negatives.

Not having experience in networking, cloud computing or accessing cloud databases led to a lot of lost time. This I found extremely disheartening. Once I started to get to grips and build an understanding of what I needed to do things started to come together quickly. After taking advise from a lecturer I switched from networking in Java, with a Tomcat server, using Servlets and MySQL to scrapping weeks of work and starting all over again with NodeJS, Mongodb and JavaScript. Once I figured out how to migrate this to IntelliJ, my API of choice and away from Atom, things started to move in a very positive way. This change allowed me to add the encryption I had wanted, to migrate the database to the cloud and to integrate the whole system.

Hosting the webapp in the cloud was not something I thought I could achieve, but with college moving online, due to the national shutdown due to Covid-19, this became critical for a mini project in another module and could easily be transferred to moving the main project to the cloud. Moving the webapp to the cloud meant it could be accessed from anywhere. In the times that we find ourselves in this seemed apt as it would allow fleet managers to work from home while still managing their logistic fleet. Keeping the supply chain moving.

My conclusion, this was a very challenging project with a very steep learning curve in many areas. The challenge was made even greater due to personal health and an unprecedented national lockdown. I believe it has improved my planning, time management and coding skills greatly. I am very proud of how this project has turned out, but I also believe that it could be improved on.

# References

1. OpenCv team (2020), About, <https://opencv.org/about/>
2. Multiple Contributors (2005-2020), OpenCV, <https://en.wikipedia.org/wiki/OpenCV>
3. Paul Silisteanu (2020), Solarian Programmer, <https://solarianprogrammer.com/2019/09/17/install-opencv-raspberry-pi-raspbian-cpp-python-development/>
4. Doxygen (2020), OpenCV [https://docs.opencv.org/3.4/db/d28/tutorial\_cascade\_classifier.htmlhttps://docs.opencv.org/3.4/db/d28/tutorial\_cascade\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)
5. Hackster.io, an Avnet Community,(2020), Real-Time Face Recognition: An End-to-End Project, <https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end-project-a10826>
6. GeeksforGeeks (2020), Python GUI – tkinter, <https://www.geeksforgeeks.org/python-gui-tkinter/>
7. Kelvin Salton do Prado (2017), Face Recognition: Understanding LBPH Algorithm, <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>
8. Multiple Contributors (2005-2020),Common Gateway Interface, <https://en.wikipedia.org/wiki/Common_Gateway_Interface>
9. Multiple Contributors (2005-2020), Salt (cryptography), <https://en.wikipedia.org/wiki/Salt_(cryptography)>
10. Erickson (2011), How can bcrypt have built-in salts?, <https://stackoverflow.com/questions/6832445/how-can-bcrypt-have-built-in-salts>
11. Bilal AlGhazi (22/11/2018), Deploying React / Node.js Application to Amazon EC2 Instance : <https://medium.com/@balghazi/deploying-react-node-js-application-to-amazon-ec2-instance-a89140ab6aab>