**Ministry of Higher Education Modern Academy**

**Computer Science and Management Technology in Maadi**

**Computer Science Department**

******

***Snap & Shape***

**Submitted to**

Modern Academy

Computer science department

**Prepared by**

Arsany Morcos Nageh (Team Leader)

Sama Amr Mohamed

Patrick Nassef Ghais

Marslino Ashraf Shnouda

Amera Ashraf Youssef

**Supervised By:**

Dr. Mona Lackousha

Eng. Nowraa Hamdy

Academic Year: 2024/2025

**Acknowledgement**

We have taken a lot of effort into this project. However, the completion of this project could not have been possible without the participation and assistance of a lot of individuals contributing to this project.

Firstly, we would like to express our gratitude towards our university **“Modern Academy”** and its board members for their professionalism, supervision, and support that helped us grow on our educational journey.

**“Prof Dr. Nabil Deabes”** For the exceptional quality of education we have received that could not have been without his supervision which in return helped us prepare for our professional career.

“**Prof. Dr. Lamiaa Fattouh”** For his guidance and keenness on providing us with the best educational facilities along with the department staff that helped us so much throughout our academic years.

“**Dr. Mona Lackousha”** For her valuable guidance and support in the completion of this project. And for always pushing us to do better. We are truly thankful to have been allowed to learn from her and be inspired by her encouragement since day one.

Finally, special thanks to **“Eng. Nowraa Hamdy”** for her help and effort throughout the year and for always guiding us with patience and care.

All the help and support that we have received during the implementation project is very much appreciated and it is the reason we were finally able to present our idea and make it real, hoping it helps many people out there.

**Abstract**

Snap & Shape" is a cross-platform mobile application developed using Flutter and powered by a .NET backend, designed to revolutionize how users manage their nutrition and fitness. Leveraging the power of artificial intelligence, the app provides real-time food recognition and analysis by processing images captured through the user's smartphone. This allows the system to accurately identify food items, estimate portion sizes, and evaluate nutritional content based on standardized databases.

The application does not stop at food analysis; it extends into providing personalized health insights and fitness guidance tailored to the user’s health profile. Utilizing secure user data, including age, weight, dietary preferences, and existing medical conditions, the system generates adaptive diet plans and exercise routines.

One of the unique features of "Snap & Shape" is its dynamic data integration. Through automated web scraping and curated APIs, the app continuously updates its recommendations with the latest health trends, nutrition facts, and fitness methodologies. This ensures that users receive the most current and reliable health guidance without manual intervention.

Furthermore, the app is designed with user privacy and experience in mind. All interactions are secured with encrypted data protocols and intuitive UI design for seamless user engagement across both Android and iOS platforms.

By integrating AI, real-time data analysis, and user-centric design, "Snap & Shape" empowers individuals to make informed lifestyle choices and adopt healthier habits, promoting long-term well-being and proactive health management.

## **Structure of the document**

This document will be split into Seven main chapters.

**Chapter 1**: Introduction.

**Chapter 2**: Software life cycle

**Chapter 3**: System Architecture and Methods.

**Chapter 4**: System Implementation and Results.

**Chapter 5**: Tools

**Chapter 6**: System Implementation

**Chapter 7**: Conclusion and References

Table of Contents

[**Structure of the document** 4](#_Toc199533719)

[**1. Introduction** 9](#_Toc199533720)

[**1.1.** Problem definition 9](#_Toc199533721)

[**1.2.** Motivation 11](#_Toc199533722)

[**1.3.** Objectives 13](#_Toc199533723)

[**2. Software life cycle & Analysis** 16](#_Toc199533724)

[**2.1** Project Development Methodology (phases) 16](#_Toc199533725)

[**2.2** Planning: 18](#_Toc199533726)

[**2.3** Analysis: 19](#_Toc199533727)

[**2.4** Software Requirements Specification: 20](#_Toc199533728)

[**2.5** system Design: 22](#_Toc199533729)

[**3. System Architecture and Methods** 27](#_Toc199533730)

[**3.1.** AI System Architecture 27](#_Toc199533731)

[**3.2. Description of Methods and Procedures** 30](#_Toc199533732)

[**3.2.1.** Preprocessing 30](#_Toc199533733)

[**3.2.2.** Predictive Analytics and Time Series Modeling 36](#_Toc199533734)

[**3.3 Backend System Architecture** 40](#_Toc199533735)

[**3.3.1** Architectural Overview 40](#_Toc199533736)

[**3.3.2** Main Responsibilities 40](#_Toc199533737)

[**3.3.3** Communication with AI Model 40](#_Toc199533738)

[**3.3.4** CQRS Implementation 41](#_Toc199533739)

[**3.3.5** Security and Configuration 41](#_Toc199533740)

[**4. System Implementation and Results** 43](#_Toc199533741)

[**4.** Dataset 43](#_Toc199533742)

[**4.1.1** Food Detection Dataset (YOLOv8) 43](#_Toc199533743)

[**4.1.2** Freshness Classification Dataset (CNN) 43](#_Toc199533744)

[**4.1.3** Weight Prediction Dataset (LSTM) 44](#_Toc199533745)

[**4.2 Experimental Setup and Results** 45](#_Toc199533746)

[**4.2.1** YOLOv8 Food Detection Model 45](#_Toc199533747)

[**4.2.2** CNN Freshness Classification Model 48](#_Toc199533748)

[**4.2.3** LSTM Weight Prediction Model 50](#_Toc199533749)

[**4.3** System Integration and Performance 53](#_Toc199533750)

[**4.4** Conclusion 54](#_Toc199533751)

[**5. Tools** 56](#_Toc199533752)

[**5.1** Mobile App development ( using Flutter cross-platform ): 56](#_Toc199533753)

[**5.2** Back-End Development: 59](#_Toc199533754)

[**5.3** Main Work 62](#_Toc199533755)

[**6. System Implementation** 71](#_Toc199533756)

[**6.1** Mobile Application Design 71](#_Toc199533757)

[**7. Conclusion and References** 104](#_Toc199533758)

[**7.1.** Conclusion 104](#_Toc199533759)

[**7.2** References 105](#_Toc199533760)

**Chapter 1 Introduction**

# **Introduction**

## Problem definition

Maintaining a healthy lifestyle through proper diet and regular exercise is essential, yet many individuals struggle with understanding what they consume and how it affects their health. Traditional methods of tracking food intake, such as manual logging, are often time-consuming, prone to error, and not intuitive for everyday users. Moreover, generic health and fitness advice fails to consider the unique dietary needs and fitness levels of individuals.

With the growing adoption of mobile technologies and advances in artificial intelligence, there is a tremendous opportunity to create a solution that can automatically recognize food, estimate its nutritional content, and provide tailored health advice. However, achieving accurate food recognition through images remains a challenging task due to the diversity in food presentation, lighting conditions, and portion sizes.

The "Snap & Shape" application aims to solve this challenge by combining computer vision techniques, machine learning algorithms, and health data analysis. The system will process images of food captured by users to identify food items, estimate portions, and evaluate the nutritional quality. It will then provide customized dietary and exercise recommendations based on the user's health profile.

The key research questions addressed in this project include:

What are the most efficient image processing and classification techniques for real-time food recognition on mobile devices?

How accurately can AI models estimate portion sizes and nutritional content from food images?

What data sources and methods can be used to provide personalized dietary and exercise recommendations?

How can user-specific factors such as age, weight, activity level, and health goals be incorporated into the recommendation engine?

What are the best practices for securing user data and ensuring privacy in a health-focused mobile application?

By answering these questions and developing an intuitive, AI-powered application, "Snap & Shape" seeks to empower users to make informed decisions about their diet and fitness, ultimately improving overall health outcomes through technology.

## Motivation

In today's fast-paced world, many individuals find it difficult to maintain healthy eating habits and follow consistent fitness routines due to a lack of time, knowledge, and personalized guidance. Manual methods of tracking meals and planning workouts are not only tedious but also ineffective for the majority of users who seek simple, real-time solutions.

The motivation behind the development of the "Snap & Shape" application stems from the significant potential of artificial intelligence and mobile technology to address these lifestyle challenges. With the increasing availability of smartphones and advancements in AI-powered image recognition, there exists an opportunity to automate the process of dietary tracking and health planning.

"Snap & Shape" is driven by the goal to simplify nutrition and fitness management by making it as easy as snapping a photo. By leveraging AI to analyze meals and recommend tailored fitness plans, the app aims to:

Empower users to make healthier dietary decisions. Encourage consistent and customized physical activity. Reduce dependency on generic health apps and one-size-fits-all advice.

Provide dynamic updates using real-time data from health websites and APIs.

This project is motivated by the broader vision of enhancing public health through accessible, intelligent tools that promote informed lifestyle choices. Through continuous innovation and integration of cutting-edge technologies, "Snap & Shape" aspires to become a daily companion in each user's journey toward better health and wellness.

## Objectives

The primary objectives of the "Snap & Shape" project are focused on leveraging artificial intelligence and mobile technology to assist users in achieving healthier dietary habits and fitness outcomes. The goals are as follows:

To design and implement machine learning models capable of recognizing food items from images with high accuracy and efficiency on mobile platforms.

To develop algorithms that estimate portion sizes and evaluate nutritional content, including calorie count and macronutrient composition, from food images.

To integrate user-specific data (age, weight, health goals, dietary restrictions) to generate personalized dietary advice and fitness routines.

To build a backend system that securely handles user data and supports real-time analysis, recommendation generation, and result tracking.

To automate the extraction of up-to-date health and nutrition information through web scraping and public health APIs, enhancing the accuracy and relevance of recommendations.

To provide a seamless and engaging user experience through an intuitive Flutter-based mobile interface, promoting regular usage and sustained engagement.

To evaluate the performance of the system through user testing, measuring metrics such as food recognition accuracy, user satisfaction, and health outcome improvements.

To explore the scalability of the solution by preparing the application architecture for integration with external health devices and platforms (e.g., smartwatches, fitness trackers).

To contribute to the field of mobile health (mHealth) by demonstrating the practical application of AI and real-time data in promoting individual health awareness and behavioral change.

**Chapter 2**

**Software Life Cycle & Analysis**

# **Software life cycle & Analysis**

## **2.1** Project Development Methodology (phases)

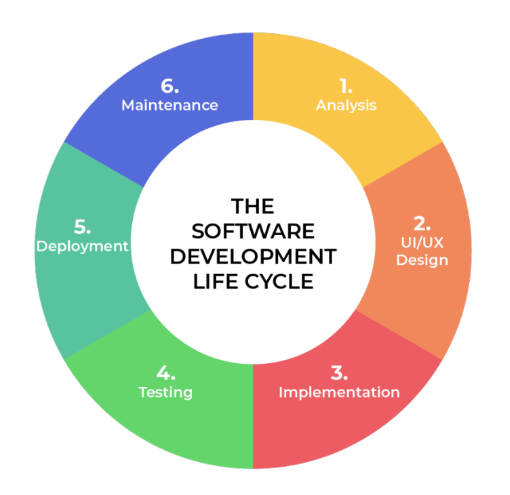
The main characteristics of the Waterfall model are sequential progression through the different stages of a project from initiation to the delivery phase. The waterfall model does have its limitations and because of this, there have been many spin-off models created over the years.

The Waterfall model consists of the following phases:

* Introduction
* Analysis
* Design
* Coding
* Testing
* Implementation And Maintenance

The waterfall model used with the SDLC progresses through six phases. You can picture this process as a waterfall with one phase flowing into the next.

* Planning Phase the purpose of the initiation phase is to conduct an initial high-level investigation of business need and come up with a recommendation for the solution. Once approved by the management team, stakeholders, client or project sponsor, it will proceed to the next phase.
* Analysis Phase the purpose of the requirements analysis phase is to conduct a detailed analysis of the current business needs and identify what options are available to achieve those business needs. During the Analysis Phase, the Business Analyst will create the Business Requirements Document (BRD).
* Design Phase the purpose of the design phase is to identify and document a solution that will be constructed including technical and procedural specifications. A design document will be created that should include but not limited to technical, environmental, data, program, procedural, testing specifications.
* Coding Phase The construction or development phase is where a resource will take the design document created during the design phase and translate it into a functional program or system.
* Testing Phase The purpose of the testing phase is to test the system and related procedures to ensure that it meets the requirements specified by the stakeholders and documented in the BRD, design plan, and testing plan.
* Implementation and Maintenance Phase the purpose of the implementation phase is to release a fully tested and operational product to an end user or customer. The product should meet all the requirements that were documented in the BRD and pass the testing phase before it can be released into a production environment.
* The systems development life cycle (SDLC), or software development process in systems engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.



## **2.2** Planning:

Our "Snap & Shape" application is a cross-platform mobile app designed to promote healthier lifestyles through advanced food recognition, nutritional analysis, and fitness planning. The planning phase focuses on identifying user needs, technological feasibility, and project scope to create an intelligent health assistant that simplifies food tracking and personalized wellness.

**Key Planning Goals:**

• User-Centric Nutrition Tools:

Enables users to take photos of their meals for AI-powered food recognition and portion analysis.

Automatically generates nutrition breakdowns, including calorie count and macro/micronutrient values, based on image processing and external databases.

• Personalized Wellness Guidance:

Offers tailored dietary recommendations aligned with individual user profiles (age, weight, goals, dietary restrictions).

Delivers adaptive fitness plans and exercise suggestions to support users’ health targets.

• Data Security and Accessibility:

Stores user health and dietary data securely with encrypted backend communication.

Ensures access across devices via cloud sync, maintaining data privacy and consistency.

• Continuous Engagement and Updates:

Integrates web scraping techniques to update nutritional data and health advice dynamically.

Plans for user retention through goal tracking, reminders, and motivational insights.

## **2.3** Analysis:

**Requirements Analysis:**

The first step in building the "Snap & Shape" application was identifying both user and system requirements through interviews, market analysis, and feasibility studies. While users have general expectations about tracking meals and receiving health tips, translating these into precise and actionable system requirements require thorough analysis and refinement.

Key challenges included ensuring accuracy in food recognition, handling user-specific dietary data, and providing real-time recommendations while maintaining a user-friendly interface.

**Software Requirements:**

1. **Operating System:**

Android 8.0 (Oreo) or later / iOS 13 or later for mobile app functionality.

Windows 10 or newer for development and backend system operations.

1. **Database:**

Microsoft SQL Server for managing user profiles, dietary data, and system logs.

1. **Development Tools:**

Flutter (Dart) for developing cross-platform mobile applications.

Visual Studio and .NET Core for backend service development.

Python with TensorFlow Lite for training and deploying AI models.

FlaskAPI for implementing AI microservices and endpoint management.

1. **Web Services and Integrations:**

RESTful APIs to facilitate communication between the app frontend, backend, and AI engine.

Web scraping tools for collecting and updating nutritional and fitness data from online sources in real-time.

These software requirements ensured the system's capability to deliver reliable, efficient, and scalable performance to users across platforms.

## **2.4** Software Requirements Specification:

Some essential specifications were established during the development and data integration phases of the "Snap & Shape" application.

* **Functional Requirements:**

User registration and login with secure authentication.

Role-based access control for users (e.g., standard users, admin).

Secure storage and handling of user health profiles and meal data.

Real-time food recognition using the device camera.

Nutritional analysis and display of food items.

Personalized meal and exercise recommendations based on user data.

Logging and viewing historical meal and fitness records.

Options to update or delete user-generated content.

Push notifications for reminders (hydration, meal times, fitness goals).

* **Non-Functional Requirements:**

User registration must be validated via email and a strong password.

The system must run smoothly on Android and iOS platforms.

UI/UX must be consistent, intuitive, and user-friendly.

All app content and interactions must be in English.

The system should support scalability and future updates.

High performance is required during image analysis and recommendation retrieval.

The backend must be secure, with encrypted API communications and data storage.

## **2.5** system Design:

**UML Diagrams**

UML (Unified Modeling Language) diagrams are a set of graphical

notations used to model software systems. UML diagrams can be

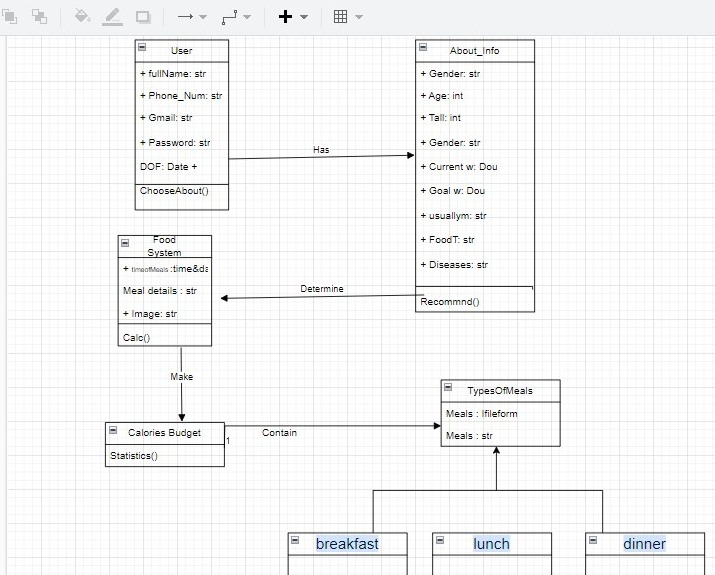
used to visualize, specify, construct, and document software

intensive systems. There are several types of UML diagrams,

**each with a specific purpose:**

1. **Class Diagrams:** Class diagrams are used to model the static

structure of a software system. They show the classes and them

relationships, attributes, and methods.

1. **A diagram of a person

   AI-generated content may be incorrect.Entity Relationship Diagram (ERD):**
2. **Data Flow Diagram (DFD):**

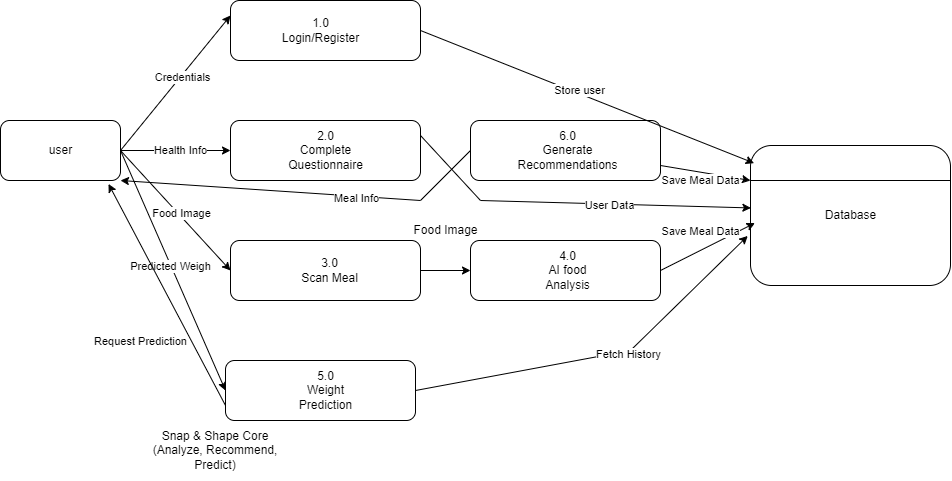
* **Level 0**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **A black and white rectangular object with white text

  AI-generated content may be incorrect.Level 1**
* **Level 2**

****

**Chapter 3**

**System Architecture and Methods**

# **System Architecture and Methods**

## AI System Architecture

**The system architecture of Snap & Shape consists of four main components: preprocessing, feature extraction, classification model, and predictive analytics.**

**Core Components**

1. Preprocessing: This component involves preparing the food image dataset for analysis. Preprocessing steps include memory-efficient image loading, image resizing, color correction & normalization, background subtraction, and data augmentation to ensure that the data is in a format that can be used by the classification models.
2. Feature extraction: In this component, features are extracted from the preprocessed food image dataset. These features include multi-scale convolutional embeddings from YOLOv8, hierarchical CNN features, spatial bounding box information, freshness indicators, and weight estimation parameters from the food images that are relevant to food classification, freshness detection, and calorie estimation. The goal of feature extraction is to reduce the dimensionality of the data and extract the most important features for comprehensive food analysis.
3. Classification model: This component involves building machine learning models that can classify food images, detect freshness, and estimate nutritional content. Three specialized models work in synergy: • YOLOv8 (Real-Time Detection Model): This model uses deep learning architecture with multiple layers of convolutional neural networks trained on 45 different food categories. The YOLOv8 model identifies and locates multiple food items per image, returning bounding boxes, class labels, and confidence scores at high frame rates with a confidence threshold of 0.3. • CNN Freshness Classifier (Deep Learning Model): This model uses several layers of convolutional neural networks trained on fresh and overripe food samples. The CNN model processes 150×150 pixel crops from detected food regions to classify freshness status (fresh vs. overripe) for supported food items including apples, bananas, cucumbers, oranges, and potatoes with a confidence threshold of 0.85. • Weight & Calorie Estimation System: This component transforms detected food regions into weight and calorie estimates using a comprehensive database of 45 food items with predefined weight ranges, reference dimensions, and calorie-per-gram values. The system calculates portion sizes based on bounding box area ratios and applies food-specific conversion factors to provide accurate nutritional information.
4. Predictive Analytics: This component provides longitudinal health tracking and weight prediction capabilities through advanced time series analysis: • LSTM Weight Prediction Model: A deep learning time series model that analyzes historical weight and calorie intake patterns to predict future weight trends over 7-day periods using a sliding window approach. • Hybrid Prediction System: Combines LSTM temporal feature extraction with Linear Regression for enhanced prediction accuracy and interpretability. • Nutritional Trend Analysis: Tracks eating patterns and provides insights into calorie intake trends and their correlation with weight changes.

Overall, our system architecture aims to provide comprehensive food analysis including detection, freshness assessment, nutritional estimation, and predictive health analytics using a combination of preprocessing, feature extraction, deep learning techniques, and time series modeling. The end result is a robust system that can accurately detect 45 different food items, assess their freshness status, provide precise weight and calorie estimates from single food photographs, and predict future weight trends based on eating patterns.

## **Description of Methods and Procedures**

### Preprocessing

**Image Resizing & Memory-Efficient Loading**

Image resizing and memory-efficient loading are critical processes used in data preprocessing that involve changing image dimensions and optimizing memory consumption during processing. In the context of the Snap & Shape food classification project, images are resized to specific dimensions while implementing memory management techniques to ensure optimal performance.

All meal photographs are resized to fixed dimensions: 224 × 224 px for YOLOv8 detection and 150 × 150 px for CNN freshness classification. Large images exceeding 1024 pixels in maximum dimension are automatically thumbnailed to reduce memory consumption while preserving image quality using Lanczos resampling.

The advantage of combining image resizing with memory-efficient loading is that it helps standardize data dimensions while preventing memory overflow issues that can occur with high-resolution images. This dual approach ensures consistent model performance while maintaining system stability during processing.

If we do not implement proper image resizing and memory management, the models may struggle with memory allocation issues, especially when processing high-resolution images from modern smartphone cameras. This can result in system crashes, reduced processing speed, and inconsistent model performance.

Memory-efficient loading includes converting images to RGB format, applying thumbnail resizing for oversized images, and implementing proper memory cleanup procedures throughout the processing pipeline. This ensures that the system can handle various image sizes and formats without compromising performance.

In conclusion, image resizing combined with memory-efficient loading is an essential preprocessing step that standardizes image dimensions while optimizing memory usage. This approach ensures reliable and consistent food classification performance in the Snap & Shape system.

**Color Correction & Normalization**

Color correction and normalization are processes used to adjust colors and standardize pixel values in images to ensure consistent analysis. In the context of the Snap & Shape food classification project, color correction involves RGB conversion and white-balance adjustment, while normalization scales pixel intensities to optimize model performance.

Images are converted to RGB format to ensure consistent color representation across different input sources. Pixel intensities are normalized to the [0, 1] range by dividing by 255, which standardizes the input values for neural network processing. When leveraging pretrained backbones, standardized channel means and standard deviations (e.g., ImageNet statistics) are also applied to maintain compatibility with pretrained weights.

The advantage of using color correction and normalization in the food classification project is that it ensures consistent color representation and standardized input values across different devices and lighting conditions. This is crucial for food recognition because the same food item can appear very different when photographed with different cameras or under various lighting conditions.

If we do not use color correction and normalization, the models may struggle to accurately classify food items due to inconsistent color representation and unstandardized pixel values. Variations in camera settings, lighting conditions, and color formats can make it difficult for the models to identify the same food items consistently, resulting in lower accuracy and performance.

The normalization process also ensures that pixel values are within an optimal range for neural network training and inference, preventing issues related to gradient vanishing or exploding that can occur with unnormalized input data.

In conclusion, color correction and normalization are important preprocessing steps that ensure consistent color representation and standardized pixel values in food images. These techniques improve model performance by maintaining visual consistency across different photographing conditions and optimizing input data for neural network processing.

**Enhanced Freshness Detection & Background Subtraction**

Enhanced freshness detection combined with background subtraction is a sophisticated technique used in image preprocessing that involves identifying overripe food items through sliding window analysis while removing irrelevant background elements. In the context of the Snap & Shape food classification project, this dual approach focuses model attention on relevant food regions while detecting freshness status that may not be captured by initial object detection.

The system implements a sliding window approach with configurable window size (25% of minimum image dimension) and stride (50% of window size) to scan for overripe food items that may have been missed by the primary YOLO detection. Non-food regions are simultaneously masked out through intersection-over-union (IoU) analysis to prevent false detections from background elements like tables, utensils, and decorative items.

The advantage of enhanced freshness detection with background subtraction is that it provides comprehensive food analysis by combining spatial attention mechanisms with quality assessment. This approach ensures that both fresh and overripe food items are accurately identified while eliminating distracting background elements that could lead to false positives.

If we do not perform enhanced freshness detection and background subtraction, the models may miss overripe food items that appear differently from their fresh counterparts, and may be confused by irrelevant background objects. This can lead to incomplete food analysis, incorrect freshness assessments, and reduced overall system accuracy.

The enhanced detection system uses IoU thresholds (0.4) to prevent overlap with already detected regions and applies high confidence thresholds (0.85) to ensure reliable freshness classification. Background subtraction reduces computational load by focusing processing power on relevant image regions.

In conclusion, enhanced freshness detection combined with background subtraction is a critical preprocessing component that ensures comprehensive food quality analysis while maintaining focus on relevant image regions. This technique significantly improves the accuracy and reliability of food freshness assessment in the Snap & Shape system.

**Data Augmentation**

Data augmentation is a technique used in data preprocessing that involves creating new training data by applying various transformations to existing data. In the context of the Snap & Shape food classification project, data augmentation is used to artificially increase the size of the dataset and improve model robustness.

Geometric and photometric transformations (shear up to 0.2, zoom up to 0.2, horizontal flips, nearest-neighbor fill) are applied during training to simulate real-world variability and prevent overfitting.

The advantage of data augmentation is that it helps increase the size of the dataset, which improves the generalization and robustness of the models. By creating new training data through various transformations, we expose the models to a wider range of variations and patterns in food photography, helping them better learn to recognize and classify food items under different conditions.

If we do not perform data augmentation, the models may overfit to the training data, meaning they memorize the training examples rather than learning to generalize to new food images. This can lead to lower accuracy and performance on new and unseen food photographs.

Data augmentation simulates real-world variations that users might encounter when taking photos of their meals, such as different angles, slight rotations, and varying zoom levels. This helps the models become more robust to these natural variations in user-generated content.

It is important to note that data augmentation should be used carefully, as some transformations may introduce unrealistic artifacts that don't represent actual food photography scenarios. Therefore, appropriate transformations are chosen and their impact on model performance is carefully evaluated.

In conclusion, data augmentation is an important step in data preprocessing that creates new training data through various image transformations. Data augmentation improves model performance by increasing dataset size and exposing models to realistic variations in food photography, ensuring accurate and reliable food classification in the Snap & Shape project.

### Predictive Analytics and Time Series Modeling

**LSTM Weight Prediction Model**

The LSTM (Long Short-Term Memory) weight prediction model is a deep learning approach designed to analyze temporal patterns in user health data and predict future weight trends. This component extends the Snap & Shape system beyond immediate food analysis to provide longitudinal health insights and predictive capabilities.

**Model Architecture: The LSTM model employs a multi-layered architecture specifically designed for time series prediction:**

* Input Layer: Processes sequential data with a 7-day sliding window, incorporating both weight and calorie intake features (input shape: 7 timesteps × 2 features)
* LSTM Layers: Three stacked LSTM layers with 100, 200, and 200 units respectively, with dropout regularization (0.2) to prevent overfitting
* Output Layer: Dense layer predicting 7 future weight values, enabling week-ahead forecasting

Data Preprocessing for Time Series: The predictive analytics component implements specialized preprocessing techniques for temporal data:

* Missing Value Imputation: Uses SimpleImputer with mean strategy to handle missing weight and calorie intake values
* Feature Scaling: Applies MinMaxScaler with range (0,1) to normalize input features for optimal LSTM performance
* Sliding Window Creation: Generates training sequences using 7-day historical windows to predict subsequent 7-day periods
* Data Validation: Ensures temporal consistency and handles edge cases in sequential data

**Hybrid Prediction System: The system combines LSTM temporal feature extraction with Linear Regression for enhanced prediction accuracy:**

* LSTM Feature Extraction: The trained LSTM model generates temporal features from input sequences
* Linear Regression Layer: Uses LSTM outputs as inputs to provide final weight predictions
* Model Integration: The hybrid approach leverages LSTM's temporal modeling capabilities with Linear Regression's interpretability

**Training Configuration:**

* Optimizer: Adamax optimizer for stable convergence in time series applications
* Loss Function: Mean Absolute Error (MAE) for robust prediction error measurement
* Early Stopping: Monitors validation loss with patience of 15 epochs to prevent overfitting
* Batch Processing: Uses batch size of 8 with 10% validation split for efficient training

**Advantages of LSTM Weight Prediction: The LSTM-based approach provides several key benefits for health tracking applications:**

* Temporal Pattern Recognition: Captures complex relationships between eating patterns and weight changes over time
* Multi-feature Analysis: Simultaneously considers weight history and calorie intake patterns
* Robust Prediction: Handles irregular eating patterns and weight fluctuations common in real-world scenarios
* Personalized Insights: Adapts to individual user patterns for personalized health predictions

**Model Persistence and Deployment: The system implements comprehensive model persistence for production deployment:**

* LSTM Model Saving: Trained model saved as "lstm\_model.h5" using TensorFlow/Keras format
* Scaler Persistence: MinMaxScaler saved as "scaler.pkl" using joblib for consistent data preprocessing
* Linear Regressor Storage: Hybrid model component saved as "linear regressor.pkl" for complete system reconstruction

**Limitations and Considerations: While the LSTM weight prediction model provides valuable insights, certain limitations should be considered:**

* The model requires consistent data input patterns for optimal performance
* Extreme dietary changes or health conditions may affect prediction accuracy
* The 7-day prediction window balances accuracy with practical utility for health planning
* Regular model retraining with new user data improves long-term prediction accuracy

In conclusion, the LSTM weight prediction model represents a sophisticated addition to the Snap & Shape system, providing users with actionable insights into their health trends. By combining advanced time series modeling with the existing food analysis capabilities, the system offers a comprehensive approach to nutritional health management and predictive wellness analytics.

# **3.3 Backend System Architecture**

The backend architecture of the proposed system is designed to be modular, scalable, and maintainable, using modern development practices and a layered architecture approach. It serves as the bridge between the user-facing components and the machine-learning classification system.

## **3.3.1** Architectural Overview

The backend follows a multi-project structure within a Clean Architecture or N-tier pattern:

* **Snap.APIs**: The API layer that exposes RESTful endpoints for frontend or external system access.
* **Snap.Core**: Contains shared models, interfaces, Enums, and contracts.
* **Snap.Repository**: Responsible for database access using Entity Framework Core, implementing repository and unit-of-work patterns.
* **Snap.Service**: Contains business logic and service implementations, including integration with external AI models.

## **3.3.2** Main Responsibilities

* Accept input from users or client systems.
* Validate and preprocess this input.
* Convert relevant input into CSV using CsvHelper.
* Send data to external AI models via HttpClientFactory.
* Receive and return predictions from AI models.
* Persist or retrieve user data using Microsoft SQL Server.

## **3.3.3** Communication with AI Model

The system integrates with a Python-based AI model using HttpClient. It sends data as multipart/form-data (CSV file) and receives a JSON prediction response.

csharp

CopyEdit

var response = await client.PostAsync("predict/", content);

## **3.3.4** CQRS Implementation

Using MediatR, the backend separates commands (write operations) and queries (read operations) for better scalability and testability.

## **3.3.5** Security and Configuration

* API is protected with **JWT Bearer Authentication**.
* Configuration values like model URLs are injected via IConfiguration.

**Chapter 4**

**System Implementation and Results**

# **System Implementation and Results**

## **4.** Dataset

The Snap & Shape system utilizes three distinct datasets to support its comprehensive food analysis capabilities: food detection, freshness classification, and weight prediction functionality.

## **4.1.1** Food Detection Dataset (YOLOv8)

The food detection component employs a comprehensive dataset consisting of approximately 2,600 images across 45 different food categories. The dataset was compiled from multiple sources to ensure diversity and robustness:

• Primary Sources: Fruits and Vegetables Image Recognition Dataset

• Manual Collection: Additional images manually collected from Google Images to supplement underrepresented categories

The 45 food classes include: Basbousa, apple, banana, beef patty, bun, burger, cabbage, capsicum, carrot, cheese slice, chilli pepper, coffee, corn, cucumber, eggplant, eggs, falafel, fries, golash, grapes, kiwi, lemon, lettuce, macaroni, mango, onion, onion slice, orange, paprika, pear, peas, pickles, pineapple, pizza, pizza slice, pomegranate, potato, soft drink, sweetpotato, tomato, tomato slice, turnip, and watermelon.

The dataset provides comprehensive coverage of common foods encountered in daily meals, including fruits, vegetables, processed foods, and beverages. This diversity ensures the model can accurately detect and classify a wide range of food items in real-world scenarios.

## **4.1.2** Freshness Classification Dataset (CNN)

The freshness detection system utilizes a specialized dataset focused on food quality assessment, compiled from two primary Kaggle sources:

• Fresh and Stale Classification Dataset:

Fresh and Rotten Classification

• Fruits Fresh and Rotten Classification Dataset:

Fruits fresh and rotten for classification

The combined freshness dataset comprises approximately 14,000 images covering fresh and overripe states of various food items including apples, bananas, cucumbers, oranges, and potatoes. This substantial dataset size ensures robust training for accurate freshness classification across different food categories.

## **4.1.3** Weight Prediction Dataset (LSTM)

The weight prediction component utilizes time series data manually collected from various weight tracking datasets available across the internet. The dataset contains longitudinal weight and calorie intake information necessary for training the LSTM model to predict future weight trends based on eating patterns and historical weight data.

Advantages of the Multi-Dataset Approach:

The integration of multiple specialized datasets provides several key advantages:

1. Comprehensive Food Coverage: The 45-class food detection dataset ensures broad applicability across diverse cuisines and food types commonly encountered in daily meals.

2. Quality Assessment Capability: The dedicated freshness dataset enables the system to assess food quality, providing users with information about food spoilage and optimal consumption timing.

3. Predictive Health Analytics: The time series weight data enables longitudinal health tracking and predictive capabilities, extending the system beyond immediate food analysis.

4. Real-World Applicability: The diverse data sources ensure the system performs reliably under various photography conditions and food presentation styles encountered in practical use.

Limitations and Considerations:

While the datasets provide substantial coverage, certain limitations should be acknowledged:

• The manually collected food images may have varying quality and consistency compared to professionally curated datasets

• The 45 food categories, while comprehensive, may not cover all possible food items users might encounter

• Weight prediction accuracy depends on the consistency and quality of user-provided historical data

# **4.2 Experimental Setup and Results**

## **4.2.1** YOLOv8 Food Detection Model

Training Configuration:

• Model Architecture: YOLOv8 (You Only Look Once version 8)

• Number of Classes: 45 food categories

• Training Epochs: 150

• Image Resolution: 224 × 224 pixels

• Confidence Threshold: 0.3 for detection

• Batch Size: Optimized based on available GPU memory

Training Results:

The YOLOv8 model demonstrated excellent performance across all 45 food categories. The final training metrics at epoch 150/150 showed:

• Box Loss: 0.6993

• Classification Loss: 1.248

• DFL Loss: 0.915

• Overall mAP50: 94.7%

• mAP50-95: 69.1%

A screenshot of a graph

AI-generated content may be incorrect.

Per-Class Performance Analysis:

The model achieved consistently high performance across different food categories:

**• Best Performing Classes:**

o Macaroni (mAP50: 97.4%)

o Fries (mAP50: 99.2%)

o Pineapple (mAP50: 99.2%)

o Pizza (mAP50: 91.5%)

**• Moderate Performance Classes:**

o Fish (mAP50: 97.7%)

o Orange (mAP50: 98.1%)

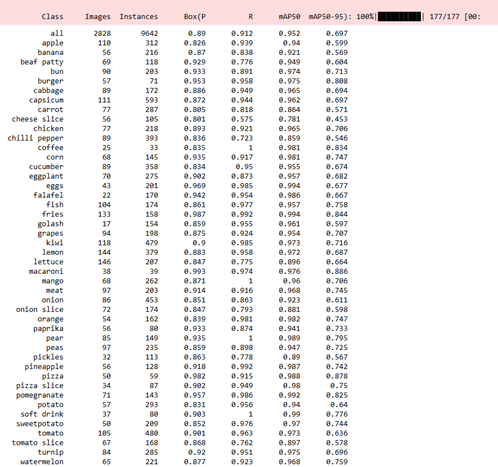
o Watermelon (mAP50: 92.3%)

• **Challenging Classes:**

o Cheese slice (mAP50: 57.5%)

o Pickles (mAP50: 77.8%)

o Onion slice (mAP50: 79.3%)



The comprehensive per-class results demonstrate the model's robust performance across diverse food categories, with an overall mAP50 of 95.2% across all classes.

## **4.2.2** CNN Freshness Classification Model

**Training Configuration:**

• Model Architecture: Convolutional Neural Network

• Input Image Size: 150 × 150 pixels

• Training Epochs: 20

• Optimizer: Adam

• Loss Function: Binary Cross-entropy

• Confidence Threshold: 0.85 for freshness classification

• Dataset Size: 14,000 images

**Training and Validation Performance:**

The CNN freshness classifier achieved exceptional performance:

**Final Epoch Results (20/20):**

• Training Accuracy: 98.22%

• Training Loss: 0.0505

• Validation Accuracy: 97.11%

• Validation Loss: 0.0764

A white background with black text

AI-generated content may be incorrect.

**Test Set Performance:**

• Final Test Accuracy: 97.11%

• Test Loss: 0.0782

A screenshot of a computer

AI-generated content may be incorrect.

**Training Progression Analysis:**

The model showed rapid convergence with stable performance:

• Initial epochs demonstrated quick learning with accuracy rising from ~71% to >90% within the first 5 epochs

• Training and validation curves showed good alignment, indicating minimal overfitting

• Loss values consistently decreased throughout training, reaching optimal levels by epoch 15

• The final epochs showed stable performance with minimal fluctuation

The training and validation accuracy curves demonstrate excellent model performance with both metrics converging around 97-98%, indicating robust generalization capabilities for freshness detection tasks.

## **4.2.3** LSTM Weight Prediction Model

**Training Configuration:**

• Model Architecture: LSTM + Linear Regression Hybrid

• LSTM Layers: 3 stacked layers (100, 200, 200 units)

• Input Sequence Length: 7 days (sliding window)

• Output Prediction: 7-day weight forecast

• Training Epochs: 100

• Optimizer: Adamax

• Loss Function: Mean Absolute Error (MAE)

• Dropout Rate: 0.2 for regularization

**Training Performance:**

The LSTM model training progressed through 100 epochs with the following key milestones:

Mid-Training Performance (Epochs 37-42):

• Epoch 37: Loss: 0.0143, Val\_Loss: 0.0151

• Epoch 38: Loss: 0.0139, Val\_Loss: 0.0120

• Epoch 39: Loss: 0.0140, Val\_Loss: 0.0162

• Epoch 40: Loss: 0.0140, Val\_Loss: 0.0163

• Epoch 41: Loss: 0.0137, Val\_Loss: 0.0200

• Epoch 42: Loss: 0.0136, Val\_Loss: 0.0510

A number of steps with numbers and numbers

AI-generated content may be incorrect.

**Training Characteristics:**

• Consistent loss reduction throughout training

• Low training and validation loss values indicating good model fit

• Occasional validation loss spikes (e.g., epoch 42) suggesting the need for regularization

• Overall stable convergence with MAE values consistently below 0.0

A graph with orange and blue lines

AI-generated content may be incorrect.

**Prediction Performance:**

The weight prediction system demonstrated accurate forecasting capabilities. The actual vs. predicted weight comparison for a test sample showed:

• **Prediction Accuracy:** Close alignment between actual and predicted values

• **Weight Range:** Stable predictions around 84-85 kg range

• **Temporal Consistency:** Smooth prediction curves without erratic fluctuations

• **Practical Utility:** 7-day forecasts suitable for short-term health planning

A graph with a red line and blue line

AI-generated content may be incorrect.

The prediction graph demonstrates the model's ability to capture weight trends effectively, with predicted values closely following actual weight measurements over the 7-day prediction window.

## **4.3** System Integration and Performance

The complete Snap & Shape system successfully integrates three specialized models to provide comprehensive food analysis:

1. **Real-time Food Detection**: YOLOv8 achieves 95.2% mAP50 across 45 food categories

2. **Quality Assessment:** CNN classifier provides 97.11% accuracy in freshness detection

3. **Health Prediction:** LSTM model enables accurate 7-day weight forecasting

**Overall System Performance:**

• **Detection Speed:** Real-time processing capabilities suitable for mobile applications

• **Classification Accuracy:** Consistently high performance across all three model components

• **User Experience:** Seamless integration of multiple AI capabilities in a single system

• **Practical Application:** Robust performance under varied real-world conditions

## **4.4** Conclusion

The experimental results demonstrate the successful implementation of the Snap & Shape system with high-performance AI models across all three core functionalities. The YOLOv8 food detection model achieved excellent classification performance with 95.2% mAP50, while the CNN freshness classifier reached 97.11% accuracy. The LSTM weight prediction model showed stable convergence and accurate forecasting capabilities.

The comprehensive evaluation validates the system's potential for real-world deployment in health and nutrition applications, providing users with accurate food analysis, quality assessment, and predictive health insights through a single integrated platform.

**Chapter 5**

**Tools**

# **Tools**

## **5.1** Mobile App development ( using Flutter cross-platform ):

Flutter using Dart: Is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase. Flutter apps are written in the Dart language and make use of many of the language's more advanced features. On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

**A person standing next to a large cellphone

AI-generated content may be incorrect.**

**What is Dart?**

Is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, mixins, abstract classes, reified generics, and type inference.

**Coding in Dart look like:**

**Like most ALGOL languages (like C# or Java):**

* The entry point of a Dart class is the main () method. This method acts as a starting point for Flutter apps as well.
* The default value of most data types is null. Dart classes only support single inheritance.
* There can be only one superclass for a particular class, but it can have many implementations of Interfaces.
* The flow control of certain statements, like if conditions, loops (for, while and do-while), switch-case, break and continue statements are the same.
* Abstraction works in a similar manner, allowing abstract classes and interfaces.

**A black letter with a white background

AI-generated content may be incorrect.**

**Unlike them (and sometimes a bit like JavaScript):**

* Dart has type inference. The data type of a variable need not be explicitly declared, as Dart will “infer” what it is. In Java, a variable needs to have its type explicitly given during declaration.
* For example, String something; But in Dart, the keyword is used instead like so, var something; The code treats the variable according to whatever it contains, be it a number, string, bool or object.
* All data types are objects, including numbers. So, if left uninitialized, their default value is not a 0 but is instead null.
* A return type of a method is not required in the method signature.
* The type num declares any numeric element, both real and integer.
* The super () method call is only at the end of a subclass’s constructor.
* The new keyword used before the constructor for object creation is optional.
* Method signatures can include a default value to the parameters passed. So, if one is not included in the method call, the method uses the default values instead.
* It has a new inbuilt data type called Runes, that deals with UTF-32 code points in a string. For a simple example, see emojis and similar icons.

**Using Dart in Flutter:**

Flutter has more app-specific libraries, and more often on user interface elements like:

* Widget: common app elements, like the Text or List View.
* Material: containing elements following Material design, like Floating Action Button.
* Cupertino: containing elements following current iOS designs, like Cupertino Bu

## **5.2** Back-End Development:

The back end of a software application refers to the server-side component that manages data storage, processing, business logic, and communication with the front end. It is responsible for maintaining the application's integrity, security, and performance. In this project, the back end is developed using **.NET Core** and **Microsoft SQL Server**, offering a scalable, secure, and efficient solution for modern web and mobile applications

**.NET Core Framework:**

**A purple circle with white text

AI-generated content may be incorrect.**.NET Core is a cross-platform, open-source framework developed by Microsoft for building high-performance, cloud-based, and modern web applications. It supports multiple programming languages, with C# being the most used for backend development.

* **Features and Benefits of .NET Core:**

1. **Cross-Platform Compatibility**: Runs on Windows, Linux, and macOS, allowing flexible deployment across various environments.
2. **High Performance**: Offers just-in-time compilation and optimized memory usage for fast and efficient application performance.
3. **Modular Architecture**: Applications can include only the components they need, reducing size and improving maintainability.
4. **Built-in Dependency Injection**: Promotes modular, testable, and maintainable code.
5. **Secure Framework**: Includes tools for authentication, authorization, and protection against common web vulnerabilities.
6. **API and MVC Support**: Supports RESTful API development and MVC pattern for clean code structure.
7. **Rich Development Tools**: Integrates with Visual Studio and Visual Studio Code, offering powerful debugging and deployment features.
8. **Community and Ecosystem**: Backed by a large community and supported by Microsoft with frequent updates and improvements.

* **Why Use .NET Core?**
* Modern, clean, and fast development platform.
* Ideal for building microservices, web APIs, and full-stack applications.
* Excellent performance and security out of the box.
* Seamless integration with Microsoft Azure and other services.

**Microsoft SQL Server:**

Microsoft SQL Server is a powerful, enterprise-level relational database management system (RDBMS) designed for secure, high-performance data storage and retrieval. It integrates seamlessly with .NET Core applications and is managed through **SQL Server Management Studio**

**A logo of a server

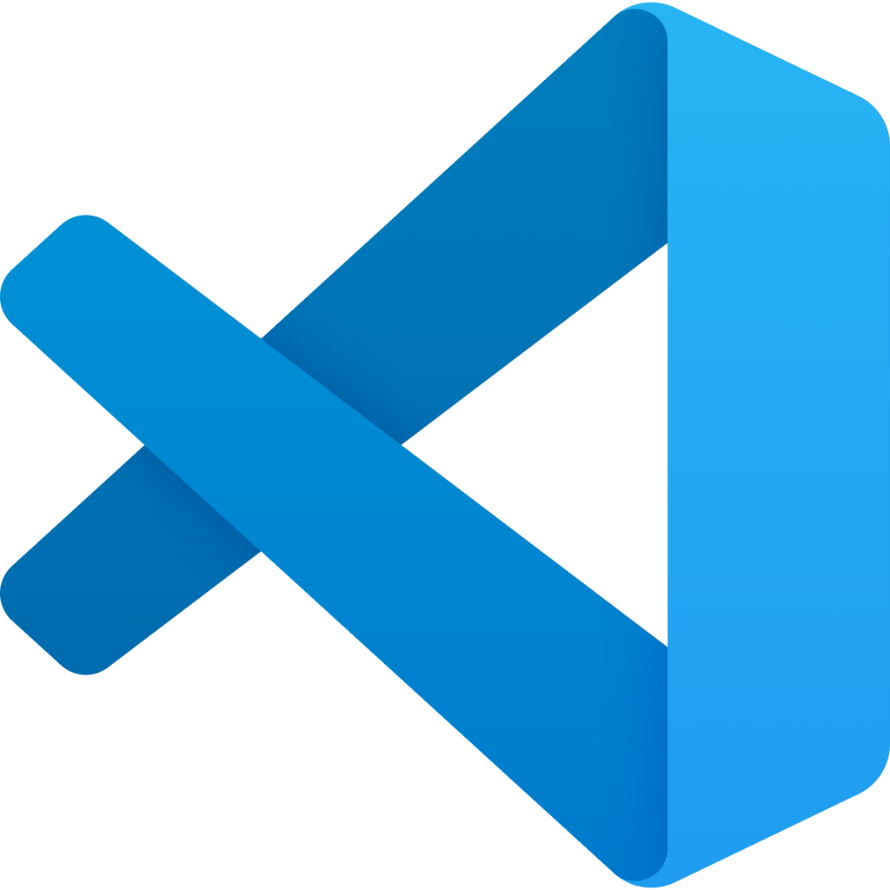
AI-generated content may be incorrect.**

* **Features and Benefits of Microsoft SQL Server:**

1. **Relational Database Model**: Organizes data into related tables with foreign keys, constraints, and indexing for optimal structure and speed.
2. **Stored Procedures and Views**: Allows developers to encapsulate logic within the database, improving performance and reusability.
3. **Security and Compliance**: Supports role-based access control, transparent data encryption, and advanced auditing features.
4. **Scalability and Reliability**: Handles large volumes of data and supports automatic backups, failover clustering, and replication.
5. **ACID Compliance**: Ensures that all transactions are processed reliably and securely.
6. **Powerful Query Tools**: SSMS provides tools for query execution, performance monitoring, and database design.
7. **Integration with .NET Core**: Supports Entity Framework Core and ADO.NET for seamless data operations in .NET Core applications.

## **5.3** Main Work

**Visual Studio Code:** Is a dual-licensed source-code editor made by Microsoft for Windows, Linux and macOS. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it. Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

****

**Visual Studio:** is a powerful and fully featured integrated development environment (IDE) developed by Microsoft, primarily for building applications on the .NET platform. It supports development in multiple programming languages including C#, VB.NET, F#, C++, Python, JavaScript, and more. Visual Studio is available for Windows and macOS and is widely used by individual developers and large teams in enterprise environments.

In contrast to lightweight editors, Visual Studio provides a comprehensive suite of tools for building, debugging, testing, and deploying software. It supports the development of desktop applications, web applications, mobile apps, cloud services, and enterprise-level systems.



**Postman:** is a popular collaboration platform and API testing tool used by developers to design, test, and manage APIs. It provides an easy-to-use graphical interface for sending HTTP requests to RESTful APIs, viewing responses, and automating workflows, making it an essential tool for backend and frontend developers working with web services.

Postman supports all standard HTTP methods such as GET, POST, PUT, PATCH, and DELETE, and allows users to add headers, query parameters, request bodies, and authentication tokens. It is widely used during the development and debugging phase of API-driven applications.



**SQL Server Management Studio (SSMS):** is an **integrated environment** developed by Microsoft for managing, configuring, and administering all components of **Microsoft SQL Server**. It provides a comprehensive graphical interface and script editor for database developers and administrators to interact with SQL Server instances, write and execute T-SQL queries, manage databases, and monitor server performance.

SSMS is widely used in enterprise environments for managing relational databases, performing maintenance tasks, and handling business-critical data operations. It enables users to perform **tasks such as database creation, table design, stored procedure writing, data backup and restore, and performance tuning** all within a single, user-friendly interface.



**Jupyter Notebook:** is an open-source web-based interactive computing environment that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. Originally developed for data science and scientific computing, Jupyter has become a widely adopted tool across education, research, and industry for its flexibility and ease of use.

Jupyter supports numerous programming languages through the use of kernels, with Python being the most popular. It provides an intuitive interface for exploratory data analysis, machine learning, and computational research. Users can write and execute code in a step-by-step manner, visualize data using libraries like Matplotlib or Seaborn, and document their processes all in one place.

Jupyter Notebooks are especially powerful in collaborative environments, enabling reproducibility and transparency in data workflows. They are also commonly used in combination with tools such as Pandas, NumPy, and TensorFlow, making them a staple in the modern data science ecosystem.

![A logo with orange circles and black text

AI-generated content may be incorrect.](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAlgCWAAD/4SCMRXhpZgAATU0AKgAAAAgABgALAAIAAAAmAAAIYgESAAMAAAABAAEAAAExAAIAAAAmAAAIiAEyAAIAAAAUAAAIrodpAAQAAAABAAAIwuocAAcAAAgMAAAAVgAAEUYc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFdpbmRvd3MgUGhvdG8gRWRpdG9yIDEwLjAuMTAwMTEuMTYzODQAV2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NAAyMDIwOjA0OjA5IDE0OjUzOjA1AAAGkAMAAgAAABQAABEckAQAAgAAABQAABEwkpEAAgAAAAMwMAAAkpIAAgAAAAMwMAAAoAEAAwAAAAEAAQAA6hwABwAACAwAAAkQAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMDowNDowOSAxMzoxNzoxMgAyMDIwOjA0OjA5IDEzOjE3OjEyAAAAAAYBAwADAAAAAQAGAAABGgAFAAAAAQAAEZQBGwAFAAAAAQAAEZwBKAADAAAAAQACAAACAQAEAAAAAQAAEaQCAgAEAAAAAQAADuAAAAAAAAAAYAAAAAEAAABgAAAAAf/Y/9sAQwAIBgYHBgUIBwcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQy/9sAQwEJCQkMCwwYDQ0YMiEcITIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy/8AAEQgAkAEAAwEhAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A9/ooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKRmCqWYgAdSTQA2KaKZd0UqSAd0YGn0AFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAV5Z8RdYuZdYOmK7JbQqpZQeHYjOT+YoA5nRtXudF1GO6t3YYI3oDw69wa94ByAfWgBaKACigAooAKKACigAooAKKACigAooAKKAPMvFnxFv9O1qbT9MjhVLdtrySLuLN3HXgV1ng/wASf8JNpDXMkQinify5VX7ucA5HtzQB0BIAyTgCuf1Pxhp9gxjiJuZh1CfdH1Nc+JxMKEeaR04bCzxE+WJzF3411SckQGO3X/ZXcfzNZj69q0hy1/P+DYr5+rmFab0dkfR0csoU1qrsdH4g1aI5W/m/E5/nWrZ+ONRhwLlI7he5xtb9OP0p0cxrU3rqhV8roVF7qszqdL8UafqZCBzDMf8AlnJxn6HvWZ4u8HHXpUvLSVI7pV2sH+647c+tfQ0K8K0OaJ83iMPOhPkmYuh/Dq5iv47jVJIfKjYN5UZLF8evHSvSK2MAooAKKACigAooAKKACigAooAKKACigAooA4LxN8OF1rVX1CzvFt3mOZUdMgn1GP5Vv6Ho9j4P0NovOyoJkmmcY3N06fgBilKSirscYuTSRyOveKLnVHaGAtDaZ4UHl/r/AIVz9fJ4rESr1HI+yweGjh6Sj16ksVtPP/qYZJP9xSatLoupsMiwuP8Av2azjh6s9YxNJ4mjB2lJIRtG1NBlrC4/79k1UkhlhbbLG6H0ZSKJ0KkPiQQxFKppGVxnQ8V1fh/xbLaOlrfsZLfosh5ZPr6itsHiXQqX6Pcwx2FWIpNLdbHfo6yIrowZWGQQeCKdX1Sd1c+PatoFFMAooAKKACigAooAKKACigAooAKKACigAJwMmvOfEWqXGval9iskeSGNsKqDO8+v0rzsyqNU1TjvI9LK6cXVdSe0dS3pvgWR9r6hP5Y7xx8n866e08O6VZgeXZxsw/icbj+tThMuhTXNU1ZeMzOdV8tPSJpqqqMKoAHQAUtekkkeVcKY8UcqlZEVlPUMM0NJ7jTa1RlXfhfSbwHdarGx/ii+U/4Vy2peCLu3DSWUguEHOw8N/ga8rGZdGa5qWjPXwWZyg+SrqibwjrUtpc/2VeblUnEe/go392u7roy+q50bS3WhzZlSUK7cdpahRXccAUUAFFABRQAUUAFFABRQAUUAFFABRQBQ1aK4ubM2ts2x5vkaT+4vc/59aNL0i00m3EVvGN38Tn7zH3NYeyUqvtH02N1VcaXs49dy/RW5gFFABRQAUUAZWsaDbarHuI8q5XlJl6g/1q7ZNObSMXIAmUbXx0JHf8awjSUKrkuv5m8qrnSUJdNvQsUVuYBRQAUUAFFABRQAUUAFFABRQAUUAFFAEC3lq121qtzCblV3NCHG8D1I64py3Vu9w9us8TToMtGHBZR7jqKAEmvLa3liimuIo5JTiNHcAufYHrTnuIYpY4pJo0kkz5aMwBbHXA70AVU1rSpL37GmpWbXQOPJE6l8+mM5qxNd21s8aT3EUTSnbGruFLn0Gep5FAEruqIzuwVVGSScACsjWPEFpp2g3Gow3NtLtjZoQZRtkYdgc8/hQBF4Y8Qw67o1ncST2ovZoy7wRuMrye2c1p3mpWOnKrX17b2yscAzSqgP5mgCaGeK5iWWCVJY2GVdGDA/QioJNU0+GKaSW+tkjhOJWaVQEPo3PB+tAFlHWSNXRgyMMqwOQRTqACigAooAKKACigAooAKKACigAooAKKAOB0//AJLNqf8A14j/ANlo0X/ksHiH/r2j/wDQUoAPHP8AyOfg/wD6+H/mlRfEa2kvPEPhW2ineBpp5YjIhwyq2wNj3wTQBf17wJoX/CN3ItLGO2uLeIyRToSHDKM5J75x3rFukufFfwjt76Qs+oWQMqyD7xMZIJ+u39aALniHxM+o/Deze1bN5q2y1AXruJw/8iPxrR8QeHtPtPh3JaPbRyGwtD5TMvKvjlh7k80AN8A6Pptt4W03VI7KFb1rcl5wvzNyc8/hXI6F4g0m7ub7Vdc0q/1K7nmYRstt5scUY6KuTgUAbngq9QeL9QttNsby10e4hEyxXERQRyggHb6A5NUvD2g2ev8AjTxJ/aKGa1t7rcLckhWc5G4464AP50AeoRxpFEkcahUQBVA7AU6gAooAKKACigAooAKKACigAooAKKACigDgNOO74zapjnFiAfb7lGtWGtaD44fxHpOntqNvdwiO4gRsMCABx+Q/WgCrfWfiXX/EuhatdaSbS1trjiDeGeNeCXY8dfT2qb4jrdv4h8KCwZFu/tEhhL/d3DYQD7UATanrHivWNPk0m38NS2lxOpiluZZgY0B4JHrXVaBo0WhaDa6YjeYsKYZiPvE8k/maAPO/CWiNJ45urMOJdL0W5klgA5AkfAA/DBP1Fej65YPqehX1jGwV54WRSemSOKAOa8DT6xa2EGhalok1sltGy/ai4KNzwMfjWfp0Gv8AgW7u7O20mTVdImlMsLQOA8ZPYg/T+tAHWaFq2o6qbhr3RZtNjTb5XmyBmkznPA6Y4/OsjwjpN9p/iPxJc3Vu0cN1cK8LEj5wN3P6igDsKKACigAooAKKACigAooAKKACigAooAKq6kl7Lp8yafNHDdlf3Uki7lU+4oAxfDfhwaFPc3l9em81W+bMs7LtBx0VR6f4e1dJRcLBXPa94em1fXNCv4544006ZpHVgcuDt6f980AdDWT4itNWvdL+z6PeJaXDuA0rrnCd8e9AB4d0C28OaUtnbkuxO+aZvvSuerGtagAooAKajrICUYEA4yKLgOooAKKACigAooAKKACigAooAKKACigAooAparZNfWEkUbmOYfNE4OCrDoawdI8Wr5n2LVh5Fwh2mQ9CR6+hrir1vYVYt/DLT5ndQoe3oyUfijr8jqUdZEDowZT0IOQadXYmnqjhatowopgFFABUc08VvGZJpFjQdWY4FJtJXY0m3ZHJ3/iaXU7tdM0bO6Q7Tceg7kf411VpbJaWscCZKouMnqfeuPDVfbzlNbLRHbiqPsIRpv4nq/0JqK7ThCigAooAKKACigAooAKKACigAooAKKACuQ8W+HDdA6hZpmYD96g/iHr9a5MdR9rRaW6OzAV/Y11J7PRnI2GsX+mN/o1wyrnlDyv5V0ln49dQFvLQN/txHH6H/GvFwuYTo+7LVHu4vLYV/fhozYh8Z6RIPmkkjPoyf4VaXxRozD/j+QfUEV7EMwoTW54lTLcRB/DcRvFOjL/y+ofoCaqT+NdJiH7syyn0VP8AGlUzGhBb3HTy3ETe1jFvPHly+Vs7ZIh/ekO4/lXOXV/farOvnzSTOThU7Z9hXj4nHVMQ+WOiPcwmX08MueWrO/8AC/h8aVb+fcKDdyDn/YHpXQ172Fo+xpKB87i6/tq0phRXQcwUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAcvr3hGK/ZrmyKw3B5Zf4X/wNcHd2N1YTGK6heNh6jg/Q96+dzHCOnL2kdmfTZZjVUh7KfxIr0V5Z64UUAXdP0m91OUJawsw7ueFH1NegaD4Yt9JAmkImuj/GRwv0r18twjlL2s9uh4maY1Rj7GD1e5v0V7586FFABRQAUUAFFABRQAUUAFFABRQAUUAFFAFDWNTTStPe4ZdzZ2ovqa4ObxJq003mfa3TnhUwAP8APvQB03hrxFJqMhtLvHngZRwMbh3/ABroJ7aC6jMc8SSIeodQRSlFSVmOMnF3RgXXgnS52LReZAT2Rsj9azn8ALn93fkD/ajz/WvKq5VCTvB2PXo5xVgrTVxY/AEYP72/Yj/Zjx/WtSz8HaVbMGeNp2H/AD0bj8qqjldODvJ3Jr5vVmrRVjdjijhQJGioo6BRgU+vTSSVkeU227sKKYgooAKKACigAooAKKACigAooAKKACigAooA5zxnbyTaSkiAkRSbmA9MYzXA0Ab3hG2km1tJVB2QqWZu3IwBXodABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAIyq6lWAZSMEEdawpvCGlzTGQLLGCclEfj9RQBrWdjbafB5NtEI0746n6mrFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQB//9n/4THkaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdG9yVG9vbD5XaW5kb3dzIFBob3RvIEVkaXRvciAxMC4wLjEwMDExLjE2Mzg0PC94bXA6Q3JlYXRvclRvb2w+PHhtcDpDcmVhdGVEYXRlPjIwMjAtMDQtMDlUMTM6MTc6MTI8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAMCAgMCAgMDAwMEAwMEBQgFBQQEBQoHBwYIDAoMDAsKCwsNDhIQDQ4RDgsLEBYQERMUFRUVDA8XGBYUGBIUFRT/2wBDAQMEBAUEBQkFBQkUDQsNFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBT/wAARCALsBTIDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9U6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAprNtzk4oJ5rzD44fHDTPg9oayyKL3WbhSbSw7n/bfuFH5ntQB3XiXxTpHhHTX1DWdRt9Ms06zXD7R+HqfpXz94u/bf8ADmlySQaBpV1rbqSPPmYQRfUdSR+Ar5M8cfELXviPq76lr1691M2RHFn93Cv91F6AVzuT3OaoD6RvP25vFskh+zaJpESdgwkY/nuq3pP7dWvxP/xMfDmnXSDr9nleI/mdwr5ipaAPvrwH+1x4H8YzR215NN4dvX4C6hgRMfQSDj8wK9tguEuI0kjkWSNwGV1OQwPcHvX5L1678E/2idc+FN9Fa3EkuqeHHbEtixJaLuWiJ6HnOOh+tKwH6IA0tYnhTxNpvi3QrXV9JuVu7G6TfG69fcH0I7itpTuXOcikAtFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGD408WWfgfwvqeu6g221sYTK2OrH+FR7k4A+tfmf488baj8Q/FV7r2qPvurh8qgOViUH5UX2H/ANevqT9uTxk9npOheGYZCgu5GvLgA4yqcKD+JJ/CvjymAmKKKKoAooooAKKKKAPev2T/AIvSeCfGEfh/UJyND1aTYu5uILg42uPQN90/UV95JwuBX5KxyPDIskbFJEIZWHYjOD+dfp58JfFx8bfDfw/rTHMtzaL5vP8Ay0X5XH/fQNSB2FFIKWkAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUnNJmgB1FNzRzRuA6im5pKAH0UylxQA6im5xQWoAdRTVbNGfegB1FN3U6gAooooAKKKKACiiigAooooAKKKSgD4N/bSvmuPjCLYnKW+mwgD6ljXglfQf7bOlNZ/Fe0vSMJeadHtPqUZga+fKoAooopgFFFFABRRRQAvTn8a+9/2Nbo3XwVgiJOLe+niX6fK3/sxr4H4+nv+f+fwr9Av2QdLk034J6c8i7Dd3M9wPcbtgP47M/jSYHta9KWkpakAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoopm7/PpQA+krnfGnxC8PfDzSm1HxFq1vpVqPumZvmc+iqOWPsBXzf4s/wCCg/hnT7h4dB8O6hq4GcT3Li2Q4OMgYYn8cUAfWW6ivi2x/wCCiZ87/S/BRMOefIvfmA/Fef0r2b4Z/tdfD/4j3EVmuoPoepyHatpqgEe8k9FkztJ9iQfagD22io/OHBHzAjIwOtPU5XrmgBaKKKACiiigAooooAKKKKACiiigAoppzRuNAC0bqYzUwyAZ54HU0m7agTbqM1yviD4keHfDOf7R1e2gYD/Vbgzn6Ac/pXmXiH9q7QrHKaXY3Gov0DyYiQ/nz+leViM1wWFX72okerhsrxuLdqNJs91JGeoprSBepA+vFfJGt/tS+KdQ3JYW9rpqf7pkb9T/AErhtW+K3i7WyftWvXm0/wAML+WP/HcV81iOL8DT0ppyZ9XhuCsxra1GoLzPua61iysYy9xdwxIP4ncAfzrnr74reELBWM+vWS7eoWUMf0r4VuL65u3Lz3EsznqzuSTUNeLU4zl/y7pfee9R4EX/AC9rfcj7Uuf2gvBFuDt1lZsdo4nP9Ky5v2nPBsOQJbyTH923P9a+Pv1o/CvOlxfjHtFHpx4HwK+Kcvw/yPrhv2qPCKkgRX5/7YUL+1R4RPWK/H/bA/418kUmKy/1ux/l9xr/AKk5f3l959gw/tN+DZcbpbuLP/PS3bFa1r+0F4GuAP8AidLET2kjYf0r4owPSlrSPGGNjvFMxnwPgZfDOS/r0PvOz+KfhO82+Vr9jlugeYKT+db9tq9neqGguoZgf7jA/wAq/O31xxn04qe3v7m0YNBcSwsO8blf5V6NLjOX/L2l9zPNrcBpL9zW+9H6KCZezCnqwIznivg/R/i14v0Mr9l166Kr/BKwkH/jwNdzov7UviexwL+1tdRTu20xt+fSvcw/F2CqaVE4nz+I4LzGjrTtL0PrfcKXdXhfh/8Aas8P32xdTsrnTnPVlAlQflz+lem6D8RPD3iaJX07Vbe4LdEDgN+INfS4fNMHil+6qJny2KyvG4P+NSaOn3UVCsm5Qc5p4fHWvT5keVqSUU3d7UtUIWiiigYUlLRQB83/ALa3gSTXPAtl4itk3zaPMfPAH/LB+Cf+Atg/ia+Ivav1f1fSrXWtOu7C8hE9pcxGGWNujKQQR+tfm/8AGr4S33wj8YT6dKrS6ZOfMsLvGBLGT0z/AHl4BHfk07gef0UfT+WKKoAooooAKKWj68AepxQBb0bSLnX9YstNs0827upkgiQDOWY4H9a/UfwV4ag8H+EdI0S3H7qxtkgHvtGCfxOTXy/+x78EZluE8dazbmJFUppcMq4LZxmYj9B+J9K+uVGFAqWA6iiikAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV5j8ePjRpvwT8HSardKLrUrg+VYWIYAzS4yM+ijqT9PWvTGzng1+YP7V3xKk+JHxh1YrLnTdLc6dZp6BD+8YfVgf09KAPP/HnxC174ma/NrHiC/a+upCSq5/dxKf4UXsK5z1GMA9aPf15ooAPfvR+tFFAH1R+yv+1ReeGdTsvCXjC9N3oU7CK0vrglns3Iwqsx6oTxk9OO1ffEcgZRjkY6+vvX4v8A6ep79/8AGv0z/ZC+JU3xG+Dtj9tk83UtJc6fcMzZZgoBRj9VIH4UAe4UUlLQAUUUUAFFFFABTWbFGeaa3SkA4NnFBbjpTC22uR8afFLw94Gt2fUr6MTdVt4zukb6Af1rGrXp0I89SVkbUqNSvJQpRbfkdd5nfpWL4i8baN4UtzNqmoQ2iejsNx+g6mvmjxx+07ret77fQ4V0m1bgSNhpmH48D9a8d1DUbrVrlri9uJLudjlnmcuf1r4TMOLaFF8mFXM+5+g5dwZicRaeLlyR7bs+kfF37V1lbtJb+H9Pe7botxcfIn12jJP6V434m+MnizxWxFzq0kMB/wCWNp+7Ue3HP5muK+vPbHaivz/GZ7jsY2pzsuyP0nAcOZdgUnCnzPu9RzyNIxZ2LMeSzHJ/Om8+tFFeDKUpO8nc+mjFRVoqyCiiipux2QUUUUrIYUUUUAFFFFABRRRTuwCiiijQNROvXmjA9KWikMP1pY5GhkDxMYnHRkO0j8RSUValKLvF2IlGMlyyV0d14Z+Nni7wuwEGqSXcIGPIu8yr+vP5GvZfCP7VmnXflw6/YPYOeDcQnzI/qe4/Wvl/rS19Bg8/x2D0jO67M+Yx3DWXY5XdPlfdaH6CaD4w0fxNarPpl/BdoRn5HGR9RWusnt+tfndpurXujXAnsLqaynXkSQOVP6V7N4H/AGoNW0cpb6/bjU7cYBniwsij37H9K+/y/i3D12oYhcr79D80zLg3FYa88K+ddup9Wh6XdXJeD/iVoPjaFX0y+R5MfNbv8si/ga6oN6191Sr060OenK67nwFWjUoT9nVi4vsyRTmlqNc470+tzIRlDVy3xC+HOi/Erw/No+t2yzwN80UgAEkD9nQ9iK6uk2j0oA/Or4tfs4+J/hfcz3IgfWNDByNQtlzsGf8Aloo5X69OK8n6cHj8x+uK/Wto1ZSCuQRg15p4x/Zx8AeNJpZ7vQY7W6kyWuLBjAxJ7kD5SfqDTuB+b9FfbN1+wz4Skcm31nVIEzwrbH/XFX9J/Yn8C2MokvbrVNR/6ZtMsan8AM/rTuB8P2NjcapdRWtnBJd3MhwkMKF2Y+gA5P8AnmvqP4F/sh3M11b6345iMEMZEkOkA/M5zwZTngf7PWvpzwf8M/C/gKHZoOi2unkjDSqu6Vvq7ZY/nXTbQOgpARW1tHawpFEixxIAqoq4VQOgA7AVKKKWkAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUtYumsdKvbhBl4YHkH1Ckivxx1GZ7jULuV2LPJM7MxPJJJzX7LXEKzxvG4ykilCPYjmvyB+IXhufwb4417RblCktleywjPQgOdp/EEGgDn6KPp0ooAKKKKACvs/8A4J13km7xtZ/8scWs2O27Mg/kBXxh3AxmvvL/AIJ8+FZNN8B6/rsq4Gp3qwwn+8kSnn/vpz+VAH1hS0lLQAUUm73o3CgBaQ0m6o551hRmdwiAZLMelJtRV2NauwrN19awvFXjLSvBti13qt7HbRKOFblm9lHc15X8Tf2krHw/5tj4e26jfg7XuM5iiOD3H3j7CvmfxB4j1LxRqD3uqXcl5cMfvOeF9lHYV8PmvE1HBt0qHvS/A+7ybhTE5harWXJD8WeufET9pjU9dM1n4fQ6XZ9PtD4aZvw6LXi1zdT3tw89zK887nLySMWYn6moqK/KsbmWJx8+etL5dD9ky/KcJlsFChC3n1+8KKKK8o9m/YKKKKBBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRRYAooooGTWd9cadcJcWs8ltMhyJImKkflXt/w8/ac1DSTHaeJI/t1r0F1GMSqPUj+L/PWvCqPfvXqYLMsTgJqdGVvI8TMMowmZQ5a8Ne/X7z9AfDfi7S/Flit3pV5FdRsOkbZI+o6g1tqx78Gvz58N+KtU8J6gt5pV5Jazg5IU/K/sR0NfTnwv/aK07xQYdP1nbpupMcLIW/dyn2J6H2r9ZyniWhjkqVd8sz8azjhXFZc3Vo+9D8V6nto6UtRRyhlDBsqeQRTt3vX2qaex8MPpMD0pNw9aWmAbR2GKMUtFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADWUntXxX+3R8Dp5LxfiHo1sZFKJDq0cY5GOI5jjt0Un2B96+1qq3tlDfQTQ3EaTW8qFHjkAKupGCD7EUAfjN+OffOaK+yvjl+w1O13cax8PGjaJyXfRJ2xsJ/55OeCP9kkY9SOK+UvEvgPxH4Ount9b0O/0yVCQftEDBeP9rGD+FAGFR9OT6VLb2dxeSrHBBLPI3RIkLMfYAV698Nv2UfiB8RrmEnSpNB01jl77U0aMAf7KEbmPsB+NAHAfDvwBqvxN8W2Hh/RoTLd3TgNJj5YYx96Rz2VRn69K/VzwB4LsPh94O0rw9pq4tLCBYlbu56s59ycn8a5X4L/AAH8PfBPRDbaShudRmC/a9SmAEs5HYf3V/2R6dzzXpi/doAKWk9aTdQAN60zeAeelK7478Vw/wASvilpnw50zzbmTzbyQfuLRPvuf6D3NctevTw0HVqOyRvRoVMTUVOkrtm/4m8V6Z4S02W/1O6S2t4xnLdT7AdzXyj8Uvj1qvjqSWysC+m6PkjYhw8w/wBo/wBK43xx4+1fx9qr3upT5TP7qBT8kS+gH9a5vA9K/Is64kqYu9LDO0PzP2vIeFKWESr4tc0+nZC/UdsdOcen60UUV8Jvq9z9FUUtAooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKT/AD0paKabi+ZbiaurM9c+Ffx+1Hwa0Vjq5k1DSM4DscyRD2Pce1fVnh/xFp/iXTor/TblLm3kGQyHOPY+hr89xjr36dK6vwD8SdX+HepC5sJWe2Y/vrVz8kg/ofevvMl4kqYVqjiNYH5xn3CtPFRliMJpPqujPvBWLdOPrT171xnw6+JWl/ELSlubGUC4XiW3b76H6dx712CNkZzX6/Rr068FUpu6Z+LVqNShN06is0S0U1W7Z5p1bmAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUnGfelooAaV3dTTJbWKddssayL6OoIqWigClb6NYWrZgsreE+scKr/IVaMYbryPTtT6KAECgUtFMyaVwFNMZ9vNKW615b8Z/jJB8O7A2lsUudZmUmOLPEY/vt7e1cmKxVLB0nWquyR14TC1cbWVCirtkvxc+M1j8O7IwwMl1rMo/d2+eE/wBp/QV8f69r194o1Oa/1K4a5uZjlnbsPQDsB7VDqmqXWtahPe307XF1O295G7k8/h9Kq1+HZxnVXM6lr2gtl+p+/wCQ5DQyqlzNXqPd/oLRRRXzWh9cFFFFIAooopgFFFFA7BRRRQK66hRR+tOSN5yVjQs3ovJq405T+FXIlOMfidhtFa9l4P1zUcG20i9mH+zC3+FdDp/wT8a6kuYtBuI/+uxCfzrrp4HE1fgpt/I4KuZYOj/EqxXzOHor1Sz/AGafG1yuZLS3tf8ArpMp/kTWra/sqeKJgDNfafD/ALrO3/std8MjzCe1Jnm1OIsrp71k/TU8Wor3yP8AZI1RseZr1snrtgY/+zVbj/ZFnON/iJB/u2p/+KrpXDeZv/l2ckuK8pj/AMvfwZ88UV9G/wDDIh/6GL/yV/8AsqG/ZFPbxF/5Lf8A2VX/AKs5n/z7I/1tyn/n5+DPnKivoeT9kW4/5Z+IkP8AvWp/+KqpL+yVqi52a5bOewMBX/2Y1EuG8yj/AMuy48V5VL/l7+DPA6K9puf2VfFEfEN9p0p9C7qf/Qaybv8AZr8bW33LS3uT/wBMp1H/AKFiuWeR5jT3os66fEWV1Nqy+en5nllFdtqHwX8Z6apMugXL/wDXEK//AKCTXPX3hTWtN/4+tJvID/tQN/h/WuGeBxVP46bXyPTpZlg638OrF/MyqKHDRttdSjf3WHNHHHr6da45RlH4lY7o1Iy+F3CiiioL07hRRRT8kPVbhRRRQIKKKKQBR+JB9c0UUB6Gr4Z8Ual4R1aLUtLuGguoueOjD0I7g96+wvhT8X9P+I2m7QVttViUedan19V9RXxRV7RdbvfD+pw3+n3DW13CQySKcYyeh9Qa+oyXOquWVFGTvB9D4/PuH6ObU3OCtUXXv5M/Q1PWpK81+EfxetPiNpIjYrbaxAo863Y9f9pfUV6P5hC89a/ccNiaWKpKrSd0z8AxGGq4Sq6NVWkiSimg06uo5gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKRqADFNbg0bjXK/ETx3ZfD/AMPzanePlh8sMX8Uj9gKwrVoYem6lR2SNaVKdeoqVNXb2ML4vfFa2+G+kkKRNqlwCLeDPP8AvH2FfGmr6td69qU9/fTNcXU7bndjnP09queLPFV/4y1y51TUJfMnmPAHRF7KPQCsivwvPM4qZjWcU/cWyP6D4dyOnlVDmkr1Huwooor5c+wCiiimAUUdegP4c1La2s17MIreJ55TwFjUsT7YFXGnKT5Yq7IlOFNc0nZEVH4H616X4a/Z78YeItkj2a6XAwzvvGwfwUZP516v4c/ZS0ez8uTWL+bUJM5MceI0+nc/rX0OF4ex2K+xb1PlsZxRluD0c+Z+Wp8vKpk4UZPTiul0T4Z+KPERH2HRbqRW6O8ZRfrk8V9m6D8NfDXhvY1ho1rBIvSQoGf/AL6OTXVJEij5VC/QYr6/DcGKy+sVPu/zPisVx1NtrDUvmz5I0X9lnxPf7Wvrm009ejLuLt+Q/wAa7vR/2T9Ftwralqt1dkdRCFjU/Xr+hr37aKTaK+mw/DWXYfVw5n5nyeI4nzTEae15V5I850n4BeCdLHGjR3DD+K5Jk/mTXXWPhXSNLVRa6Xaw7emyJRj9K2doo217lPBYal8FNL5HgVcZia38So36tkSxooGIwPwFLtH92pKWuvlXQ47vuRbcf/qo49OakxRTEMx7UbT6U+lpjI8e1H4VJRQMj/ClH5U6ikIbx6UhxUlJQAzaG5prQxvwyBh7ipaKnkj2HdrYw9S8H6Hqm4Xel2k+eu+EH+lchq37P/grVFP/ABKVtCf4rVzH+gOK9LoxXJUwWHrfxIJ/I7KeNxNH+HUa+Z8/6x+yXpVwpbTNYubVs8LOqyL+mK4LXP2XfFemljZSWupr2VXKN+RH9a+vNopdo9K8PEcM5dW2hyvyPocPxVmmHa/ecy81c+Ata+HniTw4WGo6LdwAdXEZZR/wIcfrXOk/Ng4U1+jkkKSA7lVu3IzXI+IPhX4X8Sb2vtHt3lbrKi7HP4ivmMVwZpehU+8+rwvHTTSxNL7n+n/BPhGivpzxF+yfp1yGfRtTmsn7R3AEiZ+vB/WvJ/FHwF8X+F8u1h/aFuv/AC1szu/Nev5Zr5HF8P4/Cayhdd0fa4PibLcZoqnK+z0PO6KfNBLbSNHNG8UinBR1KkfgaZXz0ouEuWWjPp4TjUXNF3Ciiip23NLNbhSUtFIW+hoaBrt74b1aDUdPnaC6gO4N269/UHpX2d8KvihafEbQ0nQiG+j+W4tyeVPr9DXxDW54L8ZX/gbXoNT09yHQ4ljP3ZEJywP9PevrMhzmeXVVCT9xnxfEeQ080oOrBWqR/E/QFcetPrmPBPjKx8baDbapYSb0lGGQn5kYdVPvXSK3bPSv3GnWhWgqkHdM/AKlOdGbhUVmh9FNU5WnVsZBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABSGjNRSPsz3pN2V2BV1bU7fR7G4vLqVYYIULu7cAACviX4tfEm5+I3iN7new06DKWkDdAufvEepr0T9pL4qf2teHwzpkubeFgbuRW4Zx0T6DvXgvXnnB55r8h4ozj283hKT91b+Z+z8I5H7GCx1de89vLzDuev40UUetfnj0P1LzCgc0degwe3fNeheBfgd4l8cFJUt/7OsSf+Pq6BAPuq9W/SuvC4StjJ8lGN2cGMx2GwMHUrzSR55uHPPTtXW+E/hb4l8ZyL/Z2myCAnm4nGyP8z/TNfTfgj9nnwz4V8ua6iOrX6gfvLo5QH/ZToK9Qgto7aNY41EaLwFUYFfoOX8Hydp4uXyR+ZZlxvvDAx+bPA/CP7Kdhbqk3iG+e8l729udkY+p6n8MV7H4f8D6L4XtxFpmm29oq/wAUa/MfqetdBtHFG0V9/hMqweDjalTR+cYzNcbj5c1eo3+Q1Y9vQYpdvpT6K9fpZHkjAp2jinClooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBOfwpu30/nTqWgBm059/Wm7ByCM1Jiip5U9GGpzXiT4e+H/ABZEU1TS7e5PQSbQHH0brXi/jD9lKF983hzUGhPa3uvnX6Buo/HNfRuKQqMdK8jF5Rg8bFqrBeq3PYwWb43AO9Co0u3Q+BfFPw68Q+DZmXU9MlhjB4mUboz+IrmvXv347V+itzZ295G0U8ayxt1RhkflXk/jj9m3w94lLz6bnSL05wYeYyfdf8K/P8dwfOmnLCSuuzP0nLuN4SahjoW80fIdFdv42+DviTwM7SXVo13ZA4F1bfOv44GR/KuI9Rx6cHnNfAYjC1sLLkqxsfpWFxuHx0PaYeakgo9KSlrl30Z2+bO/+DvxMm+HniBPNZm0q4IW5j/u+jj3Ffaen3kOoWkV1bukkMqh0dDkMD0Ir87a+h/2aPikY5R4V1KTKtzZSMfxMf8AUfl6V+kcL5w6clg67917H5TxfkSqReOoLVbrv5n0tHnHNPqNW4zTufWv1m5+OjqKQUtMAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKQ0tI3NADGY15x8bviOngHwrI0LL/aV3mK3Qnocctj0A5/Ku/vLpLO3lmmdY4o1Lu7cAADJNfD3xa8eS/EDxfc3oZhZxEw2qn+FAev1Pf8PQV8nxBmiy/CuMX78tj63hvKXmmMXOvcjqzjppHuJHklYvI7FmZjkknqfxptGRxxjPSrWl6Xea1fRWdjbSXdzI21Iolyx9/p71+GqNStJpK8mf0K5U8PTUpOyj+RU/n2rsPAvwr174gTgafa+XaA4e8mBEan0H94/SvZvhl+zPDa+VqHirbdS8FLFW/dr/vH+I+3SvfrOxt9Nt0gtoUhiUYCIMAD0xX6JlXCs6lquL0XY/Mc44yjTvQwGr/m6fI8z+H37Pnh/wAIbbi6QarqA/5a3ABVT/sr0FeppCI1CoNoHYU9VFOxX6bhsHQwkFCjFJH5PicZXxk/aV5uTEVeOKNgpaMV2nGFLRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAM2/N0pGXsTUlJSsBDLbxTIySKHVhghhnNeR/EH9nHQfFJkudMA0i/I4aFcxsf9pe31GK9ixTG71xYrA0MZBwrxuduFxuIwU1UoTcWfBXjT4b654DuTHqdmywZwlzH80T/Ru30P51zHI5x16V+iGpaVZ6tayW17bx3NvIMNHIu4N7c188fE79mfaJdQ8K8d2sJG4/4Af6Gvy3NeFZ0E6uE1XbqfreT8Y069qWOVn/N0+Z861PZ3UtjeQ3EDtFPE4dHU4IYdCD9aL+yuNLupLa7gkt7iNtrRSDDD8Kh6V8DapQn2a/A/Sr08RSte6f43PuD4P/EKLx94St7ksq30KiG5j/uuO/49a7tWyBng96+Ifgv4+bwD4wgllcrpt0RBcrngAnh/qD+ma+2beVJ4UljbcjAEN6iv3bIc0WYYZJv3o6M/nbiLKXleMcYr3Jar/Isr0paRelLX1B8sFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTW6UM22quoX8Wn2U1zNIscUKF2ZugAGSTUykopt7IcVzPlW54t+034+Oh+H4tAtZcXd/zPsPzLCOv59Pzr5U9cn3/+vmui+Ifi+bxx4uvtVlbEcj4gU9ViH3cfh+pNd38IPgPeeN3j1PVkktdFyCqdHn9hnotfh+OlXz/MHCirpaLyP3vLY4fhnLFPEO0pavu/I5P4d/C7WPiNfCOyj8mxU4lvJB8ifT1PsK+t/h78LdG+HtkqWVuHu2GJLqQAu5789h7V0ui6JZ6Dp8NnZW0dvbxDaqRjArQ2+or9IyjIaGXQUpK8+5+XZ1xDic1m435Ydv8AMao/ClZadt9KXFfUq9tT5MRPu5xinUgGKWmMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACkpaKAGMKbtGTxUtIaXkB598R/hHo/xCs3NxH9mvwP3d5EAHX6nuvsa+SvHnw31j4eaibfUYMwOT5NzGP3bj2PY+xr70YVleIvDth4n02Ww1G1S6t5OCjjP4j0NfJ5vw/QzCDnTXLP8z67JeIsRlUlBvmp9v8j89sgqD1GeMV9c/s4fEA+KvDB0q7k33+mgJknl4+dp/DGD+FeNfFz4IX/w/me/sg93ozHIk+80BznDD0965r4WeM5PAvjaw1EOwti3k3Cn/nkxwT+HB/Cvz3LZV8hx6p1laL3P0vNqdDiPLHVwzvKOq7ruj7wT7op1QWtwk9vHIjB0dQwYdCDU2fev22MlJJrY/B3poLRSClqhBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAETH5jXi/7TPjj+wfCKaTC4W41Jtr88iIfeP48D8a9mkPU9x+VeMWfgNvid8RrjxNrEbf2LYt9n0+3bpMFPLn23Z+vFeLmiq1aHsKO8j18rlRpYhV6+sY627vojhPgh8B31rydc8QwMtnkNBZsOZMfxMPT2r6gtbZLWFYo4wiKMKqjAAHQU+G3EKhU+RQMBVHAqQKBVZbltLLaKp01r1Y8zzSvmlZ1az9F0Qq9KWkAxS17B44UUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADTSU6jFAFS+tI76F4J41lhkG1o3GQwPUYr5W+NnwJl8KSS61ocTTaUxzNbxqS1v7j1X+VfWZQGobi1S5jeORQ8bDBVhkEdxXiZlldLMaTjNa9H5ntZXmtfK63tKT0e66NHlH7Ofjj/AISbwTHYzvuvNN/cMc5JT+Bvy4/CvWx1FeLWfgR/hV8SE1XSUYeHtUb7PdQgcWzk/I3+7u49t1ezqemetVlftYUI0a28dL9zLM3RqYiVWh8Mtbdn1JV6UtIOlLXsnlBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFJS0UARMgbIPQ8UyKFIVVEVVVeAFGABU+0UbRRpuAClpKWgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAQ0gp1JQBBJBHKrK6blbgqRkGnhcYwOKkxRilYPIF6UtFFMAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAaW5rjvFnxg8E+CY2fXfFWk6dt6pNdpv/75BzXQ+IV/4kmpN3FtIQfQ7TX4dzySTTNJJI8kmeXdixP1J60Afrd4F/aw+H/xK8fW3hDw1fXWqajNHJL56WzJAAi7j8zYJ49Aa9nFflj+wH/ycro/p9hvP/RVfqcKAFooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACk3DpS18/ftjfHDxH8C/BOjar4aWze6vL/7PJ9thMihPLZuAGHOQKAPoDNJur8yP+Hi3xWIyIvD+MZH+gv8A/Ha+wf2P/jR4g+OXw41LXfEiWaXtvqb2i/YojGmwRRsOCx5y5oA94opBS0ANLYOKWvJP2nfjDf8AwN+GbeJ9OsINRuVvIbUQXLMqFX3ZOR34r5LH/BTLxbgY8H6Njt++m/xoA/Q7dTq+f/2TP2jNW/aG03xFdappVnpTaXNDHGlozMHDqxyd3uv617/+tAC0UUUAFM3gGhvvcGvgP4y/t1fEP4f/ABV8VeHdNs9CksdMv5LaFri1kZyqnA3ESDJ/CgD7+LCkDgnFfm7ov/BQ/wCJupaxYWkth4eWKe4jiYpaSZwWAOP3pxwfSv0iVTxk5oAdRRTSfWgBd1G6vi/49ftyeJ/g/wDFbXPCdt4a0y9t7BovLuJ5JA7h40fJAOOrGsz4T/t/eI/iF8SvDfhu88M6XaW2qXsdo88MshZNxxkAnrQB9yUVXnuktYZJpnWKGNS7ySMFVQOSST0FfKHxl/4KEeFvBdzPpng+z/4SzUo2KNdFzHZow9GALSf8BwPegD613ckUZ/zivyp8Uft0fF7xFcOYddi0WAk4g020RdvtuYM361z1t+1z8XreQSJ441BiOfnCOPxBXFAH67hgeadX5oeBf+CiXxD8PzoviK007xRZ5w4aMWs/4Ovy5+q19nfBL9qTwV8coRBpF2bDW0TdLpF8Qs4HqmDhx9Dn1AoA9hoqNJC2OMVJQA0tijdXOfETUdb0nwXrV94cggutatbV5rW3uVLJKyjdsIUg5IBA9yK+A/8Ah5N8Q0JB8PeHwQe8M/8A8coA/R/dTq+C/hP/AMFCPEvir4jeH9F8SaRotlpGo3a2s1xarKrxl8qhG6QjG7b1Hc194q3qMGgB9FFFABSZpao6xqlvoum3moXcgitbSF55pD0VFUsxP0ANAF3dRmvzvuf+Clnixbmb7N4V0Zrfe3lGV5dxXPBOG64rpPhj+3n47+J3j/QvC9p4V0VJdSulhaRXlJjj6u+N3ZQ35UAfdm6gNmo1XqCea+YP2tP2rPEH7PfirQ9L0bStN1GK/smuXa937lYOVAG1hxxQB9J+IW/4kOpf9e0v/oJr8OH+831r63vv+CkPje9s57Z/DOhKsqNGzDzgcFSCR89fJDfMSf5UAfRP7ApA/aU0f/rxvP8A0Ua/U6vxi+C3xY1D4K+PrXxVpljb6jd28MsCwXRYIfMXaTkHtmvov/h5h4xPI8J6Hjt883/xVAH6JbqFbdX52N/wUw8Y4JPhPRMezy//ABVfZH7OvxPvfjF8KdK8V6haQWN3evMrQWxJRQkrICMnPRf1oA9MopKWgApKgmukgjeSRljiQFmdiAFAGSST2FfMXxc/b+8EeBJ59O8OwSeMdUiyGa1k2WqMPWXBLf8AAVP1oA+o9wo3V+Yfib/goV8Utamf+zDpWgw/wx21oJWH1aXd/IVy8f7bHxlSUP8A8JgWOc7WsLcqfw2CgD9ZFbNLX5reD/8Agox8QNFnQa/pul+ILXPzfujazY77WU7fzWvr/wCCP7WXgj44MtlYXLaTruMtpWoELI3qY2Bw4+nPtQB7XRUauzN93AqSgApCwHWlqORd31oAfkUbhX58+Iv+Ci/jbQPEOqaYPDGhuLK6ltwz+duOxyuT8/tXqP7Lf7YPiP48fES48P6roml6fbRWMl0JLPzN5ZWUY+ZiMfNQB9a5pNwrjPih8WvDfwh8MS634m1BbG1U7IowN0k79kjUcsfp+lfCPxO/4KJ+M9fvJrbwbZW3hjTslY5p41nu3Gep3fKv0AP1oA/SAycE+lOXpX5S/DP9pP4oeIvin4Str/xtq01rc6vbRTW4m2RujyqGUqOMEE1+rSqFGAMCgBaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvj7/AIKWf8kt8Mf9hY/+imr7Br4+/wCCln/JLfDH/YWP/opqAPzn5xyc8d6/ST/gmyB/wpXWzjn+3Zf/AERBX5t/w/hX6S/8E2f+SK63/wBh2X/0RBQB9Z0tFFAHzL/wUMA/4Z5k4/5itr/7NX5g1+n3/BQz/k3mT/sK2v8A7PX5g0Affv8AwTH/AORb8ef9fVr/AOgSV9t18Sf8Ex/+Rb8ef9fVr/6BJX23QAUUUUAIfTtX47/tPf8AJwnxA/7C8386/YjuK/Hf9p//AJOE+IH/AGF5/wD0KgDhPCnPirR88/6ZB/6MWv3Dr8PPCf8AyNWkf9fkH/oxa/cSgApKWigD8q/29IPK/aY19sY8y2s3/wDICD+YrzT4F3q6f8aPA1yxCLHrNq7FuAFEq5Oew4/Q16r+3+Nv7SGqH1sbQ/8AkOvnJJGhcOhKuuCGU4IoA+p/2xf2rr74j65feD/DN21t4Ts5GhuZoWIbUZFOGJx1jBzhe+Mmvmnw94a1bxZqkWmaLpt1ql/NwlraRtI5/BR09+1dX8E/g7rPxv8AHlp4d0geUjDzbu8YfJbQA/M59T2AzyTiv1Z+D/wU8KfBXw/Hpfh3T1jl2gXF9IoM9y3dnb+g4HFAH5/+Fv8Agn38U/EFss99Fpfh5XG4R31yGlHsVjDAH6kVB4t/YF+Knhu1kubS107xDGgyU025zL+CuFyfYEmv1I2gYwMY4FG0HqM0Afhrq2j3/h3Up7DVLK406+t2/eW91G0bpzzuBAI470zS9UvdE1K2v9PuprK+t3EkNxAxVkYd1IPrX61/tD/s4+Hvjx4blguYIrLxDCh+w6siDzEbHCP/AHkPcHpyRg1+T/irwtqfgrxJqWg6zbG21OwneCeHrhx3B7g8EHuDQB+lv7H/AO1EPjboj6Jrjxw+MNNj3S4AC3kXTzVH94HAYe+a+lFOfevxO+HPjzUfhn440nxNpUpjvLCdZduSBImcNG3sRkH2Nfsz4P8AE9n408L6Vr2nP5ljqVtHdQt/ssAcH3GcfhQBrSDJHXnivyO/a2+F/wDwqv44a5YwxCLS9Qf+0bIKMDy5CSyj/dfIr9dOtfJX/BRD4Xf8JN8M7LxbaRZvfD0wE7KOTayHa2f919h+hagD8345HhkV42MbrhgwOCpzwfwr9i/2efiSvxW+EPhrxAXBu5bYQ3a9ds8eFkB9yRn8a/HLpgdxX3F/wTX+JIhvvEXgW5m4kUalZKT3GElUe+NjfnQB96jpS0i0tABXzr+3Z8RD4F+BOoWcEnl32vSrp0QBwdh+aU/98Aj8a+iCa/Nf/gon8Rf+El+LFj4ZgkzaaBajzQp4+0S/O35LsH4mgD5Qx0PX0r7V/wCCbvwyN/4i17xzdQ5hsUGnWbMP+Wr4aQg+ybR/wM18WKjSOqIpd2OAOuST0/z61+w/7Nvw2X4U/Bvw3oLRhL37OLm845M8nzvn6ZC/RRQB6cOFr44/bg/Z98dfGbxl4dvvCekrqVrZ2DwTs13FDtcyEgYdhngivsijaKAPyd1D9iT4wabY3N3ceGYY7e3iaWVv7RtjhVGScCTJ714UepHfNft54448Ga//ANg+4/8ARbV+IjfeNAHRfD/4e698UPE0Hh/w3YjUNVmjeRIPNSLKqMsdzkDp79q9U/4Yd+M//QqL/wCDC2/+OVpfsBj/AIyV0f8A68bz/wBFV+pw6UAfk+37Dvxm2n/ilAP+3+2/+OV+gP7KPgPXPhr8D9E8P+IbP7BqttJcNLD5iSbQ0zsvKkg8EdD3r1+jHegAFVdQ1CDS7S4urqeO2tYI2llmlbCooGSxPYAVZzzXxV/wUP8AjhJo2l2Xw50m4ZbnUE+1aqUbBWDOEiP++QSR6KPWgDxT9qz9rrU/i5qd54d8OXEuneC4WKbVJSTUCCfnkPZDj5U79TXzvomg3/ibVbfStKsZ9Sv7h9sNrbx73c9gAM/4VWsrSfULq3tbWJri6ndYY4oxlnZiFCj1JJr9Wv2Wf2a9L+BfhOCe6giufGF9CGvr0gZizz5Mf91V7+poA+Zvhr/wTh8Ra5aw3fjLXIfD6SDcbGzUXE6g9mbhFP0zXpd5/wAE0/BslnstfFWswXGP9bIkbqffbgfzr7FVV4IA9QcUu0UAflV8a/2K/G/wfs59XtdniXQIvmlvLBCJYFHVpI8k7cfxDj1xXgVne3Gn3kF3aTyW1zC4linjOHRhjDAg9QOeK/c6SNJAyuodWG0qwyCO4PtX5n/tzfs72vwr8VW3ifQLUW/h7W5GEkCLhbW5+8wHorD5h7hh0oA+lv2Mv2n3+Mmhv4c8Qyr/AMJfpkIYynj7dCOPMA/vjIDfXNfTqnNfin8LfiBe/C/4gaJ4osXKy2NwsjoDxJGSN6H6rkfiK/Z7QdWt9f0ex1OzcS2d7AlxC4/iR1DA/kaANCmsu6nU0n5qAPxY+MEP2X4teNIsYC6zeAY7fvnr2n9gbWrTw38Wdd1a/lWGxsdBubmeRv4UVkJP5Zrx/wCOSiP4z+OgOP8Aid3n/o5q5zRfE2oeHrXVreymEMep2hs7n5c5i3KxA/FV/KgDsfjz8bNV+Onj6612/kkSwQmLTrME7be33fKMcfMerH1PpiuJ0Tw7qviW7+y6Rpl1qU7Hb5NjA0p+mFr379k39lCb44Xr65rxms/B9i5jZo+JLyUYzGhPRR/E3vge36VeC/A/h/wDpMemeHtItdIsowFEdtGF3Y7s3Vj7kk0Afl38I/2efiXp/wASvCGp3XgbXLWwt9WtJppprNkVEWVSzHIGBj+VfrGpyuaTaG5IpaAFooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK+Pv+Cln/JLfDH/AGFj/wCimr7Br4+/4KWf8kt8Mf8AYWP/AKKagD85/wCH8K/SX/gmz/yRXW/+w7L/AOiIK/Nr+H8K/SX/AIJs/wDJFdb/AOw7L/6IgoA+tKKKKAPmb/goZ/ybzJ/2FbX/ANnr8wK/T/8A4KGf8m8yf9hW1/8AZ6/MCgD79/4Jj/8AIt+PP+vq1/8AQJK+26+JP+CY/wDyLfjz/r6tf/QJK+26ACiiigBO4r8d/wBp/wD5OE+IH/YXn/8AQq/YjuK/Hf8Aaf8A+ThPiB/2F5//AEKgDhPCf/I1aR/1+Qf+jFr9xK/Dvwn/AMjVpH/X5B/6MWv3EoAKKKKAPy3/AOCgX/Jxuo/9eFp/6Lr5v6N7Zr6Q/wCCgX/Jxupf9g+0/wDRdfN0i7gwzgnofSgD9Rf2D/hbD4E+CtnrU0ONV8Rn7bNIeoi+7En0xlv+BGvpNelc18NbKLT/AId+F7aJQkcWl2yqo7fulrpqACiiigBu0elfA3/BSP4YR2epeH/HlnCFa7/4ll8w7uoLQv8A9871/wCArX33Xz7+3Roaa1+zd4jdlBexeC7RscgrIo/kx/OgD8qAfxB5Nfpr/wAE8vGj+JPgfJpMzlptCvpLYZP/ACyceYn5bmH4V+Zdfcf/AATH1Rxf+PdOJ/dtFaXAX3BlUn/x4UAfeg6VleKfD9p4r8O6no1/GJbPULaS2mVhkFWUg/zrVFHGaAPxG8d+D7zwB4z1nw3fgpdabdSWzFhywUkKw+owfxrZ+CfxCl+F3xS8N+Jo2Ijs7pfPVT96F/lkU/8AAWb8cV9I/wDBRv4X/wBi+M9H8cWkW231eP7HeFRwLiMZRj/vJgf8A9zXxsfu9P8A6/f+lAH7oWN1Fe2sNxDIJYJkWSOQdGUjIP4g1ZrwD9if4ln4ifArSY55hLqWik6Zc5PJ2AGNj9UK/lXvwNAGd4g1e38P6NqGq3j+XaWVvJcSsTgBUUsf0Ffir478WXPjzxprniG7Ytcapdy3TZ7BmJA/AYH4V+lH7e3xE/4Qv4H3GlwS7L7xFOtggHURD55m/wC+QF/4HX5ddRgD8z+maAPaf2Q/hgfij8ctDtZot+m6a39pXgxxsiIKg/7z7Fr9bhjjjFfIv/BOn4Z/8I78NtR8X3MW261642W5Yc/Z4iQMexbd/wB8ivrsUAFLRRQBh+OP+RM1/wD7B9x/6KavxEb7x+tft344/wCRM1//ALB9x/6KavxEb7x+tAH0R+wH/wAnK6P/ANeV5/6KNfqdX5Y/sB/8nK6P/wBeV5/6KNfqdQAtFFFAEM8yQpI8h2xou4t6DnJ/Svxj+Nnjyb4l/FbxP4ilcst3ev5K5yFiQ7IwP+AqK/Wr42643hn4R+MtVVtrWuk3Mikdj5ZA/nX4vrlcc8gYNAH0V+wb4Dh8afHqxvLqLzbXQrZ9R2kceaMJF+RbP/AK/UtemOo+pNfi/wDC34xeKvg1qV5qHhW+jsLq8iEMzyQrJuUHIGD716R/w3R8Yh/zMVv/AOAEf+FAH6tUV+Uv/DdPxi/6GK3/APAGP/Cj/huj4xf9DFb/APgDH/hQB+rVeF/treHYPEH7OPisyoDJYLFewt/dZJF5H/AWYfjXwz/w3R8Y/wDoYrf/AMAY/wDCsbxl+118T/H3hnUPD2ta5Dc6Zfx+XcRR2kaMy5B4YcjkUAeN/Xvx6+tfq3+w74sfxV+zp4dEjF5tMaXTmbPaNsqPwRlH4V+Um3+X0r9HP+Ca9403wl1+3JOItWZgP96NP/iaAPrumP8AeH+e9PHQUh60Afil8VNVbXPiZ4s1B4xC9zqt1KYwchSZWOKwdH0ubXNWsdOt1DXF5cJbR56b3ZVGf++h+FX/ABx/yOniD/sIXH/o1q679m/S01r49eBLSRQ0T6rCSD0+Ulv6UAfrX8O/BNh8O/BOi+G9Oj2Wmm2ywLxgsQMsx92Ykn610gGOlIO9OoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAr4+/4KWf8AJLfDH/YWP/opq+wa+Pv+Cln/ACS3wx/2Fj/6KagD85/4fwr9Jf8Agmz/AMkV1v8A7Dsv/oiCvza/h/Cv0l/4Js/8kV1v/sOy/wDoiCgD60ooooA+Zf8Agoacfs8y/wDYVtf/AGevzBr9Q/8AgoNbtP8As63jgZEOo2sjfTLD+bCvy85HBoA+/f8AgmP/AMi348/6+7X/ANAkr7ar4P8A+CZGrKJvHmmMf3jLaXKjP8KmRT+rCvvAdKAFooooATuK/Hf9p/8A5OE+IH/YXn/9Cr9hyfmr8b/2jNQi1T49ePriBt8TaxcbW9cOR/SgDjfCf/I1aR/1+Qf+jFr9xK/Dvwn/AMjVpH/X5B/6MWv3EoAKKKKAPy3/AOCgX/Jxupf9g+0/9F183yd6+kP+CgX/ACcbqX/YPtP/AEXXzdKTtcjqAT/OgD9uvAf/ACI/hz/sG23/AKKWt+uS+FOqR6z8NPCd7CcxT6VbMCP+uS11tABRRRQAV4n+2VcLb/s0+ONxAMlqka59TKmP5V7ZXyr/AMFEvFyaH8EbbRw4FzrOoxxqncpHmRz9OFH/AAIUAfmj619r/wDBMeBm8S+OZsEotpbLu9y78f8Ajpr4ox8vHJP4fjX6Hf8ABNLwy9n8PfFWuuuBqGopbxt/eWGPJP8A31IfyoA+yqKKKAPKf2mfhiPix8G/EOiRRiS/SI3Vl3InjG5QPrgj/gVfj80ZjZlZSrKdrKeCCDX7qMuTX5L/ALYnwv8A+FX/ABw1mK3i8vS9WP8AalngYAWQkuo+j7qAO8/4J6fEw+FPi1deGrmbZY+IoNiKTwLiLLIfqQXX8R6Cv0qzwAD171+IPhXxHdeEPE2la7ZMUu9Nuo7mPB7qwOPxwR+Nfs1D46024+HqeMRKBpTaZ/afmZ6ReX5hP5UAfnn/AMFCPiL/AMJX8ZIfD8Epaz8PWwhKqePPkAdz9cbB/wABr5x8K+Gbvxl4m0rQdPXdealdR2sC4yAzMAD9B39qf4w8TXPjPxXq+v3hzdaldy3b7jkguxOPwBx+FfS//BPH4ZHxN8T77xVcQ7rLw/BtiZhkG4lBVce4UMfyoA/Q3wZ4Us/BPhPR9A09dlnptrHbRDGMhVAz9TyT9TW5SCloAKKKKAMPxx/yJmv/APYPuP8A0U1fiI33j9a/brxyf+KL1/20+4/9FNX4it940AfRH7Af/Jyuj/8AXlef+ijX6nV+WP7Af/Jy2j/9eV5/6KNfqdQAtFFFAHkf7WU32f8AZy8fMf4tNdPzIH9a/IP+fev2P/aQ0OTxJ8CfHWnwqWmk0mdkA7sq7gPzWvxwyG5HNAHpPwT+AfiT48ahqlp4cks45dPjjkmN5KYxh2IGODk8V61/w7q+KJ/5eNE/8C2/+Jq1/wAE5fFEOj/GXVdImcI2r6ayxc/eeJg+0e+0sf8AgNfpUvSgD8y/+HdPxR/5+NE/8C2/+Io/4d0/FH/n40T/AMC2/wDiK/TSigD8zP8Ah3V8Uf8An50T/wAC2/8AiKP+HdXxR/5+dE/8C2/+Ir9M6KAPzMP/AATr+KP/AD8aJ1/5+m/+Jr6x/Y6+BniH4EeEdc0zxFJaSXN7ei4j+ySF12hNvOQOa+gtopNvvigBV6CkPWlpD1oA/ETxx/yOniD/ALCNx/6Nau8/ZTH/ABkZ4B/7CS/+gNXB+OP+R08Qf9hG4/8ARrV3n7Kf/JxngH/sJL/6A1AH7AUtFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFfH3/BSz/klvhj/sLH/wBFNX2DXx7/AMFLmx8LvC4/6ix/9FNQB+dH8P4V+kv/AATZ/wCSK63/ANh2X/0RBX5tdvwr9Jf+CbX/ACRXW/8AsOy/+iIKAPrSikpaAPH/ANrbwzJ4s/Z68a2cSF5obI3iAf8ATJhIfx2q1fkPX7nX1pFf2txbToJIZ42jkQjIZSMEY+hr8afjd8NLr4Q/E7XPDFypEVtMWtJO0tu2TG2T/s4/EGgD0b9iH4kQfDv47adHeyiDT9bibTZXfhVdiDExJ4xvAX8a/VdWJNfhWrNG4ZSyspLLt6gjpivu39nv/goBY2ej2ug/EkTJcWqCKPXbeNpVkUcATKOQw/vAHOOmeaAPuqivK7P9qD4VX1uJ4/HujbD2kn2N/wB8sAf0rzz4kft6/DTwbZzDRr2XxXqQU+XBYIVi3dt0jAAD1wCaAPTvj98XNP8Agv8ADfVvEF5Mq3XlNDYW5PzT3BHyKB3A6k9gDX47Xt5PqF5cXVy5kuZ5GllY92Y5P6mu7+Nfxy8TfHTxV/aviCYJDDuS0063z5Nsnooz8xzjLdTx9K5/xr4D1f4f3mnWmt232S6vLGG/WBvvrHJnbuHZsDOPegCPwDZjUPHfh62LbBNqFum76yLX7cq1fib8Nm8v4ieGOempW5z/ANtFr9sl60AOooooA/Lf/goF/wAnG6l/2D7T/wBF183t3+tfSH/BQL/k47Uf+vC0/wDQK+b/AK46/wCf8/SgD9Hf+CfnxqtvFHw9PgW+uAus6CC1rGx+aa0JyCvqUYlT7Fa+uVb161+IvhzxFrvw68VWmqaXPcaTrVjIGjfbh1OMlSuOQV4IPBB+lfoX8E/2/PCXjKzgsfGrL4V1wAK9wwP2OY/3g3VPo3T1oA+saKw9H8aaFr1us+na3p1/Cw3LJa3Ucgx+BNYnjD40+B/AdrJca54q0uxVP+WZuVaU/SNcsT9BQB2MkwjVmZtiryWPQD1PoPevys/bS+NUPxe+LEsOmzCXQtCVrK0dT8srg5lk+hbAHsgr0H9pn9uibx9p934W8CR3Gn6LOPLu9UkwlxOndUX+BT/ezn6Zr4+/hO4c916fh7f/AFqAHxxPPIkaIzyOQiovViSAB7nJFfsX+zr8OT8K/g74a8PSqq3sVsJrsKP+W8hLuPwLY/Cvh/8AYW/Z5l8feLovHGs2p/4R3R5t1qsgGLu6HIx6qnUnuQB61+lEeDzQA+iiigBMA18o/wDBQr4X/wDCW/CmDxTaxbr/AMOSmSUgcm1fCufopCn6bq+r6zPEOhWniXQ9R0i+jEtlfW7200bd0dSp/QmgD8OvUHvzX1tZfHjy/wBg258O+f8A8TRdR/sJQT832dj52fptyn0r5s+I3gu6+HPjvXPDl8p8/TLt4Sx/iXOUb8VINc950nkmHe3lbt5jydpbGM49aAGltueMDHGa/Wb9j34Xn4X/AAQ0WC4i8vVNU/4mV4CMMGkA2qf91Ao+ua/Or9mf4YP8XPjPoGiNGXsI5ftt8ccCCP5m/MgL/wACFfsJHGqRqqqFVRgAdAPQUAOX7ox0paTpS0AFFFFAGB49SSTwP4iWIZlbTrgIPU+U2B+dfiN3NfupNGsqujjcrLhlPcV+Lfxe8DzfDX4meI/Ddwmz7BeSJFkH54id0bD6oRQB6R+w/rEWj/tKeGDKwUXCXFqM92eFsfyr9X/xr8OvDPiK98I+I9M1vTpfK1DTriO5hkIyA6HIBHceo75xX6nfCT9sL4efErQbWe716y8Oaz5Y+0afqc6w7XwM7HYgOueRg5x2oA94pK891b9oD4baJCZbzx1oCKOcR38cjH6KpJP5V3Ol6hb6tp9ve2kontLiNZYpVHDKwyD+IoAW6tYryGaCZd8UyGN1boVIwR+tfjN8afh5cfCv4oeIfDNypRbS6b7OxH34WO6JvcFT19q/aHaPSvlX9uL9nOb4peGovFegW5k8S6NEyyQoMtd22dxUerKckevI9KAPzu8D+MNQ+H/i7SPEelyeTfabcrcRHPykqeVPqCMr9DX7EfCj4oaR8XPBGn+JNFlV7a5jHmQZy9vKPvxP6EH8+vcV+LhBXKtkMp2nsQRxgjtXpHwO+P3ib4D+IDfaHKtxYSkC80ucnyJ1HTpyrDswyfwoA/Y5W3DIpa+ffhb+2v8ADb4jWsMd1qq+F9WYASWerMI0Deiy/dI+pH0r22z8TaRfW6zW2q2VxC3IkiuUZSPUEHBoA1aKxb7xloOmxl7vXNNtFH8U93Gg/Vq878XftYfCrwZG5vfGen3UqDPkaa32pz/3xkfrQB67S186fB39svw/8avio/hTR9Lu7W2NrLPDe3jKrTMmPlCAnA25PJ7V9FCgBaaetOpp60AfiJ44/wCR08Qf9hG4/wDRrV3n7KX/ACcZ4B/7CS/+gNXB+OP+R08Qf9hG4/8ARrV3n7KX/JxngH/sJL/6A1AH7A0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV8a/8ABTKfb8O/CEX9/VXP5Qn/ABr7Kr4X/wCCnGqJ9n8Cabu+cyXNzt9AAi5/M4/GgD4P29q/Rn/gmndNJ8J/EtuR8sWtFgf96GPP/oIr85hnbjv/AF9K/S3/AIJy6K2n/Au9v2HGoaxNIh7MqpGmR+KtQB9V0UgpaAErwH9q39mO0+Pnh2O5sDHZeLNNQ/Y7luFmTBzC/sT0PY17/SMA3FAH4g+L/BeteA9cuNH8QaZPpmowuQ8M0eC2O6njcD1yMjnpWL+vav2x8bfDfwz8RtPNh4l0Sy1q2xgLdxhmX3VuoP0NeD69/wAE8/hXq87y2n9saRuOdlrehkH4SKx/WgD8xvvE55PrjJqzp9hdapfQ2djBNd3czBI4LdC8jk9goyT06V+k+k/8E5/hhYyh7q817UgP+Wct4iKf++IwfyNe3fD74I+B/hfHjwz4asdMlIwbkJvmb6yNlv1oA+TP2Vf2HbizvLHxh8RbZY3iKzWWgvhsMOVecex6J+fpXn//AAUb08Wnxx0udQMXGiQtwO6yzL09sCv0tA4rkfGHwj8G/EC+ivPEfhrTdauYU8qOa8gEjKuc4BPbJNAH42eDbwWPi3RJ8/6q9gfrnpIpr9v42DAEdO1ebx/s1/C2NldfAWhq6kMrC0Xgg5BFejxxiNQoGAOAB0HtQBJRRRQB+W//AAUC/wCTjtR/68LT/wBAr5vPQ19If8FAv+TjtS/7B9p/6Lr5vbjNAH6H/Hv9j2H4veB9D8VeFUhtfFqaVbfaLZ8LHqKiJcAn+GT36HjNfAXiLw3qvhHVp9M1rTrnTNQhbbJb3UZV8jqcHr9c1+03gNc+BvDvGf8AiW23/opapeOfhd4V+Jdn9k8T6BY6xEBhWuIwZE/3W+8v4GgD8VRlc7TjtxTePT+v/wCqv061n/gnj8KtSmaS2GtaUCc+Va3oZR9PMVj+tR6Z/wAE7fhXZSBriTXtQUHJjnvlVT9dkan9aAPzMjje4mWKNGmlc4SONSzE+gABz+FfVH7O/wCwz4h8fXlrrfjaCbw94aUq62b/ACXV2AchQuMohHc8nsO9fdfgP4A+APhmyP4d8LafZXC9Lpo/Nm/77fLfrXfrH69PSgDP8O+H9O8L6PaaVpNnHYabaRiKC2hGFRR/M+pPJPJrTpaKACiiigApGHBpajeQRqSzBVAySegHegD86/8AgpF4HtdH+IPh7xNb7Un1m0eC4QdWeArh/wAVkUf8Ar4//wA4J9j/ACr2/wDbA+Mi/GD4xXs1jL52h6SPsFgRyrhSfMk+jNyPbFec/C7wBffFL4gaL4X09d09/OsbMB/q4xy8h9goJ/AetAH3b/wTr+FJ8N+ANQ8a3kDLea83k2rMPmFrGTyP958n/gIr7BX7orL8L+H7Pwr4e07RtPiENjYQJbwoOyqoAP1OM/jWrQAUUUUAFFFFACV8iftyfsy3/wATIIfGXhS0+1eILCHyruyi+/dQjlWUd3XkY7g+wr68pjLk9KAPwtuIZLW4khljeGWNirRupVlbpggjIP4fjTMbeMYwa/Yf4lfs2/Dz4rTPc+IPDdtJqDcfbrcmCc/V0xu/HNeVyf8ABOn4Vs5K3HiCNc5CreoQPbmKgD8y25VvoQTjoDX7P/Aq8/tD4L+BrkncZNFtCTnPPlLXmvhv9hX4ReHZklfQZtYkU5A1O7eVT9VGFP4ivedJ0q00PTLXT7C2js7K1jEUNvCu1I0AwFA9KALdNwOadSUAfH/7T37Dtt8Qry68U+B1g0zxBJl7nTZDst7tj1Kn+Bz9ME+mc1+f3irwfrngXVpdL8Q6XdaRfQna0N1GUbI64J4I9wea/cEjNc94u8BeHvHli1l4h0Wx1i1PHl3kKvj6HGR+BoA/Ej16YPp0pVYgHacDvX6e+Jv+CfPwq1ySSWzi1XQmbnbZXm6MfhIrY+gNcz/w7T8D8/8AFUeIMen7j/43QB+dAO3ODijoM9s5r9KdN/4Jw/DW1YNd6p4gvsfwG5ijU/lHn9a9D8K/scfCTwnIkkHhG31CZeRJqcj3OD64clR+AoA/Pv8AZHtfEVt8dPCmraNo99qNtBd+Xdy2sDuqQuNshZgMAAHOc9q/W5e9U9N0iz0e1S2sLO3srZOFit4xGo/ACro4oAWmN97rT6Y0e4560AfiH42Yf8Jn4g5B/wCJjcf+jWrvv2UmH/DRngHnk6kv/oLV+pE3wN+H1zNJLL4L0OWWRi7u9jGSzE5JJx1qfSvg74I0PUYL/TvCWj2N7A26K4t7NEdD6ggcUAdiOlLTQvtTqACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAopkj+WpYkKB1LHj8a4fxJ8cvAPhG1efVvGGjWqJywF4jt/3yuTn2xQB27SFWOBmvy3/bv+JVv4/+N9zZ2My3FhoNuunB0OQ0oJaXB9mO3/gFesfH7/goKmoWNzovw2hnjaZTG+u3Ue0gd/IQ9+fvMRjsK+HZZXnkeSR3mkdizOxLFiTkn6k0ALDFJcTpFGpeV22Ko6sxOB+ZOK/ZX4D+AR8MvhH4W8OMNtxZ2SfaARg+c3zyf+PMa+Bf2FfgTJ8RviEnirU7fPhzw/KJBvHy3F0B+7T3Cn5j9AO9fpwg+X196AHUUUUAFFFFACUUtFACUAY6cUtFABRRRQAUUUUAFFFFAH5b/wDBQL/k43Uv+wfaf+i6+b5O9fSH/BQL/k43Uv8AsH2n/ouvm+TvQB+3XgP/AJEfw5/2Dbb/ANFLW7WD4D/5Efw5/wBg22/9FLW/QAmB6UYpaKACiiigAooooAKKKx/EvivSfB+mzahrWpWuk2UQy093IEX9Tz9BQBqeZ8xHvjpXxv8AtuftTweGdJu/h/4VvRJrd4mzU72Bv+POEjmNSP8Aloe/90Z7muT/AGjP2/Fv7W60D4aF0STMc3iCVcHB4PkKRx/vH8B3r4juLiW4mmuJpJJp3ZpHldizOxOSSeuTnPPNAEfTHYggYBxX6Ff8E9fgfJ4d8O3PxB1SDZe6sht9NR1wUtgfmcf75HB9Fz3r5h/ZZ/Z3vPjt44i+0xSReE9NdZNRuh0cDkQq3Tc36Dmv1h03TrbS7OC0tYVt7W3jWKGFBhURQAAB6YA/KgCyi7Vp1FFABRRRQAUUUUAFFFFABSUtFACUdKWigAooooAKSlooATAopaKAEopaKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAimjWaN0ddysNrD1B61+JHjrRT4c8ceINLYFZLHULi3PODlZGH9K/bl+3Ga8W1v9j74WeJvE2oa9qnh1r3UL+drm4MlzIFZ2OWO1Txk+lAH5MWttLfTLDbwvczu20RwoXZj6YHU/5xX018Df2EfF/wAQrm31DxaknhPw/kOyTAfbJ1/2E/gHP3m9fu1+g/g34R+C/AG3/hHvC+m6VIOBNDbjzf8Avsjd+tdhtHegDnfAngXRPhx4VsfD2gWEen6XaJtSFM5J7sxPJYnqa6OkpN3TmgB9FID70tABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH5bf8FAuP2jtSyf8AmH2nHr8navnBj15z+v8AWv2c8WfA/wACeOtafVtf8K6fqupMqo1zcISxVRgDr2FY/wDwy78KNuP+ED0j/vyf8aAO28A/8iL4cP8A1Dbb/wBFLW9Vayto7G1htoY1ighQRxxqOFUDAA/CrNABRRRQAUUUUAFFFFABX54/8FKfCV1Y+OvDfiESTNYahaNbNGXJRZYzkkDoCVZfyr9Dq8e/ac+Bf/C/vAcGhRXsOmX1rex3UF5NGXWPAZXGAQeVY/iBQB+RCnbznGOOvb3r2/8AZ4/ZW8S/HLUobpo5NH8KRuBPq0iYL4OCkP8AePv0Hevsb4W/sB+APA80N7rzT+L9QjwwW9UJbKw7iIfe/wCBE19NWlnBaW8UEEMcEEQCxxRrtVABgADtigDA+Hfw90P4Y+F7Pw/4es1s9NtVwqj7zserue7HufoOwFdOFA6CgADpS0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUmaO9IBaKTNJn3pgOopKWgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooASilooAKKKKAIZD8rdRx1riPBvj6PUNa1Dw9qTLBrmnyFSvQTxnlZFH0Iz75ruWFfO/wC0nod1oGraV4x0p5La5hIhmkj4PXKn+Y54OQK8jMsRPC01Xgr239D1Muw0MZW+rydm9n5n0OGDdKkWvLPg78Y7L4gaetvclbXWoVHmwdA/+0vqPbtXqCN1rswuKpYumqtJ3TOXE4Wrg6ro1o2aJKKarZFLXWcotFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABSYpaKAExS0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUmaTd7UAI3tTd2OtDNjPX8q8l+NHxptvAlm1jpzpca1MvyrnKwg/xN/hXDi8XSwVOVaq9EdeEwlbHVY0aEbtnR+IvHyJ4p0/wxpjLNql0weY9reFcF2b3I4UevWu3XOF714H+zP4duLxdT8W6kzT3d+5SOWXlioPzH88D/gNe+Bce9c+XVp4ql9YmrX29DfMaFPC1nh4O/Lu/PqSClpBS16x5oUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAxuuBWB428M2/i7wxqGl3KgpPGQG9GHKn8DXQ4FNaMHJrKpTjVhKnLZmlOpKnOM4uzWp+esg1Hwf4gdFkktNRsZtu5TgqynB/A9a+o/g58eLXxhHFpmryJa6wowrHASf3X39q4H9qDwC2m6tB4ktY8W91iG429A46MfqM/lXhKyNFIrxs0bghlZTjaR0x71+LxxWJ4dxsqK+C+3kfubweG4oy6Fd6VNr9n5n6MRtuTPWpV+6K+Y/hH+0Y1n5OleKHLRfdj1DqV/3/wDH8/WvpG01CK+to7i3ljmhkAZZI2yCD3r9Wy/MqGY0+ei9ex+PZjleIyur7KvH0fRlyio95p2a9Y8kdRSClpgFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUU0nGaTcaQCsOtRO20Zz7VDfalBpttJcXUywQINzSSHAA9zXzP8XP2jJdWEuk+GZGhtjlZNQHDP67PQds14+YZpQy6m51Ja9j18tyrEZpUVOjHTq+iOy+Mnx8tvDUU2kaHItzq5G15VIKW+e5Pc+1fNGl2eoeNvElvbq7XN/fzKvmOSTyeWPoAM1lSEyyM7Es5b5mY5zkjOTX0H+y34E865uPE93H+7UGC13Dg/wB9/p2/OvyuOJxPEOPjCXwLWx+vywuG4Xy2VWGtR9e7/wAj6A8L+HrfwzoNjplsMQ20SoOOTgck+5NbC9elKijaOPz607Ar9ohTVOCpxVktj8MnJ1JuctWwFLRRWpAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUjUtFAHPeNPDFt4w8N32lXQzHOhUN/dbGQfwr4R8Q6Hc+HNcvNNu0KT28pjYY6+h/EYr9CmWvn/8AaY+GpvtPXxLYRkz2q7bpUHLR/wB76g/pmvheKMr+t0PrEPiife8JZv8AUcT9XqO0J/mfMfVQO3Wu/wDhz8ZNb+HtwiRyte6YT89pKxwP909j+lcD/nij/wDVX5FhsVWwc1UpOzR+3YrB0MwpulXjeL7n3V4F+KWiePrRZNPuQtwP9ZbSECRD7j/CuzFfnXpupXekXkd1Y3Etrcx/clhYqw+h9K99+HP7ULwrFY+KIt6/dF/Cvv8AxL3+or9WyviqniEqeL0k+vQ/Gc34PxGFbq4Nc0e3VH0zS1laLr9h4gs0utPvIruBxkNE2fz9K0efWvvozjUXNB3R+eSjKD5ZKzJKKYGz0paskdRSUtABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUU3vQ1ADqSmFqy9d8Raf4bsXu9TvYrOBf45G2/hz3qJzjTjzTdkVGMpyUYq7Zqt78Vw/wAQPivofw+t2a+uBLdEZjtYuZH/AA7D3rxv4jftPTXhksfC8bQxng30oyxH+wvYe5/KvBL3ULnUrp7m6mkuZ5PmaWRyxP41+f5rxVRoXpYX3pfkfo2T8H18U1Vxvuw7dWdl8R/i/rfxEunWeQ2umhvktIWwuP8AaPc+3SuG/wD1Uv6j07UcDk9O5r8rxOIr4ypzVnds/ZMLg6GBp+zoRSia/hPw3deMPEFjpNou6S5k2E/3V/iY+wA/lX3h4Z0G18OaHZadbRhIbeMIox7f1615B+zb8NG0HRf+EgvosXt8oEasOUi/xY/pivdF6V+v8M5X9Tw/t6i96X5H4ZxXm39oYr2NN3hD8x46UtIvSlr7g+GCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigApKWigBlV7y0jvLeSCZBJDIpVlbkEEYq1imlRmpklJNPYabTTR8P/GL4by/DzxVJDGh/su6YvayHoATyn1GeK4L0r7t+JvgO08feGbjT51Anxugm7xuOhr4h17Q7vw3q11pt9EYrq3cqynuOx/Hr+NfiHEWUf2fW9tTXuS/A/feF87/tLD/V6z/eR/Ez8A9Rml9RRRXx22iPu3725s+F/GWs+DbwXOk3slo2fmjU/u5PqvT9K+g/Af7UVleBLXxJB9hk6fao8tF9W7rXzFx6Uv4+3WvewGdYvL2lTl7vY+azLIMFmabqRtLuj9DNJ1yw1q1S4sLuO7hcZV4mBBrRU9K/Pnw94s1jwrdCfSb+azfOSI2yp/3l6GvbvBX7VEkOyDxLY+YvT7Va8H8UP9Pyr9Ky/ivC4i0K/uv8D8qzHg/G4S86D54+W59NZo3CuV8L/Enw94vgD6bqMcz942O11+qnmum3dxX2dKvTrR5qck0fC1KNSjLlqxcX5klLUat704GtzEdRSCjNAxaKSloAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACikozQAtFNz37UhoAduoz+VMLAd65jxT8SvD3g6Fm1TU4YGH/LJWDSH6KOa56uIpUY81SSSNqdGpWko043fkdOXHrxWdrXiDT9EtWuL68itIl6vI4UD86+dvGf7VM8zPb+G7IQqf+Xq8+99Qo/qRXiXiDxVq/iq6afVr+e9kPOJW+UfRegr4vH8VYWheOHXNL8D7nLuDsbirTxHuR/E+hPHn7UdpamS28NW32ybkC7nBWMe4HU18/8Aibxfq/jC9+1avfSXkn8KscIvsF6Csb/J96K/NcdnOLzBv2srLsfq+W5DgsrV6cby7vcP1ooorw1oz6LzYn6V6T8D/hm/xA8SCW5Q/wBkWREk7dnPaMfXqfauL8L+G7zxdrltpdhH5lxO232Ud2PoBmvuLwD4Ks/A/hy20y1T7i5lkxgyP3Y/jX2nDeUPG1lXqL3InwHFWdfUKDw1J/vJ/gjoobdYY0RF2qgwFHAAqUA0qe3SnV+2KKirLY/B99WIOlLRRVCCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKaV96dRQBG6/jXjnx8+Eq+MtNOqadFjWLVT8q/8t06lT7ivZsVGyBs5rhxmEp42jKjUWjO3B4yrga8a9F2aPzlkjeGRo3UrIpIZTwQR1zSV9G/tB/BcuZ/EuiQbmxuvLaNeT6yL7+o/GvnHjoOxx7V+A5nl1XLa7pTWnRn9H5Pm1HNsOqsHr1XZi0UUV5J7vkw69aQDbnHelopb6i0H280lpIskEjwyL0dGIYfQ16L4W+P/i3wyqRPdrqdsvBjuxlsezDn8683oruw+OxOEd6M2jzsVl2ExqtiKal+Z9UeF/2qNDvgker2lxp0h6uv7yP65FeraD460LxNGJNN1S2uge0cg3D6jrXwCR696fDcS28olhkeKQdHViGH0NfZYTi/E01y11zI+FxvBGGqPmw03Hy3R+jCzAjrmnZr4c8P/Gzxj4cKiHV5LmEcCO6HmDH1PNek6D+1pdwlU1fRkmHd7V9p/wC+T1r6/C8V4GtpP3WfFYvg/M8P70IqS8j6b3Ubq8o0X9pLwdq20T3UunyH+G4iIH5jiu50nxpoeuRhrLVrS53dNky8/rX0dLMMLX/h1E/mfLV8BisM7VabXyN7dS5qGOZW6EEeuakVi3/6q71JPZnCPopM0bqoQtFJkUUALRSUZoAWikzRkUALRSUZoAWikooAWim596M0ALRupu78KjkuFj+8yr9Tipcktw32Jd3PSjdWDq3jbQtDXffatZ2y9fnmUfpmuE1z9pTwfpCv5FzNqTr2tojj/vo4H61w1cwwtBXqVEvmehQy/F4l2o0m/ker7wc00yD6fjXzPr37Wt5LuTSNGjgHQS3Um4/XAA/nXmniD40+MPEbMtxrEtvER/q7P90Ppxz+tfM4nizAUdKfvM+pwnB+ZYjWolBef/APsbXvHmheGYy+o6pa2oH8LyDcfovU/lXlPin9qrRLPdHo1pNqUg6SPiOP8+v6V8tzTSXEheWRpXbqzncfzpn1Oa+RxnF2Kre7Qioo+0wXBGFpvmxM3J/cj0bxX8e/FvijfELv+zbY/wDLGzBRv++jzXnk08lxIZJZGkduSzMST+NR7QOgH5UtfH4jHYjFS5q02z7rCZfhcDHloQS+QUlLRXB0PRCiiij0DcP0p8NvJdTRwwI00sjbERBksT0AHqaYuT0B3ZwoHfivpz9n34M/2WkXiTW4cXkg3Wtu45iU9HI/vH9BXs5XltTM68acV7q3Z4Gc5vSymg6k3eT2R1XwN+E0XgPRxeXqB9ZuwGlPURL2Qf1969WVcL1pFjXIOOak2iv37CYWng6UaNNWSP5xxeKq42tKvWd2wC7aWiiuw5AooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKTFLRQBDJGjK6suVPUEV8ufHf4HyaPLN4h0KAyWLEvdWsQ5j/wBpR6eo7V9UY61BNCskZV0EgbgrjjFePmeW0syounUWvRnsZXmdfK66rUn6ruj85v5Dg0te8/HD4DS6TLPrvh+3ZrQ/PcWaclPVk9vUflXgtfhGYZdWy6s6VReh/Q2V5pQzOiqtF+q6oWiiivLPa06BRRRTAKKKKQB9OPpR7dqKKAEp8cjwuHRmR/7ynBptFXGUo/CyHThL4lc39L8f+JNHwLPXL2BR/D5zEfka6zS/2ivG2nfe1CO7UfwzRKf1GDXmlHPrXoUszxlF/u6jXzPKrZRgMR/EpJ/I900/9rLXIVAu9Js7g9/Kcx/zzXS2P7XFiygXmhXQb/pjIpH64NfM1Ht2r1qfEmY0/t3PGq8J5VU/5d29GfWlt+1V4Ukx50N7bn0MWf5Gtm1/aQ8E3C5bUZIR6SQMP6V8ZUe9ejHi7HR3SZ5lTgjAS+GUkfbkPx68D3HTXoFP+0rD+lX4vjF4NlA2+ILM/VsV8KUV0x4yxXWmjllwLhelWX4f5H3gPix4Q/6D9l/38FI3xa8ID/mYLL/v4K+EKTFaf66Yj/n2iP8AUTD/APP1/gfdcnxj8GQglvENn+D5qjP8evA0PXX7dv8AdDH+lfEdFZy4yxL2poqPAuF61ZfgfZlz+0h4KtgSuovPj/nnEx/pWNc/tVeFY8+TBezn0EW3+dfJnPrSVzT4vx8vhSR1w4JwEfik38z6Zvv2urLBFnody57edIqj9M1zeoftZa9KrC10mztj2MjNJ/ICvC6TgcgV51TiXMqm87eh6dLhPKqf/Lu/qz07VP2ivG2oHjUI7JT/AA28AH6nP865DU/H3iLWM/bNavZw3YzED9KwMD0pa8mrmWLr/wASo/vPbo5TgcP/AA6UV8hzSPIxZnZmPViTk03oQRwR3oorz5SlL4nc9SMYxVkgoooqfIe2wUUUUhhRRRQAUUUU99g23Cj8M/5/yMUYO4AAls8ADrX0L8EPgLJM9vr3iKAqi/Pb2LryfR3Hr7V62W5bWzKsqVNadWeHm2a0MqoOrWevRCfAb4HNcSW/iLX4CsfDWtpIOvcOw/kK+l44wqgbQABikhiCRqoUKAMADgVIFFfu+XZdRy2iqVL7z+eMyzKtmdd1q3yXYVQO1LSDilr1TygooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigApKWigCGeMSKVI3KeCpr56+Mn7PYvPP1rw1Gqz5LzWK8CT1Keh9uhr6JZc03yc9ea8zH5fRzCk6VZX8+x6eX5jiMtrKth5W7+Z+c00L20zxSo0cqHayOCCp6YI7Uz/ADzX2J8W/gXp/jpXvrALYayo4kUYSXHQMP8A2avk/wASeGdS8I6nJYapavazp03dGHqp7j3r8TzXJa+V1LvWD6n73k3EGHzamo3tU7f5GXRR6UV82fV6hRRRQIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooo659qa12DbcKktbeW8uEt4InnnkO1I4xlmY9gK1PC3hHVPGepJY6TbNcTN95v4Ix6sewr6z+FPwR03wDCl1cbb3V2GGuGHCf7KDsPfrX0uU5FXzOSbXLBdT5POuIcPlUHFPmqdEcn8Gf2f49H8nWfEUKz3w+aK0YZSH0ZvVv5V73Gu1QAAB6CljiKnPb86kVa/a8Dl9HL6SpUVofgmPzCvmNd168rt/cgXgUtFFekecFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAROPmrnPGXgHSPHWmva6nbLJx8kq8PGfUHtXT7QaTaPSsatGFeDhUV0zSnUnRmqlN2a7HxX8S/gbrPgOSS5t0bUtIzxcRrlo/8AfH9a8198cfWv0Ymt45oyrqGUjBDc8eleK/En9nHS/EMkl5ohXTNRbJMYH7qQ/Tt9Rx7V+Y5twpa9XB/d/kfq2S8ZKKVHMFf+9/mfKNFbfirwXrXgy+Ntq1jJbv8AwuBlH/3W7/56VidvWvzirQqUZuFRWZ+rUK9LE01UpSTj5BRRRXPddDoCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKLrZjt1Ciit/wAI+A9c8bXiwaVZSTKThpiNsafVv6VvRoVMRNQpxu2ctfEUsNB1KsrJGBwBnr+Pf0r1P4Z/ALWPHDx3d+r6VpJP+scYlk9lB7e9ex/DX9nfSfCjRXurbdU1JcEbx+7jP+yv+NexxxIq7VUBem0dK/Ssp4VtatjfuPyjOuMXK9DAf+Bf5GF4R8EaT4L01LLS7ZYEXlmA+d29Se5roFGKcFGOlLtFfplKjCjFQgrJH5XUqTrSdSo7t9WC9KWiitjMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKSgBaKbuqOe6jtYzJNKkKD+KRgo/WgCaiseXxhocDbZNYsUPoblP8AGp7XxFpd9gW+pWkxPQRzqx/Q0AaNFN3UbuvNADqKKKAIzS7QadtHWjFAGXreg6f4gs5LTULSG8t3GGjmQMDXgvjv9llJvMuvDF15T9fsdySV+gbt+NfR20U3b81eTjcrw2YR5a8T1cDmmLy6fPh528uh+fPiPwnq/hO8Ntq1hPZy9BvXKt7gjjFY9fobrGh2OuWr2t/axXcL8MkigivGfGX7Lekalvm0O5fSpif9XJ88R9sdR+B/CvzfH8I1qScsM7rsfqWXcbUaloY2PK+62Plmiu48WfBfxZ4RZ2uNOe5t1/5b2g8xcep9PyriGUqxUghxwR6fnivhq+ErYZ8tWLR+h4bG4fGR5qE1JeQlFFFclmdtwooopDCiiigAooooAKKKKACiiigAooooAKKKKYroKKKP59aXmP1CilRWkZQg3sxwFAyTXeeE/gj4s8XOpi09rG3b/lveHYMeoGMn8q7cPg6+Kko0YNnBisfhcHFzrVEjgvrxW14b8F634vuhb6Tp810/dlGEX6seB+Jr6S8H/sv6Jo5SbWJn1e4HO37sQ/Acn869i0vR7PR7VYLC2htYF4CQqFFfc5fwjVqPnxcuVdj86zHjalTThgoXfdng/gX9lu3tvKu/Etx9qcHP2SE4jHsT1P4V7xpOjWWi2qWtjaxWsCfdSJAoH5VeUc89afiv0nBZbhsDDlox+fU/MMdmeLzGfNiJ38ugzaKXFOxRXp2PJAdKWiimMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKazbe2aAEZtv0/lXP+LvHGj+CbA3eq3awg/ciHMknsq/1rE+KnxQtPh5pQO0XGqTj/R7fP/jzewr5L8QeINQ8UanLf6lcNcXEh5JPygegHYUAeo+L/wBpHWtUeSHRIV0m25HmsBJMffkYX8q8u1LxHquszGW+1G6uZP7zzMf61nUUAH1GT70sbtHJuVmX/dOP5UlFAHU+Hfid4m8MSKbLVp/KB/495m8yM/8AAT29uvvXtfgb9pCx1SSO08QwDTJmO0XUfMJP+0Oq1810f/q/+tQB9+Q3Uc8KSxOssbAFXQ5Vh6g96mXkV8jfCf4w3fgW6SxvGe50SR/njY5MOerp/UV9XafqUGpWNvdWsgmtpoxJHIhyCtAFyikpaACmsuTmnUUARkUuOMCnYopARNCJMgjI9DXKeJvhV4Z8WjOoaVbvL2mRdjj6MMGuwxRtFYVaFOtHlqRTRtSrVaEuelJp+R87eIv2S7WbdJomryQN1EVym8fmMV5h4g/Z/wDGOgsT9gW/iH8do4b9Dg19rso9KYVzXy2K4Wy/EO8Y8j8j63CcWZnhrKU+ZeZ+eGo6LqGkybL2xuLN+gFxEyZ/MCqROD/Ov0TutKtL5ClxbRTKRgq6AiuS1b4LeDdXy02h28Z/vQDyz/47ivmcRwZNa0Ki+Z9XhuOk9MRS+4+GqSvrbUv2WfCl4WNrNe2TnptkDgfnXM337Iy8/ZNfZT286AH+RFeHU4WzGnsk/Rn0FHjLLKm8nH1R84Yor3K7/ZO1+P8A1GqWk/uysuf51lS/sveMo87Dp8g9pmH/ALLXBLIcxjp7Jnpx4myuf/L5HkVFern9mXxqv/LG0P0n/wDrUf8ADM3jQ/8ALG0H1n/+tWX9iZj/AM+X9xuuIMrt/Hj955RRXr0P7LvjKVhuawiHfdOx/kta1p+ybrsw/f6vZweu2Nm/wrSGQ5jJ2VJnPPibKo71keF0n8/TNfSFn+yPHgfa9fkJ/wCmEAH8ya6nS/2W/ClptN1JeXhXs0u0H8AK9KjwrmFR+8kvVnmVuMstp/A3L0R8j9Kvadoeo6xJssbC4um9IY2b+Qr7Y0n4M+DtFAMGiW7MOjTL5h/8ezXW2um2ljGEt4I4UHRUUAfpXt4fgyW9ep9x4GI47W2HpfefG2gfs9eMtc2l7FdOjP8AHdOAfyGTXp3hz9kyyiCvrWrS3Ld4rdQg+mTz/KvoQL705VHNfT4XhfL8PZyXM/M+TxfFmZ4q6UuVPscl4Y+FvhrwkAdO0uGJ/wDnqy7nP1Y811Qj24wAB7VJ06UbRX1FKhSoR5aUUl5HydWtUrS5qsm35jNtLtP+TTsClrcxGKpzT6KKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooASsjxR4it/C2h3mqXbYhtoy+O7H+FR7k8Vr+tfPf7T3io79P8PQyHGPtVxtOO5CA/+PGgDxnxV4ou/F2u3WqXr7pJmJVeyJ2Ue2MVj0UUAFFFFABRRRQAUUUUAHp9eK9v/Z1+IrWOoDwxfSZtbg5tWY/ck7oPYjP449a8QqazvJdPvIrqF2jmicOrr1UjkH86APvtWJxTqw/B/iBPFHhnTdUTj7TCHZf7rdGX8DmtugBaKKKACiiigAooooASjbS0UAN20badSUAJ+NJt9KfRQAzp7UYp20UbaAEC0badRSsgGYFDKO4FOxS0wGKoFB+lOxRQAm0MtLtHpRS0gE2gUUtFMAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAGnvXxh8YtUbWfiNrcp5WOYwLz2QBR/KvtBuhr4X8aMzeLtaLdTeS5/76NAGLRRRQAUUUUAFFFFABRRRQAUUUUAfUv7M+qG98CT2rMWNpdMoz6MAf8a9erwf9lVm/snXx/D9ojx9dpr3egBaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBK+KPippr6X8RNdt2GB9paRcdcNyB+tfa/XNfNX7TnhdrTXrDXY1xFdoIJWA/wCWi9M/Vf8A0E0AeJUUenp2ooAKKKKACiiigAooooAKPX6cUU6ON5pFjRSzOQFA6knjA/T86APp39mPT2tfBd9dkY+03Z2+4VQP6mvY+lc58PfDn/CKeD9K0wjEsUIMv/XQ8t+p/SukoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArmPiF4Nh8ceFb3S3wski7oZD/BIOVP07fQmunprDKmgD4I1LTbjR9RubK6ieG5gkMbxycEEf0qrX1J8a/g//wAJlbvqulRqusQph4xx9pUds9m9+9fL9xby2txJDPG0UsZKujDBUjrxQBHRR/npRQAUUUUAFFFHTr0oAPUd69Z/Z/8Ah+3iTxEus3Uf/Et05gVLDiSXqAPp1P4Vynw6+HGofELVVigUxWMZ/wBIuiMqinsPUn0FfYHh3w/ZeF9HtdO0+IRW0K7VHc+rH3NAGoo/OloooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigApKWigBjLmvPviN8HNI8fK1xgWGqgfLdxoPmPo47/XrXolJQB8Y+LvhD4m8HM7XFg91aKeLm2BdcepxyPxH41xbKQ2DkH0xiv0AZd3Fc9rHw78N68WN9otnOzdXEYVvzGKAPiDn/ACKOvr+VfXk3wB8FytkaY8Y/upM2P61bsfgh4MsWDDRI5XHeV2b+tAHyHpulXur3CQWNpNeTMcLHChb+Vew+Bf2br/UpEuvEj/YbXr9liIMrj0JBIA/WvonTtHstJhENlZ29pF/dgjCD9KugfnQBnaLodl4d0+Oy0+2S2to+FSMY/E+p960RxRS0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAJRS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//2Q==)

**Kaggle:** is a leading online platform for data science and machine learning, offering a collaborative environment for data enthusiasts, researchers, and professionals. Acquired by Google in 2017, Kaggle provides tools and resources to explore datasets, build models, and compete in machine learning competitions.

Kaggle hosts a wide range of **competitions** where individuals and teams solve real-world problems using data science techniques. These competitions are often sponsored by companies and organizations seeking innovative solutions. In addition to competitions, Kaggle offers a vast **repository of datasets**, public **code notebooks**, and **discussion forums** to foster learning and collaboration.

The platform supports languages such as Python and R and integrates seamlessly with popular libraries and tools like TensorFlow, Scikit-learn, and XGBoost. Kaggle is also widely used for practicing data analysis, participating in courses, and developing portfolios for career growth.



**Hugging Face:** is a prominent open-source platform and company at the forefront of natural language processing (NLP) and artificial intelligence (AI). Known for its transformative impact on the AI community, Hugging Face provides tools, libraries, and models that make state-of-the-art machine learning accessible and easy to use.

The platform is best known for its **Transformers** library, which offers pre-trained models for a wide range of NLP tasks such as text classification, translation, summarization, question answering, and more. These models, built on architectures like BERT, GPT, T5, and others, are widely adopted in both academic research and industry applications. Hugging Face also features the **Hub**, a collaborative repository where developers and researchers can share models, datasets, and demo spaces. In addition, it offers **Inference APIs**, **Auto Train**, and other tools that simplify deploying and fine-tuning models in production environments.

A yellow smiley face with hands on it

AI-generated content may be incorrect.

**Spyder** (Scientific Python Development Environment): is a powerful, open-source integrated development environment (IDE) specifically designed for scientific computing, data analysis, and Python programming. Often referred to as the "MATLAB for Python," Spyder combines the power of an IDE with the flexibility of interactive computing. Tailored for data scientists, engineers, and researchers, Spyder provides a user-friendly interface that includes features such as a multi-language editor, an interactive console, a variable explorer, and integrated debugging tools. It supports popular scientific libraries like NumPy, SciPy, Matplotlib, Pandas, and SymPy, making it ideal for scientific workflows. Spyder is part of the **Anaconda** distribution, which simplifies package management and deployment. Its rich plugin architecture and customizable layout enhance productivity, especially for users working on data-heavy or computational tasks.



**Chapter 6**

**System Implementation**

# **System Implementation**

## **6.1** Mobile Application Design

* **Splash Screen:**

"Snap & Shape" Logo

A logo for a company

AI-generated content may be incorrect.

* **Onboarding Screens:**

1. Welcome to Snap & Shape…

**Description:**

Scala, track your meals, receive personalized food recommendations, and enjoy tailored exercise plans just for you.

A screenshot of a cell phone

AI-generated content may be incorrect.

1. Smart Food Recognition

**Description:**

Snap a photo of your meal, and we will do the rest!

Get accurate measurements and analyze the quality of your food effortlessly.

A screenshot of a phone

AI-generated content may be incorrect.

1. Track Your Progress

**Description:**

Keep an eye on your daily calorie intake and track your weight changes.

A screenshot of a cell phone

AI-generated content may be incorrect.

* **Sign In Screen:**

**Description:**

This screen allows existing users to log in by entering their email and password. It features a clean layout with easy access to password visibility toggle, a link to reset the password, and a prompt to create a new account if the user doesn't have one.

A screenshot of a sign in form

AI-generated content may be incorrect.

* **Sign Up Screen:**

**Description:**

This screen enables new users to register by entering a username, email, and password. The interface is user-friendly, with clearly labeled input fields and a "Next" button to proceed with account creation.

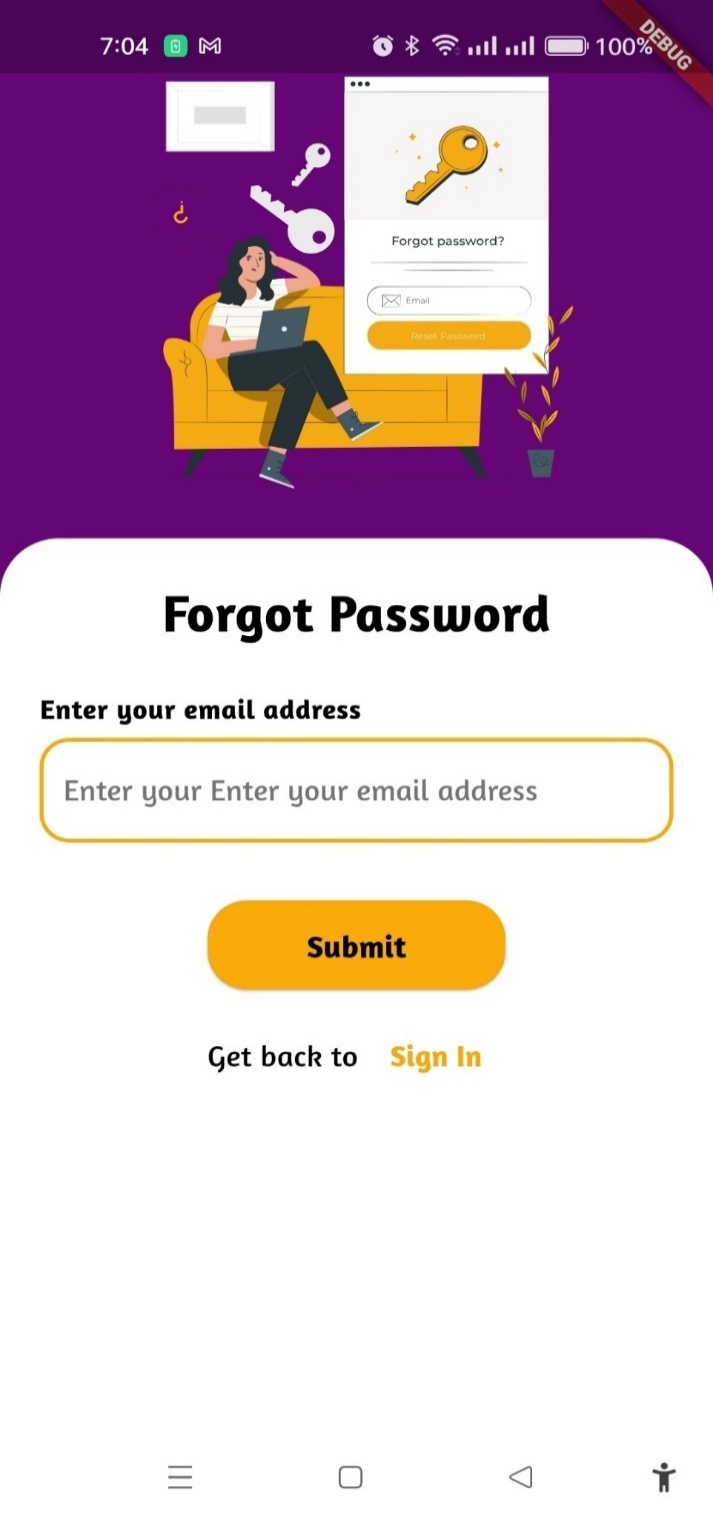
A screenshot of a sign up form

AI-generated content may be incorrect.

* **Forgot Password Screen:**

**Description:**

If a user forgets their password, this screen allows them to enter their registered email address to receive a password reset code. A "Submit" button sends the request, and there's a quick link back to the login screen.



* **Verification Code Screen:**

**Description:**

A screenshot of a purple box with white text

AI-generated content may be incorrect.After registration, users are prompted to enter a 4-digit verification code sent to their email. The screen includes a "Verify" button and an option to resend the code if not received, ensuring a smooth and secure onboarding process.

* **Reset Password Screen:**

**Description:**

This screen lets the user enter a new password and confirm it after verifying their identity. It includes password visibility toggles and a prominent "Reset Password" button to finalize the process, along with a backlink to the login screen.

A screenshot of a login screen

AI-generated content may be incorrect.

* **Questionnaire Section:**

**Description:**

This screen lets the user enter a new password and confirm it after verifying their identity. It includes password visibility toggles and a prominent "Reset Password" button to finalize the process, along with a back link to the login screen.

A screenshot of a quiz

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect. A screenshot of a quiz

AI-generated content may be incorrect.

A screenshot of a screenshot of a medical survey

AI-generated content may be incorrect. A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a quiz

AI-generated content may be incorrect. A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect. A screenshot of a cell phone

AI-generated content may be incorrect.

* **Home Screen: "MY Progress Screen"**

**Description:**

The Home Screen is divided into four main sections: Breakfast, Snacks, Lunch, and Dinner. Each section displays the suggested or target calorie intake and tracks the calories consumed for that specific meal.

Each meal section includes a "Scan Meal"button that allows the user to take or select a photo of their food. The image is sent to the backend for analysis, and the calculated calories are automatically added to the corresponding meal section.

A general "Camera" button at the top allows the user to scan any food item and view its calorie content without saving it to a specific meal.

The screen also shows the total daily calorie intake, calculated by summing the calories from all meals.

The "Analysis" button opens a separate screen that predicts the user’s future weight based on their recent calorie consumption trends, offering insights into their progress.

A screenshot of a cell phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect. A screenshot of a food order

AI-generated content may be incorrect.

A double cheeseburger with sesame seeds

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

* **Food Recommendation Screen:**

**Description:**

The Food Recommendation Screen provides personalized meal suggestions based on the user’s questionnaire responses. It displays a curated selection of meals for \*Breakfast, \*\*Lunch, and \*\*Dinner\*.

**Each meal item includes:**

1- A representative image

2- A list of ingredients

3- The total calorie count

This screen helps users make informed and healthy choices by offering meals that align with their dietary preferences, goals, and nutritional needs.

A screenshot of a food menu

AI-generated content may be incorrect. A close up of food

AI-generated content may be incorrect.

* **Exercise Recommendation Screen:**

**Description:**

The Exercise Recommendation Screen is divided into three general sections that anyone can benefit from:

1. Weight Loss Training

2. Warm-Up Training

3. Stretching Exercises

Additionally, the screen offers personalized exercise routines tailored to each user based on their questionnaire responses. These personalized routines include targeted workouts for the Upper Body, Abs, Legs, and Chest and Back, designed to match the user’s specific fitness goals and conditions.

A screenshot of a cellphone

AI-generated content may be incorrect.

* **Profile Screen:**

**Description:**

The Profile Screen provides each user with a personalized view of their account information. It includes the user’s name, email address, and other personalized details. Additionally, it features a Logout button for users who wish to sign out of their accounts.

A screenshot of a phone

AI-generated content may be incorrect.

**Chapter 7**

**Conclusion and References**

# **Conclusion and References**

## Conclusion

In conclusion, the "Snap & Shape" project successfully developed a smart, user-centric mobile application that leverages artificial intelligence and mobile technologies to enhance personal health management. The system integrates real-time food recognition, nutritional analysis, and personalized fitness recommendations in a seamless cross-platform experience.

The application was thoroughly tested for accuracy, performance, and user engagement. Results indicate that the AI models used for food identification and portion estimation are highly accurate, and users found the app intuitive, helpful, and motivating for improving their dietary habits. The backend's real-time data integration via web scraping adds dynamic value by continuously updating suggestions with the latest health insights.

This project contributes meaningfully to the field of digital health by offering a practical solution that empowers individuals to take control of their lifestyle choices. It highlights the powerful role technology can play in preventive health care by facilitating early intervention and promoting sustainable wellness practices.

Moreover, "Snap & Shape" showcases the value of interdisciplinary collaboration—combining expertise from mobile development, machine learning, and health sciences to address a real-world issue. The project sets a precedent for future initiatives aiming to blend technology with health advocacy.

The team hopes this work inspires further innovations in AI-driven health applications and contributes to the global movement towards more personalized, proactive, and accessible healthcare solutions.

## **7.2** References

1. <https://pub.dev/>
2. <https://docs.flutter.dev/>
3. <https://learn.microsoft.com/en-us/answers/questions/1282826/web-scraping-using-c-api-method> .Last Visit (2025/2/18)
4. <https://learn.microsoft.com/en-us/answers/questions/990238/scrapping-and-automation-in-asp-dot-net> . Last Visit (2025/3/4)
5. <https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/how-to-guides/use-sdk-rest-api?view=doc-intel-4.0.0&tabs=windows&pivots=programming-language-csharp>
6. <https://keras.io/api/>
7. <https://github.com/ultralytics/ultralytics#models>
8. <https://en.wikipedia.org/wiki/Recurrent_neural_network>
9. <https://en.wikipedia.org/wiki/Convolutional_neural_network>
10. <https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification>
11. <https://www.kaggle.com/datasets/sriramr/fruits-fresh-and-rotten-for-classification>
12. <https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition>