|  |  |
| --- | --- |
| Icono  Descripción generada automáticamente | Universidad Nacional de La Matanza |
| DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS |

**ELECTRÓNICA DE POTENCIA**

CONTROL DE ENERGÍA POR TIEMPOS DE OPERACIÓN

**MANUAL TÉCNICO**

Profesores: Ing. Guillermo Luis Miquel

Ing. Oscar Pugliese

Alumno: Nicolás Enrique Agostino

Ignacio Luis Mehle

Santiago Ruiz

Federico Ladislao Sokolic

Pablo Fabián Yujra Ventura

Revisión 001

**Índice**

[1. Descripción 3](#_Toc139229437)

[2. Hardware 3](#_Toc139229438)

[2.1. Diagrama en bloques 3](#_Toc139229439)

[2.2. Descripción de partes 4](#_Toc139229440)

[2.3. Circuito esquemático 8](#_Toc139229441)

[2.4. Circuito impreso 9](#_Toc139229442)

[2.5. Gabinete Plástico 9](#_Toc139229443)

[2.6. Montaje y ubicación de componentes 10](#_Toc139229444)

[2.7. Costos 11](#_Toc139229445)

[3. Consumos 11](#_Toc139229446)

[4. Firmware 12](#_Toc139229447)

[4.1. Código 12](#_Toc139229448)

[4.2. Guardado de datos en memoria 53](#_Toc139229449)

[5. Software 54](#_Toc139229450)

[6. Foto producto 63](#_Toc139229451)

[7. Video de Funcionamiento 64](#_Toc139229452)

**Control de Energía por Tiempos de Operación - Manual Técnico**

# 1. Descripción

El dispositivo tiene como principal función la conexión y desconexión de forma automática de un artefacto o carga eléctrica monofásica siguiendo un programa semanal configurable, respetando los feriados según el calendario anual vigente. Este proyecto fue desarrollado a lo largo de la cursada de la asignatura “Electrónica de Potencia”, en el marco de impulsar una mejora en el consumo sustentable de la Universidad.

# 2. Hardware

El eje central del dispositivo es un microcontrolador ATMEGA328P, encargado de coordinar cada una de las tareas del proyecto. Lo rodean diferentes bloques/módulos que le brindan al equipo funcionalidades para cumplir con su objetivo. Estos bloques se irán definiendo en particular en la sección “Descripción de partes”.

## 2.1. Diagrama en bloques

Diagrama

Descripción generada automáticamente

**Ilustración 1: Diagrama en bloques**

## 2.2. Descripción de partes

* *ATMEGA328P*

Es un microcontrolador que pertenece a la familia de los microcontroladores AVR de arquitectura RISC-8-bit. En formato DIP, es el que se utiliza en la tarjeta Arduino Uno R3. Posee características como:

* 32 KB de memoria FLASH
* 1 KB de memoria EEPROM
* 2 KB de SRAM
* 23 líneas de I/O de propósito general
* 32 registros de proceso general
* 3 temporizadores flexibles
* Interrupciones internas y externas (timers, adc, ext\_int, etc)
* Interfaces UART, I2C e SPI.
* Conversor A/D de 10 bits
* Frecuencia máxima de funcionamiento de 20MHz.
* Voltaje de alimentación de 3,3V a 5V DC.

Imagen en blanco y negro

Descripción generada automáticamente con confianza baja

**Ilustración 2: ATMEGA238P**

* *Fuente Switching 220VAC – 5VDC*

Para la alimentación del dispositivo, dado que la mayoría de los bloques requerían una tensión constante de 5V, se utilizó una pequeña fuente switching modelo WX-DC12003. Esta permite una sencilla implementación, conectando la alimentación monofásica en su entrada y obteniendo una alimentación de 5V constantes en su salida, con una variación de una centésima de volt. Posee las siguientes características.

* Tensión de entrada: 50-270VAC
* Tensión de salida: 5V DC (±0.3 V)
* Corriente de salida: 700mA
* Potencia de salida: 3,5W
* Ripple: <40mV
* Protección contra cortocircuito/sobretensión/sobrecorriente
* Dimensiones: 23,5x18,15x13,5mm

Una caja de cartón

Descripción generada automáticamente con confianza baja

**Ilustración 3: Fuente Switching**

* *Pantalla LCD 16x2*

Para la visualización del menú, se colocó una pantalla LCD 16x2, la cual es comúnmente usada en proyectos de electrónica y su manejo es conocido. Junto con el encoder rotativo, integran la interfaz con el usuario del dispositivo. Permite un control del contraste, el cual es ajustado con un preset desde la placa principal. Como el proyecto consiste en un control de la energía, se agregó como funcionalidad del diseño el apagado del backlight de la pantalla cuando se cumple 1 minuto de inactividad. De esta manera el consumo del dispositivo se reduce (ver sección 3 del informe). La pantalla, de la marca DUAITEK, posee las siguientes características:

* Pantalla de 16 caracteres, 2 líneas
* Caracteres de 5x8 puntos, color blanco.
* Permite mostrar letras, números, caracteres especiales y hasta 8 caracteres personalizados por el usuario.
* Backlight color azul
* Interfaz paralela
* Alimentación 5V

Imagen que contiene reloj

Descripción generada automáticamente

**Ilustración 4: Pantalla LCD 16x2**

* *Encoder Rotativo*

La segunda parte que integra la interfaz con el usuario, es el encoder rotativo. Este dispositivo permite que usuario pueda ir atravesando las diferentes pantallas que se presentan y de esta manera realizar una correcta configuración del dispositivo. Se utilizó un encoder para simplificar y reducir la cantidad de partes del proyecto, ya que con este se puede generar movimientos hacia la derecha, izquierda y confirmaciones pulsando el switch que incluye. El modelo utilizado es el KY-040 y posee las siguientes características:

* Tipo encoder incremental
* Ciclos por resolución (CPR): 20
* Tensión de trabajo: 0-5V
* Peso: 10g
* Dimensiones: 32x19x30mm

Un circuito electrónico

Descripción generada automáticamente con confianza baja

**Ilustración 5: Encoder rotativo**

* *Módulo RTC (Real Time Clock)*

Para el proyecto era necesario tener como información la fecha y hora actual, así como también la posibilidad de configurarla la primera vez que se utilizaba. Para ello se utilizó un módulo DS3231, el cual es un reloj en tiempo real (RTC) extremadamente preciso y de bajo costo, con un oscilador de cristal con compensación de temperatura (TCXO) y un cristal integrados. Este dispositivo incorpora una entrada de batería, para cuando sea necesario desconectar la fuente de alimentación principal y continúa manteniendo un cronometraje preciso. El oscilador integrado mejora la precisión a largo plazo del dispositivo. RTC mantiene información de segundos, minutos, horas, día, fecha, mes y año. Menos de 31 días del mes, la fecha de finalización se ajustará automáticamente, incluidas las correcciones por año bisiesto. El reloj funciona en la indicación de 24 horas o de banda AM/PM del formato de 12 horas. Proporciona dos despertadores configurables y un calendario que se puede configurar en una salida de onda cuadrada. La dirección y los datos se transfieren en serie a través de un bus bidireccional I2C. A continuación, se enumeran algunas características:

* Tensión de Alimentación: 3,3V – 5V
* RTC de alta precisión DS3231 con oscilador interno
* Exactitud Reloj: 2ppm
* Dirección I2C del DS3132: Read (11010001) Write (11010000)
* Memoria EEPROM AT23C32 (4K\*8bit = 32Kbit = 4Kbyte)
* Interfaz I2C
* La batería puede mantener al RTC funcionando por 10 años.

Imagen que contiene electrónica, circuito

Descripción generada automáticamente

**Ilustración 6: Módulo RTC**

* *Módulo microSD*

El dispositivo necesita conocer la fecha de los días feriados, así como también la programación semanal con los rangos de trabajo por día. Esto puede configurarse de forma manual, pero teniendo en cuenta una posible reproducción e implementación de este prototipo, aparecía un problema a la hora de la programación de cada uno de los dispositivos, ya que debían ser configurados en forma individual utilizando el encoder rotativo. Esto podría llegar a resultar molesto, pero sobre todo demandaba mucho tiempo. Es por eso que se incluyó un módulo para tarjeta microSD, el cual se cargará con un archivo .txt que contiene los feriados anuales y la programación semanal.

El módulo utilizado cuenta con las siguientes características:

* Soporta tarjetas microSD, microSDHC
* Posee un conversor de nivel para utilizarlo con 3,3V o 5V
* Posee interfaz de comunicación SPI
* Tensión de Alimentación recomendada de 5V
* Dimensiones: 42x23x12mm

Un circuito electrónico

Descripción generada automáticamente con confianza media

**Ilustración 7: Módulo microSD**

* *Módulo Relay*

La conexión y desconexión del artefacto a controlar estará a cargo del módulo relay, el cual cuenta con transistores para permitir el comando del relay mediante un microcontrolador. Dispone de un contacto simple inversor y la placa presenta un LED indicador de alimentación general. Otras características son:

* Tensión de alimentación: 5V
* Consumo de relay: 50-60mA
* Capaz de manejar una carga de hasta 250VAC-10A/30VDC-10A
* Lógica TTL
* Posee un LED indicador de estado

Imagen que contiene electrónica, circuito

Descripción generada automáticamente

**Ilustración 8: Módulo relay**

* *Módulo Interlock*

La alimentación del prototipo será monofásica, para lo cual se empleó un módulo interlock que cuenta con fusible y llave incluidos. De esta manera, utilizando un cable interlock se puede realizar una conexión rápida y segura del equipo.

Una cámara de video

Descripción generada automáticamente con confianza baja

**Ilustración 9: Módulo interlock**

* *Tomacorriente*

El lugar para la conexión del artefacto a controlar será en el lateral del dispositivo, a un costado del módulo interlock. Se utilizó un tomacorriente normalizado de 220V-10A de color blanco, embutido en el equipo.

Imagen que contiene reloj

Descripción generada automáticamente

**Ilustración 10: Tomacorriente**

## 2.3. Circuito esquemático

El diseño del circuito esquemático fue realizado utilizando el software KiCad versión 7.0.

![Diagrama, Esquemático

Descripción generada automáticamente](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4RD0RXhpZgAATU0AKgAAAAgABAE7AAIAAAAOAAAISodpAAQAAAABAAAIWJydAAEAAAAcAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFNhbnRpYWdvIFJ1aXoAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzEwAACSkgACAAAAAzEwAADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIzOjA3OjAyIDE1OjU1OjI0ADIwMjM6MDc6MDIgMTU6NTU6MjQAAABTAGEAbgB0AGkAYQBnAG8AIABSAHUAaQB6AAAA/+ELIGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjMtMDctMDJUMTU6NTU6MjQuMTAyPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPlNhbnRpYWdvIFJ1aXo8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgCqgP4AwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+haKKKgYUUUUAFFFFABRRRQAUUUUAMmEjQuIGVJSpCM6llVscEgEZHtkfUVw/gPx3e+IvEXiDQ9Zt4IbrSrl1t5YUaNbqFZXjLhWLdGTBIJGTjtXd15Dq8beG9JtPHdvHJ5mj63qEd9HGvzT2ct/KrADjJBKsuSB1Nejg6cKsJU5L3nZRfZ6tffa3zIk2nc6T4geObvw3q2gaPootmvtVv4beV7iFpUt4pGKhiqspySDtGedj+nHUarcalZ6Cz2KWtzqWEjQSt5MLSMwXJySQuTnGSewya8r8Qqb/T/CPiS5iljudb8YWM6rMAGjth5gt0wM4/d4YjJ+eRz3rsLHUJm+Nmq2OokqF0mFtNXcQDGXPnEDoSX2An0UV01cNCNKHKtYpt+bTSa9NfuVxKWp2UIlFvGLhkeYKPMaNSqlsckAk4Ge2TWINZuf+E8h0lbi0ms57Ce5CpEwliaKSKPBfeVYEu/G0EFa8svdQu9A0fULRJ2/sCPxtFbS7+FW0bDzQ4PHlhsLjoQWHQmvS7m2iX4raXeJzJNol5GzZ4KrPbEf+htWU8IqLvJ3TUradlf5P8h81zp6x5tSvIvG1npf7g2Vzp09x9w+YJIpIV65xtIm6Yzkda8i1Oyk0v8AZKu7C4XEtpO8LgjHzLqRU/qK9Sv5Uj+Jehq7YMml6gqD1Pm2hx+QNTLBqk3rzK9SO38qTv8AO/yDmv8AgdCZEEqxl1DsCyrnkgYyce2R+Yp1eQabqn9rfA3xjc6wduqJJfm9VmOY7gZMYAPK7QIgvptFejeEJLmXwPoUl8zvdPp1u0zOcsXMa7iT65zWOIwjoRbb2ly/he6/r8xqVzk7bxH4zvvFfijRrS60EHQFt2VprGZRcebEXwT552YxjOG9cdqteFfGmreP/B9jqvh62t9Nme8aC7N5GbhIlWNiWUK8ZbLbB14yeDjNctc6VrWq/ET4mJ4d1KS1uki0/wD0YIhjvAbY/u2YjemQCAyMpG4n0x3Hw1udCn8E28fhmyfTrW3kkiksZXLSW0ocl0ckk5yc89iPpXo4qFKnR54xTl7my2vBN373but1o/QiN2/vMvR9d8Yat408RaB/aOhxf2J9m/f/ANlTN53nRl/u/aflxjHU59q0J/FmpLqGk+G4LW2fxNdWK3V7kn7PYrgB3IDZcbzhUDc92A5rP8If8lm+Iv8A3DP/AEnasy3s20b9pu6vb1ikGt6PstHc/K0iGIMg9wIyce/vSdOnKpJNLSCklbd8sW9vVv5dgu7fM62TUPEOiXtkdZaw1KwurhbeSeytZLd7VnO1GKGSTerOVU8rtznkZx01Vru/trFoFupQjXEohhXBLSOewA5PAJPoAScAE1Zrx6kuZJuNv1/ry/PfRBRRRWQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACisI+OPCYm8o+J9GEmcbDqEWc+mN1SXHjDwzaXj2l14i0mC5RtjQyX0aurZxgqWyDmtvYVv5X9zFdGzRWTf+KvD2lXZtdU17TLK4UAmG4vI43GRkcMQehpl14w8M2Mix3viLSbd3RXVZb6NCysAQQC3Qggg+hpKjVdrRevkF0bNRG2gMDwGCMxSFi8ewbW3Elsjvkkk+uay5/GHhm2hhmufEWkxRXC74Xkvo1WRckZUluRkEZHoaG8YeGUskvH8RaStq7lEnN9GEZhjIDbsEjI49xTVGt0i/uYXRoXWn2V8IBe2kFwLaVZ4BNEr+VIv3XXI4YZOCORVbVPD+ma1NbTahbb57Ri1vPHI0csRIwdroQwyOuDzTB4p8PnTDqQ1zTTYiTyjdfa4/K34zt35xnHOM1GnjDwzJZyXcfiLSXtomCSTLfRlEYgkAtuwCQD+Rqowrxd4p6eoaF260nTr7T5rG8sbea0nJaWB4gUck5JI6Ek859eaz7Xwb4es/sXk6XCTYOXtTLmQxMQo3AsTyAiAH+EKAMAYp9v4w8M3Uc0lr4i0mZIE3ytHfRsI1yBliG4GSBk9yKfZeKfD+p+d/Zuu6Zd+RGZZfIvI5PLQdWbB4A9TxTSxME0uZLruHulabwN4ZnudRuJtGtml1Pb9sYqf32GDYPtlQSO+Oc1o2+iaVafZPsumWcP2JXW18uBV8gP94JgfLnvjr3qpbeMPDN5M0Nn4i0m4lVWdkivo2YKoyxwG6AAk+gFJbeM/C95dx2tp4k0ie4kYIkMV9Ezsx6AANkn2pyWKatLm09e1vy09Be6LeeEtC1DUnv7zTo5Z5VVZgzNsmCnK+Ymdr4PI3A47VsAAAADAHQCsM+OPCYm8o+J9GEmcbDqEWc+mN1SXHjDwzaXj2l14i0mC5RtjQyX0aurZxgqWyDmplTxErKSbttuO6NGGws7e9uby3tIIrq62/aJ0jCvNtGF3MOWwOBnpTLPS9P0+e4msLG2tZbp/MuJIYVRpmyTuYgfMcknJ9TVS/8AFXh7Srs2uqa9pllcKATDcXkcbjIyOGIPQ0y68YeGbGRY73xFpNu7orqst9GhZWAIIBboQQQfQ0lTryWievrrYLo0YbCzt725vLe0giurrb9onSMK820YXcw5bA4GelQavoum69Ymz1iyhvIM7gkq52nswPUH3HNVZvGHhm2hhmuPEWkxRXC74Xkvo1WRckZUluRkEcdwakn8U+H7awgvrnXdNhtLnPkXEl5GscuDg7WJwcH0oUK6kpJO/TfoGhNp+i2OmKgto5GMe7Y9xO87oGxuVWkJKqdq/KDjgcVfrGPi/wANLZpdt4h0oWzuUSY3sexmGCQG3YJGRke4pV8W+HHs2vE8QaW1sjBWnF7GUUnJALZxk4P5GiVOvJ3lFv7wujYorLh8TaFc2M17b61p0trb486eO7RkjzwNzA4Gfeiz8TaDqKytp+t6ddCFDJKYLuN/LQdWbB4A9TUeyqK/uvTyC6NSisi28W+HLyRks9f0u4dFZ2WK9jYqqjJJAPQAEn2FNtvF/hq9uktrPxDpVxO7BViivY2ZiegADZJp+xqr7L+4Lo2aKyIPFvhy6u1tbXX9LmuGYKsMd7GzljwAFBzmibxb4ct7x7S41/S4rlGKPC97Grqw4IKk5Bo9jVvblf3BdGvRWNceL/DVpdva3fiHSoLhGKPDJexq6sOoILZBp1x4t8OWcyw3ev6XBKyqypLexqxVhkEAnoQQR7Uexq/yv7gujXorLvPE2g6d5P8AaGt6da+egki8+7jTzEPRlyeQfUVJDr+j3FxbQW+rWMs12pa3jS5RmmAzkoAcsBtbp6H0peyqWvyu3oF0aFFFFZjCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKxLq6mTx5pVqsriCXTL2R4wflZlltQpI9QHbH+8azF1qXSoY5LWyvLyfWNbuLRIJ7zcIWQSjcpIO2Mi33bR93e2AcYPRHDylFNdf+D/k2K511FcRZfEaO+W3sYbKE67PqF3p4shct5QktgWkPm+XnbtCkHZnLqMdSNSXxNe/bLPTLfSUGs3FpJeSWdxdhFijR0Q/OivkkuNvABAOSpGKqWErRdpK3zXS+vpo9dhcyOjorkbTxtNqws7XSdNj/ALWuILieW0vbrylgEEwhkDOiPk7yQMLghWORwDy/wu8U383w/wDD+n2UUOpazeQ3l9L9su2hURrdspYuEcli0gwNuDhuRgA6fUK3s5TfRpW0/va77Ll+YcyvY9Worn4vEd5d3yWtlo7+aLeyup47mYRPDHO8itkYILxiIkrnnkA8czazrd3pmtaNYW2npcpqkk0Jne48sQyJC0qAgKchtjDPbGcHpXP7CpzcvXfddFf8h3RtUVxr+PiseoXwsYX0nR7r7HqdylyxkhmATzNkfl4dELjLFlOFchTgbrVt4i1mbUtdi/siCWDTryOzt/JuG8yR3WJgzrswqBZgzMCSArfKcc6PCVkrtfivL/NfeLmR1FFeba58R9Ug07ThY6THDeXGuwaTNuuhIiM3lygoQuHSSJvvfKRuBxmpb3xg+j6145fSPDzXtzo/2Se7zqDL9pV4MllQqQhVUAIH3gM9Rg6rL67V7fiu6Xfa7/pahzo9Eorn73xO0Cm5tLWG709lto4blLoZeeeVURCu07VAkjctknDcKak0PXb3UNU1DTNX0xdPvbFYpCIrjz4pY5N211bap6xuCCoxiuZ4eooOdtF5ry6b9UO6NyiiisRhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABSc59vpS0UAFFFFAGP4SmubjwVok1+ztdSafbvMztli5jUsSe5zmtisXwdcXF34F0G4vXkkuZtNt3meQkszmJSxOe+c1tVrX0qyXmxLYKKKKyGFFFFABRRRQAUnOfb6UtFABRRRQAUUUUAFFFFABRWJ4l1O802KwFgVVrq78l5Gs5LnYvlSPkRxkMeUUZzgAk1mQ+LNQhv7XR9SsVTWLhrZlSNCYvLZVMp3ZwCCtxgE5Ij6HjNKLauZSqxi7M66iuX8D+Jb3xJp80t/AqNCIwzLA0I8xkDsgVmYkKGX5sjdngAcmPw/4h1i90zQ7rVoII21aYIESLZtQ28koYESPnOxRzg9cgdhxaFGtGSTXU6ykOeMfjxXntv8Sxf28skEtnaxwXUpluHjedRbBQyEJGc+YytnHYRSk42kD0OlKLjuOnVhU+FhRRRSNQooooAKKKKACsbW7Se51jw5LChZLXUXlmIB+VTaXCZP8AwJ1H41s1j61bXE+r+HpIFZo7fUXknIz8qG1uFBPtuZR+IrWi7Sfo/wAmJmxRRRWQwpDnjH48UtFABRRRQAUUUUAFFFFABRRRQAUUUUAFIc9uPqKWigAooooA53VNE1u58SwatpurWFsLa2ltoop9Oeb5ZGiZyWEy5OYVxwMAnOetMu/Deq6he6Pd3uuIZdNvmu2WG02JIChjCAbyVG1nySWJLHoMLVu61zULe5kih8LatdIrECWGW0CuPUbp1OPqAah/4SPVP+hM1z/v9Y//ACTXdGVeys46K32dn+PUnQ5iw+FFxp/nXNv4mnj1M6lLqVvepaITDJKhSYbHLBgykdeQQDmugvfCVzLqWl6taaw6avYRPA93cWySC5icglHRNg4IBGMY9+an/wCEj1T/AKEzXP8Av9Y//JNH/CR6p/0Jmuf9/rH/AOSa1nWxU5c0pRb9Ydenp5bddxWiiEeC4rfVtI1OwvHivLASpPK8at9sSVt8ocDABL5cFcAEnjHFZdt8Nf7I0vSIfDWtS6Te6bbyWpu0tY5fOikkEjqVfIHzjIPUc8nNbX/CR6p/0Jmuf9/rH/5Jo/4SPVP+hM1z/v8AWP8A8k0lVxS05l98fPfvu/vC0SO08M31r42k199aaeOWyjs5LaS3A3KmSG3AgBtzMeFAwxGO9WtZ0W71XVtJuY7+O2g026F0IxblnkbZJGw37wApSQjG3rzk9Kh/4SPVP+hM1z/v9Y//ACTR/wAJHqn/AEJmuf8Af6x/+Sazft3JSvG6Vt47bfkPQpz+CC8+o29vfJDpGrXgvb628hjM8nybgku8BUby1yCjHl8EZG1b/wAI39zc6vJa62LZL+9tr6KP7IG8qSL7P947gXBFsAANuN7Z3cYt/wDCR6p/0Jmuf9/rH/5Jo/4SPVP+hM1z/v8AWP8A8k1SqYlNPmX3x8v8l9wrRM7Vvh7BqdrcqmoSQ3EmtR6zDN5e4xSoiKBjIyMJ2IPP5z6J4Nk0nxpr+vzambpdbCCW1aDaqBBtTncc4XI6c+3SrX/CR6p/0Jmuf9/rH/5Jo/4SPVP+hM1z/v8AWP8A8k0/aYlwcHJWem8fLz8kFo3uQnwPZL4Hj8NQXl3FFF5JS8Bj8/dCyNGSdu0keWi9PuqBWrpumSW07Xt/PHc6hNbxwTzQw+UjiNpGUhCWI/1jfxH8Kof8JHqn/Qma5/3+sf8A5Jo/4SPVP+hM1z/v9Y//ACTWcvbyTi5LV33j/n5Ieh0FFZWi6/b60sqCC5sruA4ms7yMJNH6NgEgqezKSDzzkGtWuSUJQfLJalBRRRUgFFFFABRRRQAgzgZ5PfApaKKAMDU7XXru71OGC7jg0+SxKQeVDmbzWVwSrb1wVOw88HPUYJrm7LwpqSLbxwWMVlCbgQXf7qCBrmAGKQyOsJ2H5kljC/3ZTn+LPXeIodQu9JNnpLGKa6dYZLgMFNvET+8cf7W3IXGfmKk8ZNYnwyTU4fBVhHqcjXML2sFxa3LvuZkkiVmRu+VcsB22lOSQaaqNO1iXhITg6jl129TsKKKKRQUUUUAFFFFABRRRQAgzgZ5PfApaKKACiiigAooooAKKKKACiiigAooooAQZ78/QUtFFABRRRQAUUUUAFFFFABRRRQAUUUUANUMM7yDzxgYwKdRRQAUUUUAFFFFABRRRQAUUUUAFFFFACc59vpS0UUAFFFFABRRRQAUUUUAFFFFAGP4RuZbzwRodzcqqzTadbySKqBAGMakgKOAMnoOlbFZXha+Op+D9GvzDFAbqwgmMUK4SPdGrbVHYDOBWrWtb+LLS2rEtgooorIYUUUUAFFFFABRRRQAUUUUAFFcdrWseJZ9D1M6ZYy6XdR3KJayTwmZmj2hmO2MSZJIZc4xgjoal0PU9fudctLXUre4jtXsFvGmaDbljHEvkudoCuHMzFRg/c6DIquV2uYe2jzctmdZRRRUm41o0dkZ0VjGdyEjO04IyPTgkfiaoaveadpFo+q6jHxCUUSJAZJNxOxAoUFicyEDH94+prRqK4t4bu2kt7qGOeCVSkkUihldTwQQeCD6UtRrluubY52GTRdUuYtO1nwy1jM3MEOoWcTpJ8mMK6F4921Pubt21M4wM1tz6Tp1zZw2lzp9rNbW5UwwyQqyR7RhdqkYGBwMdKwfBeh2dpBdX/wBmX7Wb28t45W+Yw26XMixwxn+CMKq4QYA9K6qiMm1cdSNLm9zb+vwM+fQNHuofKudJsZo8qdklsjDK5C8EdtzY9Mn1rQoop3ISS2CiiigYUUUUAFFFFABWPrUtzHq/h5bZ3WOTUXW4CnAZPstwQD6jeEP1ArYrG1u4nh1jw7HAzqk+ovHMFJwyfZLhsH23Kp57gVrR+J+j/JiZs0UUVkMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAy9Z0f+0fKubSb7LqNrk21yBnbnqrD+JDgZH0IwQDRo+sf2h5ttdxfZdStsC4tic4z0dT/ABIex/A4IIrUrL1jR/7R8q5tJfsuo22Tb3IGcZ6qw/iQ9x+IwQDW0ZKS5J/J9v8Agf8ADrzXmjUorM0fVW1COSG7h+y6hbELcW5bOD2ZT/Eh7N9R1BA06zlFwdmMKKKKkAooooAKKKKACsfwgix+CNDRBhV063AHoPLWtisjwp/yJmi/9g+D/wBFrWb+Nej/AEIbfOl0s/0NeiiitCwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMfwjcreeCNDukhSBZ9Ot5BFGTtjBjU7Rkk4GccnNbFY/hGS3m8EaHJZRNDbPp1u0MTMGKIY12qSAASBgZwK2K1r/wAWXqxLYKKKKyGFFFFABRRRQAUUUUAFFFFAHF/EG915Y7O08Ozvp+bi3M9+Y9yjzLiOFIwD1zvZjjoEAP3xXT6ReXF/o9tc3to9lcug8+3frFIOGXPcAg4YcEYI4NU/FMYl0eBWyANRsW49ruI/0rZqF8TNZVYuCppar9QorndJm8Rapo1lf/2hpcX2q3SbZ/Z0h27lBxnz+evWrf2bxF/0FNL/APBbJ/8AH6lTbV0n+H+ZyKo5K6i/w/zNeisj7N4i/wCgppf/AILZP/j9I1t4jKkDVtMB9RpsnH/kenzP+V/h/mPnl/K/w/zF8NEHS5wCDjUL0H2/0mSteuL+Hdvdafp9+NR1BZ1uNUuYrcOAhdo5HVyBnqxjZtozgAn1rtKVFt01dWCipqmlNWfYKKKK1NQooooAKKKKACiiigArH1q5lg1fw9HEqlbjUXjlJQNhfstw2QT905VeR6kd62KytXvjaapoUIhik+2X7QlnXJjxbTybl9D+7x9GNa0vi2vo/wAmJmrRRRWQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooqOeeK1t5Li5lSGGJC8kkjBVRQMkkngADvSAQ28JuluTDGZ1QxrLtG4KSCVz1wSAcewqWsH/hOfCX/AENGi/8Agwi/+Ko/4Tnwl/0NGi/+DCL/AOKo5l3NfY1P5X9xvVSn1jTbWNnudRtIUQ4ZpJ1UKcsMHJ9Ucf8AAW9DWd/wnPhL/oaNF/8ABhF/8VXHWeqaTbvr63PiPwtdwalviiiOsrHmJpriQ7jtOD+/AwM9DzTTi92ZTp1k1aD+5npkUsc8KSwuskcihkdDkMDyCD3FPrlNF8X+GrHQbC0vfFmhyXFvbRxSumoxEM6qASMkcZBq7/wnPhL/AKGjRf8AwYRf/FUuZdzRUqrV3F/cb1FYP/Cc+E/+ho0X/wAGEX/xVa1lfWmpWaXenXUN3bSZ2TQSB0bBwcMODggj8KLpilTnFXkrFisjwp/yJmi/9g+D/wBFrWvWR4U/5EzRf+wfB/6LWofxr0f6GD/iL0f6GvRRRWhoFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBj+EBbL4I0MaeztaDTrcQNIu1inlrtJGTg4xxk/Wtiszw1Haw+E9Ji06cXFollCsEwTaJIwg2tjtkYOK060rO9WT82JbBRRRWYwooooAKKKKACiiigAooooAyPEv8AyCof+whZf+lUVa9ZHiX/AJBUP/YQsv8A0qirXrNfG/RfqZr+I/RfqZHhT/kTNF/7B8H/AKLWtesjwp/yJmi/9g+D/wBFrWvRT+BegUf4cfRBWLP4y8MWtxJBc+I9JhmicpJHJfRKyMDgggtkEHtW1XAeKvhkdc1k32kXek6UsiZmjk8PW100spZi0hdxnJyOPbPeqk2loddCNKUrVXZf15MzLC78HapDaXOoeLLW1lstYv51jj1kwiRGnuQpwsigZEoYOBkrxnaxr0fTtV07WLdrjSb+1voVfY0lrMsihsA4ypIzgjj3FeR6N8LtS1exkuP7c0WHZdXFvt/4ROybPlTPFu6DrszjtnHPWvXNO0rTtHt2t9JsLWxhZ97R2sKxqWwBnCgDOAOfYVFPm6o6cZ7JO0ZXevy+9I89ifxNcah4ls0XVokNveyW0rNNgyrPmDyyyAKCpxtRmBC9BXWaU1/B4v1K0n+1y2K2lv8AZpZVYoGUMHG48MxypJ6n8OIL/wATXmha1eDW4FXSY7fz4rmGP5m/eIrAjeThRIuTgE84HHNmw1q8n8Uf2fP5DW0to88TRr88bpIqvE5DspZd65x1J7Y56ndrY8KCjGVru9/8zfooorI7QooooAKKKKACsfWrlYNX8PRtCkhuNReNWYnMZ+y3Dbhg9flxzkYY98GtisfWpLdNX8PLPE0kkmousDBgPLf7LcEseORtDDHHUHtg60fi+T/JiZsUUUVkMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiuX0Tw/o19Z3Nxe6TY3Ez6hebpJbZHZsXMgGSRk8ColJppJGcpSUlGKOoorI/wCEU8O/9ADS/wDwDj/wo/4RTw7/ANADS/8AwDj/AMKV59l9/wDwAvU7L7/+Aa1eYa74hnv5vG+nwaut9p9tod2PICgNbzAMrKx8tfRgBubhST1Fdw3hLw4ylW8P6WQRgg2UfP6Vz/jLwb4YtfAuvXFr4c0mGaLTbh45I7GJWRhExBBC5BB71Mudo78FOMZWqLV2t16762Ojm8R6bb3V3BLJODZKWuZBaymOICMSHMgXbnaQcZzyO9TzazYwaqmmvKxvHVXEKRM7BGJAc4BwuVILHgcZIyM5mp+Gbi/l1UQ6tJbWup2zxy2wgRlMjRCLzC2NxwoX5QR93rUq6BM2uw61NeR/2gkC2zvFblUeLezMu0sSM5XBzkFAeQSp6LRPN5qt9v8AhvvNyiiipNwqG5uobSISXLiNGkSME92dgij8WYD8amqtqNjFqel3VhcbvKuoXhfaedrAg4/OgTvbQWG+tbhFaKZGDSvCuTjc6FgyjPUgo3/fJrh38RDw54QS5+1Q2+/xHcRSeZglom1GUPjP+zuOe2DXQR+EreOHQF+03G7RQgTEjhJQsbIMpu2g/NnOCcDGcGsvw54U8Palp95d6joOmXdzJquob5p7ON3bF5MBliMngAfhUzWvunRhZJJyrLRNbddGdkjpIgeNlZW6MpyDWV4U/wCRM0X/ALB8H/otabH4P8Mwrth8O6TGM5wtjGP/AGWtaKKO3hSGCNYoo1CoiLhVAGAAB0FTZuV2cri/aX6f8MPooorQsKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDH8IwxW3gjQ4LadbmGPTrdI5lziRRGoDDIBwRzyBWxWP4StjZeCtEtWaNzBp9vGWiYMp2xqMggkEcdcmtita2tWXqxLYKKKKyGFFFFABRRRQAUUUUAFFFFAGR4l/5BUP/AGELL/0qirXrI8S/8gqH/sIWX/pVFWvWa+N+i/UzX8R+i/UyPCn/ACJmi/8AYPg/9FrWvWR4U/5EzRf+wfB/6LWtein8C9Ao/wAOPogoorN13WBoWltfPZ3N2iMA622zKA/xEuygKO5zx1PAJF7G0YuTsin4P/5Adx/2FdR/9LZq3q4zw5rF3prQ6TqHh/UoJr6+u5438y2dRHJcSShm2SkgBXUE4xngEkjPZ0o7GleLVRvvcqLpWnI9y6WFqrXfFwwhUGbr9/j5up6+pptto+mWd9Le2mnWkF3NnzZ4oFWSTJycsBk5Iyc96u0VV2c/LHsFFFFBQUUUUAFFFFABWPrQtjq/h43LOJBqLm3CrkF/stxkHngbN/PPIAxzkbFZerQ2kup6G91P5UsV8z2ybSfNk+zTKV9vkZ25/u1rSdpfJ/kxM1KKKKyGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRUN01wls7WUUU04HyJNKY1P1YKxH5GuH1OLxHqHh7R7u6mntNTW3bNnbrcslzKyoV81oTH5R3Aj5squ48nFVGNzKpU5Nlc76isHw/Fqseo6iupNI0ETCG3MjE7wJJXDDPUeXJCpY8lkbOcZO9UtWZcZcyvYKKKKCgooooAKKKKACiiigArI8Nf8gqb/sIXv/pVLWvWR4a/5BU3/YQvf/SqWs38a9H+hm/4i9H+hqSyx28LzTyLFFGpZ3dsKoAySSegrL/4Svw7/wBB7S//AAMj/wAaXxX/AMibrX/XhP8A+i2rVobk5WQm5OXLH+t/MyG8W+HFUs3iDSwAMkm9j4/WvOtS8R2y6b47F34k0G4ttSsp2s47fU45HZ/KaNVCcEEosYI5+YEg84Hrtcp4v1nS77wV4ps7LUbS4urfS7sTQRTq8keI2B3KDkYPHPepknbVnoYOUotqSve3y1XqSXHiy7TxNLo9tpccp8428Mz3e1TIIFmO8BCVG1iARuyV6DNa+g6lJrHh+x1KW3Fs13CswiEm/aGGRzgZ4I7U2Pw/pkeuyawtr/p8n3pWdjj5VUkKTgHaijIAzirttbxWlrFbW6BIYUEcaj+FQMAflXQ3G2h58I1FJuT01/4BLRRRUmoUUUUAFYPg/wD5Adx/2FdR/wDS2at6uZ8O6jZaX4buLjU7y3s4f7W1BfMuJVjXP2yfjJIHape5tFNwaXdfqdNRRRVGIUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFUo9Y0ybUJ7GHUbSS8tl3T26zqZIhxyy5yByOvqKswzRXMEc9vIksUih0kRgyupGQQR1BHem4yW6AkooopAFFRG4hF0tsZoxOyGRYtw3FQQC2OuASBn3FS0AFFQwXVvdeZ9mnjm8qQxSeW4bY46qcdCPSpqGmtwCiqqalYyag9hHe27XiLue3WVTIo45K5yByPzq1Taa3AKKKKQBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZXhewm0rwho+nXYAntLGCCUDGNyxqp6cdRWrWP4Stp7PwVoltdo8c8OnwRyo/3lYRqCDnvkVsVrW1qyv3YlsFFFFZDCiiigAooooAKKKKACiiigDH8TMF0mEsQB/aFkOT/09RVqmaJZliaRBI4JVCwywHUgVwusWniKztdPj1XUhqFr5+nmaRkjVvtP2+3yBtVfkxnbxnltxPy4z7m80aHxxNq0+g6nLcWl2trJJqDQeXB5k6ILiMOxm25A2FRsGWwFyxrFys7nbTwSk3JSv6eW29t7na+EmDeCtEZSCDp8BBB6/u1rYrB8Df8AJPPDv/YKtv8A0Utb1XBWikcbpqm+RdNPuCkZQylWAIIwQR1pa5TWrbWj4vtriwtbqe1xaruW8McMQEsnnFoxKu5thQjKuDtxitErmc5OKulc3NO0Ww0ppGsYWRpDyXlaQqvZF3E7UHOEXCjJwBV+vN9O0bxZbx29rq41S+tbeSHfJBqWyWdTHcM53eapG15IlPzciMEZArvNIt7iz0Wxtr2Yz3MNvHHNKzFjI4UBmyeTkgnJpuKjsyYVpVXeSa9S5RRRUmoUUUUAFFFFABRRRQAVj6zFFJq3h9pZ0iaPUHaNGBJlb7LONo4ODglu3CnnsdisfWrU3GreHpA6r9m1B5SGcAtm1nTAB6n584HOAT0BrWl8Xyf5MTNiiiishhRRRQAUUUUAFFFFABRRRQAUUVwk/g/Wbm2163nktil8WW0aO8ljMamW4kDOFXnHnqNnKnbyelNJPcznOUdlc7uio4PN+zx/aNvnbB5mw5XdjnHtmpKRoFFFFABXn3jaHxFL4s0S60txFHa3myztzIALx/s08kgf+6CEWNSeQWkOMYr0GsfWkVtX8PFhkpqLlfY/ZbgfyJqJ7GlOsqL5mr9Pv0/ryNiiiirMwooooAKKKKACiiigAooooAKyPDX/ACCpv+whe/8ApVLWvWR4a/5BU3/YQvf/AEqlrN/GvR/oZv8AiL0f6B4r/wCRM1r/ALB8/wD6LatesjxX/wAiZrX/AGD5/wD0W1a9C+N+i/UF/Efov1Keq6iulae100E1yQ8caww7d7s7hFA3EDqw6kCvLtUi1uy0XxXc3PhbU9Ptbix1H5Y9SimhfzfnEzxNORGygNnywc72x0Ar0bxL/wAgqH/sIWX/AKVRVB45/wCSeeIv+wXc/wDopqmSvJ+S/wAzswteUKvJbR27935/mY2t+LNW0vxBqFqDZrBbwSXFrGYDI9ysUKySIWWXMbZYgbk6YI3Vr+DfEreLNFfVPsptoXmKQoxyxUKu4k9/n3gEdgO+arnxJ4CN1NcnWvDhuLhNk0v2qDdIuAMMc5IwAMH0qaDxj4MtYRFbeI9ChjBJCR30KgEnJ4DepJrdzhaxyRw2IU7u9u1joqKwf+E58Jf9DRov/gwi/wDiqP8AhOfCX/Q0aL/4MIv/AIqo5l3Oj2NT+V/cb1cVrPi++0zWtStUa1kMEcz21qsO6VglqJQ7t5oIUvuUYQ/dxkZzWv8A8Jz4S/6GjRf/AAYRf/FVBJ4s8ESzNLLr/h95HjMTO15ASyHqpOentVRlFbmVTD1pL3U18mYU/j7VWfdbWVvF5crxyW06vvdk+yq0asDjdvnkUHBB2j3NUbyTXtQ8IX+jaL4fv7qG41fUIrq4iuraESQNdT7xGzuSGyQp3J03Y7GuqbxR4Fe7S6bXPDzXEbMyTG7gLqWABIOcgkAA+wHpUvgieG58NyXFtKk0Mupag8ckbBldTeTEEEcEEd6mbjPRG+GjVwydSavqrXT8/Qmg1zUpIg03hbVIX7qZrVv1E1aWn3sepaZa30CssVzCkyBwAwDAEZx35qxWT4U/5EzRf+wfB/6LWp1UrXOVt+08nf8AQ16KKK0LCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoopkztHC7pE0rKpIjQgM5x0GSBk+5AoA4L4Uf8zp/2Nl9/7JR8WNYmtbHQtCtLqe0l1/VoLN57dykiQ7xvKsOh5Ue4JqDwUniDw0fEQvfCOqS/2nrlzqMJhuLM4jk27Q2ZxhvlOQMj3q74j8M6r498Ixm/hXRNVtb8XmnxsVlMJjJCCQqxVtwyx2nA3KOSpz7kvZxxyrTkuW66p9NHZX2e5lry2R0k/hfRJ/Ds+hDS7WHTJ0KPawRLGnJzkBQADnnPrzXC6/o8Fl8RvhXpLhZorOK8jU7doJit4ypx25QHHbFdjp2ta5dRCK98L3NjdeUSZJLqBrbzAuQNyuZNpPGfLz7Vzniax8QTfELwjq0GhXOoQaIl19rmtpLeNZWmhVB5ayTA4DA5DYwO5rHCSqQquM5raf2lu4SW9927Dla2hF8XbOLRPDr+NNLMlpq+lTwSebBIY/tKGVEMcoH31II4PpjpmvQLK7jv7C3vLfJiuIllQkYO1hkfoa4DxVoniP4jNBo93px0Hw6siS3jXUsclzclTnYgjd1UdOSc/lg+iKoRQqAKqjAA7CscQ4rDU6cneSb2d7LSyv63flca3bPPYbeKD9pSd4o1VrjwsJJSBy7C5C5P/AVA/AVU8W68dZ+JMHhtNL1TVdO0mAXd/baeExLM2DEku90BQLlsZ+YkcEA1pGx1kfG4a9/YV4dL/sf+zPtHnW/3/tHmeZt83ds2+27/AGa0brS7zRPG934i0ywuNSh1K0jt7q1tnjWRZIydkg8x0UrtZlPOQduAcnHYqkIzjJtNqCS16+bvpZXttrYmzt8ypcfEOe58CzeJ/DmhyX8dq0y3lldXAt7iAxnDAqFcEgAnGQcY65wNXwx4nm8TgXNvYxrprW0ciXiTsweRlBaNQUXIXJBYHGeOu4Lh6doetaH4f1CDS7KVb3X9YlvZnLQv9gjmlUOXDPhnWLJAUONwx8w5KeFNG1jwRrep6Np2kzXfhhy11YSC4iD28hXc0IDMCULcKeMHOeDurOpSwrpzVO103bXppfra66d1fyGnK6uN8X+EtMsxo7+G4rfSvEEmpRra3kUIM0vDGbzHILOPK81iWPJAyckV6DXl9tq3xBg1i51G5+HDXty5aOGR9dt0EMOeERcHbnCljkliBngAD1CufGKpGMIzkpb6qSl+Tbt2v1vYcbdAooorzywooooAKKKKACiiigAooooAKKKKACiiigApOc+31paKACiiigDD8FLIngDw+s4KyrplsHBOSD5S5rcrH8JS3E3grRJb6SSS5fT4GmeU5ZnMa7iffOc1sVrX/iy9X+YlsFVNV1GLSNGvdSuVd4bO3e4kWMAsVRSxAyQM4HrVusHxz/yTzxF/2Crn/wBFNWL0RpTipTSfcguvF89l5P23w1qdv58iwxebdWKeY56Kubjlj2A5oXxdcNMYV8M6mZVk8ooLqx3B9u/bj7R12/NjrjnpWJ4k1Uajq+jXulLqsRs5gLgTaLdlHhMsTuNhtWy37obSGQqe9ZFlPc6XaaaLfT5rq502/e48+TS9Ria6jZLgfO32ZyGDXG7+IElunfLm13O+NBOF+XXtr/n/AFc9D0zW11C8ns57K50+9gjSVra6MZYxuWCuDG7LglHGM5G3kDIzqVx1tcP4x1iK5spZtNTSk2yXMcJSc3Dqd8GJowfLVSjnKfMWjIwUNbX9jX3/AEMmqf8Afu1/+M1XM+iuefXvCVlH/gfezn7zxjripc2tlpEEmq/2qllDamUHZG1sJ97ncFLBQ/AIGRgE9TB4U03wZrOiLqeoabotzNqepXrwzXdtEZLgG7l24LDJONox9BXP6rB4csdHnmubnVGt/wC0ls5IptJ02DdMITIrkTQIAAjEAtg/Nxwa1PBWizXnh3w/qOgrFElnfXQFxJcqjtbG8l3xlIkMbgqARtIAbkYA5463vaSZ6uJj7HD89JWd137a/wDA/F9ulg8N+AbqZ4bXRfDc0scnlOkdrAzK+CdpAHB+VuP9k+lW/wDhBfCX/QraL/4L4v8A4mqdvo2tWmm6Haxw2Eh0WVAjG6dftEa28kO4/uzsb51bb8w6jPGT1Vck0k9GebTr1pL3m/xOC1nRfDmja5bwHwRpd/b3UWIYbLS4Gn8wN8zNuKgRgFecYyTkjgGPWPDGhWegaXc3WjeFtDuJHT7Y93p0DxofKZmQAkbjuAA2tnjPIzV7xdevpetR3Ni959omt1SaHTCst1JGrttIheFxtUu3zbkHzEHJCiq/ip2sfh1pH2LTLG7SMQqIfEKxL5aiFsFw7ookzgHB4y2Aa6Ixvy+f9ehyzxNVc/vvTzf/AA5b0Hwn4W1HS1uZNA0G7DnKTRaGtsjqQCCFfcWHP3gcH8KvXHg7wXaQNPd+HNBgiXG6SWxhVRk4GSV9aq/DjTdU0jwr9i1gzs0UxFu08quTFsXbja7hQOQAD0HTmukv0ll0+eOCNJXdCoR5miBzwfnUErx3AzWFRJVGk7o6Kdes6SbbTt5nMLo3w4fytmm+Fm86NpYsQW58xFzuYccgbWyR0wfSrTeFPAyNKr6D4eVoQzSA2cA2BQCxPHGAyk+m4etZQ8H6pJYSaZNFpyWAtZY4DDMweOR4miUt+7y+yNhEGJBKgswJICy3HgSe60oWNzeM4gtby3hnjmeF5TMUZWl2YDHKtuzkNwxBJwNHCnf4jJYrFW6/eyrr+j+DrWOAaLp2hw6nbapYEi0ghWaIfbYVJ+Ubl6kH64ro7HWJ7rxZq+lSQokNjDbSRSA5aTzPMzn0A2Yx9fWsLxVp+omG9uLr7PCLrUdNitru2kzNFGt5DtXYybchmd8knlsEECotY+GY1eY3E3iC6N0SD9qksLJpVI4BVxCrKw4wwOVIBHSurDLonpc7oyU6LdV66238v6+Z3VIc8Y/Hmsrw7oZ0DT5bVr6bUGedpjdXIBnkLY/1jDAYjGAQBhQq4+XJ1q7ZJJ2TucYUUUUgCiiigAooooAKytWsJrzU9DniUFLK+aeUk9FNtNHkc/3pF9ev41q1kazbTT6t4fkhDlLfUHkl2jICm1nXn0G5l/HFaUtJb9H+TEzXooorMYUhzxj8eaWigAooooAKKKKAOHePXNSudVSVNRtvL1GAW8iTPGGiM+yQoFwCvkop5zhmcjAINPu9Z8ZW91fPHp9pLaxLO0CLaSmR9ouPLGd/JJiizgDPmjGOM9rRV83kc/sX0kzgI/GGuw6jpllqX2GK5uRbboUsLgCUyXEisFZyvllYo92HByQ2Miq8es+INY8Im91u0v8ATtSguiYrfT7K7U3EaojmMgH5SxJTe2QOcetegyWlvLcxXEtvE80IIjlZAWTPXB6jNU9T16x0i4gt7z7U01wjvHHbWc1wxVCoYkRq2AC68nHUUOpFa2FHD1Jvl5rnBavLrM0d3NoMviC1hvL6FJDeW1y7RJ++d3jRQWWM5jUBcEbQDjIJ1vCen+JPOivr24uQq20IaC8uZP3rfZIgVKMp2ES7iXBJzuBUnkbn/CX6b/z661/4Ir3/AONUf8Jfpv8Az661/wCCK9/+NVDrwta6NI4GopczT+5lDxPaeJtV020t9PMWn3RuuZLe9lKhfJl5YiMHAbYcEEEgA4rprdJorOGOeTzplRVkkI27zjlsds9cVkf8Jfpv/PrrX/givf8A41R/wl+m/wDPrrX/AIIr3/41UurBq10bRw1RScrPU2Z54baIy3MqQxggF5GCjJOAMn1JA/GsrW5Amr+HFOT5mouox/16XB/pWfqmt6HrGntZahYaxNbu6O8baBeMr7HDgEGEgglRkelZFxd6PZ6posmjaVrUNvBqBnnjTRr0RxJ9knjBSMx7V+Z1BCAZJye5rOU4vqvvOiOG5o+8nf07ar8TP174pXJXRf8AhHrC5hutWme1tEv1hNvI5KpmRo5GeMozqcY5BZcZ+ZPQNDeWTRbdrh715sESG/jRJtwJBDBAE4PGVG0jBBIOT5Vqb30Umhra6dqOoOdStLnVLweHZoZFCSB5MFYF3oWAIGNw2/xbvl9I/wCEv07/AJ9Na/8ABFe//GqqNeLhyu2h0YrDKLTpR3v3f/B/q5u0Vhf8Jfpv/PrrX/givf8A41R/wl+m/wDPrrX/AIIr3/41R7SHdHH7Gr/K/uN2isL/AIS/Tf8An11r/wAEV7/8ap9t4q0y6v7ezVNQhmunKQ/atLuYFdgjORukjC52qxxnsaftIPZh7Golflf3G1RRRVmQUhzg45PbJpaKACsjw1/yCpv+whe/+lUta9ZHhr/kFTf9hC9/9Kpazfxr0f6Gb/iL0f6B4r/5EzWv+wfP/wCi2rXrI8V/8iZrX/YPn/8ARbVr0L436L9QX8R+i/UyPEv/ACCof+whZf8ApVFWvXNeNtXtdL0uzF1IqGbULbaCcZCTI7H6ALW5e6hbafpdxqN1KFtbaFp5ZFBbCKCScDrwD0pKS9o15L9RQalWlFb2X6lmiuSufE+u299ZWraHYxy6gGa2hkvpzIFBXPmCO2dUI3rnLEAnqQM1GfFuriKxkNhoey/jWW2P9rT/ALxWeNFP/Hrxlpoxz/e9jVcyOz6vUfT8UdjRXBP8Qr5NOa+NhpZgV5EOy9u2ceWQHOwWe7apZQWxgE4JzxWnb+IteurhIINN0R5JPP2qNXm58mQRy/8ALr/C5A/HjNHOhvDVVq1+KOqoqhoepf2z4e07VPK8n7baxXHlbt2zegbbnAzjOM4FX6owacXZiDOBng98GloopiCsjwp/yJmi/wDYPg/9FrWvWR4U/wCRM0X/ALB8H/otazfxr0f6Gb/iL0f6GvRRRWhoFFFFABRRRQAUUUUAIM4GeD3waWiigAooooAKKKKACiiigAooooAKKKKAEGe/H0NLRRQAUUUUAFFFFABRRRQAUUUUAFFFFADVLHO8Ac8YOcinUUUAFFFFABRRRQAUUUUAFFFFABRRRQAnOfb60tFFABRRRQAUUUUAFFFFABRRRQBleFr641Pwfo1/ekm5urCCaYlQPnaNS3AxjkmtWsbwfdyX/gfQryYIslxp1vK4jQKoLRqThRwBz0FbNa1larJWtqxLYKwfHP8AyTzxF/2Crn/0U1b1RzwRXVvJb3MSTQyoUkjkUMrqRggg8EEdqxeqNIS5ZKXY5nxJPdWviTR5LaPUbyNpUhktLcTpEm+VR57SRqVOwBso5CkHtxnkNNfxha+EYG09da1C5mW0Z47vfHLFcLFI0qZmAPlmRIdxJ2lXcKQcCu7/AOEG8Jf9Cvov/gvi/wDiaP8AhBvCX/Qr6L/4L4v/AIms3Ft3OyFelCPLa/qv+CUbtp77xdYzaAxtL77CHvjdW7GMW8hbyg6ZVjKJEbaM/KPNz1AOr9m8Rf8AQU0v/wAFsn/x+renaTp2jwNDpNha2MTtvaO2hWNWbGMkKBzgDn2q5T5E9WcNVRm1a+nm/wBDzib4eeLJlvv+K+aOS/nWaeWHTPLYlY/LwNsoAG3b26qp61F4U0zUtM0nQ7DTprp4rG/u4JESJ1t5VW/dWJZZMowQMwD7lPTkmvTKxfBv/IBuP+wrqP8A6WzVzYi1NJo66tapiKLhN6XWyS6PtY4e3XxguiqukW2rfaZ7W0Yte3Um4yxrI07DzSdmZDEuwhd4zgYyw9St5vtFrFNsdPMQPsdSrLkZwQeQfapKK46lX2nSxw0qPsutwqrqNidRtDbi7ubVWYF3tmCuy913EEgH1GD6EVaorFOzubNJqzKOj6Lp2gaetjpFqttbqxbYpJyT1JJySfrV6iim227sElFWQUUUUhmD4y/5ANv/ANhXTv8A0thrarF8Zf8AIBt/+wrp3/pbDW1XpYP4WVP+GvV/oFFFFdpgFFFFABRRRQAUUUUAFYevLI2teGTGCVXU3MmD0H2O5HP4kVuVja3cTw6x4djgZ1SfUXjmCk4ZPslw2D7blU89wK2o35nbs/yYmbNFFFYjCiiigAooooAKKKKACiiigArDuf8Akomkf9gq+/8ARtpWd4i1+FdT0KLSvE1tbzXdzEEs0MT/AG2NnQk7iCQPLEmCuNxYYOcA43hbxJNrXxMmh1C8sJ5rG2vQv2GXzIkhY2TJ82ASc7wSQOQR2rmxMrQcWd1HDz5XVW1n/kelUVjx+K9ElvLO2S+Uy3sUc1v8jBZFfds+bGATsbAJzweKSLxXpM8do8D3cgvUaS3C2E5Lou3L42ZC/OnzHg54rzPZz7M4va0/5kbNFFFQaBXC+K0tm0zxAbi9kXT/ALXEt9bLbq8sj+XDhImZwoBXZwRnJbB6V3Vcrqvhq+vPFceqWiWS7dv726CzlQB2Qxbx34WZAD82CSwO1FpSu2c+Ii5QslcuavFdSeJ9O/s2+htboWdz8txatMjpvg3H5ZEIIO3HXqelSeDvIHgvSBaTSzw/ZE2yzLtZ+OpGTj6ZP1NZXxDlhTSoFubixiQlmK3kazAhQCWERicybRkkArgdSByN3w87SeHbJiioPKAQJb+QuwcLiPc2wbccZ49B0FSv7FP+upMWvbtf10/yNKiiq2o6jbaVZNd3rOsSsifJG0jFmYKoCqCSSzAYA71zpNuyOltJXZZrA8R/8hvwt/2FX/8ASK5qx/wlWijUrewa9CXNyAYkeN1ySzKFJIwGzG42nDZU8VyfizxVbahpuhap4d1CzjMOrlFuNT320CMbKcjcWXOCrqRgc5AB5yNqcZRmm0XQlGpU5ItXd/yO/orgtF8a315deF1mhuJodSsI/PljtSY2nZW3kuOFKNFgr3E2f4a72vXjJSWgVqE6MuWf9dAoooqjAKyPDX/IKm/7CF7/AOlUta9ZHhr/AJBU3/YQvf8A0qlrN/GvR/oZv+IvR/oHiv8A5EzWv+wfP/6LatesjxX/AMiZrX/YPn/9FtWvQvjfov1BfxH6L9TnfGWl2mo6fYveW8cxg1G12h1yMNOiMMdwQxBB4p994J8P3cAEOlWdncRussFzbWyJJDIpyrqcdQQDg5B6EEVY8S/8gqH/ALCFl/6VRVr1PJF1G2u36kxio1nNLXT9Tg9WTX9UbRJLrw9eLfaXcJPNc2v2crKwwJBEWukZUcbh86k4YZAIrMsPDM2n31xcR+HPEDLLNFJFA01n5cAjuRMFRftWFG1IkwMACMHHOK9PoquRbnorFTjHlSVvn69zz9YNZs7i3udJ8ParDcQyXhJuEs5UZLmcTONou1IYMqgHOMZ45GH6Vp7aTrjarbeC9US6l+1+fKn2FXm8+ZZfmb7Rk7du0ZzweMdK72inyIX1mTVrL8f8zJ8KWVxpvg3RbG9j8q5tbCCGZNwO11jUMMjg8g9K1qztbutRs9PWTR7OO7uWmjj8uRyqhWcKWJAJwM5PHABPauQ1mDxLc+KHS3ldTMpW2EJu1jtmFsWBZxthZfOAHzKxYNg9No0jG6OGtWtK9rtv8z0Cisnw2b2XSDdams8dxdTST+TOfmhQsfLTA4XCBQR65J5JrWpPRlxfMkzM1HxJoej3C2+r6zp9jMyh1jubpI2K5IzhiDjIPPsaxvAnibRr/wAK6FZQavp8uof2dAHtYrlGkVhENw2A5BGDkdsVr6pocV7cpqNnstdWgULDeBPmKg58t/70ZycqemcjDAEJpOl3MU39oa1NDc6m6eWWhQrFCvGVjUkkAkAkkkk47AARZ81zotR5L9f68tvma1FFFWYBRRRQAUUUUAFFFFABRRRQAVyGiWur61Bc30vijVLcDUbyFLeCG08tEiuZI1UboC33UHJYmuvrH8MXK3ekTyLCkIXUb6PahOCUupVLck8nbk9sk4wOK6KcnGnJpLddE+/cT3G3Wh6hcXTyxeKtXtUZiRDFFaFEHoN0BbH1JpbrRNQnkVovFGrWwVFUpFFaEMQACx3QE5PU44yeABxWzRUe2kui+5f5BYyr3SL26WEQeItSsvLjCMYI7Y+aR/E2+FuT/s4HoBUc+iahLDCkfijVoWjXa0kcVoWlOSctugIzzj5QBwOM5J2aKFVku33L/ILGM2iag1kkA8UassiuWNwIrTewOPlI8jbgYPQA8nJPGJBpF6NMNr/wkWpGbzN/2zy7bzcY+7jydmO/3c+9atFHtZeX3L/ILGVb6Rew2U8EniLUriSXGy4ljtg8OOu0LCFOe+4H2xUdvomoQxzLJ4o1acyJtV5IrQGI5B3LtgAzxjnIwTxnBrZoo9rLy+5f5BYxbXQ9Qt5S8vinVrpSrLsmitAASMBvlgU5HUc4yOQRxUljo99aXizXHiTU76MZzb3EdqEOR6pCrcfWtaih1pO+33L/ACCxjQ6JqEV6k7+KNWmjVwxt5IrQIw/ukrAGx9CD71GfD+pGbf8A8JdrIXOdghssfT/j3z+tbtFP20uy+5f5BYxrjRNQmvHmj8Uatbxs2RBHFaFFGegLQFsduST70+/0e+vLszW/iPU7BCBiC3jtig49ZIWb361rUUvay02+5f5BYxrrRNQuJFaLxRq1qFRVKxRWhDEAAsd0BOTjJxxk8ADiifRNQlhhSPxRq0LRrtaSOK0LSnJOW3QEZ5x8oA4HGck7NFHtpaaL7l/kFjGbRNQaySAeKNWWRXLG4EVpvYHHykeRtwMHoAeTknjEg0i9GmG1/wCEi1IzeZv+2eXbebjH3ceTsx3+7n3rVoo9rLy+5f5BYxk0TUFs5IW8Uas8jsCs7RWm9AAeABBtwc55BPAwRzkt9E1CGOZZPFGrTmRNqvJFaAxHIO5dsAGeMc5GCeM4NbNFHtpdl9y/yCxk2WkXtp53n+ItTvfMjKL58dsPKJ/jXZCvI98j2pltomoQTM8vijVrlSrARyxWgUEjAPywA5HUc4yOQRxWzRR7WWu2vkv8gsYttoWoQXcc0virV7hFYMYJYrQI4/ukrAGx9CD70w+H9SM2/wD4S7WQuc7BDZY+n/Hvn9a3aKftpXvZfcv8gsY1xomoTXjzR+KNWt42bIgjitCijPQFoC2O3JJ96ff6PfXl2ZrfxHqdghAxBbx2xQceskLN79a1qKXtZabfcv8AILGNdaJqFxIrReKNWtQqKpWKK0IYgAFjugJycZOOMngAcUTaJqEsMKJ4o1aFo1w0kcVoWlOSctugIzzj5QBgDjOSdmij20tNF9y/yCxlT6Rey2EFvH4i1KCWLO+5jjtjJNk/xBoSox0+VR75qNtE1BrJIB4o1ZZFcsbgRWm9gcfKR5G3AwegB5OSeMbNFCqyXb7l/kFjGTRNQWzkhbxRqzyOwKztFab0AB4AEG3BznkE8DBHOZLfSL2Gyngk8RalcSS42XEsdsHhx12hYQpz33A+2K1aKHVk+33L/ILGTZaRe2nnef4i1O98yMovnx2w8on+NdkK8j3yPas21XUtM8cWVhca7e6la3en3U7R3UVuux45LcKVMcSHpKw5J7V1FY9zcqvjfTLUwoWk068kEpJ3KFkthtHOMHeCeM/KMd86U6jlJppap9F2fkDRsUUUVzDCiiigAooooAKKKKAMfwlcpeeCtEuYoUgSbT7eRYkztQGNSFGSTgZxyTWxWP4Slgn8FaJNZxNFbyafbtFG7bmRTGpAJAGSB3wK2K1rfxZerEtgooorIYUUUUAFFFV7q+tLFYje3UNsJpVhjM0gTfI3CoM9WPYdTQk3sBYrF8G/8gG4/wCwrqP/AKWzVmv4slPi63trW0nutLmlFil1CU8v7Ttd5M5+ZgqxgZX5QfMB5UAUdN8SS6LZ29nFDaSSX+raikInnlQs32+VcfJC4Ay6DLEda48TGUrRS1OiS9jR5pvR2/U76iuHT4g3TW8dy2lwi1kKhZ832xtxAXDGzwckgDnkkV3FefOnOHxI56dWFT4XcKKKx/EtzrltYwN4atI7q4acLKsiq22PaxyAZIwTuCj73QnipiuZ2LlLlVyxrl0bTTNyySxNJcQQCSHbuQySpHkblYcbu4rhLzxHqKJeHTdZ1G6ewF810k0trEUFqyA7cWrZLBwR0HbNdhoZ1bUdPnHiiyjgkS6DQKihNyLsZWIWSQA7weN3YcVKfC+jm/nvTZA3FxbyWsr+Y3zxu5d1xnHLEnPXt0rohKFPSSuc1SNSrrB2/AqXviG4063MEVhLqN7aWS3N5HHKoKAggAHADMWVuABwrHg7VaJvFrkW7WttaXcTPbLPPb3u+MCe4aFTGwT94QVJYHbjpk1rxaNaxJejdMWvpvOnkErI5OAAAy4IAVVUAdhznJJof8IT4fLWDNp++TT5TNbSPNIzo5k8wksWy3zkn5ietSpUuq/r7ynGt0f9fcczrHjew1rR2RI2tl0+902XUZLhwi2jG7t22NnHT94CeNpibOMioLnXvDk3xLsdXt9c0OK1gtpY7mZ7+3HmuwUIVw+4kAFfmAwBxnNdD4n0qysNIE1nbrFJPqunmRgT8xOoRv8A+hSOfx+ldLjHSuygoyT5NFc7qFSVKl7+rfMu2jSMn/hK/Dv/AEHtL/8AAyP/ABqO58Z+GrS0muJde04xwo0jBLpHbAGThQck+w5NbdJ1610Wn3X3f8E4Up31a+7/AIJwHiDUda0uy/tbU9dn0oX1z5Vpp4ktIo4AI2YK88kbjcwjY9du5goOPmrKvfE+p2iG4GqeIjYq00Tzt9gBWWCMyzJgW5BwqSKCCQzoVyBh66iTQdXsrF9J06CC70xZWkty2rT2MsCkk+SDFGxKLkheRhdq4+XJyF8D3EU9w9p4U0S0ju43iube31qZIplZGQ/J9mwpwx+ZNrckZwSDLUj2ac6VveS/D/O/5leDXLnWPA2veIfD/i3WpYdNimNu81taBZmSBZOVNuDgMxU/7pwa9Mrg38Oa3b+C77w3ovh/QtOtbq2mgBTV5n2GRSpY5t8sec8nt1rvKqN+py4hwfw932vbTewUUUVocgVj61cywat4ejiVSs+ovHITGGwv2WduCfu8qOR9O5rYrK1e++yapocPkxyfbL5odzjJixbTSbl9D+72/RjWtL4tr6P8mJmrRRRWQwooooAKKKKACiiigAooooA4WL4VaQviBNdklmXUpRIL6a1kktRcbjngROpTBAxy2QPm3N84LvwnaR+L4LKyub2L+0NGvYZ5p7ya5cJ5tqDt8122nDMMjuQSDjFd1WHc/wDJRNI/7BV9/wCjbSs8VUlKDnLc3pSk7wvpZ6fJla08ELa31tePqVxcTwsrP5udk+JpZSZEBCsQZsqcDaygjj5aXQvBY0mHS1utQmun0yJ44nSSaPcG8vgjzG+UeUPk+5yPlGOeooryXWqPqcyoU1sv6/pBRRRWRuFFFFAFa+06y1O38jUrSC7iznZPGHGfXBp1lZQafZx2lmnlwRDCJuJCjPQZ7DsOgHA4qeind2sTyq9+oVR1fS49Y042c7uiGWKQsjMrfJIr8MpBB+XqDkdavVkeKdH/ALf8Pyaac7ZpoC+3bkKsyMx+YEHgHgg59DTh8S1sKp8D0v5GBffDe3n8Q6dqFnem3gsZIpRBKJJmd0neZm3mTksZGGWDYycdaxNX8KXHh/Q9D09jPrs0msq6R2shsnYJp8kaqH8zK4EQJO4dxjsb0Hga40fxZpzaTFJJZWsMS/aJRCP+XiaWVcrtdD864CLtYfKwx0zPEkN3p3hLRI9cu/7EMevzyC6kld9kbxXLrkxSK3O7bgOD9eh7XKWi57rTsRgIR+sp8lnr37eVvw+Rb0l7C91LQ4Tba1Bb6rYxXUVw2vag6eayO5gBB2FlWPJ3MuQenauw/wCEasf+e+qf+Da6/wDjlcdounubzRLbTNa09za2i3ljGuk3xjjhkWQK5zc7ASpkA3/MOnpXY/ZvEX/QU0v/AMFsn/x+uhK61jf7jpzBQdRWi3p19X/N5W+Yf8I1Y/8APfVP/Btdf/HKP+Easf8Anvqn/g2uv/jlH2bxF/0FNL/8Fsn/AMfo+zeIv+gppf8A4LZP/j9Plj/J+R53LH/n3+X+Yf8ACNWP/PfVP/Btdf8Axyr1hYW+m2a21orLErM3zyM7EsxZiWYkkkknk96o/ZvEX/QU0v8A8Fsn/wAfoNv4hAJOq6WAOp/s2T/4/TVk7qH5f5jilF3jTt93+YeK/wDkTNa/7B8//otq164PUPEdhqWl3VjcePvCqxXULwuyKoIDAg4zc9ea6KDxj4YuriOC28R6TNNK4SOOO+iZnYnAAAbJJPGKIyvJt+X6m8aVbncnBrRfqP8AEv8AyCof+whZf+lUVa9ZHiX/AJBUP/YQsv8A0qirXql8b9F+pC/iP0X6hRRXH61p2q3F54nig0+aaLUtLFtbSCaPZ5ixy8EFwVyZFA46gk4HNapXHOTirpXOworhNK8O6nbxaBMtgbQWU920lmlx5YjSW5DKP3bYO1Mtt+ZSFKnkgju6JJIVObktVYK860O/8TR+PZbm/wDtUllfQQSzacwJ+wLK86xMB2IWCNXA43SO3Ra9FrGtYwPHOqSc7m02zU/hLc/41nLdHTCrGmnFr4tDZoooqzI4m+8Wan9o1G3057EXazrDZ2lxbSkgefFCzyuDjGZNwUc7HRu/LdL8a6hqfl3iwQw2ZNuDBJEwmPn3ktup3bsDaqK2Npyc8ius/snTvt8l99gtftcoAkuPJXzHAIIy2MnBVcf7o9KBpOnC9S8Gn2ouowVScQrvUEkkBsZGSzH6k+tXzRtsc3s6vNfmLlFFFQdIUUUUAFFFFABRRRQAUUUUAFY/hiS3l0mZrOJooxqF6pVmBJcXUoc8AcFgxHse/Wtisfwz9n/smb7G8jx/2he5Mi4O/wC1S7x1PAbcB7AcDpWq/hP1X6i6mxRRRWQwooooAKKKKACiiigAooooAKKKKACsjV/ENvo9/YWs0F1K15KYw0NtLIEGx2zlVIJ+TG3rg7ugNa9QXT28MJurzy1jtg0pkkAxGApy2e3yk8+hNCt1FJSatHc42/8AFuuaJqNy2t2kMNlFJJJGEt8mS2WdI94kWZiGCyo5BjGcEcda7DTpLmbS7WTUIlhu3hRp4lOQjlRuUcnocjqa4zwNqsHiO61KbVNAs7K5uWmMDC02PNbCZo2SQtkmRXjy46fOnHNd5T5oyV0L2FSjUcZsKKKKRQUUUUAFFFFABXN6j4pkS41Ww0qzeS/s7ZHh+0KYY5ZXdkVcvtBG4DkHDZIByK6SuU8fa2+gaDPNYWFve300Ls0c8e5DBCrSSGTHO0LlR23yKO9F1HVgqc6slCD1ZDY+LdSuNR0uxltI0lvndC7x7djQvKtwNqu/TZGo+bBMmecFR2NY/hs2VzodncWumRacY1kjNssQT7O4ciVBgDjep5HDYBrYptp6oiMJwvGbu/6/UKKKKRYUUUUAFFFFABWXcXNuni2wtXt91zLY3Mkc+77iLJAGXHfJdDntt961KyLmaFfGmmwtG5nfT7t0kDfKqiS2DAjHJJZcHPGD1zxpSV5fJ/kxM16KKKzGFFFFABRRRQAUUUUAZfhk2TeEtIOkiQWBsYTbCQAN5XljZnHGcYrUrI8JfZv+EL0T7AzNa/2fB5LOgRinlrtJUEgHGOMnFa9aVv4kvViWwUUUVmMKKKKACs/WtGttf01tP1B5xaSH99HDKYvOXB+QsuGAzg/KRnGDwSDoUU03F3QHKaV8N/DGm2ZtZND0q7iRz5DTafEZEj7IzkZcjkbjyRjOTljT8PeFbO+sNOuYhHa/2Pquoi1jjiG1B9vfKhegGxCo443ZGCBXb1z0Xhe5tDMth4m1a0hluJbjyI47VlRpJGkYAvAWxuc9Sa568ZzfNHc2uqlN06j003v0H2/hMRrcie6jJnubeciC2EKsYpfM3soOGlY8M4wCAvyjFdFXPf2Dqn/Q4a1/35sv/kej+wdU/wChw1r/AL82X/yPXHLD1pbijTpx2kvx/wAjoa5zxtDo15osVj4ivBZWl3P5QnJiARvLdvvSAheFIBHOcYPNO/sHVP8AocNa/wC/Nl/8j1ygs/iHI07jX72CKLUBFEWtrWRriA3BjLY8keWVQBy3zBgQQBzSjQqQkma/V41otc6Xrfr8jf8AhuLGLwu1rpxRltpvLkMc0Uy7/LRsCSNEDkBlBYjOQRk4Brra4zw3ZazrHhXStTufF2rJNe2UNxIscFmFDOgYgZtycZPrWl/YOqf9DhrX/fmy/wDkenPD1Jycu5nGjGkvZuS006/5HQ0Vz39g6p/0OGtf9+bL/wCR6P7B1T/ocNa/782X/wAj1P1WoVyx/mX4/wCQ/wAZf8gG3/7Cunf+lsNbVc9L4Xubowrf+JtWu4YriK48iRLVVdo5FkUEpAGxuUdCK6GuzD05U4tSJqOPKop33/QKKKK6TEKKKKACiiigAooooAKx9auVg1bw9G0KSG41F41Zs5jP2Wdty474UjnjBPfFbFY+tSW6at4eW4iZ5H1B1gZWAEb/AGWcknjkbQw4xyR9K1o/F8n+TEzYooorIYUUUUAFFFFABRRRQAUUUUAFYdz/AMlE0j/sFX3/AKNtK3Kw7n/komkf9gq+/wDRtpXPiP4TNqPxfJ/kzoKKKK8gQUUUUAFFFFABRRRQAVleJRqh0GVdAk8q/aSJY3wDtBlUMeVYAbS3O04644rVrK8S6jc6Vob3ViFMwmhjG6B5toeVEY+WhDMQGJwDk4q4fGrGdS3I79jjLjVfHGh69pcF8ZdQtgkZvWt7J5lYNPKGIaOEfMsflk/d+7wpzRrF14gnt9Hn0yUNetr7CxGu25gCp9imBDhFUnJ3lSB/EoPIIE1z4/1aDxBp1gLALFdmKJmls5UZDJPNCspBI2KTHGwRsMQ/Xis7xff3uoWNtba1ZLqL2fiI2vk2a3FuZk+wyODiLzJB98n5c5A5wM12yTVm4penqRgHF4hJSbWu/p5vUZpPgnxdYal4fvp7bQZrjR4VtjN9smDSQqkiKo/c/LxKc9c7U6Y57v7T4i/6Bel/+DKT/wCMVxmjNZLa+GI9Usrt5tHgEbyx6FelndF2RAOYR8gBZiD/ABBSOhrs/wDhJbH/AJ4ap/4Kbr/43W0eVLe33HZmNRyqLnnZ69Eur739fmH2nxF/0C9L/wDBlJ/8Yo+0+Iv+gXpf/gyk/wDjFH/CS2P/ADw1T/wU3X/xuj/hJbH/AJ4ap/4Kbr/43Vc0f5/yPM5o/wDPz8v8g+0+Iv8AoF6X/wCDKT/4xWRq93qE17Y2XiO1tdP0iaUi4mjufOjuDtIW3fdGuxWLZ3Hg7Nn8Yq3feIPtk1pp2l/araa9lMTXFxayW/koEZiyeagV3+UAJyeS2CqtXJyPo9148uPBl/p+uvFIwiSeTW714rhDbmWTIL7fl+RSMnO8HjpR6Sv93+R34Wi5P2nM2km9LbbX6bfobmoa1MNc1awfxANKvAJItLtXSIRyAWqyee25CxCuz85Cfu8YyDmEaqda+G+k37yvNJJqdisjyNGx3rfxKw3RgKwDAgEDBABrlhrEFpo8N5Po+pw2721peFoPE2pSpBbzpMweTYuV2mAA4Uj5xzxXRahpVi+i2+o27aqjw67bQCO51a5nU+XqKRbijyFSDtyMg4yO4zQm3c75U4wcU1bVa6fo+vzOv1yzuL7SxFZCIzpcQTKszlFby5kkIJAJGQpGcGoftPiL/oF6X/4MpP8A4xWvRWjjd3TPGcLy5k7fd/kcPc2E2u+O9Lt/F+haRJCmm3jwxiY3ilvNtgSRJEgUgHgjPU9O+1/wg3hL/oV9F/8ABfF/8TV/U9D0nWvK/tjS7LUPJz5f2q3SXZnGcbgcZwPyFcJNYeH4PFGoaaPCOhTCDcYYV0dVJRbdZDIZiChG9gm0AEbgaag22dc8TGMY3k49NPm+5p+MvB3hi08C69cWvhzSYZ4tNuHjkjsYlZGETEEELkEHvW7FdeJ2z5ukaQvpt1WVs/8AkuKzdD8K+FtU8P6dqEnhXRFe7tYpmC6dEACyBiBx711lLld+wqlVVKaSbfm/l5syPtPiL/oF6X/4MpP/AIxRplrqP9s3l/qcNrB51vDCkdvO0v3GlYkkouP9YPXpWvRS5NU2zl9m7ptt29P8gooorQ0Ciiquo38em2TXMyu43JGqRgbnd2CIoyQMlmAySAM8kDmgTaSuy1RWbba1Dcahb2TQXEFzNbvceXNHjYEdUYE5wTlh0yCOQcEE6VGwKSewUUUUDCiiigAooooAKKKKACsvw9DaQaZMlhP58Rvrt2faRiRrmRpF5/uuWX3xWpWP4Ziih0mZYJ0nU6hesXQEAMbqUsvQcqSVPuOp61ov4T9V+oupsUUUVmMKKKKACiiigAooooAKKKKACiiigAooooAxPCkXk6NOuc/8TK/bOPW7lP8AWtusjw1/yCpv+whe/wDpVLWvUU3eCZMJynFSluwoooqygooooAKKKKACsXxlF53gTXos436bcLnGcZiatqub8Wa1pQ8M6vZnUrMXUmn3WyDz13tsjcNhc5ONrZ9MH0qKjtFg+dRcqe61OjpaKKsAooooAKKKKACiiigArHuVtT4201ndxdjTrsRKF+Up5ltvJOeCCEwMc5PTHOxWPcxRHxrpszTqsy6fdqsJB3OpktiWBx0BVQef4hwe2tH4vk/yYmbFFFFZDCiiigAooooAKKKKAMfwjBFa+CdDt7adbmGLTrdI5kztkURqAwyAcEc9BWxWX4as00/wnpNlFLFMltZQxLLC25HCoAGU9wcZBrUrSs71JPzYlsFFFFZjCiiigAooooAKKKKACiiigArnLnwfDdaq97LqF8/+lQ3UMT3cxjiZHDMoTfsKnbwCvyknqMAdHRUtJ7mkKk6bvB2MXwbH5PgTQYs7tmm265x1xEtc1b32qmxuY9Ts9bJtbW1t9kfmqbiUGYO++MFtpGxmZOeFHJ+U9R4U/wCRM0X/ALB8H/ota5C38beILnSUurZ9KnJjtpZSsPl+QJIpZJFxJOoYqEjOdw4LHHFXRu4J+h59So5KMpvVr79DutJie30WyhknluHjt41aaZWDyEKAWYNyCepzzmrlcPYeMZp7l/N13RXsHsftMV8ts8ao4kVWR1MxwQHTgkHLj0wZpPEmtW0uhpcGwLX115c6GExyCIyMI5AjS7wWUL8uG2kNnpiqcHc0jXhbQ7KiuEfxjrt1DZNpGn287zf2gsyBGfY8L7IRkMOC7JuPoSflAJHWaHezal4e06+ukWOe5tYppEUEBWZASADyBk0nFrcuFaM3ZF+iiipNQooooAKKKKACiiigArL1U2Y1LRPtgczG+b7LtxgSfZ5s5z22eZ05zjtmtSsfWltTq3h43Lusi6g5twq5DP8AZZ8g88DZvOeeQB3yNKXxfJ/kxM2KKKKzGFFFFABRRRQAUUUUAFFFFABWHc/8lE0j/sFX3/o20rcrDuf+SiaR/wBgq+/9G2lc+I/hM2o/F8n+TOgoooryBBRRRQAUUUUAFFFFABWfreoQaXpbXd1bvcossSLEgXcztIqpjeQowzKckjGM1oVna9Dpk+jyrr0kEdgHjeU3LKIztdSA27jBYAYPXNVC3MrkTvyuxzC6/wCFfEurWUV/oCXN1dW4aKS7t4JWCea8YAwzMw3KxygZQDuJAyaz/E76Sug+HpPD4XSrS51Uln062zIqyWk671SEEhyhBU4yMqSBgit0WXhfU/EFjd2urWjyeTF9ms7eaEo6wvIUdABvGGZwdpAOCDkZFc/4g0LRp7fSh4RtNFv0v9a3mCXa1mWWxmGCEVgAAu7GPvHPGcjr926Sv0FguZV05W666dv63KuiSeJZ/GPhGS6/tiO1XSo471JUnEbSiO4DtIT8m7cIvvDJypB9fUa8m0lNLvdf8LRDwfoC2WrWEc8syaOOJjFM7qkn3V2mNPlOThutehf8Ip4d/wCgBpf/AIBx/wCFdVPmtpqd2Zc6qRtFbenV+pr0Vkf8Ip4d/wCgBpf/AIBx/wCFH/CKeHf+gBpf/gHH/hWl59l9/wDwDzL1Oy+//gFXxkFutGOmW6tJqd0Q+nqgOYpo2Vkmb0SN9jMTx0XBLBTz9tq2qnxI+jSazojatDdJG/8AxJHJEj2zSB8/aR1jRlzjPGMY5roNQ8K2sVo8/hi1s9K1aJSba4hhWNS2PuSbR80bcAjB7EfMARxcUhbxAfFl5o+gWt3HeyW6alca2YEnkEfkMi7rfJA2MBnnKtglTzLvfU9PDe9Safn236b9GiqJ7SDQ7q5h07w//ZfkR3E7xeFh5bSeSJo4mUXX3yko2kjZl9obccVsTX2sanokUs93YxwW+uWUN1ZDSXt5kl+2Qsfm+0OoyWDZwcg++azbvw+mq5tpl0J1sdPij8uDxAI3t4YwrI7OloJNo+V8MxTJDbeFI17621yx0i3W9s7F4rnWNOluL1dTaaaRvtMCq20W8anhEXqOBnk9ZR1yabj3uu3l/X3WPQqKKK6DxQqvLY2k+7z7WGTcSzb4wckrsJOfVfl+nHSrFFAWTK9nYWmnwtFYWsFrGzFykEYQFj1OB396sUUUAklogooooAKKKKACsbW5tKvrebTbvWIrKaForhik8ayQ7JEdWIcEAbgnUYOQO9bNcnrbaxYPqGryXGi2ghi8mxuL65ZIoUaRN3mfIOW2jncRkIAOpZx3M6l7WSvcuW9jp009p4j/ALckvEs7eWP7WZovKeMklyxVQvBUcjGNg9869jqFlqdv5+m3kF3Du2+ZbyrIufTIOM81wWoaBqb6xZ6fNqNvaR3thcRvavqTYupmW4MjLGFUSEtKjsdg27eK7+ztIrGzjtrfzPLjGF8yVpGx7sxJP4mqlbuRS5+sbL8yeiiioNwooooAKKKKACiiigArH8MWps9JmjLq+7UL2XKOGHz3Ur4yO43YI6gjB5FbFZXh6wm03TJoLhQrvfXc4AOflkuZJFPU/wALj/63StU/3TV+q/UXU1aKKKyGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZHhr/kFTf9hC9/8ASqWtesjw1/yCpv8AsIXv/pVLWvWdP4F6GdH+HH0QUUUVoaBRRRQAUVU1W+/szRr2/wDL837LbyTeXu27tqlsZ7Zx1qn9p8Rf9AvS/wDwZSf/ABioc0nYiVSMXZ/ky0NVtDrP9l7pVu/KaYK0DhWRSoYq5G1sF1yASRmuF1zwfNrOral4f0m+g0u2K/2ncStatNJNNdC5gfkSKAoQcDB5xk9cweIm+JOkpqut2tppN5ITGlnDatNPPDH5qZRUKBWBHLEbT3zhVA0vAWra3rPiPVbrxLo39j3n9nWKiHzN29fMuiHx1TJJG088Z71F1J2aPWhCVGm6tKS2XVb6dPmb2j3urf8ACQ6hpesXFlc/Z7W3uI5LW1eD/WPMpBDSPnHlA5yOprRbWNNS+azfUbRbpSqtAZ1DgtjaCuc85GPXIrOtP+Sh6v8A9gqx/wDRt3WZqngX+1tfvb24njEUyyPAPLUvBOYY40lVtu4EbWON2MhCORkbQSe55mJc4tOCvt+R1aXVvJII4543c7sKrgk7W2tx7Hg+h4qauR0vw1qel65Yahm1uWFrLDeM0zIS80/nOyDYcgNkAEgkHkjHPXU2ktjOEpSXvKwUUUUjQKKKKACsq4slk8XaffGeNWhsbqEQlvmcPJASwHcDywCe24etatY9zbF/Gum3W6MCPT7uMqWG47pLY5Azkj5OTjjI6Z50pfFv0f5MTNiiiisxhRRRQAUUUUAFFFFAGP4RtmsvBOh2rukjQadbxl42DKxWNRkEZBHHUcVsVjeELSaw8EaHaXMbRTW+nW8UiN1VljUEH3yK2a1ra1ZerEtgooorIYUUUUAFFFFABRRRQAUUUUAFFFFAGR4U/wCRM0X/ALB8H/otasx6JpUNi1nDplnHaswZoFt1CEjGDtxjPA/Kqng+QTeBtCkXID6dbsM+8a1s1FO6gjOEOWCjJapGfLoGjztIZ9JsZTKgSQvbId65B2nI5GVBx6gelOg0TSraaGa20yzhlgTZE8duitGuScKQOBlmOB6n1q9RV3ZXJG97Fa106ysf+PKzt7fkn91EqdcZ6DvtX8h6VPHGkUaxxIqIgCqqjAUDoAKdRQNJLYKKKKBhRRRQAUUUUAFFFFABWPrUUUmreH2lnWJo9Qdo1YEmVvss42jg4OCW7cKeex2KytWslutT0OZp44jaXzTKjtgyk20ybV9Thy2PRSe1aUnaXyf5MTNWiiisxhRRRQAUUUUAFFFFABRRRQAVh3P/ACUTSP8AsFX3/o20rcrDuf8Akomkf9gq+/8ARtpXPiP4TNqPxfJ/kzoKKKK8gQUUUUAFFFFABRXO+KtKvdRn02TT4TJ5Epa6Xzggmt8qXhweCXZU6gAhWBYBucC/8LaxY2emp4asbaGS21aa+m2MsW+ISyeXFx1BjkIA6ADHFbRpxkl71rnPOrKLdo3SPQao6zpUOt6Y1jclhE8kbttJBISRXwCCCD8vUHI61yF/p+uTWsNtDpupiSz1iS9Ro7i3W3uUN2ZQHy+/Gw8DA+bGelX9Y0S98SfbJp4giLLFbWltcpvTYlyjyyumQGV/LX5c8ogwQXIFKmotPmJdVyi1y9CK4+HkY1yxu9K1B7K0tTCXtT5shl8u4ac5YyjO5nP3g2OorP1vwZZSWuhaFq4iu9OudbLi2gR4EiVbGbCjDlvvJvJ3clj61VvfCHiK38Z6NdWn+kQ2zQebdW6JCqp9omZ4grSbljWOUDaA2QqjjFU4oda8OeG9Hk1ySLTbmHWZLgz3cCTgKNOmMjssLKZCWEmCW3njOeh6HdpNzTWg8FdYhckHF62evby/T5E3gXSfA+t3Vy3h7TNV06axjQMTqkqkLIWOz93O2OUOVOO2RXcf8I1Y/wDPfU//AAbXX/xyuK8PwxaTr4Gi6zpEV74jRL5tmiXhW4VhJIrZa4KxkgSkLx0PHArtfs3iL/oKaX/4LZP/AI/W0Uraxv8AcduZJSrX5XJW0vv5/F53D/hGrH/nvqn/AINrr/45R/wjVj/z31T/AMG11/8AHKPs3iL/AKCml/8Agtk/+P0fZvEX/QU0v/wWyf8Ax+q5Y/yfkebyx/59/l/mQXnhK0urSWGLUNYtmdColi1W43JkdRucjI+lcrr1xJcaENEu9EvLfUNLmX7JPp0F3FbqRGQssLwxSY+VyPLIIHIJOMnsfs3iL/oKaX/4LZP/AI/R9m8Rf9BTS/8AwWyf/H6LW2jb7v8AM6aFX2O1N/Ky/U4cXt49lr+n3C3csOrRnddnRL/zHc2ccBzH9n2oN8e7OW4ONvcWraS3t/CGl+HdPstRZ7bUbQR40a7hTykvY33szxgBvLG5iSBu3Y7V132bxF/0FNL/APBbJ/8AH6Ps3iL/AKCml/8Agtk/+P0e92/L/M3eMbt+7elnuuhr0VkfZvEX/QU0v/wWyf8Ax+j7N4i/6Cml/wDgtk/+P1fM/wCV/h/mcnPL+V/h/ma9FZH2bxF/0FNL/wDBbJ/8fo+zeIv+gppf/gtk/wDj9HM/5X+H+Yc8v5X+H+Zr1ly+JtBt5nhn1vTopY2Kuj3cYZSDgggng0z7N4i/6Cmmf+C2T/4/VnR7CTTdP8iedZ5WmlmeRI9ikySM5AUk4A3Y6npSvNuyVv69SeapJ2St6/8AAZW/4Svw7/0HtL/8DI/8aP8AhK/Dv/Qe0v8A8DI/8a16oa1rWn+HtHn1TWJ/s9lBt8yXYz7dzBRwoJPJA6U/fXVfd/wS1GrJ2TV/R/5lL/hNfDIufI/t2x34znz12/8AfXT8M1L/AMJX4d/6D2l/+Bkf+Nebw/EX4eH4hXmszalD5RsYUgkNhNky75PMP3M52iMZI6Hg8GvUNF1qw8Q6RDqmjz/aLOfd5cuxk3bWKnhgD1BHSs4SnK92v6+Zc8NjKSvVVl6P9balSTxl4Yhx53iPSY89N19EM/8Aj1cR8W/Ffh7Uvhbq9pp2vaZd3Mnk7IYLyN3bE6E4AOTwCfwr1GuE+NI/4tDrX/bD/wBHx1cubkdzrwXL7empb3X59v8AgkHji/tNN+KXgO71G6htLaP+0d808gRFzAoGWPA5IH410n/Cc+Ev+ho0X/wYRf8AxVYXiz/kr3w//wC4j/6IWu7pq93/AF0Q6rh7OnzLp3/vS8jI/wCEr8O/9B7S/wDwMj/xo/4Svw7/ANB7S/8AwMj/AMa16KLT7r7v+Cefap3X3f8ABMuLxNoVxMkMGtadLLIwVES7QszE4AAB5JrUrI8S/wDILh/7CFl/6VRVr0Rbu0wi5czjL+twooorQ0CiiigArI8NW89rpU0d0rK7aheyAMuDta6lZfw2kYPcVr1h+EVkTRbgTAhv7T1AjJzwbyYj9MVqv4UvVfqLqblFFFZDCiiigAooooAKKKKACiiigAooooAKKKKAMjw1/wAgqb/sIXv/AKVS1r1keGv+QVN/2EL3/wBKpa56+0O+l8TX8kOm3UK3VyzvqVrLCjtB9h8oICXD5EuCARtBAbPFTRV4L0OeM3GlCyvt+R3FFcFpuk67DfaTIdIjtILPUpWaC1KQxeS8ITzPKErgHczHCnsTjLc97VyVjSnNzWqsFFFFI0MjxX/yJmtf9g+f/wBFtWvWR4r/AORM1r/sHz/+i2rXrNfG/RfqZr+I/RfqFcdqviDT/C3jW/vdcNzb2l1p1rHFcJZzSxs0b3Tuu5FIBCkMQe3PQGuxri/GHhKzm0XWNSlvNReRbO6kSF7x3hQtBIh2xsSBw7Yx0z6cUTbSujqpSpR5nVbtbp95f0S7GpeMtWvoLa9itm0+zhV7qylt9zrJclgBIqk4Dr09RVLWrzVF1LXraBdSMT2tuLcwQSHABcz+U4GBJsI25PL7R2wOx6UtaR90wrL2r00/4axxOnjXl1W1a5N++mF1UodwkVfOuDCzE/MRs8kOpw3KlzgMG7aiim3cinDkVr3CiiikaBRRRQAVj3NrM/jfTLtUYwRadeRO23gM8lsVGfcI3Ht7VsVj3NtO3jXTblUcwR6fdxu4+6GaS2Kg+5CNj6GtaLtL5P8AJiZsUUUVkMKKKKACkzzjH40tFABRRRQBieC0kj8A+H0m/wBYumWwfnv5S5rbrI8Jy3E/gvRJr55JLmTT4GmeT7zOY1LE++c1r1rX/iyv3YlsFFFFZDCiiigAooooAKTPOMfjS0UAFFFFABXPeJNDlv5oL63l1CQ26lZLK21CW2E69flKOoEg7EnByQccFehrmr7xvYWMd27Wt1OtsSoEJiLzkTCA7Iy4c4kO3JUDg4JyMnLzaB7ZUXzXIvCnhZ9Huje/aNQtbf7P9nt9IkvpJ4raMEbSdzMPMAXHyHaoJUZ+8eqrmovHFhczRiytbu6t3eNDdReX5aeZcPboTlwxBeM/dB4wa6Wjl5dBuv7Z817hRRRQIKKKKACkJ6cZ/pS0UAFFFFABRRRQAUUUUAFZWrWE15qehzxKCllfNPKSeim2mjyOf70i+vX8a1ayNZt559V0CSFWKQag8kxVcgL9lnXn0G5l59cetaUtJb9H+TEzXooorMYUhPTjP9KWigAooooAKKKKACiiigArDuf+SiaR/wBgq+/9G2lblYdz/wAlE0j/ALBV9/6NtK58R/CZtR+L5P8AJnQUUUV5AgooooAKKKKAOa8Xavf6ZJpq6Y5Xz5wt2TGCsNtuVZJ8kYDIWXGcjDMSpAyMO/1XxLYafZvaXNzeyJqM8l4ZEgXFlbzMj/wryylCcc5HGK9BoraNWMUlypnPOjKTb5mjg7/xbFNps1pF4jjsdZF5dW0UUbQfKVndYzKHU7VCqCehIzjJIq/rWr6rdRX8mgyPHBZzxWYaJYzJcTNNEJNhcFRtUsmTxvLZxsyetpscaRKViRUUsWIUY5JyT9SST+NP2kU7qIvZTaacv6+888j1fxPZ+JNO0nUtQJmmtYmmjUwNIrSzTDdsCAybUVVLLgKQGKsMiq/iQazPDpUkTS6tJH4iKWUN3bwlv3NvcBnIVolJ3q7DLLwq98g+m1geI/8AkNeFv+wq/wD6RXNXGopySUUjfC05UqnNzN2vv6HE6Poet6frGhXcmlaxLFpdrHbNGY7NfN8tJ0Rs/aztwtweMH7vvXdf2zff9C3qn/fy1/8Aj1a9FegqdtmVia1TESUpO1u1jI/tm+/6FvVP+/lr/wDHqP7Zvv8AoW9U/wC/lr/8erXop8r/AJn+H+Rzckv5n+H+Rkf2zff9C3qn/fy1/wDj1H9s33/Qt6p/38tf/j1a9FHK/wCZ/h/kHJL+Z/h/kZH9s33/AELeqf8Afy1/+PUf2zff9C3qn/fy1/8Aj1a9FHK/5n+H+Qckv5n+H+Rkf2zff9C3qn/fy1/+PUf2zff9C3qn/fy1/wDj1a9FHK/5n+H+Qckv5n+H+Rkf2zff9C3qn/fy1/8Aj1H9s33/AELeqf8Afy1/+PVr0Ucr/mf4f5ByS/mf4f5GR/bN9/0Leqf9/LX/AOPVNZaq+oW16YbGaC5tJDEbe5dAS+xXA3IWGCHXnnHPFaNZGjf8hXX/APsIL/6SwUtYtaktSjJe9e/p2fkVNP8AFqagmlyCxlt49SupreMTMA8ZjV925RnB3RsMAnjBzniqtppEOteIfET313qf+j38cMSQapcwIifZYGwEjkVfvOx6d62pdA02eS1kkt/mtJ3uISsjLskZizNwecknr6kdDWLr+i6tYxapqnhi91OW9u5Y5jp0EtrGkjhY4iQ80L7fkQHk84461pO3RHThedNxlKzez27fd1Lw8H6aQCbjWh7HXb3j/wAjVRu9Ih0XxD4dexu9T/0i/eGVJ9UuJ0dPss7YKySMPvIp6dRXFx6z8SZNRmsU0jWjcwxRzSJ/a2mfKjlwpz9lxyY3/L3FdpoGiarfRaXqnia91OO8tZZJhp08trIkb7ZIgS8MKbso5PB4zz0rNNPZHdOnUpK86ifknfdHWUUVheNPEn/CIeEb3XPsn2z7L5f7jzPL3bpFT72DjG7PTtWjdldnBCEpyUY7s3aKwtV8Sf2Z4u0DQ/snm/2z9p/f+Zt8nyow/wB3B3ZzjqMe9btF7hKMopN9f+G/QKKKKZJkeJf+QVD/ANhCy/8ASqKtesjxL/yCof8AsIWX/pVFWvWa+N+i/UzX8R+i/UKKKK0NBAcgEjHse1LRRQAVjeF7ie50id7qRpHGo3yAsSSFW7lVRz2CgAdsDjitmsfwxcy3WkzyTqqsuo3sYCxhPlW6lVeB7Ac9+p61qv4T9V+oupsUUUVkMKKzNU1+w0e4ggvPtTTXCO8cdrZzXDFUKhiRGrEAF15PrVT/AITDTf8An21r/wAEd7/8aqbo0VOo1dRZvUVg/wDCYab/AM+2tf8Agjvf/jVH/CYab/z7a1/4I73/AONUcy7j9jU/lf3G9RWZp3iCw1S6a3t/tUU4TzBHd2U1szKCASolVdwBIzjONwzjIrTp3uRKLi7NCA56jH1paKz9c1qy8O6Jc6rqbslrbKGcquTyQAAPUkgfjQ3ZXYRi5SUYq7ZoUVx7XNr4t13QzHc3iWFxp97OEtNRaLcyS26AloJMNjc/G44yehrW/wCEN0v/AJ+da/8AB7e//Ha5p4mMJWsbOko/G7P087dzarB1PVdYg10WumaaLm3ht1nlLqy+cCzKUjk+4JF2g7W+8HHK4yZP+EN0v/n51r/we3v/AMdqrqXhvRtM0q7v7i41sxWsLzOF1y9JKqpY4/fegrP61GWiTHH2ULuWvy/4JW+GN1c33w/tLq/VluZrm7eVWzlWNzLkc+h4pt/ruuW/jRNGgtA9vcTROl1xiOMhSVPuRDddj1TkVl+HLXQNQ1qfSLaS4GI3u0k0zxVdXEb5k+ctiRSrFnBORyWJznNdT/whul/8/Otf+D29/wDjtX9ZVL3ZRMqvscRJzpyaV+i/DcwT4tkt/CllfSaiJLtru1F+rRri2MjDzYBgAKUAbIbLqB8x5BqlqPirWrVtZW6u4LWGO6VUmgdGksYhdRxDzI2QgGRHZ1LEggcAV1f/AAhul/8APzrX/g9vf/jtZPivw1Z6Z4N1q/sr3Worm10+eaJ/7bvG2usbEHBlwcEd6ax1O/wmSwvO1FVH22/+2KcXiu4bT9Jni1Lzr6PSRfXtkxiRbg/Zy4A43ByxVvl4Cg5AyM9F4T1G81HQUbVlZNQgkeC4V4wjblY4JUEgErtbgkfNwcYon8T2tv4utfDstvOLq6ieWKTMewqoyT97cO4+7yQewzW1XXzKWiRKoVKbTm91/X5fmY/ili3gzW8qVxYXAGcc/u25rYrI8V/8iZrX/YPn/wDRbVr1kvjfov1Ev4j9F+oVkeK/+RM1r/sHz/8Aotq1643xf4psJtJ8ReH9P+23esRWMkZgtdPnm2PJETGCyoVGcjqcfkadT4WjX2M6ycILf/htTsqKydM8S6dq9/LZWf2xbiGMSvHc2E9uVUnAP7xF6kHHrg+hrWqr3HKMou0lYKKKKZIUUUUAFFFFACZ5xj8aybmS4HjPTYkdhbNp920iZOC4kt9px04Bf8z71r1jXVxcL450u2R3FtJp15JIg+6XWS2Ck+4DPj6mtaKvL5P8mJmzRRRWQwooooAKKKKACiiigDL8MXlzqPhHSL2+Obq5sYZZjgcu0YLdOOpPStSsbwhdvf8AgfQryYIJLjTreVxGgRQWjUnCjgDnoK2a1rK1WStbViWwUUUVkMKKKKACiiigAooooAgu53t7ZpIovOcEBU3YySQOv40ywvkvrfeMJIOHi3ZKHPQ/lVqoYbaKCSV41w0zbn+uMf5+tPSxDUudNPQmrL/4RvShqRvvszeec/8ALZ9i5kWUkJnaCXRWJA5I5zk1qUUrtFOKlujF/wCES0X7RDMlrJGYShRIriVI/kkMqZRWCth2YgEHGa2qKKG29xRjGOyCiiigoKKKKACiiigAooooAKKKKACiiigArD15ZG1rwyYwSq6m5kweg+x3I5/EitysjWZLhNV0Bbd2WOTUHWcAkb0+yzkA46/MFPPp9K1o/E/R/kxM16KKKyGFFFFABRRRQAUUUUAFFFFABWHc/wDJRNI/7BV9/wCjbStysO5/5KJpH/YKvv8A0baVz4j+Ezaj8Xyf5M6CiiivIEFFFFABRRRQAUUUUAFUtW1JNJ083UkE1x+9jiWKHbvdpJFjUDcQPvMOpFXazteh0yfR5V16SCOwDxvKbllEZ2upAbdxgsAMHrmqjbmVyJ35XYhg8R29zcxQ21rdSsTicoqt9lPmNHhwGz9+N1yoYDaSSBzXKeM/EcF1pmh6jp8t+nlasQDYwpLOd1lOybFIdTuV0PI4D87SDjcs4PD9tqVpeaNrVtZQ3cSwQWlnJbiC5EcjnCjaSTulcHYR19a5LxN4esdD8PaHZ2Fxa6jbya2DJJrSpPAgSzliO4KFBCLF07MvJrdRgndfIvBynKvFSs1rf7jV0i78TfbPDIvr66nE7XH9ox/2eypsxKYmaQwpsYYRSMLk4IGDk91Xmtrb+H4da0ez1DwNptut+HVbgaOFDPucxEErhMpEXZGO9fMTrhiOz/4RTw7/ANADS/8AwDj/AMK9CLlbTX5m2O5lNOMUt/Lq+1/Q16x9X1G9W+ttM0URfbJ8vLNNEZI7WIA/OyhlzuYBANwPJbkIRS/8Ip4d/wCgBpf/AIBx/wCFZt9HB4Y1Sy/4R+yjDXsjefpdoio1wiof3iKSqKysYwzsVBXCklvLFVeXU5qPM5e8l9//AAP69Ctc61qtqsBk8TaKTcTyW8Mceh3EjySRsVdVVbgk7SpB47Ulhresalq0mnWniDTGuY/Nzv8ADt0iN5Ugjk2u04V9rkA7SetYLw6gIo7lrPV7Q22qTzWl1bPYIyGedzJA5e6ZHyz7B8oOVHGeKr6SNYj8Q3t9okd5dXMM11H9mdLKZLVZpzLIjKt6G37woySBhT8uTkRd3PU9lDlb5l+H+Xp+J0M/iTUbbU77T5/E+jpc6fA9xcK2g3IVUVEdsObjaxCyISASfmHFWptS8SWVjaahNqOk3EMl5bW00Cae6OvmXCQuu8XLqGUsf7wyuK4ufSU1TW9YvPFVpq15s2z3ljaXVnaQ237lULyAXjOAYk/ibGC3rWxHdrD4P06y07RtQXTn1m1kjvZLq2mjAbUUfBZJ3ZuTtzySeuOcCb1CVOC5eVp7X0j+HU9Kooorc8kKKKKACsNbfWbHVNSlsrSwuILy4WZWmvHiZcQxxkECJh1QnOe9blFTKPMRKPNbW1jI+0+Iv+gXpf8A4MpP/jFH2nxF/wBAvTP/AAZSf/GK16831TSfE9z4KisLbTrg30Woy3D+ZJbOsyuZnTIZipAdosgjqMrnANEabf2vy/yMKjnBaNv7v8i8tz4si8fXE8fh238q4sooJJDeExfu3kYMJNmR/rWBXaTwD0ro/tPiL/oF6X/4MpP/AIxXMa94W17UPEOoXMEyf2e4m8m3BIZ5JLLyt5O4DAZVA/3mPHWtBbLxPaatGySpNHcamJbyS3jSNGh8mJRhZHZh8ykHBzgMepFKNFr7f5f5GcZVbvmcnr5f5GpJdeJxjydH0lvXdqsq4/8AJc1xHxcufET/AAt1ddR0vTILY+Tvkg1KSVx+/TGFMCg84/iHrz0r1Gua+IXh278WeBNQ0XTpIYrm68rY87EINsquckAnop7UpRfK1c9fCTjTqw5ujWvz+78DC8cvdx/FPwG2nQQz3I/tHZHPMYkb9yucsFYjjP8ACfT3rpPtfi3/AKAmi/8Ag4l/+Rqg1vw7d6l478L61BJCttpH2vz0diHbzYgi7QBg4I5yR+NdLTSd2VUqR5KasnZef8z8zI+0+Iv+gXpf/gyk/wDjFH2nxF/0C9L/APBlJ/8AGK16KXK/5n+H+Rwckv5n+H+Rw3jjUvEVr4bLf2Ra/wDHxAVe2vWlZHWVWX5TEucsAOD3rtLd5JLWJ54/KlZAXjznY2ORnvisvxQypo8Jc4B1GxX8TdRAfzrZqYxaqNt30X6kQpzjUcm200v1CiiitjcKKKKACsvw/e/b9MlmMMUO2+u4dsXQ+XcyJu+p25PuTWpWP4YuVutJnkSFIQuo3se2POCVupVLc9yRk9sk44rRL903bqv1F1NiiiisxmDef8lD0j/sF33/AKNtK5Btb1OfSvDX2mz8TyQWtn/xN0trSeKdp/LUICSFZxuEmShxnbnAIrvtT0PSda8r+2NLstQ8nPl/ardJdmcZxuBxnA/IVQ/4Qbwl/wBCvov/AIL4v/iazcW3odlKrTilzb/8P/n95ymg6pq+l+L9Tl1aHXn01pbpSXtZ7iPf9pYweUoVmC+VnJUbMBO55dJLq+sXfiOTSJ9ZtLie7sxpz3FtdxRxQg24kYJIoTG4PkYDEB8/Kc11P/CDeEv+hX0X/wAF8X/xNQXfhTwRYRCS+0Dw/bRk4DzWcCAn0yRS5ZbGjxFFPn6+itvfv/SMh7jUrrwrp9repJB4ludQea3hEu5rYC5ZixOBmFIzgjgMpCAgsprpfs3iL/oKaX/4LZP/AI/UFtZeEvDrSX1nbaLpRCtG9xFHFD8odVZSwxwH2gj1wOuK24pY54UlgkWSNwGV0bIYHuCOtPk7nFWlTqu0fN9t/R7GX9m8Rf8AQU0v/wAFsn/x+sPxjaas/he6TVNfsLSzfajywaRO8qszqEKBZyd28rjAPOK2IfEUl35rWOh6jcxRTyweajW6qzRyNG2A0oONynqBVLXrm91XSJbB/CeoTpNjcsptXThgfmX7SpI46Z+voY91rS/4k4blhiI7qz63/XT7+py+naPrFitrYWmom8lbw7qa2DizksrgSmS3+ZzJITuLkHJC4OT3GO78K293baDHHqMV1Hchj5n2qYyMx9QTLLge27qDwM1wlgupaK6W0enTaZdWmiaxPbvJFAke954JAEjjllCqhIGGPTHXnHQWWtao2lXNrHPdSapdXrW9nHfxwia3QLGJJJFiAXahLOP7wZBn5hnkqRlJNLy/L8jXGziq/M3fR66Wer/FnaVT1j7YNDvv7KGb77NJ9mHH+s2nb146461zml+JtRvvG8/h6W2kiFi80810yDbNHn93Go7YWSNi2c/LjHzZrrZI0miaOVFeNwVZWGQwPUEVzSi6clf1MYzVWL5fQ4fwdqM134rvYhdNFHHb5m06e9uZpEcspVwtxEjKu0nOCRyvAruqgs7G00+3FvYWsNrCCSI4Iwignk8Dip6KklOV0gpQcI2kzD8UWWr3dnD/AGBcGC4LtFKfMKgRSKVL47shKuOh+QgEbjnkddsdXh0XxTNew6iludK1fzJZ77zIZdz7oPLj8xtuIww+6uOlelVg+Ov+Sd+I/wDsFXX/AKKanGo1HlsaU6SdeMr9UYR8I+KX8TjW5fFlrKyZMVm+mSG3hJQIWVftGckA9z95vWug+zeIv+gppf8A4LZP/j9a9FetyL+mxVpyrW5+mitp+VjA1DSde1LTLqxn1bTliuoXhcpprhgGUg4zOeea36KKcYqLujGMIxd1+YVm6rpAv3huraX7LqFscwXIXdgHqjAEbkOBlcjoCCCARpV59YeLNUudA1eW2eS61qzvEV7B/LCRBrtkEKkLnJRcZbJ6GtFHmCVZUpL5/gdnpelxaXatGjvNLI3mTzynLzPjBZj+AAHQAADgVeri9M17VZvEWjWs9wJPtOm2s1zAqLjLx3DPKMDcPnjiXrtG/GMkV2lDjy6BGr7W8gooopFhRRRQAUUUUAFY9zdOnjbTLUBPLl067kYlRuyslsBg9cfOc446Z6Ctise5uVXxtplsYULSaddyCU53KFktgVHbB3gnPPyjHetaPxbdH+TEzYooorIYUUUUAFFFFABRRRQBj+ErlLzwVolzFCkCTafbyLEmdqAxqQoyScDOOSa2Ky/DNzBeeEtIurO3+y289jDJFBv3eUhjBC574BxmtStK38SWnViWwUUUVmMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKydXvbi11TQobc4jvL9oZxtzlBbTuOe3zIvP4d61qx9aunt9W8PRoEK3GovG+5QSALWd+M8g5UdOcZ7ZrWkry26P8mJmxRRRWQwooooAKKKKACiiigAooooAKw7n/AJKJpH/YKvv/AEbaVuVh3P8AyUTSP+wVff8Ao20rnxH8Jm1H4vk/yZ0FFFFeQIKKKKACiiigAooooAKzde0t9Y0k2cU727GeGQSoQGTZKj5GQRn5eMgjNaVY/iq0vb7w9Ja6XPLbXMs8CiaIuGRfOTefkIbG3dnBHGeR1q4fGjOp8D0voU7Tw/d6FfxNoXkTW8kUFvObyVvMVUlld3BCne7ec3XaAR3zxyfiHwzcxeG9C0bUoDdb9emlSLTxE7FWhuZQALgCM4JOdw6DjnFayWt/4YvrYagNf8QBYAI5bKSTYjCWV23o0vPyvGo3biQmPasPxXbS3Xh/S2fXLzw39p14yx3V/LJvsVFnKAm6RlPzbeQG2gyMAWA56Xe+99vmPA2WIjZWav8ALTy3+XyLGh3FtLq1hoUGs63Yzosk0NtNBpkYiKyTRMFVY8k5jk5TPytnOCa7P+xr7/oZNU/792v/AMZrhtO0e9m1fQjb/EjSLprVN1lYw2MYSdVEqFlRJh0VpFJUAfLz92u5+zeIv+gppf8A4LZP/j9dUbtap/f/AME3zFR9ouVN6dLrq+/KH9jX3/Qyap/37tf/AIzWN9ll8LeI21XVb+61eG9iW1iZ4g89swy5SOONcuHAywUZHlAkFQSuz9m8Rf8AQU0v/wAFsn/x+se1i1C38dQDxTcQXglhY6TJBAYo4ZFB81SCx/eMjAqck7VkAx826ktdn9//AATmwytJ6PZ7v8tX6+ifo+Xae7e/tZJbOaa0s76+mFrLpGoFbqO5dyCw+zYRlVyuMODubkd4tP1Wbw94qk1T/hHNXvoX+2rvsNNujNKJrrz496yxRqoQbhgO3LdOtal9Pquo3FtY3J8R21jHf6gt7PYxSxyKDM32Yh9u549mcGPIHyg1b0fUbzTNamnvk1t7ZrnUEug9rdXA3m5BtfLXa2E8kPzGNnTdyVpdT2W1y7X30v6+XXX715W5ecy69r+p6heaTrFnY3hZltpNPvI7iUNDBC8MpjgkURMsTN8rFshMFOa2otUEXg/TtKvE1I3f9s2rKZrO9ZI0Ooo6IZ54lztQquWxnGPSsvX7/wAV3up+I5/CVtrbyzRlIFu4bmCBbcwwqzRK+0LOJNxXAzjzCc/LXRQy303w10ttWjmjul1e0RhMsquyrqSKrESszjcoVsEnrxxihbsdS3LC/daX20O9oooroPFCiiigAooooAKKKKACiiigArL8RWk19oskFvH5x82J5IcgedGsqtJHzgHcgZcEgHOCQCa1K57Vdf8AsviG10+21HT4yySPcRToS8SrGzeZneBgHZ8uMlSTnAyHG99DOo0o2l10+8rw2lze+JtNmuvDv2CGwjLQ3KNCxUlGURkhtyqFblQpG88HC7m6muCh+IUiXdhC32LUYdQvzbQ3MLNb4UNAh/dtuJcNM3BK5VM9DXQeEr7WdT0SO916KGCWdVdIoodmARnOfMfIOeM7TxyKqUXa7MqVSDfLHV/18jdoooqDpOG8S6PrbX7y3Osatd6TJKksNrpttb+bbzRuJIwd0Z3qXVRuJG3jdlSzLv8Ahq01iC0ll1zULi5knYNFBcLDutlx91miRVZieTjIHAGcZL7TxRpN5DbzRTyxxXUiR273FtLCszMpZQpdRuyFPI9vUUWXinRtQ1BLG0vQ9xIgkRTGyhwUWT5WIAJ2OpwDkA80uRp3Llioygoafh/X+Zr0UUUyAooooAKx/DElvLpMzWcTRRjUL1SrMCS4upQ54A4LBiPY9+tbFZfh82Z02X+zQ4h+3Xe7fjPmfaJPM6dt+7HfGM81ov4b9V+oupqUUUVmMKKwdYvdW/4SHT9L0e4srb7Ra3FxJLdWrz/6t4VAAWRMZ8085PQVlT63q1ta2FxJ4m0Vk1JQ1msWhXEjzgruyqLcFjwQenFTzWNo0XJJ339f0XkdnXOeK9E1PU2tLrQp7WC/tFkEUlwudhcD5lJVgCNuOVOQWHBwRQ07WtW1bUrrT7DxNosl3alhLE2h3CEbXMbEbrgbgHUrlcjI61BN4l1G31e80yXxNpAvLJA80Q8P3RxnZgKRPh2JljAVSWJcADNCnbUJYVzTg3+D/wAgm8EX8OoT3thcQSNJcNc/Z5nKJ5hvIpydwUkZjiVTgdVzg5JrQu9B15NASLRNZ/s+8FzPcyKkaOkvmytIY9zo23G7aH2+5U9BJa6zqFjDBea5cW8+n3LpCs0WnzWbwSM4RA8cjsdrM2M/Ljjghsr0tU5OSszGFFYefPH/AIH4nC+G9J15NQia11LVbTSEmee4i1FLbzZ5XcvIgRYQVXe7Zffzg7RtKvWr4w1C2l0q+0IXF5BdXdqVMlrpk935SPlckRqQCcMBkjpntXS1kWx/4rLUf+wfaf8Aoy4rJq1l3/yNZ4h+1jO3X/NnG6f4muk1KK/1OG6mvdN0fV5ZPPsJbJZkW4gMe0OvdFXpuK5G7nrowePrmeG0vpNChhjknmtWu5p5VjgwYhhn8jK7nYrhgozH16VsajbxXfjvTbe4XfFNo+oRuuSMqZbQEcexrSuvD+m3kZiubdnhM3ntD5ziNnyGyUB2n5lDYIxnJ6k54qjpxm1NX/4YWK9pUalTdlbb5vy2NKiiiuIYUVE1zCl5Has4E8sbSIncqpUMfwLr+dYjeOdATSV1OS4uUsnYKkzWE4D5VnyuUyRtRiSOAByapQlLZESqQj8TsdBWD46/5J34j/7BV1/6KatKLVbWaW9VX2x2JUTTOQI1JQPjOeyspJ6fMOeDjnvGmrwS+GfFmkGOWO4g0SecFwNsqNE4yvOTgjByB1HrQos1pSj7SOvU6mivJ/8AhNNVu/D+mXVhrsE00c15BPFEyRveuj5hWMtDKASgOFJBbcOSQTXrFe1GalsVXw06Fufrf8HYKKKKs5gooooAKK5Y6vq+pXQk0290/T7Ke5NtYm7sZJ2vCsZcyKVlTC/LIACOQm4EhhWY/inUUivJW8Q2HlWU3kTSDwxelfM80RbFYTYdt5C4XP5VHMjoWHk/6f8Akd5RXG3Gr65B4Uk8RJr+jzaelobxWj0WUM6bdwwGuhyRxg454OKtq3it7ZnXVdJNwsAma0/sh/NAIOFI+14BJUjk4yDzxT5hextvJfj/AJHT0VyVnrGtG48P3EupaZfadrMu1DBp0tu+w20kysC0zY/1Y4K9+1dbTTuRODg7MKKKKZmFZdxc26eLbC1e33XMtjcyRz7vuIskAZcd8l0Oe233rUrHuZLceNtMjeJjctp120cu4YVBJbblxjJyShznjaeuRjSkry+T/JiZsUUUVmMKKKKACiiigAooooAx/CT28vgrRJLGMxWzafbtDGzbiiGNdoJwM4GOcD6VsVz2kaJrOi+EbHR7XVbF7ixijt4rmTT3KGJECgFBMDu4zkMB7VYS18TCzkWTV9Ja5LAxyLpcgRVwcgr9oJJzjncMYPBzx01IwlOTU1a/n/kJXsbNFY1va+JljmF1q+kyOUxEY9LkQI2RywNwdwxkYBHJBzxgvsrfxBH539panplxmMiLyNNki2P2LZnbcPYYPuKzdOK+2vx/yC5rUVjW1r4mWZjeavpMsW1gqxaXJGwbHynJuGyAcEjHI4yOtJbWvihbuNrvWNIltwwLxxaVKjsvcBjcMAffB+lHs4/zr8f8gubVFYRtPFnnZGtaMI8/dOjy5x9ftP8ASpLi18TNeO1rq+kx2xbKxyaXI7hc9CwuACcd9o+lP2cf51+P+QXNmism/t/EMl2W0vVNMtrfAxHcabJM445+ZZ0HX/Zpl1a+JnkU2Wr6TEmxQwl0qSQlsDJBFwuATkgYOBxk9aSpxdvfX4/5Bc2aKxprXxM0MIt9X0mOVVxM0mlyMrtk8qBcDaMYGCTyCc84Ek9v4gawgS21PTY7tc+fLJp0jxvzxtQTgrx6s2faj2cf5l+P+QXNWisZrXxMbJFTV9JF0HJeQ6XIUK8YAT7RkHrzuOcjgY5EtfEws5Fk1fSWuSwMci6XIEVcHIK/aCSc453DGDwc8Hs4/wA6/H/ILmzRWVb2/iBbKdbvU9Nlumx5EkWnSIieu5DOS2e2GXHvSWVv4gj87+0tT0y4zGRF5GmyRbH7FsztuHsMH3FLkjr7y/H/ACA1qKxra18TLMxvNX0mWLawVYtLkjYNj5Tk3DZAOCRjkcZHWktrXxQt3G13rGkS24YF44tKlR2XuAxuGAPvg/Sn7OP86/H/ACC5tUVjwW3iNb1WudV0uS13DdFHpkiOR3Ac3BAPvtpZrbxG147W+q6WlsWJSN9MkZwvYFxOAT77R9KPZxv8a/H/ACC5r0VjXFt4la7drTVtKityxKRyaXI7hewLC4AJ99o+lOuLbxG0ym01XS4otq7ll0yR2LY5IInGATkgY4HGT1o9nH+dfj/kFzXorLvLfXn8n+z9S06DCAS+fp8ku9+5XEy7R7HOPU1HcW3iRooha6rpccgXErSaZI4ZsnkAXA2jGBgk9Cc84CUIv7S/H/IDYorIe28Rm0RYtV0tbgMd8jaZIUZeMAL54IPXncc5HAxyi23iMWbK+q6WbncNsg0yQIBzkFPPyT053DGDwc8P2cf51+P+QXNiisuGDXVsZluNS06S6bHkyx6e6InruQzEt+DDHvTIbbxGsMouNV0uSUr+6ZNMkVUbI5YGc7hjIwCOSDnjBXJH+Zfj/kBr1j61crBq3h6NoUkNxqLxqzZzGfss7blx3wpHPGCe+KLW28SI7G91XS5V2ttEWmSRkNjgkm4bIBwSOMjjI602303WZL+2m1i/026itpDJGtvp8kLqxRkyGM7Do56jv+NaRjCLbck9H37PyDU2qKKK5xhRRRQAUUUUAFFFFABRRRQAVh3P/JRNI/7BV9/6NtK3Kw7n/komkf8AYKvv/RtpXPiP4TNqPxfJ/kzoKKKK8gQUUUUAFFFFABRRRQAVk+J57230F30x5Y7kzwIDCELlWmRWC+Z8m4qSBu4yRWtWZ4h1f+wtFkv9kL7JIo/383kxrvkVNzPg7VG7JODwKqF+dWVyKluR3dtDmf8AhIb6LVo7PXdZtfD/AJVr5pjvGg82YNNKikndt3COMN8vAdhkEDacLVdU1nULO1/sXUzqt1b+JGhsrqJbeUmP+z5GJGDHGxG9zyR6c4weysPEuo6jLFDb6RC0qlHucXvyJC80kayRts/eAiJn5C5BXGc8cb478UaLr/hTQtQMdpqFvHq6PLaPcoi5NvOyLI7lVU/d3A8A5HzY56tU/ht+n6hgbPEQSlfffrp5tJ/gLp+gatB4s0nXLjQ9Tkmso2WY+RZo0zsZyz7vtZ2BjcFmUAglRjA4Hdf2zff9C3qn/fy1/wDj1cVYal4Wnk8PNcaxoyyWkktzLJJf24kRst5cCqrtiMNKWVQxCiJRya7X/hK/Dv8A0HtL/wDAyP8Axrqjyr7VvuOjMajc4+0nbTyXV97/ANMP7Zvv+hb1T/v5a/8Ax6s+4j/4SbVorHWLN7Syth576feIj/biQVBbaWRo0JztyW3hCQoCl9D/AISvw7/0HtL/APAyP/Gsa61Gz8X+ILWx0C/jb+y83U+pWrI72rsrRoiBgVLOrSZJDAKp4DFWFpp7Sv8AccuGac3ad9PLTz0+7597GFcWOhZ09YvC3hi2S5vr+3lvLnTkMUKW7yAcYHzFYyeWA+VjUfh6z0DVPGU2lXPh/wALvEFvmEEOjoksAguhChZixDbhuPCryKzLK7NzHIdNlvGubnUys9qP7NSYsgnlEkqfZcrLm3ZlDYB3KQ/XGjpGk6L4pvmtU1C7ZLqCPUj59hppaSRlV2Lr9mPzqtxESxznzSATg1K1PblGUU03+f8AX/DeoradpT+Or7RovDXheK1jdoombR1eSMrBBM0j4YAp+9KfwkFk6/NieDRdJl8DaJro0DR7S/n1DT5o5rLT0gKxvexbCMFipMbDI3HkkZIqz4U8HK+j3jW+qz2STXd3azR2un2EaypFPJEu4C3+b5V5zxknp0roJPCs9z5C33iXVrqCGeGfyJI7VVZopFkUEpApxuQdCKpRvrYwqVlGSjzbWvvrbfp1OiooorY8sKKKKACiiigAooooAKKKKACiiigCJoI2ukuCuZY0ZFbJ4VipI/8AHV/KpaKKACiiigDkYfhzpcVs8Mt5ezkxwxxySGIPD5UbxxlSqD5lEhIY5OVX0q3pPgnTtH1KG8guLuVoIljjjlkXYpWFId2AoyxSMdeOTgDNdHRVc8jFUKatZbBRRRUmwUUUUAFY/hhbVdJmFi7vH/aF6WLrtO/7VLvHU8B9wB7jB46VsVj+GYoodJmWCdZ1OoXrFkBADG6lLLyBypJU+46nrWq/hP1X6i6mxRRRWQzntaF/a+KtM1Oz0m61KGKyureRbWSFWRne3ZT+8dARiJuhPauKh0fxdBa6QsFrqUFzo1obOzmisbLBjKqreYrXrbj+7Qggrgg8HPHq1FQ4JnVTxMqaskv6v/mzy6z0rxPpuuXGq2Gnan5rvcCGK4trOSOKOeczOMC9Ult20A5AwD8uTkPtNI1uLV9X1a88PXd1fatH5Vyr2dn5DoFiVVKG8LYAR8jdhjJyPlAr06il7NF/XJvojhNL0zWr7S7Xw9qWmXFjpqu8t3O/kRrLH5u9LWOKOaTbHghDkkbEK/xcdJ/winh3/oAaX/4Bx/4Vr0U+SL31OWtJVneSX/DmR/winh3/AKAGl/8AgHH/AIVcsdK07TN/9m2FrZ+Zjf8AZ4Vj3YzjOBzjJ/OrdRXMzwW7SRW8lyy4xFEVDNz23ED35NChFapGMaUE9EjIuf8Akomkf9gq+/8ARtpXQVyOn6omteLdC1CGCaCOfStQKpNt3YE9qA2VLKQcZBBIIIPeuury8RrUZ2VIuKjF7r/NhRRRXOZmVqelLf6paTxarc6feQwyon2fyiXjZoy+VkRuAUTkAYz71xw8P+DD4SGmL4ujXTYZwRLHdWi7XaJ0KllQAlkZvvZY7cg8Gut1W11T+27LUNKgtLgQ288EkdzcND99omBBWN8/6s8cdRXJXXgzxNd+A7PQpprB7izOxJXuWKmM28kOBthUjG8EA7iectXZRdoq8rfd5/18zgrK8naN/v8AL+vkdB/wj9hqDa5pMt0GtriWN5IbS6aOSHMCR+W4U/dKxgjPBDHjgE43ibTlh0fxW4vIpksfDs1tDE1y01wm9Hd2lLZYbiq7eTkKelbel6LcW82rWyXl7aO88LpdoiFpUW2jj6urKTuRsjGRx2Izg+KrOeOz8VSH7VKkHhmeGa8njVBdOyuy42qqsUAYZUcb8Gpu9Vfp/l/wx0UUvawfL9r9X/w5zEmpaBF4TsLyTwLo39ozyXRuY4NLimW1itnPmMRlQxC7R94DLZGeFPcr4R0q01Rf+Kf0+5juGYljbriIbvTGOmO3rzVQfC/TfscVqdQnaCEs0cb6dp7BS2NxGbbqdq5PfA9K6D+xr7/oZNU/792v/wAZruimvijf7jpzJ0a/LyK9m36Xfn5aB/winh3/AKAGl/8AgHH/AIUf8Ip4d/6AGl/+Acf+FH9jX3/Qyap/37tf/jNRW63tj4mtrSbVbq9gns55StwkQ2sjwgEFEU9HbrStBbw/I87lgt6f5f5kv/CKeHf+gBpf/gHH/hSN4S8OOpVvD+llWGCDZR8j8q2KK09nDsjZUqad1FfccLqdn4ltEsLa0trrUJNMvXubS+2wzK8TRyxrHKslxE5dVlxuyc7VYkliBiRaJ4jTSdQtBok8c17eLfG7t9PtI5DMLkXA3n7cS6g5UDggY5459Voo5Ed8cXOKskv61PPjBq8HgeHRrfwxrNxeW3l3CXE01monuUlEwZ/9IYhWlXLAZOCcHPNSzrqkfiC/1+w8La81/cWkcEUMl9bRwhk8zBYLcEMPnHVTjBI6mu8op8gvrL191a+vX5+Rwmh2us/Y/BunXegXtn/Y2z7VcTTW7R/LZyw/LslZjl3XHy9OuK7uiimlYyqVHUd2rf1cKKKKoyCse5W1PjbTWd3F2NOuxEoX5SnmW28k54IITAxzk9Mc7FZk8dqfFlhK84W7WyuVih2cvGXg3tntghBjvu9q0pO0vk/yYmadFFFZjCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKw7n/komkf8AYKvv/RtpW5WHc/8AJRNI/wCwVff+jbSufEfwmbUfi+T/ACZ0FFFFeQIKKKKACiiigAorI1LxDFp+qRadHayXl3LH5iQw3FurkZI+7JIrHoeQMe9Z8XjeKe3juINJupIZJBEkiXliVZz0UEXGCeRx1rRU5NXMnVgnZnT1Q1q7sLDS2utWLLbRSRMSqO53+YuzCoCSd+3gCshfG9uWkEmnTxCGTyZWe9ssRvz8rYn4PynjrwfSr3iD+yL7RPL1a8s47GRop2a4dfLkRJUbnJwVJ2r6fMOucFqm4yXN+AnUUoPl/Ezxf+GvEWpWksQu57ldkqNFBcxgbJXVPNwAMCRJMCTphjjBNYXizWdDSx0PxBb6hcaZZ3WsCSXUY7YhyTZTIrBZI2yCAq52kYyeOTXRJplha6vp0+iarbaZbzRhBZW0UIS8RSzgLxkAF3J29dx+tc8/h3SI49B05/7N1SCbXXN3HDbRrbmVLGdSPKBKqR5akj+9k8dBtFRuuVvpYvCyarJzStre1u3Tf8V/wcvS/F+qnx9oGmya9Hd2F1CzSJ+4ZpN5ujHkxrgsoSFWKMFDDvuzXqlcdofh/wAL6nJfiTwVpdhLY3fkFJbOBn/1aSBsqCBkSDgE/nwNr/hFPDv/AEANL/8AAOP/AArujzpdzpx1RSmvZwSstdl530Xma9clrxl/tdR4O+z/APCQiMeYJAfs3k5OPtO3nHDbMfNuzj5d9a3/AAinh3/oAaX/AOAcf+FZzaAvhrWDqvhjS0MVxH5V/YWgjjafB/dyLuKruTLAglcq5OSVUF+891/X3HPh5SU/eS27/notP1seY3EHiIWs2sXM/hyNGni23r3OsI0rhZFj8l925wVkkx5ZIIbPPBrY8M6Z4yV/7S8OQeCg6W0NqZc3wYRCKNo0Ifn/AFZjPPPPPJNaMWl+IIdSuL9dF1DzTcx3VrAsFt9mglCypI/lG+5LiYklSvzKrHJznZ0q71/T9V1K5n8O6nPDfypO0SJaIySCGOI4b7WflIizjGRnqcVCjrqexUre61Fp/P00ND4fNPN4F068vAgnvxJfyCMEKDPI02BnnH7z3+prpKyfCllcab4N0WxvY/KubWwghmTcDtdY1DDI4PIPStato7I8ms06kmu7CiiiqMgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKyvD1kthpk0KTxzhr67m3xtuAMlzI5X6qW2kdiCK1ax/DNsbTSZo2aNi2oXsmY2DD57qVsZBPPzYI7HI46Vov4T16r9RdTYooorMYUUUUAFFFFABRRRQAV5tcN40jR3/ALbs9D8M/MsN9qUW+9iU8KZt7qoUnhTkycx7xu3kek15L4o8Sa14qsf+Ee0rTGu5Gl3DUrORBaXskLLII4naVCOmWwxZCjqN20uNY83s58qvoa0EnWipOyv5froXvKvoUgj0vU7LUYovDGpQ6Y+jwOhUo1soAbzZC7ZAAwQQQep6WbpNclhtrqzi12KMXMkdjAZZCY2/c+W1xubc0JdJSxbJCyHgcYpadcX3hxoi+jXWnzafoer3caXs0cyszTwS7QySMzBSQCzbS3XrnG7Z+OSt9BDe6jpNzavqMlq1/bny4igtklBGXYZ8xwhO7HQYya8x8120v6sRmHJ7VJy+en8zs9PxO3ooorziwooooAKwfHX/ACTvxH/2Crr/ANFNW9WD46/5J34j/wCwVdf+imoW5pS/iR9UbVFFFe+cwVi3UuPHelxY+9pt62c9MS2v+NaV9ZQalp1zY3ieZb3UTQypkjcjAgjI9jXDD4fpG39mnw/4cnt5AQurrp8Mc8K8cGLYVd8Z+cELk5KYGDErs2p06c177/rvv07HoNFV7Gyg03Traxs08u3tYlhiTJO1FAAGT7CrFUZO19AooopiCiiigAooooAKKKKACse5gibxtps7TqsyafdosJzudWktiWHGMAqo6/xDr22Kx7m2L+NdNut0YEWn3cZUsNx3SWxyBnJHycnHGR0zzrS+L5P8mJmxRRRWQwooooAKKKKACiiigAooooAKKKKACiiigDA0nxjp2r+LNX8OxR3EGoaTsMyzqoEityGTDEkY2nkD7y/gvi7xjpPgjRRqeuSSLC0giRIk3O7HJwB9ATXnHirf4b8War48soSz6XrsVtqARfmks5LK2DA+u1iCPQkmo/itLB4i8CeKtfhuI7mzslt9P06SFg8ThpoJJpVccNuYpH3wYDzkkD3qeX0qlai9eSXKn6uya/FP09DJzaT7nrst80WjPqBs7l2S3M/2RFDTEhd3lgA4L9sZxnvViGQy28cjRPCzqGMcmNyEjocEjI9ia5XV9S/tH4gW3hGY3Ntavprai89rdNC8xWQIItyYdQM7iVYE4A6ZB5TVPGGs+FtG+INtHcPcr4f+zLptzcMGkT7QgwGZs79hYEFslu5NcNPBTqpRj8Ts/k2or8X9xTkkdzqfiyHTPG2i+G3s53l1dZnjuMqI0EaFmHXJPA7AfN17V0FeV6ppkWn/ABd+HMyXl5ey3Ed+ZZ7m5kkEhFsvzKhO2POckIFHtwKqaZeXrfDT4hyXmrajJPpOq6jHZ3H22USReXEBGAwbOATnaTjPJreWBhKFOVN7pX825yj+iFzO7ud/qfiyLTPGuieHJLKd5dXWZo7nKiNRGhZh1yTwOMAfMOe1dBXkcFzLd+J/g1c3Usk082mXTySyMWZ2NlGSxJ5JJOc1saW174z1Lxa82qX1lPpeoSWOnRW9w8aW2yNcSsqECXc/zbZNw4xjBIM1sFGMYu9ko6vz55R/Rf8AD6DUj0SiuZ+Heu3fiX4eaPq2pD/S7iD982AN7KxUvgcDdt3YHTNclp1lrPiD4h+L9Gu/F+vQ2uk/Y/INq8EbHzYy7A4ix1HGAOOua544N89SE5Jcm+76qOlvNj5tF5nqdZ/2zUf+Ek+xf2X/AMSz7J5v9o/aF/12/Hk+V977vzbunavO7YeIZfDvhHxfc+Jr9p7iSwS6skQLbTRzyJGfkA+9iQEsSQSOAMrjQudc1DR/jDqNtc6heXumweGpdTW0bYAjicAhQqrn5VwC2SMnnmtVgWm0mpWT77rtt/kLmPRKK4fw/p9/4g0HQPEtnr93bX155F5fgu80EsZXMlukLN5cYzhQ4XeAvJJJJxdA0/VNV+K3iexvfFevNa6HcWM8ESzxosu9PMZJAqAFMrjC7cjOc9ahYOL57zXuLXR90vzY+bbQ9SrH1TxFDp2pQ6bb2lxqWoyxNOLO0aIOsSkKZD5jooG5gOuSTwDg487ttQ1XXPhTq/jaLWL611qA3c9uizN5NqkUjYh8kfupPkUjc6sfmzngY2dZ0C88YQ6P4p8N6qNE8TwafHKqHEiNHKN3lyL/AHdwYBsHoeDjjVYKFOdqslZNxe9lJd7a281/ldcza0Oh0jxcNWn1W3Oh6tY3OmRpI8F3HGrTBwxURlXKt9wjqBkj3xm6Z8SI9Y0KLWtN8Ma9cabMHKXEccDZCMVb5BMX6qf4eccUvgnxXd6zqF/pXiXR10nxFYxR/aEVgy3ERztdGHVcluMnG7ryapfA3/kjOh/9vH/pRJVVKFOlCc5w1Tit7qzjJ3TXe116iTbaOr8PeJNJ8VaSmpaDeJd2zHaWUEFG7qwPIPsa1a82+HcVvF8TviGNPiSKzW6tERYlCoJBG/mYA6Hd1/yK9JrjxdGNGs4Q2snrvqk7fK9iou6CiiiuUoKKKKACiiigAooooAKKKKACiiigAooooAKw7n/komkf9gq+/wDRtpW5WHc/8lE0j/sFX3/o20rnxH8Jm1H4vk/yZ0FFFFeQIKKKKACiiigDndU8O3l74kj1GC+MVvi1Wa14xOIpZJBk7SVILgjB+bBBxnIwJPBevP4Q0yxM1pJqumo0dvcSTgxxqYxHt2G3IdSuRhhn/aOePQaK2jWlE55YeErnKWvhrUvst3bX1zG0R1hb+1CuCI0+1eeRjywQSOOWbk8ECpL3w7d6zd313dSG1lM9vHZMrKWhhimSUuMhhudlJwQQQsYI4NdPRS9tK9x+xha39f1qctZeF59O8QrchINStzCFNxeygTpIbiSZ2CrFtOWdCMFcFB9a5y08M6xo2m+G9K1HVHhnOtEwz2k3nmFBp8wIXzkIALK527SAH4r0ysDxH/yGvC3/AGFX/wDSK5rSnVlKST6tGtGjGE+ZdE/yOXuPhfLFrOn3mja5PYtDN5lzcR2trFK6j/lmPKhTIbJzuYr8oyrcFe00WxOm6PDaNFFGYtwIidnDfMTvJf5izZ3HJJyTkt1M1vqNleXFxb2l5bzzWxCzxxSqzRE5wGAOV6Hr6GrNevzvkUFsOpKcpXnuFFFFIzCiiigAooooAKKKKACiiigAooooAKKKKACiiigAoorL8Rz6pbeHryXQoEnvVhcxq7EEHacFQFbc2cYU4B9RQtWKT5Vc1KK4/VrfWJtat5RcXEM1zbC2eztPPeKHLOGnWb5Ywyq+7DoSfLAHJBrV8Ktq02kyXPiBTFeTzEtBg7YdoEeF5+6xQuP9/v1NONlczjUvLlt/X9dDboooqTUKKKKACiiigAooooAKx/DFrNZ6RPFcIyO2o30oDLj5XupXU/irA571sVh+EVkTRbgTAhv7T1AjJzwbyYj9MVqr+yl6r8mLqblFFFZDCiiigAooooAKKKKACuB8V+L9LvdLuNAvryXw/qVyrbo74pGVjUgk7vNVWWQDYCkmcsem19vfV53Pd+F9D1rwtPpE9hp8F3eXPnyuRHLMfJcP5ruQ5bzEQMH53queVqvd5JcyNqCl7VOO61+7Xsyh4c1y306HTpU1Gz1a50nQNUnlitrtZCiia3eONirPtO0Bc5b7p5bGT1k3jC/t7V5pNC/49bo216gvF3Rt+7KeWCB5u5ZAQODxjGTXE6Vr39rfE7XbTxFqBu9LstJ1NZovLVfs8f2sIyZjAY/uo0POW5461197F4Q8iyvrwX4a0mZIps3gnilkCDbIR+8DMDGAH5IxjivLfI5axb/4b1/r8C8zVSFX4ktL/e2+39fidlRRRXEZhRRRQAVg+Ov+Sd+I/wDsFXX/AKKat6sHx1/yTvxH/wBgq6/9FNQtzSl/Ej6o2qKKK985jL13Wf7GtoHS3+0S3EwhRTII1Hys7MzH7qhUYk89KwLz4i2tpJdQfZVmngjMiGOcGKcC0NySj45AwFzj+NT3xXVX1hb6jb+TdK5XOQY5GjZTgjIZSCOCRwehI71QPhTRTYrZmxUwrEYVG9shDEISN2c8xqB17A9eauLj1MKkazfuMl0XVZdUhuhdWZs7m0uDbzReZvGdqsCrYGQVdT0HXpWnVeysYLCDyrYPgnczSStI7HAGWZiWJwAOT0AFWKh2voaxul7wUUUUFBRRRQAUUUUAFFFFABWVcWE0ni/T9RUDyLexuoHPGd0kluy+/SJq1ax7m2nbxrptyqOYI9Pu43cfdDNJbFQfchGx9DWtL4vk/wAmJmxRRRWQwooooAKTPOMfjS0UAFFFFABRRRQAUUUUAFFFFAGLD4W09LPWrW4Mt3BrdxJPdJOQR88axlVwBhQqDHU+9ZuqfDrRtU8A2/g9pLq20y3VFBt3USNt5ySVIyT8x45NdZRW8cTWi01J6NP5rZi5UZGoeHLLUdd0/WWaaC/0/csc0L7d8bfejcEEMp/MdQQarX3g3SdS07WbK8jlkGttvvJtwDtgBUAIHAVVUDj+HnJJJ6CilGvVja0ttvvv+evqFkcXF8NbZdV0W/ufEOt3kmhsxskuJICEDABlJEQZgQoHJz6EVYvfh7p17Fqlsb7UILDV5nuL2yhkRY5ZHVVLbtm8fdU4DAEjkEEg9ZRWjxldtPm29O9/z19dRcqORj+HOlxN4ZZL3UA3hnzBZHzE+ZX25V/k5G1QvGDjvnmtS28MWdjq+qajp09zZzaog+0LEylPMAwJgrAgPj8D3BraoqZYmtLSUv6vf89R8qKum6ba6Rplvp+nxeVbW6BI0yTgD1J5J9SeTWZB4SsrbWvEGqQ3F0tzr8cSXHzLiLy4zGpj+Xg4Oec8/lW7RWaqzTbT3389b/mrhZHLyeA9Pk8H2HhsXl8lpp8kckMqunmgxtuQE7cEA47fwjOec328MWT+ND4mZ5mvDp39mmIlfKMXmeZnGM7s8dcY7Vs0VbxFV397e/47/eFkctoPgS08NnydJ1bVYtOEvmrppnUwod27CnbvC5/hD4PIIIJFaOm+GrPS/E2ta7byTtdaz5H2hHYFF8lCi7QACMg85J/CtiiieIqzbcpb7+eqevzSCyRz1t4L02y03VNNs5bqGw1SRpJrZZBtTf8A6xUyMqH5BAPGTt2nmnt4VgXxVBrlpfX1pJFaJZG1gZBA8SMzKGUqT/ERkEYB4x1reoo+sVW23Lf+v0CyMjT/AA3Y6fr2oaypmn1C/wALJNO+4pGv3Y0AACqPTqTySTWbofgdPDnhRPD+j69q1vaxE+VJ/o5kjBZmYAmLHJcnkEjAwR36mij6xVtZvt+G33BZGbomg6d4dsDaaTbiGNnMsjFizyueruxyWY+pNaVFFZSlKcnKTu2MKKKKkAooooAKKKKACkJ6cZ/pS0UAFFFFABRRRQAUUUUAFYdz/wAlE0j/ALBV9/6NtK3Kw7n/AJKJpH/YKvv/AEbaVz4j+Ezaj8Xyf5M6CiiivIEFFFFABRRRQAUUUUAFZPifUL/TNAludIhSe882GKKN13Al5UQ8bl7MerAepxWtWV4k1NdH0KS+e3huBHLCNk8ojQFpVUOzEHaFzuzjjbV09ZrS5nU0g9baGBZ+JPEMd9BbazbWcNxi3ElokZ8yQzXE0e5SJGA2xRCQj5ujcgDIy/8AhINavrPw/fahpy3V7Hq48q2slERkD6ZJJj94+MgyEElhwvAzxWtpfi6x1TUrfUBpMSCS3hie+Do8kZlnliWPIHzJ5kZGQx5cHGMkVbvX9Lks/DGryiDSLIa5ciTz3WJUYQXaEseACzc/Vu9ddrS1jbVfIjCPmqK0r6PTvoM8P6p4qTULy817wXdrc3JVBLbXtqyrEpbYm0yKQBvY5yxJY9BgDpP7Zvv+hb1T/v5a/wDx6uG0T4h6Npvim9hvdUI02+aFLJ59TiuvJkwQyEJLIwVsbt54BOCR8ufR7C7+32MdyIJ7dZMlY7iPY+M8Er1XI5wcEZ5AOQO2MJOPNd2+X+R142Eva/y6Lb0877bblE6zfBSf+Eb1T/v5a/8Ax6sOK5vfFUmqpdW95p8umsVt9LW9a1kmJQFJZJImzsY7lXB2/KxO4gbezrI1bw9Dqk/2hLqexuGiMEs1usbGaI5PlusiOrLkkjIyMnBAZgTlfe5jQ9xu7/4H3HnMuq6fYJpUN7e6lLeXGqCyvI31S/tDbjCBgFa5cMymaI5VmBUkjgFhbuHij8KWmpxvqImuPDdxq5RtbvyoljSFgo/ffdzKc856c9a6SH4fQW0EkNtrF3DFJv3xx2GnqrbwA+QLbncFAPrgZ6VDa/DSwsfO+xajNb+fE0M3k6bpyeZG33kbFtypwMg8Go5X2PT9vS35vz1/DQ5lNX0qDxHo+k6jc6hGbuW6trmU+IL1NrxzSRRuqmfO12gfI52lkBbpuuavbrB46tNF06/vvLmjtZFjk1LUZGnV5JfNxKtwEjKxRM4DA7iCBmuik8BpNbiCbXL2SFY1iEbWNgVCL0XH2bGBk4HQVHZfDy201XXTtXurRZGR3EGn6egZkOUJxbclTyD2PSnyvsL21PdS/Pv6fL/LcwfB9tLrEOjHWv7Tgk1OwmvQI9YvU+VTb7SuLl8qfObk4bp8o5zj2Gu2+sQxNpC6jctNrH9mrs169YJHIvmRXDYn6BA25OCSCMpjFdg/ww02S3jt5L6RoYmZo420zTiqFsAkD7LgE7Vz9B6Veg8FvasrW3iHUYSmNpjtLBduAQMYtuwZh9GPrS5WP29PV83526/18h+iWg03xlq1jBcXslsthZzKl1ey3G12kuQxBkZiMhF6egrpaydK0I6bqF1fT6pe6jc3UUULPdCIbUjLlQBGiDrK3XPatatVsedVkpSve+35BRRRVGQUUUUAIDkAkY9j2paKKACiiigDg7RvE7/FC8mluC1nDHCP7NWX5Et5HnQSjsZMwRvzyFd1GTjPeVj2yKPG+puB8zadZgn2Elzj+ZrYqIbGk6yq2aVrafdoFFFFWZhRRRQAUUUUAIDkAkY9j2paKKACsjw1JcS6VM147SSDUL1QWJOEF1KEHPooUfhWvWT4cvbjUNLmmvDukW/vIQdu35I7mRE4/wB1V579a1V/ZP1X6i6mtRRRWQzJ1XXTpuoWtjBpd7qNzdRSzKlqYhtSMoGJMkiDrIvTPeq//CQal/0KGtf9/rL/AOSKqa7qunaP460a41e/tbGFtNvkWS6mWJS3m2pwCxAzgHj2NeenxNaTaT4Zju0029j0Wy8m6sbnXbNVuZvLRVdcSsrBSj/fx98EAnOMpSsz0KOH54ppf1r5rt+J6b/wkGpf9ChrX/f6y/8Akij/AISDUv8AoUNa/wC/1l/8kV5xoXiOw8P+L9S1MT6bJazyXautrrNpvuzJdNLFKVaZVAVCVySG+cDbgEiv9v03XPGHiG8fVNJ0KLVbCa3W+n1GzlkAdLVAhWOcnjyZiOcDdnqcVPPoa/VPed0rW31/K56rZ+IPOv4rPUNMvdKnnB+zi8MJE5UZZVMcjjcBzg4JGSMhWxsVwVjrdp4itrLRZNUtda1lZ5bn7Xps0ai0jSUhZxtLhTsdQE+Ytu2twWNdN/Y19/0Mmqf9+7X/AOM1fM+iuefiIum17v8AXfV31NYHPUY+tedah4E0eDxrpep302ovrerSyQ3N9Z3bWok2xM4G1DkKBGFABHCgszHk9f8A2Nff9DJqn/fu1/8AjNc94k8M29/q2iRal4p1+O7aeRbJrWSCIq3lMzElYhxtUj/gWO5qvazimkrX9BYZ81Vcya0evbR9mZsGmWWp+NNR0mye9gdtL1HTpJbu+muyM/YyHAkc7R+9PAIzjk9hvXXhC5mtYLa615N9xeG4nka0TzJpAF2eUSSY2VIyAwy2MnPFcdp/hmCPxvrOl2V5q8upajomoLKdZdCG3SQxJMuxeFYq3UZwg4AxnpY/BOqrOtxYvpmjBdQ+0rZ2sZljRdkCkq21NrfuXB+XlZWGeTXC5Wk/et8h5jFe0ilFy0XV+f8AW+x3lFFFcABRRRQAVg+Ov+Sd+I/+wVdf+imrerB8df8AJO/Ef/YKuv8A0U1C3NKX8SPqjaooor3zmGqxbOVK4OBnHPvTqKKACiiigAooooAKKKKACiiigAooooATPOMfjWTcyXA8Z6bEjsLZtPu2kTJwXElvtOOnAL/mfetesW6uJ18daXbrJILeTTbx5EB+UustsFJ9wGfH1Na0VeXyf5MTNqiiishhRRRQAUUUUAFYniO+1S2k0m20WS2hmv7027zXVu0yxoIJZM7VdDnMYH3u9bdY+tXMsGr+Ho4lUrcai8cpKBsL9luGyCfunKryPUjvWtH49uj/ACYmJb2viZY5hdavpMjlMRGPS5ECNkcsDcHcMZGARyQc8YKWtr4nSUm91fSZo9rALDpUsZDY+U5Nw3AOCRjkcZHWtqij20tdF9y/yCxk2Nv4ijvFbU9U0y4tud0dvpskLnjj5zO46/7P5UyG18TLeo1xq+kyWocF449LkVyvcBzcEA++0/Stmij20r30+5f5BYwjaeLPOyNa0YR5+6dHlzj6/af6VJcWviZrx2tdX0mO2LZWOTS5HcLnoWFwATjvtH0rZop+2l2X3L/ILGTf2/iGS7LaXqmmW1vgYjuNNkmccc/Ms6Dr/s0y6tfEzyKbLV9JiTYoYS6VJIS2Bkgi4XAJyQMHA4yetbNFJVpK233L/ILGNNa+JmhhFvq+kxyquJmk0uRldsnlQLgbRjAwSeQTnnAknt/EDWECW2p6bHdrnz5ZNOkeN+eNqCcFePVmz7Vq0Ue1l5fcv8gsZQt/EH9mFDqem/bvMyJ/7Ok8rZjp5fn5znvv/Co0tfEws5Fk1fSWuSwMci6XIEVcHIK/aCSc453DGDwc8bNFHtZeX3L/ACCxjW9r4mWOYXWr6TI5TERj0uRAjZHLA3B3DGRgEckHPGC+yt/EEfnf2lqemXGYyIvI02SLY/Ytmdtw9hg+4rWoodWT7fcv8gsY1ta+JlmY3mr6TLFtYKsWlyRsGx8pybhsgHBIxyOMjrSW1r4oW7ja71jSJbcMC8cWlSo7L3AY3DAH3wfpW1RR7aXZfcv8gsY0Nr4mW9RrjV9JktQ4Lxx6XIrle4Dm4IB99p+lFxa+JmvHa11fSY7YtlY5NLkdwuehYXABOO+0fStmij20r3svuX+QWMa5tvErXjtaatpUVsW+SOXS5HcD0Li4UE++0Ut1beJHlU2WraVFHtXcsulySEnHzHIuF4JzgY49T1rYoo9tLy+5f5BYyJ7bxG0cIttV0uNwuJTJpkjh2yeVAnG0YwMEnnJzzgPuINeazgW11LTo7lc+dJJp8jo/ptQTAr+LNn2rUope1l5fcv8AILGQ9t4jNoixarpa3AY75G0yQoy8YAXzwQevO45yOBjlFtvEYs2V9V0s3O4bZBpkgQDnIKefknpzuGMHg542KKftZeX3L/ILGXDBrq2My3GpadJdNjyZY9PdET13IZiW/Bhj3pkNt4jWGUXGq6XJKV/dMmmSKqNkcsDOdwxkYBHJBzxg69FL2svL7l/kFjHtbbxIjsb3VdLlXa20RaZJGQ2OCSbhsgHBI4yOMjrSW1t4lS6RrzVtKlgDDckWlyRsR3AY3DAH3wa2aKftpeX3L/ILGRBbeI1u1a61XS5LfcN0cemSI5XuAxnIB99v4UTW3iNrx2t9V0tLYsSkb6ZIzhewLicAn32j6Vr0Ue1le+n3L/ILGPoepXl9f65b33kf8S/Ufs8JhQrmMwRSruyTlv3pBIwOOlbFZek3UFxqeuRwW/lPb3yxzPuz5zm2hfdjt8rKuP8AZz3rUoq/FtbRfkvzBBRRRWQwooooAKKKKACiiigArDuf+SiaR/2Cr7/0baVuVh3P/JRNI/7BV9/6NtK58R/CZtR+L5P8mdBRRRXkCCiiigAooooAKKKKACqOsXNlZ6cZ9T3/AGdJYiNkbuxfzF8sBUBYnft4Aq9VHV9Lj1jTjZzu6IZYpCyMyt8kivwykEH5eoOR1qo25lfYmV+V23MMar4U1LxBbTyTSfbplSONZlniRiksgRWRgE3rIku0MNwKkjtVPbpMV/oEGhpLFHFr0onjmSVWWU2Nw5yJOeQyn0Oc1a1TwQ17qkdzY6m1lCiQnyRD5haWKSSVHZi2Tl5Mt/EcH5hk1n3/AIZFofD9lfXctxLea489zdW8kttJK/2GcFsrISv3BwpAxxjGc9cXC65ZPpoZ4dVPa+/FLR6/J+YzQ9f03xHry2HiS0EOp2U8k2nxagpiaTMsu0xxMBuZIkQ7wCRuPIyc9hpsVnBYrFprK1vG7qoSQuFIchlBycYbI2/w4wAMYrziy8NWHjjUrqw1e31W10+xkDvYXuo3zyXal3EbusxCqh8sthdxyMFhghvRdH086Vo9tYGVJRbJ5SMkKxAoOF+VflBC4BwAM5IAHA9SLTpL3rvt9/8AX/D6dWKio1mlG3kXaKKp6lqtppMCS3ruPMfZHHFE8skjYzhUQFmOATwDgAnoDUHOk27IuUVhHxhpo/5dtZ/8Ed7/APGqT/hMNN/59ta/8Ed7/wDGqXMu5p7Gp/K/uN6iqGlazZ61FO9iZv8AR5fJlSe2kgdH2q2Csiq33XU9O9X6Zm04uzCiiimIKKKKAOaXxcH8YXegJbwtPDkJ+/O44hWTe67MImXC7skk/wAPXGj4e1ga9okeoKkaCSSVAIpfMQ7JGTcrYGQduRwOtR3+k6JcyXj3yxb2hP2pjOUIiYLncQRhSIV9vlPvV/T4raCxjhsZWlgjJRWaZpTkEggsxJOCCOTxjHaqfLbQxgqnO+Z6a/18izRRRUmwUUUUAFFFFAGRbf8AI56l/wBg+0/9GXFa9cVrfiC60Lxpdmz0yXVJLi1sIEt4nVGBZ7w7snt8gB9Ovaul0TUJ9T0sXF5araTiWWJ4Vl8wAxyMmQ2BkHbnoOtZwaWnr+Zf1epTp88tm+/fXbc0KKKK0ICiiigAooooAKKKKACsbwtdyXmjzyyhAy6jfRDYgUYS7lQcDvhRk9Sck8mtmsfwxcrd6RPIsKQhdRvo9qE4JS6lUtyTyduT2yTjA4rVfwnp1X6i6mxRRRWQzBvP+Sh6R/2C77/0baVhWniLVNQ07RRdHWtPIswdTmh0aTzPtRVNqBXhYFD++JZVwCqDcNwDaWu6bBqnjvRoLl7pEXTb58213LbtnzbUfejZSRz0zj8hXM/Y7vU9L8Orob3KX+r6ab+V7vXdQMUQVYsgKs2TlplHXgA9ayle/wDXkejSjBwjzdvL+8aGneJrrT/FEv8AwkurXVpYs98jDUrZLW2jZblRbCKYxrvLQ72xvbIBPasvUvEPinV9d15PCOrObFYAbG7SyE9qrA2udsiROXY7rkEfNjHQYyLfh+wk1DxBNa6g2ofYpWvfsssGt36sv2e68kq+ZyCSCrAjH8XFZ2q3H2LXtc0uyGpXdxboUsIhrd8rGUrahBITPgqz3fbbtEZ5OciHexvCNP2llG7t2Vt/6RsQ3OpyeCNE1PUjMPFIuRBCzW/kvcsZiGjZCqkRMiljlflUCTBKCuq+0+Iv+gXpf/gyk/8AjFchp8KadoNr4siN2NQs5Zra8gu7qaVp0WRont41eaUBzIiFdrHeyKuQGOOv/wCElsf+eGqf+Cm6/wDjdO6W8rf15nm4xxTXvWV32+70X6h9p8Rf9AvS/wDwZSf/ABisLxFYeINTvNMlWOxsbm2lka2aHV9rM7RlTgPaOGOzfxjgZPbI3f8AhJbH/nhqn/gpuv8A43XO+JtYaXVtEv8ATLG6nbT5pZWSfTb1NwaJo8Bkt3x9/PI/hH4JuP8ANf7jPByTrJRnd66aa6M5TTY74eJtXlt9YTVdRfw9fiNrTVVvZ43DQ7VBSGMpyflUZGd2ADnPTvFrFs1vd2l1ql3bwXE7WdrcR3Ie5VliKRyYwUxIJFDTcBTkhhzWDb6tfp8RdW8SSWP2WWPwxdNFFLFchC0Ukbf8toYiR8wyFz35GRXTyeOpra5gima1lgSeUXF9EmFaONYd22NpAQ2+bZgF2ynCnOBz+9zPlV/+GNM0cfapydtF+f8AX3HcUUUVwCCiiigArB8df8k78R/9gq6/9FNW9WD46/5J34j/AOwVdf8AopqFuaUv4kfVG1RRRXvnMFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFY9zcyr430y2VV8mTTryR22AkMslsFG7qBh2474HoK2Kyri+MfjDT7DyYiJ7C6m80r867JLddoPofMyf91fStaXxbX0f5MTNWiiishhRRRQAUUUUAFZWr3xtNU0KEQxSfbL9oSzrkx4tp5Ny+h/d4+jGtWsfWrlYNX8PRtCkhuNReNWYnMZ+y3Dbhg9flxzkYY98GtaSvLbo/yYmbFFFFZDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooqpqOqWOkWZutTuo7WAMFMkrYGTwBQJtJXZborktU8dwafqlzbx2vn28FmbgXIZ9sriJ5fLUhCudiBslhweAcVv6VqS6ray3MUeyFZ5IYyWyXCOULe2WVsdeMHvgNxaV2RGrCT5YvUvUUUUjQKKKKAMfRZLd9X8QrBE0ckeoos7FgfMf7LbkMOOBtKjHPQnvgbFY+ii2Gr+ITbM5kOoobgMuAH+y2+AOeRs2c8ckjHGTsVrW+L5L8kJBRRRWQwooooAKKKKACiiigArDuf+SiaR/2Cr7/ANG2lblYdz/yUTSP+wVff+jbSufEfwmbUfi+T/JnQUUUV5AgooooAKKKKACiiigArH8VaOdf8PSaaMhZp4C5AUkIsyMxw2QflU9QfoelbFY/imC9ufD7w6WZBcvPAB5czxEr5ybxvQFlBXcCQDgZq6d1NW7mdRJwkn2OYXwM+nX/ANl0uxS4t47WHyr26aNSNtxNNLD8q5Ak8xUOFC7GbOcbWhbQrqzTRLZDLopuvEEs8KW/lO9spsp8oMqyEZDAcfdI4B4Fiyh8S6frtrHe3V9c3KW1vHsTdLayqbmXzWdyir5iQ7Dn5SSBw3Ss+fVdfttP0GTWrCK0vLLVD+81K8Eccp+wTmR2kXeFG4uRgYxtAA7dt5N6tPVGWDjH2qUYtOzt923/AA5d8NeG7T7VfXGheLdfcq4tp2eKHy/3TOBHHvg2hVJcYj4GfpXS/wBjX3/Qyap/37tf/jNcT4E0DxB4Ye5kt4fD16NU2zRNDqLJmBfukbbf95jzOX77lrtvtPiL/oF6X/4MpP8A4xW6tbW/yv8AodeYcntvicl0ev42+7UP7Gvv+hk1T/v3a/8AxmsQQt4U8UvqevapJeafeW4hW/vjGi2Dqc7CQAFWTd97j5o1ByWUVt/afEX/AEC9L/8ABlJ/8Yo+0+Iv+gXpf/gyk/8AjFP3Vqr/AInJSqQptuz10+0eU2+qabZWl5bxy6HqFpBcRzpBqGp2UU2qbkmWRbh0d0k2s6OCygkjuRmrHh++hsrrRba58V6DFoyQ2Ut9bDUYC5uIrYLjeH4CSW8JwvDbznIzXp32nxF/0C9L/wDBlJ/8Yo+0+Iv+gXpf/gyk/wDjFTp5/cz0HmMGn7j19f8AL+vuKPhG+tNS1DxLd6ddQ3dtJqq7JoJA6NiztgcMODyCPwrpayPtPiL/AKBel/8Agyk/+MUfafEX/QL0v/wZSf8AxitFNLv9zOGdeMndJ/c/8jXorI+0+Iv+gXpf/gyk/wDjFH2nxF/0C9L/APBlJ/8AGKftF5/cyfax8/uf+Rr0VkfafEX/AEC9L/8ABlJ/8Yrnrqwm13x3pdv4v0PSJIY9NvHhjExvFLebbAkiSJApAPBGep6d3zp7G1FKq30S/rra5o3/AIdvrzU/ELh7dLXVtNFouGIcOFcBj8uMfvD3PQfhr6HYSaVodrp8rrJ9lTyY2UEZjU4TP+1sC57ZzjiqP/CDeEv+hX0X/wAF8X/xNYvjLwd4YtfAuvXFr4c0mGeLTbh45I7GJWRhExBBC5BB71TlKw6dCjzqzd35d36ncUVixXXids+bpGkL6bdVlbP/AJLin/afEX/QL0v/AMGUn/xip50v+GZlOShLlf5X/I16Kx2ufEe07dK0snHAOpyDP/kCoftni3/oCaL/AODiX/5GoUk/+GLpr2iuvx0/Oxa1/Wl0PTfPW3e7uZXEVtaREB7iQ9FGfbJJ7AE9q4rxDret6FrmlQ65rs1ompxXczrpyQCOzWCISbcyxO0hPTOV9cdqpaf4t17xH8RNPin8Kwabf2Nhcyx22oX7x7g7QjeCIDyAGGAO7ZxjBn8QpqOofFrwJF4h02whRXvniW3u2uAxWFWBO6JMYKqR1/DHOblzK6PUp4f2MuWols29U+jtbftcp+C9S8XLptz4v1mzv7qXy47O6sLmARTSRR5fz4QqICB5rAIVJOG+c8LXq1tcw3lrFc2kqTQTIHjkQ5V1IyCD6VLXPeB0ji8MtFbhRDHqF8kYXoEF3MFA9sAVpFcuhyVqirJ1LW16dv8AgWNOXWtLguZbebUrOOeBA8sTzqGjUkDLDOQMkDJ9RU9peW1/apc2NxFcwPnZLC4dWwcHBHB5BFcm+ieKLHXJLnSf7HltkS6FqLiSWN0a4lWVy+FYNhlIAG3g10mi2clhpEUFwiLPl3l2SeYGdmLM27auSSSThQMngYrVpJaHnwnOUrNWL9FFFSbBRRRQAUUUUAFZfh66gu9MmktLf7Oi313GU3bsulzIjtn/AGmUtjtnFalY/hiS3l0idrSJooxqN8pVmBy4upQ7cAcFgxx2zjnrWqX7pvzX6i6mxRRRWQzltds573x3o0dtqN1pzjTb5jLarEzEeba/KfMRxjnPTPA561y8XhjRTp2mf2WdQuhq0K38VrBpelIdiqP3jb4FXK+eB1J+c4GNxrs9YstW/wCEi0/VNHt7K5+z2txbyRXV08H+seFgwKxvnHlEYwOorkF8D+Jls7GASsradB9nspo9e8t7eIqFKgpYgMCFT74Y/KCCOc4yWp6VGXuL3kvu7v8A4A7T/D9lq+o3lpba34j0ua4nuXae2WytWvWgl8qWTfBEGOHYffwTuyAecVrPQ7G08UanpFleeJLvUEgZZ7uaHTpnu1AgLR+bLHvbCzRcOQMDA6AVcg8G+JLPVLnUtPWK0u55JWEia3uEayyGWRFV7FlwXIOSC3yqN2MgvsPCXiex1y91hhDc315A8Lyy6yBs3CNS6hLFQGAhjA6jjpzSt5G3OtfeVrbabjdJt47DT9C1oSXcnh6CWV/st1b2sIsCQ6rdD7OgVo+W5yV2zCTPy5rsP+Er8O/9B7S//AyP/Gsm00LV7iKy0nUbWwsPD9pbiJrS0vZLh7jaFVI3Z4k/d4B3DJLcA5BYHrapKXQ87FScmnFrr56X02e/cyP+Er8O/wDQe0v/AMDI/wDGua8UfEHQdK1zw8x8Q24tGuZTdC3lEqlPJYLvCZON7Jj8+2R3lFNqb6/h/wAEyoycJ3qarXTbdW31/I8i0nxJFdfEbVJb+/sta0m00C/uV+yTC43QvdB/LcMAAwQKuw8YA55rsm1HSYrRZH8IzK2m3i2qQC3ti1tJJ5TKVxIQN3mIcqexzjFWNVs4L/xtYWd2nmW9zo2oRSpkjcrSWoIyORwe1alp4e02y2mKGSRluftQe4uJJn83y/L3bnYk/J8vXArgqOMJtPf+vM1xbdaUZwSSt+r8vmadVNWupbHRb67txGZoLeSSMTOEQsqkjcxIAGRySR9RVuqOuW8N34e1G2uXEcM1rLHI5k2bVKEE7iDt474OPQ1yRtzK5nK/K7GZpWt39/4g8iQW62pglkaAIRPb4kUQmTLceYhZwpRSMYOSDXQ1xHga3txfSyf2Rf295GsvnajcQ7ReeZIH5Zo43cgjA+QBQCOeCe3rSslGVkZUJOULsKwfHX/JO/Ef/YKuv/RTVvVg+Ov+Sd+I/wDsFXX/AKKasluddL+JH1RtUUUV75zBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABWPc3Kr430y1MKFpNOvJBKSdyhZLYbRzjB3gnjPyjHfOxWPcyW48b6ZG8TG5bTrxo5dwwqCS23LjGSSShznjaeueNaPxfJ/kxM2KKKKyGFFFFABRRRQAVj61Jbpq/h5Z4mkkk1F1gYMB5b/ZbgljxyNoYY46g9sHYrH1oWx1fw8blnEg1FzbhVyC/2W4yDzwNm/nnkAY5yNaPxfJ/kxM2KKKKyGFFFZOoeJdO03UDYzi9luViWZktdPnuNqMWCkmNGAyUbr6GlexUYyk7RVzWorB/4TDTf+fbWv8AwR3v/wAao/4TDTf+fbWv/BHe/wDxqlzLuX7Gp/K/uN6iqem6pa6tbtNZNIQjmORJYnikjbAOGRwGU4IPIGQQehFXKZm007MKKKKYgorM8Qa5B4c0WfVLyGaW3txul8ooCo9fmZQecDA5JI4rGS4vvEPiKRLLWr/TLNdLtLuOO3jt2JaZ58li8b9o1HBx161Lkr2No0ZSjz7Lv/XqdZRWD/wj+pf9DfrX/fmy/wDkej/hH9S/6G/Wv+/Nl/8AI9F32FyR/mX4/wCRvVm67a3F3pYS0XfJHc28+wMAXWOZJGUE8ZIUgZwMnkjrVP8A4R/Uv+hv1r/vzZf/ACPR/wAI/qX/AEN+tf8Afmy/+R6abT2FKnGStzL8f8jnbfw9fLrVnrE2jzSXZuGHlzC2dbeJryWTczFywZY5Mjy++ASeg9ArB/4R/Uv+hv1r/vzZf/I9H/CP6l/0N+tf9+bL/wCR6cpuXQzp4eFNaT/P/I3qKwf+Ef1L/ob9a/782X/yPR/wj+pf9DfrX/fmy/8Akepu+xryR/mX4/5G9RWD/wAI/qX/AEN+tf8Afmy/+R6o2HiK703whNf6jBfav9kub5Jp4VgVxHDcSICy5QE7UH3Rzg8ZPJzW3GqLl8LT6f1ex1lQXVpBexpHdRiRUkSVQezowZT+BANT0VRjvucLYeI9I1z4iS2E9nciS2LNY3T3cjQXDASQyARZ2hhsnA4OQjsK7KwsLbTLCGysIVgtoF2Rxr0UVztl4f06LxfPBHaxpbWOnaf9kRSQYWjkugCD16NjryCwOQTnqqmLbWrNKio816S/4fr/AMEKKKKozCiiigDL0mG0i1PXHtZ/Nllvle5TaR5Un2aFQvv8io3H96tSsfRYYotX8QvFOsrS6ijyoM/um+y267TkdcKrcZ+8PoNitavxfJfkhIKKKKyGFFFFABRRRQAUUUUAFYdz/wAlE0j/ALBV9/6NtK3Kw7n/AJKJpH/YKvv/AEbaVz4j+Ezaj8Xyf5M6CiiivIEFFFFABRRRQAUUUUAFZXiXUbnStDe6sQpmE0MY3QPNtDyojHy0IZiAxOAcnFatZniHVrTQ9Dm1HUYWmt4Hjyihc7jIqqfmIUYYg5JGMZqoayWlyKjtBu9jml8Wa3a63ZWOoR2Lec0KugieGSRZZ5kV1DMduEjRih55bnjFYPirUL3WfDelQ+JNHa4vINa+z3enwebGGJsZXwGgMrsuJAcgcjIIAzXT6f4r0LWL621GPSJPMaONBfyRQsYQ88kKJuVyxDSI2NuVw2SRk1zPiDXvDNx4V0PVbSd/DultrJM1xZJEJYnNnMMkR7wGOVByNwB7cGutqz1jbYWAk3iI8s77237ff935jvDtzNp+o+HpblL1k0/Sm02fZo9/83zLtZcwAf8ALNMk46n057j/AISWx/54ap/4Kbr/AON15zoHxD0i/vvB8UuvrBPBZA6g91fSKszeU8ewj7hffhyXwfu4JzXo3/CV+Hf+g9pf/gZH/jXTGaS0kv6+Z1Znz06iU2ouz39X6B/wktj/AM8NU/8ABTdf/G6yF1ZfF/iCXSrR760sLCFZb3IltJpnckRKjfK4QbHLEEEkIMkbhWv/AMJX4d/6D2l/+Bkf+NY9zqVr4n1+3Xwpe20t/pX7ya9B8yGJJFZfLdVYGTcVDbQygFFYn5QrUpXduZP+vU5MNLmk/eT06dPPd+nq16HHSazaaXZ3F34h/ta2WyuFF7bx6pqYltonWYRjLSBZCXjQb1O07m6ABmlZb26SK6sZruG2mhsZoo5tU1B3Zbu6eNMsLkBSkYRmUBvmLAHGDV5TLabLbT4dBkm1S5mhNta+GiDMbWQozvm6C7VYcFj/ABDHJq/pN7rOo3zWEVxpNqYzLHAZdCOyZbWYRsU23ZwqSMMBgp5yB1wrHtSmkrp/i/npb1/pGUu4aHqGvypqR0tL6K3tIoda1GS5KC6MEzOvm9cDKquff0rX0u38qPwXqcU2pRTanIpuYJtTuZo/msppCuySRhwyg88jArOutWudCuNdeWTRbaGwuYDqF3H4fwsksjRsjkC73sQZQ24r1DYJPW1cNqtxZaNqn9qac9lputRwRwWumPAWPnNZMAxncBcSORxzgdKaIldrfR7b9Vov1PQ6KKK3PICqGp6HpOteV/bGl2WoeTny/tVukuzOM43A4zgfkKv0UhpuLujB/wCEG8Jf9Cvov/gvi/8AiaP+EH8J/wDQr6L/AOC+L/4mt6ilyrsae2qfzP7woooqjIKKKKAOIfwRPodvFqGhXt3qmq2UrTRf2jKm64DgLLG8gQEllC4Zs4KJ2BFY+v3+jeIfF/hq+1PWLbQzo/2kX1ne3rWVyjSRqFCMMbgCOSGwR0yDivT6Kz5FayOuOKlzc09Xrrtvp/XmcJbeMbq8W50Xw5cWniDU4kQw30EqtCsbFlEkzL8odduSi/eyCoAJC9bo2lQ6Jo1tp1u7yJAuDJIcs7E5Zj7kkn8aw7mLVG8a6sdFktYbg6bYAPdxM8ePNu8jCspzyD1/nmtrRLa/tNKSLV7hLm7MkrySR52/NIzAKDyAAQAMnAGMnrRHcKri4e5bpp12+7Q4FtO8Sy3+q3EE+rXCG4mDWnmT25CC9jZRG7uFbdCsmDHtwDjJJq9rJ1ywvrK/8MxapOphWOOyu45pVI3ybpHLTrhtrKQsg3fcAxhgvoFFdHtDyvqytueTXumeLpNBsI7S61Z3EskPnKZ45W3JFtklUzZXa6yjOSvcLhq9ZooqZS5jSlRVNtp72/AKKKKk2CiiigArH8MC2XSJxZM7R/2jfFi64O/7VLvHU8b92D3GDgdK2Kx/C8MVvpE6W86zodRvnLrnhmupWZeQOVJK/h361qv4T9V+oupsUUUVkMKKKKACiiigAooooAKKKKAMzVNAsdXuLe4u/tSTW6OkclreTW7BXKlgTGykglFOD6Cqv/CIab/z9a1/4Pb3/wCO1u0VDhFu7RoqtRKyk/vML/hENN/5+ta/8Ht7/wDHaydWtPDmi3EcF/eeJBJKu5BDqGpzZH1RyPw+nrXZ0UnTj0S+4uNeSfvNtetv8zz7wBptv4l8F2ur311rQluZrghf7bvBsUTyKq8S9lAHqcc810n/AAiGm/8AP1rX/g9vf/jtO8KQpb6LLHHwo1G+IHpm7lP9a26mEIOKdhzxMqk3ODaT1WvToYX/AAiGm/8AP1rX/g9vf/jtRz+CdIureS3uZdXmhlQpJHJrV4yupGCCDLggjjFdDRVezh2RHtqv8z+8KKKK0MgooooAKKKKACiiigAooooAKKKKACiiigArLuGsh4u09ZBJ9uNjdGEgDaIvMg8zPfOTHj8fatSse5Ft/wAJvphdnF0NOvBGoX5SnmW28k54OdmBjnJ5GOdaXxfJ/kxM2KKKKyGFFFFABRRRQAVl6tDaS6nob3U/lSxXzPbJtJ82T7NMpX2+Rnbn+7WpWPrUMUur+HnlnWJotRd4kOf3rfZbhdowOuGZucfdP0OtL4vk/wAmJmxRRRWQwrBtP+Sh6v8A9gqx/wDRt3W9XJ+IF0e+8QtaXfgz/hILyC1jkeb7Pav5cbvIFXdM6nqjnAyB+NSzakr3Xl+q9DmvDlp48ikgbxPqM3lNFAQ1mkjtFGLW6B3KyfNL5nlMwIbLbfYU0S6++jPFFBrCSQfbIrWWM3v7+8AhNq7CUl1jwZAQ5MOVO4ng1sf2Nov/AESj/wAlNN/+PUf2Nov/AESf/wAlNN/+PVlZ2seg6sXLm0/8lt+Zq6oLqbxrax6FOsF7Dalr5poy8HkOT5eVDKWk3xnbzwvm56qDf+zeIv8AoKaX/wCC2T/4/VbwtPpYW9sdN0I6BLbyhprJoIombcoxKBExVlOCu7PVCO1dBV8qlr/meXWgnJJ30Xf/ACf9IyPs3iL/AKCml/8Agtk/+P0fZvEX/QU0v/wWyf8Ax+teij2a8/vZj7KPn97/AMzkfEXhvxTr1itrD4rt9MUNuaSy051kb5WUqSZz8vzZ6dQDmsV9M8Q2dxrOnafqFu2pR6dpUVpPb2zwpGguZxhgWkJAAbJ6Y7dc9R4z8UReEdCXUJgjbrhIwjsBuGdz4JI+by1cgdyAOScFlndQf8J1qlwZkEMmlafskZsA7proLyfUkAfWiPKpq2532qPCOL+Fp277q+u5hXkHiQ+HoBqV3eRXIW5+0yafFNL8/wBtiKbUUq5Xyw4GCDsJxT7D/hILzUNLeaPUbab/AEad8u4t1gMC+bE4YnLiQN975/nHJCnHYzavptvLBHcahaxSXEhihR5lUyuG2lVBPJDHBA78U6LUrGedoIb23klV9jRpKpYNgnBGeuFY49j6V0cztseT7GPMve7FqiiiszrCiiigAriPEVn4pn1bUotHuLhYBbC6t2R/L/eny0MIJyPuRynngGYHGVDV29FOLs7mdSn7RWvY5W1W7OoadNPYax9kkhjWKI3Z3W0u5zI0/wC8+cEeXjO/G0jC5wecg8Ka/wCIYUuLHxLDpttZaxqMkVv/AGeZg7fbJOZA0m1yGXKnaNueOea9NrB8H/8AIDuP+wrqP/pbNUTtJ2Z1YeUsPByg9bron0fkSQ2niZIlWbWdMlcdX/suRc/h59P+zeIv+gppf/gtk/8Aj9a9FTyLz+9nNKCk23f72Zenade2+p3V9qN5BcSzwxQgQWzRKoRpD3dsk+YfTpWpRRVRioqyHGKirIKKKKooKKKKAMfRbVrfV/EMrOjC51FJQFYEqBa26YIB4PyZwcHBB6EVsVjaJaT22seI5ZkKpc6iksJI+8otLdMj/gSMPwrZrWtrJei/JCQUUUVkMKKKKACiiigAooooAKw7n/komkf9gq+/9G2lblYdz/yUTSP+wVff+jbSufEfwmbUfi+T/JnQUUUV5AgooooAKKKxNT8U2mlSTme3upLe2kEVxcRKrJCxVWww3bvusrEhSAO9OMXJ2RMpRiryNuiiikUFZ2vTWFvpJm1Z3jtop4HyiFzvEqFAFAJOX2jgd/xrRqjq+lx6xpxs53dEMsUhZGZW+SRX4ZSCD8vUHI61UbcyuRO7i7bnOzT+Fta8TW10dUkuLiS1huobSJm2TLDJI0bgKMuysX+TJ6crwKx79YLb+xbPwWFN7DrOTHqEUkCofsE4VW+TPESqBxkgKScndXRL4Mit75fsVz5On4ty9u6vJIXhuHuFYSs56u5yCDx0IrM8W+FrDU5NH07WVN9BqOuyT3AJMW7FpcbBlCD8qoi5zzt56mupSi2lFvp/WwsJGSqqVRLrfr0fn/l8h+l+L7u9ufCu42Jg1myElwsYO+KbymcADdwpKOBkH7hGTnjsq4+2+FfhGzuba4ttPuYp7UYgkXUbkNEMk4U+ZwMk8D1PrWz/AMI1Y/8APfVP/Btdf/HK7l7Rb/n/AMA2xU6cpL2C089Or7X6GvXMa2h0/wAT6bc6IkbapdrJA9qzGOKeILv3ysoJXYQAr7W5l24G/cuh/wAI1Y/899U/8G11/wDHKoS20nhjV1voI7y70y4iMd0P3t5cQuD+7ZfvSMhywKjIBKsAMuafvdUZ0HPm1S2fXfy2OXm0jU7KePU5bnT7Uw3ly9tK3iEIkDzSM00Q/wBDwwL5yH3EFeCMGm2Ntqk2s3Nvo2p6X/aUE1zIYbbX4pJbVZZt88ew2Rwpk253AsCoAIyQUF1NAwuodPllvrG8vZrD7RpGoNG63MzyZbFvmN1yoyN2RvHG4EQ6ffarpfiCXUY7ea7hjkvVtrZ9L1CLCXNz57Ozi1b5htRduD1J3cAVnoevaTT0/Pz/AK+b8y1b2mo2+oanbX11o99dPLDe30F1rsYdGjMRR3VLJSoxHGOeMHPU5qQQ3un+ENMsrHT9ITRV1azKTWerPcFc38bYUeQob5iR978yOca7sxq2qajqer6ZNcfb5CZNMm0zUprcARQojg/Z12y/upF3hcqs2ctt2N0X9otc+GdO0ryNWuLwalZv8+l3YSONbyOTaZZIl3BIxje2C2zJ5NC6ikmuXTqr72X420PQKKKK6DxgooooAKKKKACiiigAooooAKKKKAMi2/5HPUv+wfaf+jLitesi2/5HPUv+wfaf+jLites4bfN/mZ0vh+b/ADYUUUVoaBRRRQAUUUUAFFFFABWV4eslsNMmhSeOcNfXc2+NtwBkuZHK/VS20jsQRWrWP4YtWs9InjZ0ctqN9LlGDAB7qVwMgnkbsEdQQQcHitV/CevVfqLqbFFFFZDCiiigAooooAKKKKACiiigAooooAKKKx/E2i3OuaWkFlqE1hPFKJVeOSVFfAI2P5bo5U5zww5APOMFMqKTdm7CeFpBLo87LnA1G+Xn2u5R/StmuA0PwzdXF7bzW8ep6Da2dyXuIZdTupmu5hIWfYGl2eUzZJZlJkDN8q9T39TDSNi6lOFN8sHoFFFFWZBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVj3MMTeN9MnadVmTTrxEh5y6tJbFm6YwCqjr/EOvONise5tWfxvpl2HQLFp15EULDcS0lscgZyQNhycYGRnqM60fi36P8mJmxRRRWQwooooAKKKKACsfWrVrjV/D0quii21F5SGYAsDa3CYAJ5Pz5wMnAJ6A1sVlavp815qmhTwj5LG/aeXn+E208f8A6FIta0naW9tH+TEzVooorIYVyc+uaTovxD1L+2NUstP87SrLy/tVwkW/Et1nG4jOMj8xXWVz15ea5deKrnTNJu9PtIbayguGa5snnZ2keZcZWVAABEOx6mpkbUra32t+q8meXadpen2SRHVfGPhXxII57dltNR1ONYhDFBPEqfdccGZWzt5Kk8Hk2YLDQreOKOHxF4SSSXRhp15dHU4WkLi18kPHmPevIUff27QTt3Gurt/FuoXVnbXVv4gsniu5BHbH/hFb7M5KM/yDzssNqMSQCBjk8itKHUtcm1W70628QaBd6hYhWmsBp8sLkFQwG4zttBDAbwrAZ6HpWKiun9fiepKtO7cv1X/tthZQPFWvWd5ol4IrGxjkJ1WzMbtO7fL5KMQytGBuL8EbxHg5Vsan9jX3/Qyap/37tf8A4zVzTNRg1WxF1a7wm942WRdrI6OUdSPUMrD044yKt1pyp6njVYqUrO6tpuzI/sa+/wChk1T/AL92v/xmj+xr7/oZNU/792v/AMZrXoo9mvP72Zeyj5/e/wDMwbnw7e3SFJPE+q+Www0fkWbK31BgNc7d+FpdW1fxBol5q13di40/TSbicRI6ItzcMVXy41XopxkdT1x02fHGoXmn6dYNp+r22lPLfLE8tzKsYkUxudgdopFUkgHJXsRnmqWm6ylnq93qmtCa2LaHpZmR0MkiSPLcjaQijLbmA4Uc9hRHSaS3O502sLJytytW8911/wCCU4/AeqWun2dlHc2l7mN4Lm6u2YzQgztJ50BwcSkOSST95I+Tt5veHvCeo6RcW0Ek0a6XY3Ans7YT+a0ZaOdZMt5SE5MykAk9Dz2roLbX7C71BLKFrj7S8Qm8t7SVCiEsAW3KNmSpwGwT+IqbTNVstZslu9MnE9u33ZApAbgHjI564PoQQeQRXQ5StqeXCjSUk4vbz7FyiiiszqCiiigAooooAKwfB/8AyA7j/sK6j/6WzVvVg+D/APkB3H/YV1H/ANLZqnqar+G/Vfqb1FFFUZBRRRQAUUUUAFFFFAGPottcQav4hknVljuNRSSAnPzILW3Uke25WH4GtisfRZbmTV/EK3Lu0ceootuGOQqfZbckD0G8ufqTWxWta/Nr2X5ISCiiishhRRRQAUUUUAFFFFABWHc/8lE0j/sFX3/o20rcrC1ez1YeItP1TR7eyufs9pcW0kV1dPB/rHhYMCsb5x5RGOOorGtFyptI1o/H8n+R0NFc/wDbvFv/AEBNF/8ABxL/APItL9u8W/8AQD0X/wAHEv8A8i15nsKvYv2b7r71/mb9FYH27xb/ANAPRf8AwcS//ItH27xb/wBAPRf/AAcS/wDyLR7Cr2D2b7r71/mb9eaeMrG3Pilry6tbOKx3pHc3lzHp7732AiMedhw23nLMeBwpGDXWfbvFv/QD0X/wcS//ACLXK6/YavPrFr9u0izzq1yIDDFrknls6QyPubNrlcxxshKkEg4PBNb0Kc6crtGFfDTqx5Y2+9HpNFc+L/xb/wBAPRh/3GZf/kal+3eLf+gHov8A4OJf/kWsPYVexv7N9196/wAzforA+3eLf+gHov8A4OJf/kWj7d4t/wCgHov/AIOJf/kWj2FXsHs33X3r/M36wPEf/Ia8Lf8AYVf/ANIrmj7d4t/6Aei/+DiX/wCRarSW/iHU9Z0ebUbDTLS2sLt7l2g1CSZ2zBLEFCmBB1lBzntWlOjUU02iox5bttbPqux0lFFFeqcoUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBkW3/ACOepf8AYPtP/RlxWvVG90TStSmE2o6ZZ3cqrtDz26OwGc4yR05P51W/4RTw7/0ANL/8A4/8KySnHRL+vuMUpx0SXXr/AMA16KyP+EU8O/8AQA0v/wAA4/8ACj/hFPDv/QA0v/wDj/wp3n2X3/8AAKvU7L7/APgGZq+v6iE1W70qa1ttN0iBnnup7Nrjz3QMZEjAlj+4FAJyQWYrwUaqGq674l0m6v4pb2xmTT9P/tC5mh0Y7UjxLgfNeBix8lhwMdMkVc1Tw1exWep6boVtaNpOqWzQyWb3RtRaswZXeLbFIPnDAkYADKWwS7Vi3/g3X9VvZbzUbOxuLmbTn02SV9VXcY28zJBFj8rfvTyuAcDIPOU7nq0vZ21tb5eXf5/8Ma+iah4l121mlg1TTbeS3na3ngudElWSKRQDg4uiDwykEEgginaZqGv6rptrfW+v6KkN4T9n87R5UaUAkAgG65BAyPYimaLYeK9GW8JstOvpr24+0TTXOrkMW8tI+BHZqoG2Ne3rWLp/gnWk0nRbXXtA8N6tcaOojhuZL1wXQIyqrBrZsqN2cZxuUHtRr/VwtHXVf+S9n+tjSu9c8SQeH9c1O31XSZW0YTiaB9HlQl4k34z9pPDAqQeeGH0rua89OgeM5dE8TWN1baJJJrzzN5q6jMBAHgWEDaYDu2hAeoz7V6FVRMK/KkrW+VvLt8wrG8LWk9lo88VyhR21G+lAIx8r3crqfxVgfxrZrH8MW1xaaRPHdqyyNqN9IA2fuPdSsh57FWU10J/umvNfqcnU2KKKKyGFFFFABRRRQAUUUUAFFFFABRRRQAVQm1UJqy6fDaXFzKESSZo9gWBGZgrNuYE5KP8AdDH5eRyM36xtW8MWWsanZ31xJPHLaSRyKI2G1zG+9cgg9DnkYOGYZwTTVupE+a3ulG18d6bqMVo2l293evd+X5ccQRWG4THB3sAMfZ5M8+mM5roLK7h1Cwt7y1bfBcRLLGxBG5WGQcHkcGucm+H2kPEqW0l1ZsPJzJbOqs3lxPEvJU4+SQ9McgEY5z0ttbx2lrFbwLtihQIi56KBgCnLl6EUva/8vCWiiipNgooooAKKKKACiiigAooooAKKKKACiiigAooooAKyrjT5pPGGn6io/cW9hdQOc/xSSW7L+kTVq1j3NtcP430y6RWNtHp15HI3OA7SWxUemcI/5GtaXxb9H+TEzYooorIYUUUUAFJnnH9KWigArH1q2uJ9X8PSQKzR2+ovJORn5UNrcKCfbcyj8RWxWHr6yNrXhgxglV1RzJg9F+x3I5/EitaPxP0f5MTNyiiishhXNXT6jpvjK9voNEvdRtrqwtoVe1kgG145JywIkkQ9JV6Z710tFJq5cZct9L3PLItA1i21Cxv7Pw/fW0+n3TzWsENvaraxI8TJIoi+28FtykkMFBQYUFnLb1k2p2XibUNWi8Oa9sv/AJprYy2ewsEiRG/4+eCBG3TGd4z90VStbrV/+EU8PajPr/iC8vNZii229lDp4/eNA0zY8yJQFARurZ6das6bdTal4kvdCXxTr8F7YxpJMky6bkbkRsBViLHHmAFsbc5GTxnJWPSm5yvzcui89r/5/iaml+DtPMbX3iHTrC/1m8CyX07wK6l9oG1Nw4RQAo74GTkkk3v+EU8O/wDQA0v/AMA4/wDCqemeLbFbU2viLULCw1e0IhvYHuFRfM2htybjkowYMO+Dg4IIFz/hK/Dv/Qe0v/wMj/xpfuutjx6zpOo/aWv8v6tbbyD/AIRTw7/0ANL/APAOP/Cj/hFfDucf2Bpn/gFH/wDE0f8ACV+Hf+g9pf8A4GR/40f8JX4d/wCg9pf/AIGR/wCNH7ny/Ax/2f8Au/gRyeC/C0zbpfDWkOcYy1hEf/ZaxYPDOmJ4n1/S9N0+wsrebTbCQxJZRmJnE1yctHgBvugHocDgggEZPxP17T9V8Mwafo/iDSzJJdxvOovrbd5ahmBxJIoI3qnerWk+LIZtQ1LWImt9RuRpGlxTQ2Fwkii5ea4XytysVHzOO54OaqLjzpRR6vs7YN1Ob3Wtuyuvz1NvRPCM2gS2H2HWbloIYvLu4ZUDC7OCFYk8qwyoyM/Kir0Aq/4a0FPDWjJpkE7S28ZzEpXHl5A3DuTl97cnjdjoBUFnrt8+sW+n6pp9rp8k0JkCvfBmdt0g2xrs+fCorNyMBx6VvVvJy6nm04U94dPUKKKKk2CiiigAoork7vxylouoSNY7ks5hEkInH2iceesLOsWM7AzEA55Ix3zTSb2InUjBXkdZWD4PONEn99V1Ht/0+zVn23jtbu+iFtp/nWDypG13HODt8y4lgjbbjlS0QOQeN44OM1yeo61py6TDbJrcEN9Y+Ibq5kszqNvArBdQd8TCRg3KZKgd8E9qipeDuzqwqjiU4xfVfqetUVjp4v8ADUiB4/EOlOp6Fb2Mg/8Aj1L/AMJX4d/6D2l/+Bkf+NT7SHdHK6kIuzaua9FU7HVtO1TzP7Nv7W88vG/7PMsm3OcZweM4P5Vcq001dFJqSugooopjCiiigDF0O4uJtY8RpO8jRwakiQBicKn2S3YhfbczH6k1tVj6Lcyz6v4hjlVQtvqKRxEIFyv2W3bJI+8cs3J9AO1bFa1viXovyQkFITjHv7UtFZDCiiigAooooAKKKKACiiigAooooAKQnHX9BS0UAZuv6NDr+iTadc7CkjI+JE3oWR1dQy5G5cqMjIyMjI61yGn+CLbwnq+giyZJA+o7CywrEfLS0vNm/H33zKQX4JAXI4JPoNZGs/8AIV0D/sIP/wCks9ZTS0l10/M1+tVaVJ04v3W9V66GvRRRWpkFFFFABRRRQAUUUUAFITgEnt6DNLRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAgOQCO/qMUtFFABRRRQAUUUUAFFFFABRRRQAUUUUAIDkAjv6jFLRRQAVj+GJbmbSJ2vXd5BqN8qlzk7BdShB9AgUD2ArYrE8J3U13o08lzK8rrqd/GGc5IVLyZVH0CqAPYVql+6k/NfkxdTbooorIYUUUUAFFFFABRRRQAgOen6iloooAKKKKACiiigAooooAKKKKACiiigBqsGzjPBwcginUUUAFFFFABRRRQAUUUUAFFFFABRRRQAmecf0rEu1kPxA0lwD5Q0u9DHPG4y2uP5NW5WJdXUy+PNKtVlYQSabeyPGD8rMstqFJHqA7Y/3jW1G/M7dn+TEzbooorEYUUUUAFFFFABWLrlxcQ6x4cSB5Fjn1J0nCk4ZPslwwDe25VP1ArarH1q5lg1fw9HEqlbjUXjlJQNhfstw2QT905VeR6kd61o/E/R/kxM2KKKKyGFFFFAHCQeH/EH/AAjOj6Rf6Rpk/wDZMUSxTwa9c2z70iMW8GOAEZVmGMn73erDaX4livW1Gx0fSU1JbU2sNxc67dTqiHafuGEA5KKSeCccnk12dFRynT9Ybe35/wCZR0fSodG01LSBnk+Z5JZZPvSyOxZ3PuzMTxwM4GBgVeooqjnbcndhRRRTEFc2trDeeOtat7hS0b6VYZ2sVIImuyCCMEEEZBBBB5FdJWTqHhrTtS1E305vYrlolhZ7XUJ7fcilioIjdQcF26+ppO/Q0g42al1X6kq6FYrdWdwwuJJbHcYGlu5ZNpYMCSGYhjh2GTk4PsK0awf+EP03/n51r/weXv8A8do/4Q/Tf+fnWv8AweXv/wAdovJgo0ls39y/zN6isH/hD9N/5+da/wDB5e//AB2j/hD9N/5+da/8Hl7/APHaWo7U+7+7/gm9RWD/AMIfpv8Az861/wCDy9/+O0f8Ifpv/PzrX/g8vf8A47RqFqfd/d/wTerKHhrSBqkmoraAXMh3OwkYAncj525253RoScZO0Zqt/wAIfpv/AD861/4PL3/47R/wh+m/8/Otf+Dy9/8AjtO8kS4UZbt/cv8AMsweGdJtruO4t7VomjUKqJM4jwHZ1zHnacM7EZHGeMcVW8Hj/iSXH/YV1H/0tmo/4Q/Tf+fnWv8AweXv/wAdrT0zTLXR7BbOwR1hV3f95K8jFncuxLOSxJZick96NW7sdqcIOMO66W7+Zbqrqd6dN0ye8FrcXfkpvMFsoaRx32gkZPfGfpzVqighWT1OCbxXcWGuTa1ceHtVax1C2tbW1EMtrK8kiPOzbUSYlgVkU/KCcKxOAM13UMhlgjkaN4mdQxjfG5MjocEjI9iaoWnh/SrDU5dQsrGKC5m3F2jG0EsQXbb0DNtXcwGW2rknAxpUopo1qyhK3Irf18woooqjEKKKKAMrSL43eqa7CYYo/sd+sIZFwZM20Em5vU/vMfRRWrWPotys+r+IY1hSM2+opGzKTmQ/ZbdtxyevzY4wMKO+TWxWtVWlt0X5ISCiiishhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXDal4ku766sb238Pamlvo1/JJetLNax7VFvNGQQ04PWRW5wCvIJ4z3NZ91omn3t/HeXMBaaPHSRlWTHK70B2vtPK7gdp5GDUyTZrTlCL95X/r5E2m3p1LTILw2txaecgcQ3KhZEB6bgCcH2z9atUUUzN2voFFFFMQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFZC+E/DqamdRXQNLW+MvnG6FnH5vmZzv34zuzznOc1r0VUZyh8LsBjXHg/wAM3d493deHdJnuXbe00ljGzs2c5LFck5pLrwb4Yvrp7m98OaTcXEjFnllsYndiepJK5Jraoq1XqraT+9isjGuvB/hm9kWS88O6TcOiKitLYxsVVQAACV6AAAD0FSXvhbw/qSwjUdC027EEYihE9nG/loOirkcAegrVoo9tV0956eYWRlT+FvD9zYQWNzoWmzWltnyLeSzjaOLJydqkYGT6VG3g/wAMvZJZv4d0lrVHLpAbGMorHGSF24BOBz7CtmihVqq2k/vCyMZPB/hmOzktI/DukpbSsHkhWxjCOwBAJXbgkAn8zUlv4X8P2llPZ2mhabBa3OPPgis41SXHTcoGDjtmtWih1qr3k/vCyMa38H+GbWOaO18O6TCk6bJVjsY1Ei5BwwC8jIBwe4FJa+DvDNjKZbLw5pNvIVZC8NjEhKsMMMhehBII7g1tUUe3q6+89fMLIybHwp4d0u8W70zQdMs7lc7Z7ezjjcZGD8wAPIpkPg/wzbXqXlv4d0mK6Rw6Tx2MaurDoQwXIPvWzRR7are/M/vCyMI+B/CZm80+GNGMmc7zp8Wc+udtPuvBvhi+unub3w5pNxcSMWeWWxid2J6kkrkmtqin9YrXvzP72FkY114P8M3siyXnh3Sbh0RUVpbGNiqqAAASvQAAAegqS98LeH9SWEajoWm3YgjEUIns438tB0VcjgD0FatFL21XT3np5hZGNP4P8M3MMMNz4d0mWK3XZCkljGyxrknCgrwMknA9TQ3g/wAMvZJZv4d0lrVHLpAbGMorHGSF24BOBz7Ctmij29X+Z/eFkZQ8L+Hxph00aFposTJ5ptfscflb8Y3bMYzjjOKjTwf4Zjs5LSPw7pKW0rB5IVsYwjsAQCV24JAJ/M1s0Ue2q/zP7wsjGt/B/hm1jmjtfDukwpOmyVY7GNRIuQcMAvIyAcHuBSWvg7wzYymWy8OaTbyFWQvDYxISrDDDIXoQSCO4NbVFHt6uvvPXzCyMmx8KeHdLvFu9M0HTLO5XO2e3s443GRg/MADyKZD4P8M216l5b+HdJiukcOk8djGrqw6EMFyD71s0Ue2q3vzP7wsjCPgfwmZvNPhjRjJnO86fFnPrnbUlx4P8M3d493deHdJnuXbe00ljGzs2c5LFck5rZop+3rfzP72FkZN/4V8Pardm61TQdMvbhgAZrizjkc4GBywJ6CmXXg/wzfSLJe+HdJuHRFRWlsY3KqoAABK9AAAB6Ctmikq1VWtJ6eYWRjTeD/DNzDDDceHdJlit12QpJYxssa5JwoK8DJJ47k1JP4W8P3NhBY3OhabNaW2fIt5LONo4snJ2qRgZPpWrRR7ar/M/vCyMoeF/D40w6aNC00WJk802v2OPyt+MbtmMZxxnFRp4P8Mx2clpH4d0lLaVg8kK2MYR2AIBK7cEgE/ma2aKPbVf5n94WRjW/g/wzaxzR2vh3SYUnTZKsdjGokXIOGAXkZAOD3AqXTfDOg6LctcaPomnWE7IY2ltbSOJipIJXKgHGQDj2FalFDrVWmnJ6+YWQUUUVkMKKKKACiiigArK1e+NpqmhQiGKT7ZftCWdcmPFtPJuX0P7vH0Y1q1j61crBq/h6NoUkNxqLxqzE5jP2W4bcMHr8uOcjDHvg1rSV5bdH+TEzYooorIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFcz42t9dnsbVvD0cs8kUhZoYp/J3Hadu5t6Ep1BUMDllbnaVbpqKadnciceeLiziH0XxRC2ttFf3E0d1a37WsZuPmind/3IUlvlUIq7f7pZ+ma3vCkF/beHIYtWWdLkSzErcT+dIqGVygL7mzhCo6mtmim5NqxEKKhLmTYUUUVJsFFFFAGPoslu+r+IVgiaOSPUUWdiwPmP9ltyGHHA2lRjnoT3wNisfRRbDV/EJtmcyHUUNwGXAD/ZbfAHPI2bOeOSRjjJ2K1rfF8l+SEgooorIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBn63Dd3GkyR2DOJS8ZYRvsdow6mRVbI2sU3AHIwSOR1HJ2ukeK5bqxvXnnsxGYYp7NrlnUxC6kdm5kf5zGIgSWPys456DvKKpSaVjKdJTlds5PwVZavZ+Z/a8OoQ/6Fao/229+0eZcL5nnOn7x9qnKf3enTiusoopN3dyqcFCPKgooopFhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVj61Jbpq/h5Z4mkkk1F1gYMB5b/ZbgljxyNoYY46g9sHYrN1bRYNYm02WeaaJtOvFvIjCQNzBHTa2QcqQ7Zxg+9aUnFS97s/yEzSorGuNE1Ca8eaPxRq1vGzZEEcVoUUZ6AtAWx25JPvSXWh6hcXTyxeKtXtUZiRDFFaFEHoN0BbH1JpqnH+dfj/AJBc2qKxrrRNQuJFaLxRq1qFRVKxRWhDEAAsd0BOTjJxxk8ADiibRNQlhhRPFGrQtGuGkjitC0pyTlt0BGecfKAMAcZySezjp76/H/ILmzRWVPpF7LYQW8fiLUoJYs77mOO2Mk2T/EGhKjHT5VHvmo20TUGskgHijVlkVyxuBFab2Bx8pHkbcDB6AHk5J4wKnH+Zfj/kBs0VjJomoLZyQt4o1Z5HYFZ2itN6AA8ACDbg5zyCeBgjnMlvpF7DZTwSeItSuJJcbLiWO2Dw467QsIU577gfbFDpx/mX4/5Bc1aKxrfRNQhjmWTxRq05kTaryRWgMRyDuXbABnjHORgnjODSWuh6hbyl5fFOrXSlWXZNFaAAkYDfLApyOo5xkcgjij2cdffX4/5Bc2qKybHR760vFmuPEmp30Yzm3uI7UIcj1SFW4+tMh0TUIr1J38UatNGrhjbyRWgRh/dJWANj6EH3o9nG/wAa/H/ILmzRWNcaJqE1480fijVreNmyII4rQooz0BaAtjtySfekutD1C4unli8VavaozEiGKK0KIPQboC2PqTQqcf51+P8AkFzaorGutE1CeRWi8UatbBUVSkUVoQxAALHdATk9TjjJ4AHFSXukXt0sIg8RalZeXGEYwR2x80j+Jt8Lcn/ZwPQCj2cdPeX4/wCQXNWisafRNQlhhSPxRq0LRrtaSOK0LSnJOW3QEZ5x8oA4HGckjaJqDWSQDxRqyyK5Y3AitN7A4+UjyNuBg9ADyck8YPZx/nX4/wCQXNmisoaRejTPsp8Q6kZvM3/bDHbebjH3MeTsx3+7n3pq6NfCzaE+JdUaRmBFwY7XeoGeAPJ24Oe4zwMEc5XJH+Zfj/kBr0VjwaLfxQypJ4m1SdpFwskkVqGjOQcrthAzxjkEYJ74NFrot/buzS+JtUugysAssVqApIwCNsIOR1GeMjnI4p+zjr76/H/ILmxRWVY6Te2l0JbjxDqV8gz+5uI7YKeP9iFW9+tNg0a+hu1mk8S6pOgYEwSR2oRh6ErCGx9DmlyR/mX4/wCQGvRWINB1ETbz4s1grn/VmKz2/wDojP61JNo19LePMniXVIY2YkW6R2pRR6AmEtj6kn3p+zj/ADr8f8gua9FZV7pN7dXhmg8Q6lZRnH7iCO2KDj1eFm9+tMutFv7iVWi8TaraqFUFIorUgkDBPzQk5PU8454wOKFCLt7y/H/IDYorHuNFv5oolj8TapbmNdrPHFakyHJO47oSM844AHA4zk0r6NfPaJCviXVEdWJM6x2u9wccEGErgY7AHk5zxg9nH+dfj/kFx+kw2kWp649rP5sst8r3KbSPKk+zQqF9/kVG4/vVqVl6Jog0Vb0tf3WoT31x9omnuhGGLeWkYAEaIoAWNe3rWpSqtOWjvt+QIKKKKzGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRWdq2u6dof2L+1LnyPt13HZW/yM2+Z87V+UHGcHk4HvTjGUnaKuwNGimu4jUFgxBIX5VLdTjt9evbqa5mx8dWep/D9PFum6Xqt5bSZ2WdvbiS6bEvlnCBiDggnr0GfarhSnNXitLpfN7fkK6R1FFFFZjCiiigAoopkU0c6F4ZEkUMyEowIDKSrD6ggg+hFAD6KKKACis/SdW/tX7b/AKBfWX2S7ktf9Mh8vztuP3kfJ3RnPDd8GpLTVbO/vr60tJTJNYSLFcDy2Co7KGChiMMdrAnBOM84q3Tkm01tuFy5RRRUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVHceb9ml+zFFm2HyzIpKhscZAIJGfcVJRQBwvwt+IUvj3Rrp9Rs1sdSs3XzoUDBTHIu6Nxu5wwz+WehFLfePp4/ivpHhWxtYns7hp47q7fORLHB5vlpz1AeIknI+fHUHHGfabjwJong3xhpVm94NS0W10u8tI+DNKYA1u313DaT6HjNbN7ow0H4lfDOxMxuJx/aslzcFdpnmeEPJIRzjc7Mcds4HAr6KeFw/tpzjH3ZRnyrs4xd/ua08muxipO1jrPF+qx27WFpD4tsPD05uVknNxJD5ksG1gVRZMjJbbg44x36HevdQstNs2u9Ru4LS2XG6aeUIgycD5icckgfjXCeD7SDUvFPjyy1+0hub3+0RuaaIEtaSRAQpz/DtVvzNcn4Yub/TNH+ElzrNzJHZh7yOVpWIXMkbrbZJ4+4cD2PGa5vqUZLkT1j5LW8XPR79LK9916Fcx2XhnxxFqnjXxn52uWc2g6YLE2k6yReTF5kbb/3g65cY5JweBiusfxFosclokmsWCPfBGtVa6QG4D/cKDPzbu2M57V5rJaPqPij4x2tuokkn020SMYzljZvt/XFUdR1S2u/h78JEt5A7HV9MXgggGJdjj6hjitqmBp1ZxcdLqP8A6bUr/PW/cSk0v67no+lXeqH4geIrG+uhLYR21lcWMIRR5IcSq/IGTloieScdq0YvEeiT6odNg1nT5L8Eg2qXSGUEdRsBzxj0rmb2+ml8T+NhokgbULTQbWOPbyVn/wBLdBgd/mU496428tLfUP2atKu9DtY49Rj+x+RNGn7xboXCRu4IGdxYvk9eTWEcJGrKLlpfkjp0bitX+fnqx81vxPaq84s/HetTeCPFur3P9nRXXh7Ubu1ULbyGOdYUBAx5mQzMeuSAOxr0evnw+FL3WNN8XaxpjXV3LpnjC6ml0fzWa3v4kkjdkaI/KW/A5Axzxicto0qvN7XTWOvz2+ewTbWx7R4e8TQa14Ls/EV0g06Ce2+0SC4cKsK9yWOBt4znjjmrmm65pOs+Z/Y+qWV/5RAk+y3CS7M+u0nFcB4212C4+F+jav4buHsNKt9QspLn7LGge1gSQZTZgqpRgnykEDbggitTw2nh248Uz+IbLxfceIL1dNNvK5eB44YBIHG4QxqFbJON3JG7AOOInhI+zlVs1q0lq7Wto9PPuPm1sSfDbxTNqnwp0zX/ABTqUIll83z7ufZCvEzoucAKOABXSHxHoi2X2xtY08WvmCLzzdJs3kZC7s4zjtXk3gWfTn+B3hCG6+ztdf2syWT3UrpDDdCWaRGfYQWAAJCZwzFQSM5G34QW31/4i/Euyu5EuredLK1uPLG1XP2d4pcDJxypGMkiurFYOmqteeqUW9trc6jZff8AhbzJjJ2SPQn1vSo7CO+k1OzS0lbZHcNcKI3bngNnBPB/I1erynwNHe3Go2/grVEZ4fCEsjXDLgLP902RYdwUeR8dmiQ5GMH1avNxVCNCfInfr8un3rX0aLi7oKKKK5CgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvOPF+izeO319LaRgdCh8jTljbDG/2xzmT04HlIp6jdMOhFehzLI0EiwOscpUhHZdwVscEjIyPbIrP8O6P/YWix2TzC4m3ySzXAQoZpHcuzkZOCSxJ5x6ADAHVh63sH7WL95Wt+r/T5ktX0KfgXxJ/wlvgrTtYZPLmmj2zx4xslUlXGO3zA49iK8p0fW9b8N/ss6drPh++htJbUy7/ADLYSswe8ZBtJO1cbieVbPTivRvDfgq+8N3HiE2+u+ZbavdT3cMBtdv2OSQ5BB3/ADAcZHGcDGOc5c3wreT4PweA49cMaRyFnvPsud6+c0oXZv45K9/4ffj06VXCU6jV1yOcHazfupSutV0vb8UQ1Jr5Glf63qlz8Uz4WsrtbO3OhG+84Qh3EhuFjyM8ZChsZyMtkhsYrlrHxH4xvvh14yupdbgi1Dw5e3MEN3DZpmYW67n3q2V+YEYwBgjv0rtpvCcj/EyDxdHqPliPTf7Pez8nPmLvZ92/dxyV4wfu+/GXB8Op7fQ/F2mR62QniW7nud32b/j284/OuN/zfLgZ46Z9qilWwsYpO32d431v73Tt8htSKN1421jWLzS7PRLC9Bu9Dh1Z2097Yyq0pwiETkDyxtbcRzkr07tuPFfi7SfDvh4eILGOx1K+8R2+mSsfLbzbd2J8wKjMqswG0jJwckdq0Jfh1LFH4cudJ1ySx1XQrFLEXQg3x3MSoAVeIsOCRnAbj1zgiz4j8DXPiHS9Ohk1ySO8stWj1Q3LQ71LoGARE3DYgyMAE9OcsSxpVcHeMUo8vXR36/enp/kFpFPTNT123+J2qeFtU1Y3sdxo6ajaTLbpF9lbzDG6qAMkFiCNxYgADJ5J5/4cahqNh8EdP1Z9X/4+bpjcTX0kSraxG7cSyIxUbnIJbDlvm6D+E95/wi//ABcj/hLPtn/MJ/s37L5X/TbzN+/P4Yx757Vz+l/C7+z/AAFpvhubWPtJ0vUEv7a4NttXcr79jJv5UktnDDrSWIwzp8raV3BvTspJ9PT189RWlf7yjYfEWfTh45a+kuNQtfDkVvJavd232aeYyRE7XXYuAXHB2Dg55GDXWWlp4ii1fTrhdYh1DTZo5GvkljWPaSq+WbcImdu7dkO7HBHJIrK/4V7LPr/iS+vtXWS08R20UF5aRWgQ5SAxZDlmwPmZsYyCF+YgHdc8NeFNX0OPTbe98TzX9npsTRxQC2EXmDG1fMbcS21eg45wTyBWdeWFcW6TSdlpZ/ypO2mjUr66J3T6DXN1OduPiNqNj4G8ca9LFFNLoutTafaREYUKpijUtjk/M5Y+vI44xfur3WfBOoeGo7u+TU7TWNQ+x3w+yxxFJ5VLK8QjC4BcMWLlzz2qez+G0Eek+K9L1HUXurLxHfTXjIkXltbtIcnB3HcRhecD7vTnFasXhm5lGhx6tqEN7Fo5Eq4tNjyzKjIjk7iAAr5wFBLAHIGVqp1sIrqNrXfTdcqtbTSzv23C0jkJ/E3iXVfCfjXVNO1WPT5fD+p3sUKrapJ5kcCIQpLZ4Pzk8ZywwQBg994a1KXWfCek6pcKqy3tlDcOqdAzoGIHtk1ztj8PpLLRvGGnHVvMi8TT3U6n7Pj7K06sp43fPgFf7udvvx0ugaX/AGH4b0zSfO8/7BaRW3m7dvmbEC7sZOM4zjJrDF1MPKHLSto9NLaWXl37jinfU0KKKK80sKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKTIzjPPXFLRQAUUUUAFFFFABRRRQAVBd2dtqFq9rf20N1byYDwzRh0bnPIPB5FT0UJtO6AoWmh6Tp9mLSw0uztbYSiYQw26IgkBBD7QMbgQDnrwKhvvC3h/U737ZqWhabd3Rxme4s45H44HzEZ7CtWitFVqJ8yk7+orIzNS8N6HrMyy6xo2n38ijar3VqkpA9MsDVu7srO/tWtL62guYHHzQTRh1YD/AGTwe1WKKn2k9NdtvIdjP0vQdH0Pzf7F0qx07zseb9ktki34zjO0DOMnGfU02Dw7olrax21to9hDBFMLiOKO1RVSUYw4AGAwwMHrwK0qKbq1G23J6hZGbF4e0W31VtTg0ewi1Bixa7S1QSktnJ3gZ5yc896SPw1oUOqDU4tF09L8MWF2tqglyQQTvxnJye/etOin7Wp/M+2/QLIZNFHcQvDPGssUilHR1BVlIwQQeorP03w5oejXDz6Ro2n2E0i7XktbVImYZzglQCRkVp0VKnJJxT0YGcmgaPHcXE8ek2KzXSGOeRbZA0qk5KscZYE84NSW2jaXZWU1nZ6baW9rPuMsEUCqkm4YbcoGDkdc9au0U3Um92wsjJfwr4el0/7BJoOmPZ+d5/2drOMx+ZjG/bjG7BxnrTL/AEyfT9NuZPB9hpNvqkixIpuIzHE6IwG1zGN2AhYL1wSO2a2aKpVp31d/J7CsjO0e1v7eCeTVbkT3NzMZmjQkx242hRHHkAlQFHJxlizYGcDQJAxk4zwPelorOUnJ3YwooopAFFFFABRRRQAUUUUAFFFFABSEgYycZ4HvS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUhIHU47c0tFABRRWPr+r3mltpsOm2UF5c6hdm2Rbi5MCLiGSUsWCOekRGMd6qEJTlyxDY2KKx1ufEZsmZ9J0tbncAsQ1OQoV5yS/kZB6cbTnJ5GOSC58RtDKbnStLjlC/ulj1OR1ZsjhibcbRjJyAecDHORfspeX3r/MVzYorLs7jXnWX+0NN06AhCYhBqEku5+wbMK7R7jOPQ0y2ufEbSMLzStLiTaxUxanJIS2OAQYFwCcAnJwOcHpS9lLXb71/mFzXorGtrnxK90i3mk6VFAWG54tUkkYDuQpt1BPtkU0XXinzsHRtHEWfvDVpd35fZv60/Yyva6+9f5hc26KyJrnxGt462+laW9sGISR9TkVyvYlBAQD7bj9ade3Gvx3hXTtM02e24xJPqMkTnjn5RAw6/wC1S9lLy+9f5hc1aQkKCScAckntWRdXPiRJVFlpOlSx7V3NLqkkZBx8wwLduAc4OefQdKWe58RrHCbbStLkcrmUSanIgRsnhSIDuGMHJA5yMcZL9lLTb71/mFzXorLuLjXls4GtdN06S5bPnRyahIiJ6bXEJLfiq496Y9z4jFojRaVpbXBY7421OQIq8YIbyCSevG0YwOTnhKlJ9vvX+YXNeiub1XXNd0bwvqGrXmkacXsYWuGhi1KQh0RGZvmMA54GBjBz1GOekolTlBXf5p/l6hcKKKKzGFFFFABRRRQAUUUUAICGAIOQeQR3paKKACiiigAooooAKKKKACiiigAooooAQEMAQcg8gjvS0UUAFFFFABRRRQAUUUUAFFFFABRRRQAgIPQ57cUtFFABRRRQAUUUUAFFFFABRRRQAUUUUANV1fOxg204ODnB9KdRRQAUUUUAFFFFABRRRQAUUUUAFFFFACZGcZ564paKKACiiigDnL271+68V3Om6Rd6fZ21tZQXBe6sJJ2keR5lIBWZAABEvY/eq/Nb6+2nwpb6lpqXi586Z9OkaN+eNqCYFePVj+HSkt74yeMNQsPJiAgsLWbzQvztvkuF2k+g8vI/3m9a1a6Kk3FpJJWS6Lt6CRjm28SfYVRdW0oXe8lpTpchjK8YAT7RkHrzu5yOBjlEtvEospFk1bSmuiwKSrpcgRVwcgp9oyTnHO4YweDnjZoqPbS8vuX+QWMq3t/EC2U63ep6bLdNjyJItOkRE9dyGcls9sMuPeo7e18TLHMLrV9JkcpiIx6XIgRsjlgbg7hjIwCOSDnjB2aKPay8vuX+QWMW1tfE6Sk3ur6TNHtYBYdKljIbHynJuG4BwSMcjjI61JY2/iKO8VtT1TTLi253R2+myQueOPnM7jr/ALP5VrUUOtJ32+5f5BYxobXxMt6jXGr6TJahwXjj0uRXK9wHNwQD77T9KLi18TNeO1rq+kx2xbKxyaXI7hc9CwuACcd9o+lbNFHtpXvZfcv8gsYt1a+J3unay1jSYrcsdkculSyOo7AsLhQT74FLdWviZ5FNnq+kxJsUMJdLkkJbAyQRcLgE5IGDgcZPWtmij20l0X3L/ILGVe2/iCRYf7O1PTYCsYE3n6dJLvfuVxOu0exzj1NRz2viZoYRbavpMcqriZpNKkdXbJ5UC4G0YwMEnoTnnA2aKFVku33L/ILGM1r4mNkipq+ki6DkvIdLkKFeMAJ9oyD153HORwMcyC38Qf2YUOp6b9u8zIn/ALOk8rZjp5fn5znvv/CtWij2svL7l/kFjGS18TCzkWTV9Ja5LAxyLpcgRVwcgr9oJJzjncMYPBzwW9r4mWOYXWr6TI5TERj0uRAjZHLA3B3DGRgEckHPGDs0Ue2l2X3L/ILGLa2vidJSb3V9Jmj2sAsOlSxkNj5Tk3DcA4JGORxkdaksbfxFHeK2p6pplxbc7o7fTZIXPHHzmdx1/wBn8q1qKHWk77fcv8gsY0Nr4mW9RrjV9JktQ4Lxx6XIrle4Dm4IB99p+lRm08Wedka1owjz906PLnH1+0/0rdop+2l2X3L/ACCxjXFr4ma8drXV9Jjti2Vjk0uR3C56FhcAE477R9Kff2/iGS7LaXqmmW1vgYjuNNkmccc/Ms6Dr/s1rUUvay02+5f5BYxrm28StIps9W0qJAqhll0uSQlsDcQRcLgE5wMHHAycZK3Ft4kaKIWuq6XHIFxK0mmSOGbJ5AFwNoxgYJPQnPOBsUUe2lpt9y/yCxkPbeIzaIsWq6WtwGO+RtMkKMvGAF88EHrzuOcjgY5eYNd/s3YNR04Xu/PnHT38vZjp5fnZznvu/CtSil7WXl9y/wAgsZC23iMWbK2q6WbksCsg0yQIF5yCnn5J6c7hjB4OeEgtvEawyi51XS5JSv7po9MkRVbI5YG4O4YyMAjnBzxg7FFP2svL7l/kFjLs7fXkWX+0NS06clCIjBp8kW1+xbMzbh7DGfUUy2tvEayMbzVdLlTawURaZJGQ2OCSZ2yAcEjAyOMjrWvRS9rLXb7l/kFjGtrbxKl0jXmraVLAGG5ItLkjYjuAxuGAPvg06C28Rrdq11qulyW+4bo49MkRyvcBjOQD77fwrXoputJ9vuX+QWMia28RteO1vqulpbFiUjfTJGcL2BcTgE++0fSnXtvr8l4W07U9NgtuMRz6dJK445+YTqOv+zWrRS9rLy+5f5BYx7q28SPKpstW0qKPau5ZdLkkJOPmORcLwTnAxx6nrSz23iNo4RbarpcbhcSmTTJHDtk8qBONoxgYJPOTnnA16KftZabfcv8AILGXcQa81nAtrqWnR3K586STT5HR/TagmBX8WbPtVK4v9a03U9Btry4sLlNQvJbedorV4iALeSVdoMjYOYjnOcg9sZroax9akt01fw8s8TSSSai6wMGA8t/stwSx45G0MMcdQe2DdObk3FpPR9F2f5AzYooornGFFFFABRRRQAUUUUAFFFFABRRRQAVj61crBq3h6NoUkNxqLxhmJzGRaztuGD1+XHORhj3wa2Ky9WuoLfU9Djnt/Ne4vmjhfdjyXFtM+7Hf5VZcf7We1a0vi26P8mJmpRRWBrlzdw65p0UdxqFvZPDOZXsrPz8yAx7FY+W+0YMh7Zx1rJK7FKSirs36K4WTxD4xln1aKz0yBfstyy27T2MwWSNRcEc7hvJ8qH5l4/e9DxW1J4gl07xFNb601taabIWW1uZT5QLKkR2lmO1ixkkwBjAiPXnFcjM1Wi1c6CiuC1DWvEoutTbT4rxwtncNbBLVZLd28yMQNGyrkkozZBY8gnAArR1nXtaa4vrfRbCcQro8t3BdNaSbjON6iIKwxuzsYAg8A8HcCHyMXt466M6yivPRruswWGjy3l5qNszXTNdzXdiI4hbLcMB5hEPyM0eP7gAG4kd79/4jmn8TWD+H759U0+R44Z4bFPMjjYyfM7SiJ14VgSu9CAAf4hRyMX1iFr+n4nZ0Vw+meJfFN1DbtqOnJZxSIrNcrp1y5WQgboTBkOuOf3uSny46nitd+KvF8E0lvbaLI+7ULq3juJLKV1SNGzHIwXGUYOqgj/nmx+cnAPZsX1mFr2Z6DRWL4XudUuNNnGuj/SobuaHeIDEJEViFcA9QRggjIraqGrOxvGXNFMx/F8lvF4I1yS9iaa2TTrhpolYKXQRtlQSCASMjODWxWP4vFs3gjXBqDOtodOuBO0a7mCeW24gZGTjPGR9a2K1f8Jer/QfUKKKKyGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBj21yreN9TtRCgaPTrOQygncwaS5G084wNhI4z8xz2xsVj20lufG+pxpEwuV06zaSXcMMhkudq4xkEEOc553DpjnYrWt8XyX5ISCiiishhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFZ767pyAFrkc+aAAjEsY5BG4AAySHZVwOSSMZoE2luaFFFFAwooooAKKKY8qRsiu6q0jbUBOCxwTgepwCfoDQA+is0a/pxnmiEz/uZVgeTyH8vzGcIFD42k7mAIBODnOMGrdpdwX1qlzaP5kMgyjgEBxnqM9QeoPQjBGQaLMlSi9EyeiiigoKx9aFsdX8PG5ZxINRc24Vcgv8AZbjIPPA2b+eeQBjnI2Ky9WhtJdT0N7qfypYr5ntk2k+bJ9mmUr7fIztz/drWk7S+T/JiZqUUUVkMKKKKACiiigAooooAKKKKACiiigArH1qS3TV/DyzxNJJJqLrAwYDy3+y3BLHjkbQwxx1B7YOxWPrQtjq/h43LOJBqLm3CrkF/stxkHngbN/PPIAxzka0fi+T/ACYmbFFFFZDCiuX1TxBqK3Ot2UVlLYraacZ4dRmMZjjYiXEjAMxK/uxgBSeDkAYzjaV4l1i9NnHb3SXqm8FrJcwyRzpK2baU4ZY0A/ctcZGOCpGc7RV8jZhLEQTseg0UUVBuFFFee3GleK7e31rVbO5ma4WWU2VqbmaTzQLveMxsQqjy0CAJ1UnnnFOKuZVKjhsrnoVFeZWj+JJdH8P6pq41kTJLNNqMVuJVfIuIgg8leGUopG3j5WZwCQQeilurjXvFelHTm1G0tbWKaW5W4s7mCOQ74gi87ATjeRncBzlTmqcLGca6ktu34/5HV0VwcOta02n+GxBY6rK8EKPqbSW0kZDAIr7w6gyDa0xATJLqmM99XwfPen7RbalaX0dzFBA1xcXMsrLLOyt5qoHG0BWH/LMlSGGMcCk4tK5UaylJJL+rF/xfDFc+CNcguZ1toZdOuEkmbOI1MbAscAnAHPANbFZXieyXUvCOsWMk8dutzYzQtNK21Yw0ZG4nsBnJNatW3+6S83+ht1CiiishhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZdu1kfF2oLGJPtwsbUzEgbTF5k/l475yJM/h71qVj2wtv+E31Mozm6OnWYkUr8oTzLnYQc8nO/IxxgcnPGxWtX4vkvyQkFFFFZDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDmrHxbd6lp9vfWXhTWpba5iWaJ/MsxuRhkHBuMjg96n/4SDUv+hQ1r/v9Zf8AyRR4G/5J54d/7Bdt/wCilreqFdrc6Kkoxm4qK09f8zB/4SDUv+hQ1r/v9Zf/ACRXLwaTeNc3bar4c128gmkleGFJLSHyA9x55w63W7O8JyCPujivRqKpXWzMZezn8UF+P+Zymj3+paTotpYf8IvrtwbeJYzNJNZlnIHJP+kVd/4SDUv+hQ1r/v8AWX/yRW9RRq+pSlFKyivx/wAzB/4SDUv+hQ1r/v8AWX/yRVvRtZGsLeBrG6sZrO4+zzQ3JjLBvLSQHMbspG2Re/rWnXK2OqW2k6r4ikuywE+u29tGFGSXktbVF/U5PsDUvTqaRSqJpR18r915nVVlaxoUOs3WnTTzTxfYZ2lAhlaMvmNkxuUgj7wPXsR0JrVqOeIz20kSyvCZEKiSPG5MjGRkEZHuDV3a2OZxUlZnA6bpWkeJPEeu3Gj6jAk1nLJbuYrGNJI7gyRTLIWXHmqrxcbhyQ2SR17uxs4dO0+3srVSsFtEsUak5wqjA578CuV8IeG7DSfEeuzWML24tp47KJQx2tELS1K7s/eKlTgnpuf1NdjSUpSWpU6VGnO9LZhRRRTEFY+tQxS6v4eeWdYmi1F3iQ5/et9luF2jA64Zm5x90/Q7FY+tWrXGr+HpVdFFtqLykMwBYG1uEwATyfnzgZOAT0BrWj8W/R/kxM2KKKKyGFFFFABRRRQAUUUUAFFFFABRRRQAVl6tDaS6nob3U/lSxXzPbJtJ82T7NMpX2+Rnbn+7WpWPrUMUur+HnlnWJotRd4kOf3rfZbhdowOuGZucfdP0OtL4vk/yYmbFFFFZDI554rW2kuLiRY4YkLyOxwFUDJJ/CuO8FeLNU1rXNWstbsntcSCeyUpgpEYonMb+jqJoyQe7kDheO1rF0KLy9Z8SNnPmakjdOn+iW4/pUS3RrGdOMXGS1extUUVieIPEkfh+OR5bdpgljPd/K2M+UYxt6dzKOe2KtJt2RhKSguaWxt0UUUFHF6x4r1Sx1S6hkS10iC3/AOPc6hbyyjUj1CxyRnajHBAQeZIeT5fGD1GlXk2oaVBdXNo9nLKuWhfOV59wDg9RuVWwRlVOQKGvQCbV/DZbGI9TZ8EZz/olwP65/CtuojuzWVSnKKUVqt/6/EKKKKsyMfxdatfeCNctEdI2n064iDyMFVS0bDJJIAHPUnFbFY3jC0nv/A2u2dohknuNOuIokUZLM0bAD8zWzWr/AIS9X+guoUUUVkMKKKKACiiigAoorF8Q32oWsuk2uky2sM2oXpt2luYGmVFEEsuQquhJzGB17mk3YqMeZ2NqisH7J4t/6Dei/wDgnl/+SaPsni3/AKDei/8Agnl/+SaV32L5I/zL8f8AI3qKwfsni3/oN6L/AOCeX/5Jo+yeLf8AoN6L/wCCeX/5Jou+wckf5l+P+RvUVg/ZPFv/AEG9F/8ABPL/APJNH2Txb/0G9F/8E8v/AMk0XfYOSP8AMvx/yN6isH7J4t/6Dei/+CeX/wCSaPsni3/oN6L/AOCeX/5Jou+wckf5l+P+RvUVg/ZPFv8A0G9F/wDBPL/8k0fZPFv/AEG9F/8ABPL/APJNF32Dkj/Mvx/yN6isH7J4t/6Dei/+CeX/AOSaPsni3/oN6L/4J5f/AJJou+wckf5l+P8Akb1FYP2Txb/0G9F/8E8v/wAk0fZPFv8A0G9F/wDBPL/8k0XfYOSP8y/H/I3qKwfsni3/AKDei/8Agnl/+SaPsni3/oN6L/4J5f8A5Jou+wckf5l+P+RvUVg/ZPFv/Qb0X/wTy/8AyTR9k8W/9BvRf/BPL/8AJNF32Dkj/Mvx/wAjeorB+yeLf+g3ov8A4J5f/kmj7J4t/wCg3ov/AIJ5f/kmi77ByR/mX4/5G9RWD9k8W/8AQb0X/wAE8v8A8k0fZPFv/Qb0X/wTy/8AyTRd9g5I/wAy/H/I3qKwfsni3/oN6L/4J5f/AJJo+yeLf+g3ov8A4J5f/kmi77ByR/mX4/5G9RWD9k8W/wDQb0X/AME8v/yTR9k8W/8AQb0X/wAE8v8A8k0XfYOSP8y/H/I3qKwfsni3/oN6L/4J5f8A5Jo+yeLf+g3ov/gnl/8Akmi77ByR/mX4/wCRvUVg/ZPFv/Qb0X/wTy//ACTR9k8W/wDQb0X/AME8v/yTRd9g5I/zL8f8jeorB+yeLf8AoN6L/wCCeX/5Jo+yeLf+g3ov/gnl/wDkmi77ByR/mX4/5G9RWD9k8W/9BvRf/BPL/wDJNH2Txb/0G9F/8E8v/wAk0XfYOSP8y/H/ACN6isH7J4t/6Dei/wDgnl/+SaPsni3/AKDei/8Agnl/+SaLvsHJH+Zfj/kb1FYP2Txb/wBBvRf/AATy/wDyTR9k8W/9BvRf/BPL/wDJNF32Dkj/ADL8f8jeorB+yeLf+g3ov/gnl/8Akmj7J4t/6Dei/wDgnl/+SaLvsHJH+Zfj/kb1FYP2Txb/ANBvRf8AwTy//JNH2Txb/wBBvRf/AATy/wDyTRd9g5I/zL8f8jeorB+yeLf+g3ov/gnl/wDkmj7J4t/6Dei/+CeX/wCSaLvsHJH+Zfj/AJG9RWD9k8W/9BvRf/BPL/8AJNH2Txb/ANBvRf8AwTy//JNF32Dkj/Mvx/yN6isH7J4t/wCg3ov/AIJ5f/kmj7J4t/6Dei/+CeX/AOSaLvsHJH+Zfj/kb1FYP2Txb/0G9F/8E8v/AMk0fZPFv/Qb0X/wTy//ACTRd9g5I/zL8f8AI3qKwTaeLf8AoN6L/wCCeX/5Joou+wckf5l+P+RNbQxL431OdZ1aZ9Os0eHnKKslyVbpjBLMOv8ACenGdise2tWTxvqd2XQrLp1nEEDDcCslyckZyAd4wcYODjocbFb1vi36L8kYoKKKydQ8S6dpuoGxnF7LcrEszJa6fPcbUYsFJMaMBko3X0NY3sXGMpO0Vc1qKwf+Ew03/n21r/wR3v8A8ao/4TDTf+fbWv8AwR3v/wAapcy7l+xqfyv7jeorB/4TDTf+fbWv/BHe/wDxqj/hMNN/59ta/wDBHe//ABqjmXcPY1P5X9xvUVg/8Jhpv/PtrX/gjvf/AI1R/wAJhpn/AD7a1/4I73/41RzLuHsan8r+43qKztI1yw1yO4bTpJWNtL5MyTW8kLxvtDYKOobowOcVo09zOUXF2aCiiimIKKKKACiiigAooooAKKKKACiisiDVrt/E76VNaWwiFu0/nQ3Rd0w6hA6FBt3AsRyfuN6ZpN2LjByvboQeBv8Aknnh3/sFW3/opa3qwfA3/JPPDv8A2Crb/wBFLW9Sjsiq38SXqwoooqjIK5HxND4il1oxaM8y281mXR43wscsSzEA+m55IDj+IRsDxkHrqKadmROHOrXscEllcSafp6Taj4mhtJfMa7JEpuROVjCrlVJWP/WNlflz35xUd3p2r6trfiC30+1tngh1q3uhK+oGCQSpaWzKNvkSAqCFPXk8Yx19BrB8P/8AIc8U/wDYVT/0itqmfvWRvhV7HmktdF+aHW8/idYFW40zS2dQAW/tSQluOpxbAZPsMVL9p8Rf9AvS/wDwZSf/ABiteip5X3/L/IxlFyldO33f5GXo1pe28moz6ikEUt5dCYRwStIqgRRx/eKrz8hPTvWpRRTirKwRjyqwUUUVRQVlavp815qmhTwj5LG/aeXn+E208f8A6FItatY+tW1xPq/h6SBWaO31F5JyM/KhtbhQT7bmUfiK1pfFv0f5MTNiiiishhRRRQAUUUUAFFFFABRRRQAUUUUAFY+tWrXGr+HpVdFFtqLykMwBYG1uEwATyfnzgZOAT0BrYrG1u0nudY8OSwoWS21F5ZiB91TaXCZP/AnUfjWtHST9H+TEzZooorIYVyngzX7XW9S8RNbHhb8NyMZXykjBx7mI11dctp3h3RbrWNea70mxnYagMNLbo5GbeFjyR6sT9Sawqc3NFx/rQ563Pzwce7/JnU1l3/hzTNTnllv4ZJmmhMDq1xJt8slSVChsDJRc4AzjnvWfqWm+CtGjV9WsdBsVfOw3EMMe7HXGRz1H51keDvEGiR2viC80/amnNrRhtY7KBn3kW0IPlxxqSclWb5R0yemTVqck7P8AM6o4erVi5Shp9/6HbW8CW0CxRmQqvQySNI34sxJP4mpa42W60PxX4205ImtNVtbezvkcFBLGsqvadMjBID9R6ketbf8Awinh3/oA6Z/4Bx/4Uc0m9Pz/AOATUhVptK36foJrbrHqmgFzgf2iw/E204H6mtiuH8UeD7Ca+0GPS449L36gPMNnGse/bG8gJ2gZYbCAT03H1ruKmm5c0rr+rHPSc3Oakuq/JBRRRWx0GP4utri98Ea5a2Ss9zPp1xHCq5yXMbBQMc5yRWxWP4uluYPBGuS6e7x3UenXDQPGcMriNipB7HOK2K1d/ZL1f6C6hRRRWQwooooAKKKKACsHxB/yHPC3/YVf/wBIrmt6sHxB/wAhzwt/2FX/APSK5qZbGtL4vk/yZvUUUVRkFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGVb6fNH4w1DUWH7i4sLWBDn+KOS4Zv0lWtWse2trhPG+p3TqwtpNOs4425wXWS5LD0zh0/MVsVrV+LfovyQkFcnPrmk6L8Q9S/tjVLLT/N0qy8v7VcJFvxLdZxuIzjI/MV1lFYtGsJKN01ueHx6rFbWOl2mkaj4fsbTSNS+0RKmqW0EtxFGkSr5m2Rw8jqZ0OSoJ2k7RwbC+IrS/wDCsOmy3thbTWvhW601mn1iyKz3DpAqhdszHBMb8tj9a9oorP2b7nb9di9eT8f67ng9rqSWV/4W1CyufD8UOiy30r2P9s2qMFuZpPkTDlfljKnrjnGeuNi61zRvEPji31m+m0K1tYVslP2vULN7pGimlk3QukzBVDNHvB5ZdwHIAPsFFHs/MJY1N35Ndevd3/M8T8Naxovhvw7oYtTZG9sbpZb1P7T02Np82s0Z2skoDKsjjlzvw2eag0U2kljqQ1jxBoVpPe6z9uItdZhO+C4ZVuoGYMDjyxnjuFIORx7nRS9n5jeOTbfLq/M5Pyv+Ej8UQapoGpSW9jDaSQz31iYmW8cshRAWVg4jxJk4+UvgH74Gr/Y19/0Mmqf9+7X/AOM1r0VfInuebUSqNeXmzI/sa+/6GTVP+/dr/wDGaP7Gvv8AoZNU/wC/dr/8ZrXopezXn97M/ZR8/vf+Zkf2Nff9DJqn/fu1/wDjNH9jX3/Qyap/37tf/jNa9FHs15/ew9lHz+9/5mR/Y19/0Mmqf9+7X/4zR/Y19/0Mmqf9+7X/AOM1r0UezXn97D2UfP73/mZH9jX3/Qyap/37tf8A4zR/Y19/0Mmqf9+7X/4zWvRR7Nef3sPZR8/vf+Zkf2Nff9DJqn/fu1/+M0f2Nff9DJqn/fu1/wDjNa9FHs15/ew9lHz+9/5mO2i3zKQPEuqKSOojtcj/AMg1h6f8ORpN/dXmneKvEEMt3M1xOoltykkjHLNsMO3J+ldpRT5Im9KpKknGD0e/X8zyuy8Maxq3wu0yPS7lreWeO2uEcbTmL+zUhK8kfebI9s5966DUfB0kl94q1i38xtQ1GyltLWMMiqymCMDnrnemOSAPTnJ1vA3/ACTzw7/2Crb/ANFLW9WkZtRRjXw8HVm31b/X/M5PSPDs41Sw1DVLSHzVS8eRWw5jeS5WWIZ9VBcZGcEnB556yiihtsIQUFZBRRRSLCsHw/8A8hzxT/2FU/8ASK2rermtKvrTT9W8Ty391Daxtq0ah5pAgJ+xWxxk9+D+VS+htTTakl2/VHS0UyORJoklhdZI3UMrqchgehB7in1RiFFFFABRRRQAVh6+sja14YMYJVdUcyYPRfsdyOfxIrcrF1y4uIdY8OJA8ixz6k6ThScMn2S4YBvbcqn6gVtRvzO3Z/kxM2qKKKxGFFFFABRRRQAUUUUAFFFFABRRRQAVj61bXE+r+HpIFZo7fUXknIz8qG1uFBPtuZR+IrYrH1qW5j1fw8ts7rHJqLrcBTgMn2W4IB9RvCH6gVrRvzadn+TEzYooorIYVkaN/wAhXX/+wgn/AKSwVr1kaN/yFdf/AOwgn/pLBWct4+v6Mzn8UfX9GaU1tBcBRcQxyhTuXegbB9RmuZ8N6bph1fxJCtpaslvrazxJ5akRSG0gO9R/C2S3I9TXV1zPhvwcfDmqXd+urXd5JqAL3qz42yTZyJFH8GBldvPG305bWqOunJKEk5W7feOnit7f4lacY9qy3GnXskg3cs2+zUHH+6g/L610lcze+DjeeMYfEh1a7jurYokES48pYcfvIyv8W8knd2+X+7z01NdRVHFqNnfQyNZ/5Cugf9hB/wD0lnrXrI1n/kK6B/2EH/8ASWetepjvL1/RHLD4pev6IKKKK0NDF8Y3FxaeBdeuLJ5I7mHTbh4XjJDK4iYqRjvnFbVY/i65ls/BGuXNsqtNDp1xJGrIHBYRsQCp4IyOh61sVq/4UfV/oLqFFFFZDCiiigAooooAKwfEH/Ic8Lf9hV//AEiua3qwfEH/ACHPC3/YVf8A9Irmplsa0vi+T/Jly58QaVZnURcXiKdLgW4vAAWMKMGIJAHUhScDnpxyM0rnxtoFlpsV9fXktpBMzLF9ptJonfau5iEZQxUDndjHvXPXvw+1JtU1XVbTUke71HzWMc/CRSoyvaSBlXJ2GGIFW3DlyOpDPl8MeKpbmbU0ubBL9p7nyYLi7muYoYJoolKq5VSMSRbgoXABxU80ux0xpUNLy/q3p369jqU8RabLcmC2lmuWWMSM9tbSyxgFN4BdFK7ipBC53EMuB8wzBD4v0Se1nuFu3WO3S4klMtvJGVWAqJjhlB+UsAffjsayrDwpeaXrGi/2cscFnYokd1Kl/Mpu1S2MS7rfb5Zbd5Z3klsRrz0AS68FTT6fd263EYNxaatbgkng3k4kU9OwGD/Wi8iOShdav+r/APA+82bjxVo9vCsv2mSdWnW3QWtvJcM8jQ+cAFjVif3Z3ZAxjvWhY3tvqWn219ZSeZbXUSzQvtI3IwBU4PI4I61xtn4L1LQYYBpD2179m1g3kMV3O0QEH2M2yR7lRzlRt7HIXk5NdLoOlz6X4Q03Sp5gtxa2EVs8sHIDrGFLLuHPIyMj6imm76k1YUoq8Hf+uxq1SuNXsbW5eC4uFSWPyNykHjzpDHF2/icEfhzUen6bdWU7yXOt3+oKy4EdykCqvPUeXEpz25JrB1/wTLrXi231pNTurdbf7Fi1inKxT+TcNK3mrtO7Ab5eeDmm27aEQhBytKWhrw+KdKnlmRJbgCG4+zNK9nMkRl80Q7VkKBWPmHbwT37A1bi1exm1STTorgNdx7t8WDkbVjZucY4E0R/4F9a5WHwhqMllqWm3YiS2utU+3LcRanOXK/bfPCiPaBCduRmNuWGevIfbeFdT0LxO2paIsF5DI04ZNQ1CbzFEiWi58xkkZiDbNwT0YYPGBN5Grp0btKXe2x0Wn69Y6rd3FvYm5drd3jkdrSVI9yOUYLIyhWIYEfKT0PpWlXOeGNH1HRrrUku4oTb3N5cXKTJfyyk+ZM7qPJZAkeFfB2nkjJznNX7zSry6vPOg1/UbOPj9xBHblPzeJm/XvVJu2plOMFNqL0/rsWdT1O10ewa8v3dYVdE/dxPIxZ3CKAqAsSWYDAHeqUXirR5bD7YLpkhENxOfNgkjZUgcJMSrKGBViAQRn2p/iTRzr2imwErQhri3lZ0kKMBHMkh2svIOEOCO+OR1rmbrwXqraPZ6dDPbyj+y77Tbu6uLhzITcNGfPA2Hex2FipK8tjccZpNyT0LpQpSiuZ2d/wALf5nSaj4o0rSo7mS8ln2WjlLhobSaYQkIr/MUU7RtdTk8c+xrTgmW4t45o92yRQ670KnBGeQcEH2PNcnq3h3VpdTmksorK8srnUVv57e4ungLNHFAsQysT5UPCWI4yQnONynp5Yri60/yzO9lcOo3SWxVzG3faXUgjtkr07Cmm7kzjBRi4vf+vUs1jTeK9Jt2u/MkufKs0lee4WymaFBGCX/ehChK7WBAJOQV68Vf0+0ms7YxXOoXOoPuJ825WMMB6fu0UY/DPNchrXhHWdQudUTTZbfTrK8sbyB4BeyyR3Esy4RzEV2w4YlmKZLFjnPUjbtoFKFOUmps34fFuiy27TtdPbIk7W7/AGu3ktykixecQwkVSP3Y3ZPGO9PPijSF1KGwe4kS5mijlCvbyKqrI22MOxXajMwwFYgk4GMmucvfBmqa3DPJqElpp922sDU4DbStcLAyWYhjbLIm4iRVfaQAQMZNXJvD2sT6nNuNillfXVnqF0wkcyRXEDRbkQbQGjZYEAJKkFmODwKm8jX2dDv+P/A16nQpq9jJ5Gy4B+0XMlrF8p+aWPfvXp28p/b5fpUWl6/p2szzxabM8xgVHZjC6KyvkoyMwAdTtOGUkVk6d4f1GKe1W8+yrFY6xdX8TxTMzSpN9pwGUoNpHnr0J6Hn1t+FtP1bSNNh0vURZtaWFrBbWssEjmSbYu1ndSoC5wpCgtjnk002ZyhTSdn/AFf/ACsbtQT3cFtNbRTyBHupDFCpH33CM+P++UY/hVabTrqXUluk1m+hhUgm0RIDE2OxJjL8+zd+MVW8W+H4vFPhPUNHmC5uYSI2fokg5RvwYA1TvYzjGPMlJ6P8Ca28RaVd2P2y2vFkt/Mhi8wK2N0wjMY6fxCaP/vrnvSXviPTLByk80rSC6Fn5cFtJM/mmLzdu1FJ/wBX82egHesW+8MarP4kkkt7m0XSLq+tr+4jYP5okhCABcfLgmKL0xg9ag1LwZqGtW87T339m3VxqbXwexuH3W5FmbZNsgCljkK5GFHJXkDJi8jeNOjfWWn9f8E6OHxBpdxrUekwXayXklmL9EVWIaAttDhsbcZ7Zz7VpVxmm+B5LHxdaeIFuTFJBbpYi0SXdElqsIGwZQEsJRnOQNvbPFdnVRv1MqsYRa5HfT8QoooqjEKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDDtFkHxA1ZyD5R0uyCnPG4S3Wf5rW5WPbS3J8b6nE7ubVdOs2jQn5Q5kuQ5A9SAmfoK2K1rfEvRfkhIKKKTAznHPTNZDFooooAKKKKACiiigAooooAKKKKACiikwM5xz0zQAtFFFABRRRQAUUUUAFFFFAGD4G/5J54d/7BVt/wCilrerB8Df8k88O/8AYKtv/RS1vVMdka1v4kvVhRRSEA4yM45HtVGQtclrHiq80+91RFjgSO1jkW1R4pHa6mS288jIwqAAjrknDAV1tVJ9K066u1urmwtZrhUKLNJCrOFIIKhiM4IJGPc04tLczqRlJe67HHf8Jxqcwu5orWC3jsYLqeeCaNmdzbx25eMMGAB3yyLvww+UYBqDUtQtvtPi7SrjTrq/F1qMYkijguxGyGzt+DLBE+DkYK9cHng89vNpOnXNxHcXGn2ss0TmSOSSFWZGIALAkZBwq8+w9KzfD/8AyHPFP/YUT/0itqmpaSSRvhOak5Sm77eXVEOjeJIl0Wzj1CLUnu44ES4ZNGu0VpAo3FQYhxnOOBV3/hJbH/nhqn/gpuv/AI3WvWZrz6xHpLyeHVt3vEIYR3CFg69wvzqN3pkgZ4JGciLTXX8P+CRJTnU91pJ+X/BJ9O1W01VJms2kJgl8qVJYXiZH2q2CrgEfKynp0NXK8+0KbxHLrVz9gvbFmvJ/tOpJcaTJEbJxEkQj4uGy5EakLyCCW3bWTd6DTi21qa1Kfs2k3cKKKKsyCsfWrmWDV/D0cSqVuNReOUlA2F+y3DZBP3TlV5HqR3rYrK1e+NpqmhQiGKT7ZftCWdcmPFtPJuX0P7vH0Y1rS+La+j/JiZq0UUVkMKKKKACiiigAooooAKKKKACiikIB6jPfmgBaxdcuLiHWPDiQPIsc+pOk4UnDJ9kuGAb23Kp+oFbVY+tXMsGr+Ho4lUrcai8cpKBsL9luGyCfunKryPUjvWtH4n6P8mJmxRRRWQwrC8P3KXOreJPL5EWqLGT6kWtvn9cj8K0dVs5tQ0qe1trt7OWVcLMmcrz7EHB6HaytgnDKcEc3oPhjU9N1e2nC2Om20CstyllcTSjUSVx5jo+FjfIVi5MjnAUvgZMS3RrGFOUW5PVbf1+B2FFFFWZBRRRQBi65Ls1jw2uM+ZqTr16f6JcH+lbPWuHvfhrpsV5NLomlaJtuyTKNSsRcCBs/fjHBORkFNwXkMMYKv1Gi6FY6BpYsNOhWOLcXfCgb2PUkAAD6AAAYAAAAqI3u9DacKUY3g9X5f8H5GjRRRVmJleKb46Z4P1m/EMU5tbCeYRTLlJNsbNtYdwcYNatZfie6gsvCOr3V5b/areCxmklg3bfNRYySue2QMZrUrR/w1p1f6C6hRRRWYwooooAKKKKACsHxB/yHPC3/AGFX/wDSK5rerB8Qf8hzwt/2FX/9Irmplsa0vi+T/JmFrfiTxRp+tX6aXbWNzZ2bzTyrdFlYwRQWrFYyv8ZaWTG4Y46gDm3pviy7uvHV3pl3JYQ2McFzKsJBW5gEMkUe+XLYCuWdl+UZUKcnmtSbxNbWv9tF7K6W20ONmuLgCPYxWFJSiDduJ2OOqgdRmkaTw5D4our2TUbYataWjC4SS/JNvB8jMTEWwi8IS2B2OeanW+50Xjy2cOm6+RL4omu4NBd7C6azk8+3ElyoQmGIzIJX+cFRiMuckEDrXLL8Qo4bbw6LWdruPV5pLdJr3y/NldLyGBseVhD8skrDb1CqfWunbxboa31vbDUbdhcWs12k6yqYvLiIDkvnA6n/AL5b0plp408PahrVvpWm6vZ3tzcRySILW4SUDZtyDtJwSGyPUK3pQ99GTTTjG0qd+v4ehy8XifXNPjvI9RuJZjbyROszW6jfCupyxTt8oA4gERP90HJxnNQ/8LD1/UNU8QWOkadpu+xultbNpbtXZmFykDmSJXDqDuLA4A475Ge1i8TaRdfZG0+/t76O7uWtUltJVlRZFiaUhiDgfKh/MetIfFfh1bNbtte0wWzMVWY3kewkYyA2cZG5fzHrSt5mnOr60tf66fL8zlNW8T+M7DXrqG0ttIubKxikmnEkcsckqRQ27ybGDMAzNOwUEYG3knrWl4f8Zzax4yvtIk/s/wAqL7T5SW9wXuIvImWJvOTHy7y25cdhXUXt9aabZvd6jdQ2ltHjfNPIERcnAyx4HJA/Gqen+I9K1GKwMV7DHPqFql1b2ssqrM0brkHZnPY9PQ+lO1nuZ88ZU/4flcqN4it7XxzcaPf6hbW4ksraS0glkVHlkeSdX255b7kfAzj8a5vR/H99rnh+21azit3VLoxzxxuqK+NNM7IWc4XEx25yMBee9dbquunTdRtbGDS73Ubm6ilmVLUxDakZQMSZJEHWVeme9ULv4heFbPS2vZtbsh/ooultzOizujRiRQI2IbcykYBAJyKH6jgrq6p3v/w23mL4G8R3PiXw4bnUVtY7+Gd4LmOzlWWJGB3KA6swPyMmcHqSOMYHMp4v8VT3niG3Onm2h0601RrO8YoRdSRyr5OFxwEVtp5IJ68g12EPjLwxcSFLfxHpMrhGcql9ExCqpZjw3QKCSewBNSDxR4eksRdrrmmPaNL5AmF3GYzJjOzdnG7HOOuKN1uNPlk26e/roc1qvj25s/EniHTrdbVho+l3N35TBjIXjigkRmIYYRvOZcYz+7zu5wKt78R75rTRn0Kxtb+TVbu7SFTOkayxQ3AiUKzuo3MrBgfm5GNuDleot9b0ebVZRcFbG7sbh9OiW6kSMyblikPljd8wIaLHf2Hfdos31E504KN6f476f0/wMXwxqzavYXjzSCSW11K7tXwANojndUHH+xsqa01W8ub4QTaBqNpHz/pE0luUHHokrNz0+7WpXODxrYLFqEs9reQR2Ud3KGdUP2hLV9kzJtY9GwAG2k5GB1xW27MUnNtxiL4l1bUtLmkk05EkWLR766Eci5Vpo/J8oHkHHzPwCP5Viy+ONTiTTIbu2srG8lu4be4WZyyzs12bd0t+QWICtISc7QUyDuyu9e+LILXUhYw6fe3lwbw2QWARgeZ5Cz4y7qMeWxP/AAFh6Zm1y30y6vNIi1SOaRnvR9lWOZ0QTIrTKzhWAYDySRkHnHHJqXd7M2g4xUVOHn52NZJEkUtGyuASpKnPIOCPwIIry7Q/Gmqv4Zvb+21yDWZ4dN026l+0G3CW0sskguE+QxDIVQQjsDkAZ5rrU8e6K89lBumWW+tLW8gRkGWjuJREnfqGYZHYHjNW77xZpmm+GbzXbtpEs7OWWGT5csXjlMRAGe7rgfUZxQ7PqFOMqatKF7tW/wAvmYB8ezot3eOdLFrb6YL2K2+0sJ7sfZ/OMkQYAmMH5BlQThjwV2mLwl461fxRpGnz/ZLGCd9Yaxu/Lk86MRC3abcjK2Mn5V6kA549Okv/ABZpWnaillcykTSG1EQGP3huJGjTbzzgoxPoBnmtuhJ33CU4Rj/DtfZnIeM7nxCmp6fbeHDekyWtxI6WbWylnVoQhdp1YBMO+doJyRwazLnx/PoMUun6nPZ3mqaexS4iVwssyR6b9paUKOgMgK5A24OMZr0KmSxiaF43LBXUqSjFSAfQjkH3HNNxd7pkQrQUVGcE7fecPonifV9fhhVbywJOqtZyXOmYkjCfYDN8rMWDFZWAyOu3kDkVnaJ8QPEVzfGwfSra7Fjo8VxdzveRQyNcG2SbJUkbULOF3AYBOcgCu80nRLDQ47hNOjkQXU5uJmkneVnkIClizknooHXtWhS5X3Ldald2hdf12POdP+IWq3Q8Kl7eyf8Atie7inwyx4EVwsSlCZSM7WLEKZCxGF4OR6NRRVJNbsxqzjN+7G2/5/psFFFFUYhRRRQAUUUUAFFFFABRTVRUzsULuOTgYyfWnUAFFFFABRRRQAUUUUAFFFFABRRRQAUUmBnOOemaWgAooooAybe9uZPGOo2Ln/RYbC1mjG3o7yXAfnvxGnHb8a1qxrW7kfxzqloQnlxadZyqQgDZeS5By3Uj5BgdBzjqa2a1qq0tui/JCQUUUVkMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOT0ODxbovh3TtL/ALK0Wb7DaxW/m/2tKu/YgXdj7McZxnGTV77X4t/6Ami/+DiX/wCRq3qKm1ups6ik7uK/H/Mwftfi3/oCaL/4OJf/AJGo+1+Lf+gJov8A4OJf/kat6iiz7i54/wAq/H/Mwftfi3/oCaL/AODiX/5Go+1+Lf8AoCaL/wCDiX/5Greoos+4c8f5V+P+Zg/a/Fv/AEBNF/8ABxL/API1SeHrHUbWbVrrVorWGbUL0XCxWs7TKiiCKLG5kQk5iJ6dxW1RRYHU0aSSv6/5hVDUdc03Sbi0g1G7jgmvZVht4zktI5IAAA56kc9Bnmr9c/q/hl9S8QWWrW2oPZy25jEgVMmRFk3lNwIIDAspBypB5UkAi1a+pz1HNR9xakknjPQIrSK6bUFaGZVdHSN3+VhIQTgHAxDJnPTac4rZiljnhSaF1kjkUMjqchgeQQa4+1+Hy6W0Emj30cMkJhO24t2ljOyGWI/KHXG7zmbr97J5zXTaPYf2Todjp3m+d9kto4PM27d+xQucZOM46ZpyUehnSlVb/eKxdoooqTcKx9auVg1fw9G0KSG41F41Zicxn7LcNuGD1+XHORhj3wa2Kx9akt01fw8s8TSSSai6wMGA8t/stwSx45G0MMcdQe2DrR+L5P8AJiZsUUUVkMKKKKACiiigAooooAKKKKACiiigArK1e+NpqmhQiGKT7ZftCWdcmPFtPJuX0P7vH0Y1q1j61crBq/h6NoUkNxqLxqzE5jP2W4bcMHr8uOcjDHvg1rSV5bdH+TEzYri7rx7dafHqF7faG8elWFw8Mtysz7ztlESlVaNVbJOflc4AOTnAPaVXnsLO6tHtbm0gmt5G3PDJGGRju3ZIPBO7n681nFpboipGbXuOxyMPxCe81bTLaw0hprfUSpjnaR1+QyyJuAEZH3IjJhmU4OO1dDfatNa+INN0yC2SX7aksjyPKV8tYymcAKck7+5A4681am0yyntWt2t0WNkVP3fyFQvK4K4K4JJBBGDyMVFY6JZadcGeD7RJMVKCS5u5bhlUkEhTIzbQSBnGM4Gegptx7ERjVWjdzHPjSJbvwtE9rsTxFC0iu0v+oIjVwuMfNkuF7Vd8NeIl1+zV5Ikt7n7PDctAjs+yOVdyHcUUEnDAgZxjrVy00PSbBCljpdlbKzq5WG3RAWU5U8DqD0Panafo+maT5n9ladaWXm48z7NAse/GcZ2gZxk/nQ3G2gRjVUk5PT/gf56/Mu1S1jUP7J0O/wBR8vzfsdtJP5e7bv2KWxnBxnHXFXajmhiuIJIbiNJYpFKPG6hldSMEEHqCO1SbO9tDlNT8f22kRt9rtd8kK3K3HlM3lwzRKjCMsyLneJFAbGMkAZzV/Q/Ez6vrF/YyWawC2eRYnSfzDIEleJiy7RsOVBAOcgnBO1sXk8O6JHZPZpo9gts5JeAWqBGJKk5XGDyi/wDfI9BVi102xsZ7iaysre3luW3zyRRKjStknLED5jknk+pq2420RjGNbmu5aFqiiioNzH8XSW8PgjXJL2JprZNOuGmiVgpdBG25QSCASMjODWxWP4vFs3gjXBqDOtodOuBO0a7mCeW24gZGTjPGR9a2K1f8Jer/AEF1CiiishhRRRQAUUUUAFc14tvIrDUPDVzOszxpqrZEEDzOc2dyOEQFj17D36V0tYPiH/kOeFv+wq//AKRXNTLY1o25tez/ACZzMy6Pc6h4jmuItS8rXoTDI8fhi8W5iUwpEVExjOV+QsF29W/OKKHTHvtdm1FtVuE1q1+zyhPDl+rxZhjifYxVgA3l5IxnO3LHaKr6zo13/a3iFNNt7nUpL60vStzPaXEc9qzQlViSdhslhLfdjHQsGAIGRe17VfGGl3kNhYC5uY1nkCXz2rNvCxQMgl8m3cbS8kwO1UJCDDqQc5ep6SV7KEt13S2t5eQzWHtdde3l1CfUxKun3djcGHwxfKsqzhRlQVJXbtHGTnJ6VZW40xfFb6v5mumF51uvsp0C9+WcQGDeG8roYzgrg8qpBHIOVrM/i/Vl1N3S8gs7LVYikEMDmWRI7tVRo18kZQxhpGIeTlVI2LlT6RZTSTfaPNfdsnZF/wBGeHCjoPmJ3/744PaqWrMqj9nBXfdaNeT7eZ57b6d4fg0aPTca55KvvJTQL1Sf9B+yH/lnwdvzZ9fzqv4lsYvFXhqLS9U1XW5JIlZPO/4RzUFSQHbh3jQKHkUpkE/KNx+XvXZaIZda1ibxA5dLMRtaafEcjcgf55j6iQom3r8qKQfnIHRUct0RLEOE09brzWj+45PWdd0rWNLe0aDWEPmRyozaBesoeORZFyBGMjcoyMjIzyOtYOnWmiWN/ot1NHql22madBZET+HL4hmhDeXKg2YRwXfkhjg4BHJPpVFU43dzGNZRjyxvb1X+RxWrXXhrXNUs7vV9K1G9S0hmiSC58OXcq5kaM7sNCQCPLx0/iP45mrWmiazqVxJdNrosbjzJHtE0C8GZntxbb93ldBGMAY4JJyeAPSKKHG441+T4b/ev8jy2fT9Pv77ULvVJL+afUIJUmkj8K3ylHeHyMx7g3lr5YXIX5mK8vt+QaVxcWg1t9W0y51q1unuGlYSeHL2SN1aKKMoy+WCf9QjAgjFegUUuQp4m+9+3Tb7jy3xZpeleJre+SO6160N5cGdl/sLUGjDeXCgYoqpuZfIBUtkDc3ynrXX3XifSby1e3mg19UccmHSL+JhznhkjDD8DXR0U+WzuRKspRUXey21X+Xkcvp/iDRtNtjBbQ+InQsWzc6ZqM7Z/3pI2OOOmcVzog0+WHUYb2fV5YrmHUIIBH4bvUMSXkvmvvOw7yCAARt4zkc8elUUctwjWUXdX+9f5HE/btH/tj7f5et7/AO0ft+3+wb3732T7Nt/1Xpz/AE70/X9XsNZtrYW51q0ubS5W5gm/4R+9cBgCpBHlg4KswOCDzwRXZ0UcrF7WN07PTzX+R5q2neG2/s3MeuudMj09LZn0G8LD7I7lSSIhksJCDx789Kintre+RbDUb3VX0ddRnvxb2/hrUYZy0jySKDMP7jyBgQoyUX3r0+ilyGn1p+f3r/I8rGgeGZLzSry9j1y5vNHitYbKdtCv1MaQTM4BATaxKkKSR1GQOcDtv+Ew03/n21r/AMEd7/8AGq3qKajbYipWVT47v5/8Awf+Ew03/n21r/wR3v8A8ao/4TDTf+fbWv8AwR3v/wAareop6md6fZ/f/wAAwf8AhMNN/wCfbWv/AAR3v/xqj/hMNN/59ta/8Ed7/wDGq3qKNQvT7P7/APgGD/wmGm/8+2tf+CO9/wDjVH/CYab/AM+2tf8Agjvf/jVb1FGoXp9n9/8AwDB/4TDTf+fbWv8AwR3v/wAao/4TDTf+fbWv/BHe/wDxqt6ijUL0+z+//gGD/wAJhpv/AD7a1/4I73/41R/wmGm/8+2tf+CO9/8AjVb1FGoXp9n9/wDwDB/4TDTf+fbWv/BHe/8Axqj/AITDTf8An21r/wAEd7/8areoo1C9Ps/v/wCAYP8AwmGm/wDPtrX/AII73/41R/wmGm/8+2tf+CO9/wDjVb1FGoXp9n9//AMH/hMNN/59ta/8Ed7/APGqP+Ew03/n21r/AMEd7/8AGq3qKNQvT7P7/wDgGD/wmGm/8+2tf+CO9/8AjVH/AAmGm/8APtrX/gjvf/jVb1FGoXp9n9//AADB/wCEw03/AJ9ta/8ABHe//GqP+Ew03/n21r/wR3v/AMareoo1C9Ps/v8A+AYP/CYab/z7a1/4I73/AONUf8Jhpv8Az7a1/wCCO9/+NVvUUahen2f3/wDAMH/hMNN/59ta/wDBHe//ABqj/hMNN/59ta/8Ed7/APGq3qKNQvT7P7/+AYP/AAmGm/8APtrX/gjvf/jVH/CYab/z7a1/4I73/wCNVvUUahen2f3/APAMH/hMNN/59ta/8Ed7/wDGqP8AhMNN/wCfbWv/AAR3v/xqt6ijUL0+z+//AIBg/wDCY6Z/z7a1/wCCO9/+NUVu0Uahen2f3/8AAMi2uVbxvqdqIUDR6dZyGUE7mDSXI2nnGBsJHGfmOe2Nise2ktz431ONImFyunWbSS7hhkMlztXGMgghznPO4dMc7Fb1vi+S/JGKCiiishhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFZerNZDU9DF4JDM18wtNgGBL9mmzu9tnmdO+K1Kx9aFsdX8PfaWcSDUXNuFXIZ/stxkHngbN5zzyAMc5GtL4vk/yYmbFFFFZDCiiigAooooAKKKKACiiigAooooAKx9akt01fw8s8TSSSai6wMGA8t/stwSx45G0MMcdQe2DsVl6s1kNT0MXgkMzXzC02AYEv2abO722eZ074rWl8Xyf5MTNSiiishhRRRQAUUUUAFFFFABRRRQAUUUUAZfieG0uPCOsQ6lP9ms5LGZLibaW8uMxkM2BycDJ4rUrH8XQxXPgjXILmdbaGTTrhJJmziNTGwLHAJwBzwDWxWr/hL1f6C6hRRRWQwooooAKKKKACsHxB/yHPC3/YVf/wBIrmt6ua8W3ElrqHhqaCzmvZF1VsQQFA75s7kcF2VeOvJHT14qZbGtJXlbyf5Mgm8QaxFqfiYyQRxaZpEYaGcWvmFyIY5XB/fKWOGYABQOnzZBBq3PxHh06OFJ9J1LUrqe8voEh0m285lW3nMeWBYHlSh47k9Kbf8AxC0nT7m5stT0hrWd/wDj4t7i/wBNRmyoHzqbnJyu3r2x2qnHrvhjW5YFi8G2+oSTXEzwBZdLlLzYV5Sv+kHL42sxHOME9qzv2Z2xpKyc4aeTXb1+Z38Ery+Z5lvJBscqvmFTvA/iG0ng++D7CuYsfEWpJ4YtfEepeTc2t/FbPb2Fja7Zka4dFRDI8u1sGQAnCZ68dKdpfiX7TZm70XwnfzW1y7SGa0nsGSV84Zty3GCcjBPtVZraxdrkv8M52N3/AMfJMOnHzvmDfP8Av/m+YA89xmqbvsYRpqLalb718+uhZg8eQXE0ca6LqoBlSGdyIMWzvcvbAPiXJ/eRtnZuwMGoLLx6P+EYsNT1XSL2F7jS5dRk8oRsgjijR2YfvDgNvAUHn1C02XXLHw5p6CbwbPpVmpVVDy6dBGCGLqB/pAHDFmHuSaTS9Q0+40f/AIkngSaXTZ0kT/RDpxhkVsK4+WfaQdoBHfbg9KV33NPZwtfl0v3X+Ze0/wAXf2h4oi0l7CewcJcLNFdlDIJI1tnUAxuykFLjPXPHaptZ1bVLbxZommabbxywXkc8t2xjDNGkbRLkZkQAfvTk4Y9MKeax306wNvHDb/DnUbIRO0kT2MlnayRswAYq8VwrKSFAOCMgAHIo0+8srnBtvCGqanJp9y6rNcX1neSWsw27lDyXTMjDC5XIxgcUXYuSF+aK6d1567ly5+INla2esXEulaqV0qVIiqwozXJed4AYgH+b50brg4xxzUl743SzFv8A8SHV5TNBaysoSFGhNxIY443EkikPvGCBkDuaS4ujdW8sF14A1CaGbHmxyCwZZMMXGQZ8H5iW57knrVGbUbfX9Ju9PtvCN7LAhW0mEDaZKIjEQREytMygqT90jjPQGi7BQpv7PXuttPP1J73x/b6fpo1S5sr77PsmQ2SWyvcCWK6S3YbhIUPzyDCjOQCc9q2dC11PE/huPVNNimtPP8xES9iw8bo7Id6BuzKeMj8K5SBdO8N+D107VfCF/JpVvv3yanJpoQCSbzMH98qAbyuAABwvoKsLp0MLD7J4M8TWUSnK29lq0VvCn+7FHdqi5PJwOSSTyTQnK45U6XLp30d1t95ZtfFl7bafcXetmErajUp5EtbRgXhtJPL4LSYVyfmwcgg44xkyTfECCHTdKum0HWt2plhHbvBHHLFiZIh5geQBdzSpjk8HPAo1HX4rPTZ5tX8H3kFlsdJnupdPWPbIw3Bi1xjDMRkHqaq6P4n0vUoBDoHhWS7is8YjsrjTnWDLbhwtx8uWTd9Vz2ou9rhyRa5nHr3Vv62Lkfj63kuI4To2qIfMiinZhBi2aS5e2QP+9ycyRtygYYwaSX4h6ZDpd3qEtnepa28NrcpI3lKJ4bmQpFIuZPlGVJO/aQOoqvJBBPrS6rcfD2/mu41QRPIunsYSru+5CZ8qxZySQeuD1qR0tpI7yN/hvdMl8ytdqY9PxcFTlS/7/wCYg8jOcGi8g5KN9vxXz6/1+Jr6vrZtfA99rtgI3aLTpLyAOwdCRGXXJRsMOn3Wwex71hXXjmXw1p+pS+Jo/tz2eo/Y1bTLcRbl+yLcliskpxgF/wCLnAwKgkay0a2luLvwprtvYrG0ckN3q1v9kWNhtKeU935QTBwF24A4AFSy6dYS2cFn/wAK2u0sobk3P2VItOETybGTcyedgnDdevA9KG2OEKa+JXXqv8/6uac/ja0g1fXLA6bqb/2Jatc3FykAML4jWTy0bdzIVcYU46Gs6++KOl6drVzpl3pWrpJbQNNLL5CeUCtsbkx7t/39gxj1745q59p/0y4u/wDhX9/9puk8uebFhvmXAG1m8/LDAAwfQVDOltdX8l9c/De6mu5UKSXEkentI6lNhBYz5IK/Lj046UNy6ExjS+0un8y3+/1Lt54oexhN1f6fd2EdvZ3V7PbzRxySPHCEztZJSqnLjAOc4P3epqXHxAitVuhL4f1jzrMTNcwKLctEsUcMjMT5205WdMBSSeeOKW3aK0s3s7X4dXkFtIro8EaaeqMrgBwVE+CCAMjvgVQn17Q/Ctutvc+DhosNyJAsckum26yghQ+AbgA5ATP0XPahtjjTg3blv81/n/mdDqniD+zPFGj6W8LSJqcc4QovPmIYyBkkADY0jHPPycZPBbaeJku9QNja2V1dSpI4neNERbdBNJErNvcEgmGTG3cSFzgZArnrLxtoGpX6W+m+H0u7uWRpVjgvNMeR3yHLAC5yTlFYnr8oPatSa4+0XEE9x8P7+Wa2dpIJHFgzRMx3FlJnyCTySO/NPmvsS6SikpLX1Xn5/wBfgU9C+KGmeIY3+waTq6zkxiC3nhjje58xJXUoS+3G2GQ5JA44JJxW7puutceBrTX7qAs0mmpeyQ245JMQcqu4/gMn8axYbWxtrS4tbf4ZTxW9ztE8UcOnKku3ONwE+DjJxnpmqzaXAyGI+DfFH2cjb9lGsRiDb/c8r7ZsCY42Y244xjiknLqVKFFv3VZeqf6mzpfjO01mLR5NPsL2VdWhlnj4iHkxxuiMz5f1kXhdx68dqk8OeJH1q81OzuLGa2msLiRPMZQI5o/PmjRkO4k8QnOQOenBFZ8YgiS0SL4c3aLYsWtQsengW5J3Ep+/+UkgE4xzzV2LV7uB2eHwTq0bP95kexBbktz/AKR6sx+rE96ab6kShCzUV+K7+vY6Kiual8XT295b2k/hrU4rm63eRC91Yh5doy21TcZbA5OOlT/8JBqX/Qoa1/3+sv8A5IquZGHsZeX3r/M3qK5i+8ZS6ZB52peHNSs4s48y4u7GNc4z1Nz6CpofE19cQJNb+FNXlikUMkiT2TKwPQgi45FHMh+xna+n3r/M6GisH/hINS/6FDWv+/1l/wDJFH/CQal/0KGtf9/rL/5Iouheyl5fev8AM3qKwf8AhINS/wChQ1r/AL/WX/yRVSPxs02qS6ZF4f1B7+FN8tot5YGWNeOSn2nIHzLyR3HrRzIaoze1vvX+Z1NFc3beLbi8837H4Z1SfyZDFL5VzYt5bgAlTi44OCOD6ipv+Eg1L/oUNa/7/WX/AMkUcyE6Ulvb71/mb1FYP/CQal/0KGtf9/rL/wCSKP8AhINS/wChQ1r/AL/WX/yRRdB7KXl96/zN6isH/hINS/6FDWv+/wBZf/JFH/CQal/0KGtf9/rL/wCSKLoPZS8vvX+ZvUVg/wDCQal/0KGtf9/rL/5Io/4SDUv+hQ1r/v8AWX/yRRdB7KXl96/zN6isH/hINS/6FDWv+/1l/wDJFH/CQal/0KGtf9/rL/5Ioug9lLy+9f5m9RWD/wAJBqX/AEKGtf8Af6y/+SKP+Eg1L/oUNa/7/WX/AMkUXQeyl5fev8zeorB/4SDUv+hQ1r/v9Zf/ACRR/wAJBqX/AEKGtf8Af6y/+SKLoPZS8vvX+ZvUVgnxDqQ/5lDWv+/1l/8AJFFF0HspeX3r/MtW7WR8XagsYk+3CxtTMSBtMXmT+XjvnIkz+HvWpWJNol6vii71mx1CKI3FhFaeRLbmRVaOSRw5w654lYY49c9qdaWviZJib7V9Jmj2sAsOlSRndjg5Nw3AOCRjkcZHWuqcYS1Uui79vQx1Nmismyt/EMd8G1HVNMntecxQabJE59PnM7D/AMd59qZDbeJVvEa41bSpLYOC0celyI5XuAxuCAffafpUezj/ADL8f8gubNFYRtPFnnZGtaMI8/dOjy5x9ftP9KkuLXxM147Wur6THbFsrHJpcjuFz0LC4AJx32j6U/Zx/nX4/wCQXNmism/t/EMl2W0vVNMtrfAxHcabJM445+ZZ0HX/AGaZdWviZ5FNlq+kxJsUMJdKkkJbAyQRcLgE5IGDgcZPWkqcXb31+P8AkFzZorGmtfEzQwi31fSY5VXEzSaXIyu2TyoFwNoxgYJPIJzzgST2/iBrCBLbU9Nju1z58smnSPG/PG1BOCvHqzZ9qPZx/mX4/wCQXNWisoW/iD+zCh1PTft3mZE/9nSeVsx08vz85z33/hUaWviYWciyavpLXJYGORdLkCKuDkFftBJOcc7hjB4OeD2cf5l+P+QXNmisa3tfEyxzC61fSZHKYiMelyIEbI5YG4O4YyMAjkg54wX2Vv4gj87+0tT0y4zGRF5GmyRbH7FsztuHsMH3FDpxX21+P+QXNaisa2tfEyzMbzV9Jli2sFWLS5I2DY+U5Nw2QDgkY5HGR1pLa18ULdxtd6xpEtuGBeOLSpUdl7gMbhgD74P0o9nH+dfj/kFzaorGhtfEy3qNcavpMlqHBeOPS5Fcr3Ac3BAPvtP0ouLXxM147Wur6THbFsrHJpcjuFz0LC4AJx32j6Uezje3Ovx/yC5s0Vi3Vr4ne6drLWNJityx2Ry6VLI6jsCwuFBPvgUt1a+JnkU2Wr6TEmxQwl0qSQlsDJBFwuATkgYOBxk9aPZx/nX4/wCQXNmisaa18TNDCLfV9JjlVcTNJpcjK7ZPKgXA2jGBgk8gnPOBJPb+IGsIEttT02O7XPnyyadI8b88bUE4K8erNn2o9nH+Zfj/AJBc1aKxmtfExskVNX0kXQcl5DpchQrxgBPtGQevO45yOBjkS18TCzkWTV9Ja5LAxyLpcgRVwcgr9oJJzjncMYPBzwezj/Ovx/yC5s0VlW9v4gWynW71PTZbpseRJFp0iInruQzktnthlx702G28RrDKLjVdLklK/umTTJFVGyOWBnO4YyMAjkg54wTkj/Mvx/yA16Kx7W28SI7G91XS5V2ttEWmSRkNjgkm4bIBwSOMjjI60ltbeJUuka81bSpYAw3JFpckbEdwGNwwB98Gj2cf51+P+QXNmisiC28Rrdq11qulyW+4bo49MkRyvcBjOQD77fwomtvEbXjtb6rpaWxYlI30yRnC9gXE4BPvtH0o9nG9uZfj/kFzXorGuLbxK127WmraVFbliUjk0uR3C9gWFwAT77R9KdcW3iNplNpqulxRbV3LLpkjsWxyQROMAnJAxwOMnrR7OP8AOvx/yC5r0Vl3lvrz+T/Z+padBhAJfP0+SXe/criZdo9jnHqajuLbxI0UQtdV0uOQLiVpNMkcM2TyALgbRjAwSehOecBKEX9pfj/kBsUVkPbeIzaIsWq6WtwGO+RtMkKMvGAF88EHrzuOcjgY5eYNd/s3YNR04Xu/PnHT38vZjp5fnZznvu/Cjkj/ADL8f8gNSisuGDXVsZluNS06S6bHkyx6e6InruQzEt+DDHvTIbbxGsMouNV0uSUr+6ZNMkVUbI5YGc7hjIwCOSDnjBOSP8y/H/IDXrH1qGKXV/DzyzrE8Oou8SHP71vstwu0YHXDM3OPun2BLW28SI7G91XS5V2ttEWmSRkNjgkm4bIBwSOMjjI61U/sTXbvWdLu9W1fT5bfTrhrhYbbTXid2MMkWC7TuAMSk/d7CtacYRbfOtn37PyBnR0UUVzDCiiigAooooAKKKKACiiigAooooAKydc0m51ObS57G7itZ9OvDcqZoDKr5hkiKkB1I4lJznqOla1FVCbg7xDcyJrbxG147W+q6WlsWJSN9MkZwvYFxOAT77R9Kde2+vyXhbTtT02C24xHPp0krjjn5hOo6/7NatFV7WXl9y/yFYx7q28SPKpstW0qKPau5ZdLkkJOPmORcLwTnAxx6nrSz23iNo4RbarpcbhcSmTTJHDtk8qBONoxgYJPOTnnA16KftZabfcv8gsZdxBrzWcC2upadHcrnzpJNPkdH9NqCYFfxZs+1Me28Rm0RYtV0tbgMd8jaZIUZeMAL54IPXncc5HAxzr0UlVku33L/ILGOtt4jFmyvqulm53DbINMkCAc5BTz8k9Odwxg8HPEkMGurYzLcalp0l02PJlj090RPXchmJb8GGPetSih1ZPt9y/yCxl2dvryLL/aGpadOShERg0+SLa/YtmZtw9hjPqKZbW3iNZGN5qulyptYKItMkjIbHBJM7ZAOCRgZHGR1rXoo9rLXb7l/kFjGtrbxKl0jXmraVLAGG5ItLkjYjuAxuGAPvg06C28Rrdq11qulyW+4bo49MkRyvcBjOQD77fwrXoputJ9vuX+QWOZ8Q6H4h1zTdR01Na0y2sr6GW3IOlyPIiOpX732gAsAeu3Ge1dNRRSlUlKKi9l5Jf1sFgooorMYUUUUAFFFFABWD4g/wCQ54W/7Cr/APpFc1vVzXi57uPUPDTadBDPcjVW2RzzGJG/0O5zlgrEcZ/hP4damWxrRV5W8n+TOH8dWWnS/Fg3N1pdjPHpPh251edJLZGN2wBjVZMj5toVSM5xjiuBsre28M3nhuW6u57a7g8OnVNOS1jLC5vJJnaNHCqchk2ox44QDcMCvcH8TaxFfGylt/DKXaqWMDeIHEgAG4nb9mzgDn6Vd+1+Lf8AoCaL/wCDiX/5GrFwTdz1aeKnTgoSjpb+Zef+bPBNW8XappHhvwj4f0LWf7FtG077RcXPzgiUyyq4YopbAZGGAOp5r13wPq+qar4p1iG98SW+oR6ZDDazWEFsV8m5ChZXMjRqWzJHLjBIwe3Are+1+Lf+gJov/g4l/wDkaoP7a8Q/2j9g/s7w/wDbPK877N/bknmeXnbv2/Zs7c8Z6ZpxjyvczrVlVg4qCT11vF7u99v1OJ+JV7pll8UNCm8ZwtN4djsJditAZYxOxKksMHPy7cY5Bwa5Pxl4xttFm0y38Ba9caVoUWmteWMVtC7LcTm6kVkYOQQuVY/NkAAgA5Fezxaj4onhSaDSNDkikUMjprUpVgeQQRbcinfa/Fv/AEBNF/8ABxL/API1Dje9n+A6WIUFFSje2luZW69LPXU8tv8AW/ENp8Wi8usXcYj1bSLGaxSZjbEXFuTLhDwOU4PuT1rmI7O8t7rUrLRvE2sWN6/jAabIsN4yRGOUuBKyLglyYzznkDFe8/a/Fv8A0BNF/wDBxL/8jUfa/Fv/AEBNF/8ABxL/API1Jwv1LjjHFWUVsluuh49beMtVh+IFppzeJ7ySaz8QxaOmnuGIltFYxtNK2MO7HGTnOcnHTHonwkZ5PC+oy3aeXqEur3Ul8hGNsxYEjnn7u3rW79s8W/8AQE0X/wAHEv8A8jVU0zXfE+q2r3FvoWkqiXE1uQ+ryg7opWjY8Wx4yhI9sdOlVFWerMa1X2tO0YpbdV5+h5T8Uda1HxB8T18IPqUlvp4vbK1FnGGCziUK7O5HXaxUAH2I5BqXRPE3iTVvj9c2I8QFLa11OeJNPmklWOaJC6sFVFK7lRSw3Y5A5zXqt5rPiLTrR7rUNO8P2tumN80+uSIi5IAyTbYGSQPxqSHUvFFxAk1vpOhyxSKGSRNalZWB6EEW3IqeT3r3NVibUlDkVrW3W/V/195yvx5vXt/hwlqsiRR39/DbyyODhF+aTPAJ6xjoDXDarr7XPh7RV8I3um6LLdG+fVrnw/ZvbKwtohKFXcFdiInOASBubqOo9ki1HxRPCk0GkaHJFIoZHTWpSrA8ggi25FEuoeKYIXlm0jQ4441LO761KAoHJJP2bgU5R5ne5FGv7KCg4p2bfxLqrbeWn3HhHizxj4qWy8J2UviG40oXGix3E1wJZF3lncK7sgZySqJzzyT6mtbVdS8Zy6d4s8XSeIbmwbR7qPTlsrO4Z7dp0aOKRgrDAU7iwGM5b259ObxreLYm8ZvCYtVl8gznxIdgkxnZu+z43Y5x1xVTWda1HW9Mk0sXmhWD3ESzLNYeKGinWMYk3o32Y/KVwScY2n3qOXzOlV37q9kl31W17s8o0C+1TxX4u0zSLjUr3WNIutXiliub+YkObWIvMqofuhhID2425yc42fhb4x1zxP8AFVyt9qBspRc3NzbT3RlhSMt+7WNCPk2llHBOfYV3PgYXWmaPNY+F4tB1RBM01zOPEb3EjyP1Z2W26nHt0rcv9e17Srf7Rqlj4dsoc48y412SNc+mWtgKcY2s7irYhScqcYLVWWq++33fdc5z4y6+uk2Ph+0bVr3SkvNUQ3FzYytHItuoIlIK5JxvU4weQODXlr+PtbDahp8vizU7bSLZbmXSrx438++IJEKNJszg8EkgDqOOMet38l7qnjrSZZo9AGs6MJjDYx+IW3nzYwG3x/Zd33cEYx68it651XxLZW7XF5peg28KY3SS63Iqrk4GSbbA5NOUXJt3M6NWNGnGDgm/Vd+mj6HD+FbnV/HHjq3ubrWdRtLXR9J06aeC0uTGl1cSos2XT7pUgspGASAOa6D4xxWafDPVL+4sLS6uLeIRwSXEKu0PmuiMyEjKnBzkdwPSp5vG13b2dvdzv4Sitrrd5Ez+JCEl2nDbWNvhsHg46Vrfa/Fv/QE0X/wcS/8AyNVJLlaOeUpqrGpypJW0uum+vmeL61YN4T1DxNdaCkNlP4Z0Sx08XEESo0k07R7puB98qXG7rUHh3VvFPiO30bRY/Et00OsatJL9shvZhdWyxQhpYGYqoI2yIRtyu4Gvb/tfi3/oCaL/AODiX/5Go+1+Lf8AoCaL/wCDiX/5GqfZ67nR9cfJaUU33uu3+ev4Hh3hTxdr+neH9Ym1bxBfXcV74bnvIDNO7PbzfaTbIVckkHdzkY6j0zVKw+IPiW20Zo7PxPcNpkklpBeX18rTT2ksyMXMZxuKjy39T0xnqfafE+keIvFnhy60XUdI0yK2utm94NakDja4cYJtCOqjtWr9r8W/9ATRf/BxL/8AI1Lke1zR4um05Omrt946bW6HhfiLxr4msPCOmiy1zUZre61K/W0u9xSe4gTyxExYqCRl2OMDPTjHH0FoL3kvhzTX1Tm9a0iNxxj94UG7jtzmqP2vxb/0BNF/8HEv/wAjUfa/Fv8A0BNF/wDBxL/8jVpFcrvc48RUVaCjGKVm9brW58/eG/FGvaz4guvEN3qt1qmoaVo91dQReXxbysfJ+VB8vR1ckDGOD0r0f4FeINW13TtX/trWW1GSKSJlilLmS33bwcllAwdoICk4wc47919r8W/9ATRf/BxL/wDI1H2vxb/0BNF/8HEv/wAjVEYcrvc2xGJVaDgoJXtbVaW6HA/Eq60nTfiNaaj4vtzd6TBo0hs7eSFpYnvPN6FcbeVI68cD2rgtS1vV/DfhLwfpkGuX9gJtKnujDaMUYs7yPBkjJIY7Vx6A9M1739r8W/8AQE0X/wAHEv8A8jVk/wBkeIf+Ey/4SX+yNM+2fYPsHl/21J5fl+Z5mcfZM7s++MdqJQu7oqhiVCKjOKdv7y87em55d4R8beMPFnxN0u3uNY/s7yjEsmnTCRRcqiZmIAQruIV2+bGMgA8VQ1HXfFNv4FtvF513W/M1nUpv3MF2fs8FupPygEHY24EAg4CjpXu32vxb/wBATRf/AAcS/wDyNR9r8W/9ATRf/BxL/wDI1HI7bj+txUk404paaXXS/wDwPuPIl8a3174S8Hp4h8Ralo9nc29097qdqr+ZKySNHEgZVJyMZPqO+cVW1fVry08Qavq2gXs0iW2iadpzaqYWd388Ryi5cH5gDGjAk5bJUY5yPZvtfi3/AKAmi/8Ag4l/+RqPtfi3/oCaL/4OJf8A5Go5PMlYpRd1BddOaNtXft8v0PDPDFzqd/8A2f4T8L6rdWujX2uTNbajFK0F0Y4og8qtt65WRSM915HTHsfws1a91z4Z6Rf6pO1xdSLIryv95tsroCfU4Uc1ofa/Fv8A0BNF/wDBxL/8jUfa/Fv/AEBNF/8ABxL/API1VCPK9zLE13XjblS1vuvP89PuN6isH7X4t/6Ami/+DiX/AORqqXmu+J7G7sLebQtJL39wbeIrq8uAwieTJ/0bpiNh35I+tacyOFUZPa33r/M6miuavda8Q6bZvd6jp3h+0to8b5p9ckRFycDLG2wOSB+NFlrXiHUrNLvTtO8P3dtJnZNBrkjo2Dg4YW2DyCPwo5kHsZ2vpb1X+Z0tFYP2vxb/ANATRf8AwcS//I1H2vxb/wBATRf/AAcS/wDyNRdC9lLy+9f5m9RWD9r8W/8AQE0X/wAHEv8A8jUfa/Fv/QE0X/wcS/8AyNRdB7KXl96/zN6isH7X4t/6Ami/+DiX/wCRqPtfi3/oCaL/AODiX/5Goug9lLy+9f5m9RWD9r8W/wDQE0X/AMHEv/yNR9r8W/8AQE0X/wAHEv8A8jUXQeyl5fev8zdorCN54tH/ADBNF/8ABxL/API1FF0HspeX3r/M3qKKKoyCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACua8W31ppuoeGrvUbqG0to9VbfNPIERc2dyBljwOSB+NdLWXqwzqWie183/pNNUy2NaTSlr2f5HnE+vWlrqmtQ+Hte0nT7TUEuppJTrtvIslw8eElQHEiPuC5BbYoUgDJBpdY8QWN9ca+dN8SR2jyRRiy3+JYxDKUm3uRtm3Rlx8uAAAg6gsQPWqKjkfc6liopp8v4/8AA8jznwt4i0PTLiJbzxTBLCtlsE2oeJYrh95mkbayDClwpXL5PG1RkAsc7Ur7Trzx9Pr1n440OzEludOjYXULtHD5DsJM+aAf37j5dufkHIBOfV6KfJpYlYmKk5KO+m//AADz7wh4i0nRYr6z1TxjoE1pFJFHYCG9RVSJII1wA0rkDKngsSSGP8VdH/wnPhL/AKGjRf8AwYRf/FVvUU0mlYynUhOXM0/vX+Rg/wDCc+Ev+ho0X/wYRf8AxVH/AAnPhL/oaNF/8GEX/wAVW9RT1IvT7P7/APgGD/wnPhL/AKGjRf8AwYRf/FVjeFfGXhi20edLjxHpMTnUr9wr30Skq13KynluhUgg9wQa7eiizvcrnp8rjZ/f/wAA4fxd4u8O3nh/ZYa7o95NHeWk/kLqdupkWO5jkYAs4H3VPU1yK3llpiaDBZ+K7FI7Wa3kkj0zXIYrdSblpLhZUeQboxG+ECDJwQwxtFezUVLi273NqeJjTjyqOl77/wDAPFNG1CGzvNFspfE9rb6XpwjUmDxPH88kb8swLndC8e0eXgYZMAIrFj2em+NtE03wrLnW9Lnv4RcPFaza/FM0p3uUXzmI4I2gEj5QQD0ruKKFFrqFTEwqfFH8f+AeRr4ss7nytfmudLOuW9zHIbSfW7SKJ4xFPGscbpI/CfaHYswBYk9BhVi/4SHS0Oi6Td3Wk3thoK2rWUseq2qrLcRxBRJJIZwyopZgVETH5dwz92vYaKXI+5X1qH8n4/15/eeZ6J4h0iW41ieXxJpnhwXs8cwEGsWtzK8gUh3LSb1CkbFCgcCPgDJzD4mufDupWSBvGGmaxN9murRXu9QtEeH7Qip5q7AqnZt6Yztd8EkBT6lRT5HaxP1mKnzqP4+Vu1/xPMl8SaOPiBd3Nlrum2Nncndc351y3ZZ8QBFCQbiFYNglmXP7vGSDir954u07SNMu7nTvG9nrt4VCw2t1fWKqCWALfKIs7Rk4LjOMZBOR31FHK+5LrwbV47W69vl+R4n4nGh6/wCH9Ojh8QafFclbtZ1Gr28BjkuZ45JHbbK4MeFk+QM5+ZR2JHp3/Cc+Ev8AoaNF/wDBhF/8VW9RQotO4VK8akVGSel+vf5GD/wnPhL/AKGjRf8AwYRf/FUf8Jz4S/6GjRf/AAYRf/FVvUVWpjen2f3/APAMH/hOfCX/AENGi/8Agwi/+Ko/4Tnwl/0NGi/+DCL/AOKreoo1C9Ps/v8A+AYP/Cc+Ev8AoaNF/wDBhF/8VR/wnPhL/oaNF/8ABhF/8VW9RRqF6fZ/f/wDB/4Tnwl/0NGi/wDgwi/+Ko/4Tnwl/wBDRov/AIMIv/iq3qKNQvT7P7/+AYP/AAnPhL/oaNF/8GEX/wAVR/wnPhL/AKGjRf8AwYRf/FVvUUahen2f3/8AAMH/AITnwl/0NGi/+DCL/wCKo/4Tnwl/0NGi/wDgwi/+Kreoo1C9Ps/v/wCAYP8AwnPhL/oaNF/8GEX/AMVR/wAJz4S/6GjRf/BhF/8AFVvUUahen2f3/wDAMH/hOfCX/Q0aL/4MIv8A4qj/AITnwl/0NGi/+DCL/wCKreoo1C9Ps/v/AOAYP/Cc+Ev+ho0X/wAGEX/xVYuueMvDE2seHHi8R6S6w6k7yst9ERGv2S4XJO7gbmUZ9SB3ruKyNZVTq3h8kAkag5Bx0/0Wek7mlN0+bZ7Pr5ehgeKvE3hzV9B+yWHi7R4Jzd2sqyrfwExiO4jcsAxwSApOCD06HpXKWV9p39t6FcXnjNLeOKNWvEsNft4rdrlZJZZpZY93zrK7JgKAdrHdtwBXsNFJxbdy4YiMIcij+P8AwDzPQPEdlaX1i+seLtMuhF5yPJ/b6BFCxwxIzR7j5hkMbyAEjZ5jDk8nrv8AhOfCX/Q0aL/4MIv/AIqt6qWqyPHZxtGzITcwKSpxwZkBH4gkU0miJ1IVJXcfx/4Bnf8ACc+Ev+ho0X/wYRf/ABVH/Cc+Ev8AoaNF/wDBhF/8VW9RT1M70+z+/wD4Bg/8Jz4S/wCho0X/AMGEX/xVH/Cc+Ev+ho0X/wAGEX/xVb1FGoXp9n9//AMH/hOfCX/Q0aL/AODCL/4qj/hOfCX/AENGi/8Agwi/+Kreoo1C9Ps/v/4Bg/8ACc+Ev+ho0X/wYRf/ABVFbp6UUahen2f3/wDAP//Z)

**Ilustración 11: Circuito esquemático**

## 2.4. Circuito impreso

El diseño del circuito impreso fue realizado utilizando el software KiCad versión 7.0.

![Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4RD0RXhpZgAATU0AKgAAAAgABAE7AAIAAAAOAAAISodpAAQAAAABAAAIWJydAAEAAAAcAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFNhbnRpYWdvIFJ1aXoAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzAxAACSkgACAAAAAzAxAADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIzOjA3OjAyIDE1OjU4OjM3ADIwMjM6MDc6MDIgMTU6NTg6MzcAAABTAGEAbgB0AGkAYQBnAG8AIABSAHUAaQB6AAAA/+ELIGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjMtMDctMDJUMTU6NTg6MzcuMDExPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPlNhbnRpYWdvIFJ1aXo8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgCggKFAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A8BtLZ7y9gtoioeaRY1LdAScDP510q+BLjy5PN1K0DqAyhNzBh65x/IGsTQf+Rj03/r7i/wDQxXoFyAJiAAOOQFK8+47H6cV7FGnGUW5Hm1Jyi7I5YeBb5tuL6xO4ZHzvz7fc6+3WnReBrp03NqNkN3C4L4z6HKjFdFRW3safYz9pPucz/wAIRejrfWI+bactIMH3+TipD4Fulidn1CzDoQNo3kfX7vT3xiuioo9jT7B7Sfc5s+Bb4Z/06x4Xdw7nI9R8nNOXwJdtHn+0LPewygBcg/jt6+2M10VFHsafYPaT7nNL4HvWVSL6wwxwCXcc+/ycfjTl8DXRidjqFmHVtu35+Pr8uR+VdHRR7Gn2D2k+5zLeB71d269sfkxn5pP/AIjp71J/wgt0EbdqFnvA3ALvII9c7f5ZroqKPY0+we0n3ObHgW+bbi+sTuGR878+33Ovt1pw8DXTQoy6jZbmbGCXAH47ePxFdFRR7Gn2D2k+5zQ8D3xBP26xG07TlpBg+/yfrTz4Fugjf8TCz8xOWX58Y9QdvP5V0VFHsafYPaT7nN/8ILfeZs+3WO4jI+d+R7fJzS/8INdeWm3ULMu+cD58H8dvX2xXR0Uexp9g9pPuc0vge9ZVIvrDDHAJdxz7/Jx+NO/4Qa68ksNQs96ttZTvAX/x3P6V0dFHsafYPaT7nNf8INfBnBvrHKDJG6Tp6j5OfwqVvAlxtRY9RtGmIyVIcL+DY5/Kugoo9jT7B7Sfc5oeB70lP9PsPn+6d74/PZxTn8DXQRCuoWZJO1g28bT6fd/niujoo9jT7B7Sfc5n/hCL0KzG+ssKcMN0mR+Gynt4Gugp2ahZl1GXB3jH0+XkV0dFHsafYPaT7nN/8ILfeZs+3WOSMjDvyPb5Oac3ga62xbNRsiZB/EXAH47f54roqKPY0+we0n3OaPge+ABN9YgFtvLScH3+Tj8afJ4FukjyNQsyV4cHfx9MKcj3roqKPY0+we0n3OaHge9O7F/YHaMnDuf/AGTmnN4Gui6rDqFm2V3fNvXP0+U5ro6KPY0+we0n3OaHge9JT/T7D5/une+Pz2cU6TwNdLtKahZkE7Tu3rg+h+U4/Gujoo9jT7B7Sfc5pvA18oYtfWOFODhpDj3+5096fJ4Eu1J8vULNgFDHJcflhTmuioo9jT7B7Sfc5seBb5tuL6xO4ZGHfn2+519utK3ge5LJ5Wo2bBxwXLrz3H3a6Oij2NPsHtJ9zmh4IvM/NfWSjdtPMnB/74p0vgW7Rn2X9myocEkuD9cBTx710dFHsafYPaT7nN/8ILegndfWOANxwznj1Hy80+XwJc+YBbajaSKwypfepPtgA8+3Wuhoo9jT7B7Sfc5tPA14+0/b7Haxxne/H/jvH40SeCLkSEJqFkRu2/OXU59xt4rpKKPY0+we0l3OaPge9Vm331kAhG7BkOPf7lPk8C3au3l39kyqN2SXBI9cBTn8K6Kij2NPsHtJ9znI/At4zpuv7IKwyCpcn8BtHPt1pP8AhBrxsGO/sSrHAJZx+fy8V0lFHsafYPaT7nOR+Brst+8v7NVVtrEFyV/AqKRvA16HcLfWR2cnLPnHrjbz+FdJRR7Gn2D2k+5zi+BrsSbZb+zC43EoXbj1Hy80h8DXu5dl9ZMH+6dz8/8AjnX2rpKKPY0+we0n3Oci8DXblC2oWSoxxuDOcfmo/XFIfA958xW+ssK207mcEfUbOK6Sij2NPsHtJ9znE8C3e9xLqFmuwZO3ex/LaMj3FJ/wgt95mwX1iTjI+d+R7fJzXSUUexp9g9pPuc5H4GumZfM1CzAcEqU3tk+n3R/jTf8AhCL0qrC/sdrHaDuk6+/ycfjXS0Uexp9g9pPuc5H4Gum3eZqFmgU4ON5IPuCowPemnwNfBnH26xynJG6Tp6/c5/Culoo9jT7B7Sfc5+LwJcNIyTajaKdm5Nm9t36D9M/Sov8AhBr3KYv7DD/dJd8f+gcV0tFHsafYPaT7nOL4GugpMuoWa87RjeefQ/KMU1vA18gYtfWPynB+aQ4/8c6e9dLRR7Gn2D2k+5zqeBro+YJNQswyDOF3nP8A4709xmmDwPfMwUX1jkruHzScj2+Tn6V0tFHsafYPaT7nOp4FujjfqNkpcfJguQT6E7RimHwPfBQTfWIBbby0nB9/k4/Guloo9jT7B7Sfc50eBbpVkMuoWalOy72/9l6e9NPgW+Gf9OseF3cO5yPUfJzXSUUexp9g9pPuc7H4GunJDahZhiu5AN53f+Oj+tRr4IvW2YvrH5+hLyAf+gV01FHsafYPaT7nOf8ACDXQi3PqFmG3bQPnxn3O3j8qa3ga+QMWvrH5Tg/NIcf+OdPeuloo9jT7B7Sfc53/AIQS7EbltQs96gMAC5BHrnb/ACBpo8C3zbcX1idwyPnfn2+519utdJRR7Gn2D2k+5zqeBrp4g39o2QZjhRl8fidvFR/8IRejrfWI+bactIMH3+Tiumoo9jT7B7Sfc5w+BrpY5N+oWgdCOBvIwe+dv9Kq6j4TvNNsZbqW6tZEiUMREzkkFguRlQDyfWutqDXAf+EPv8F9u1MjcCufNTt2P86UqMOVtIaqSujkdL0/SLu1Z9S1v7BKHIEX2R5crgfNkH1yMe1FZVFedZ9zrv5F/Qf+Rj03/r7i/wDQxXoVwp3ZZsYUYUtuH0U88V57oP8AyMem/wDX3F/6GK9Aucedx1xg/JsOfcdB+Fehh/gZy1fiRFR68/8A16KLm9stI0177UVaVm3JaW6kjzpBgncR0QBhnkE5AHcjWUlFXZmk27IKO3X8K5FvGviTzGa31q8s1P8Ayzs5jAg/4CmBT7Lxhqi3YOq3c+o27N+9W4cyPgnkqzZKt+h75FYLEK+qNfZO251dHbr+FPfy3SKe2YtbXCebCzdSpJHPuCCD7j0pldO5iFHbr+FFFABR26/hRRQAUduv4UUUAFHbr+FFFABR26/hRRQAUduv4UUUAFKQAoO4Ent6UlFABR6c/wD1qKKACj05/wDrUUUAFHpz/wDWoooAKPTn/wCtRRQAUHjvmiigAoPHfNFFABQeO+aciNI21evXk4p32d/WP/v4v+NF0NRk9UiOg8d81J9nf1j/AO/i/wCNH2d/WP8A7+L/AI0rofJLsR0Hg9c+9SfZ39Y/+/i/40fZ39Y/+/i/40XQckuxHSsArYDBh6in/Z39Y/8Av4v+NH2d/WP/AL+L/jRdByS7EdB4PXPvUn2d/WP/AL+L/jR9nf1j/wC/i/40XQckuxHQeD1z71J9nf1j/wC/i/40fZ39Y/8Av4v+NF0HJLsRjk9ce9FSfZ39Y/8Av4v+NH2d/WP/AL+L/jRdByS7EY5PXHvRUn2d/WP/AL+L/jR9nf1j/wC/i/40XQckuxGOT1x70VJ9nf1j/wC/i/40fZ39Y/8Av4v+NF0HJLsRjk9ce9FSfZ39Y/8Av4v+NH2d/WP/AL+L/jRdByS7EY574oqT7O/rH/38X/Gj7O/rH/38X/Gi6Dkl2Ixz3xRUn2d/WP8A7+L/AI0fZ39Y/wDv4v8AjRdByS7EY574oqT7O/rH/wB/F/xo+zv6x/8Afxf8aLoOSXYYADnLAYGee9JUn2d/WP8A7+L/AI0fZ39Y/wDv4v8AjRdByS7Efrz/APXoqT7O/rH/AN/F/wAaPs7+sf8A38X/ABoug5JdiP15/wDr0VJ9nf1j/wC/i/40fZ39Y/8Av4v+NF0HJLsR+vP/ANeipPs7+sf/AH8X/Gj7O/rH/wB/F/xoug5JdiP15/8Ar0VJ9nf1j/7+L/jSPE0YBbbg8Ahgf5U7oOWS1aGevP8A9eiiigkO3X8KKKKADt1/Ciq2q6tb6Jp4dl8y/mAa1jOCirkgu478ggD1BzxweZ/4TXxOrN5Ov6jApOfLt7l4kH0VSAB7AVhOsouyVzWNNtXZ1/br+FFc5pPjG6a+jj8R3E99aO/7yVzvnTPcOckj/ZOR16HmukZQuNrK6sodXU5DKRkEfUEVdOopkyg4iduv4VX13H/CJX33c7EIG3Df61O/cVYqDXCf+EPvwN2NqZw/APmp1Hr71cvhfoyFujg4LeOVC0l3DCc42yByT7/KpFFQUV5R3F/Qf+Rj03/r7i/9DFehXG0v97oo2gNuGP5gex5rz3Qf+Rj03/r7i/8AQxXoN0MTYIwccgx7D+I6flXfh/gZzVfiRDWJ4+ChtDB4/wCJYWGB1P2qcc/gOvsK26q65pMviDToEtmX7XYowiV3x5sZbPlr/tBixA77j6AUV4uUdBU2lLU8/pfl2nJO7IwMcY+v5UhBViGGCOCD2qzYafc6ldLb2cTSOx59FHck9gBk59q4TpOy8NOzeD4AxyEvZwvsNkR/qav8Y96SC0j03T4NOhKsLfJkkU5Ekh+8wPpwAPZRS16VNNRSZySd5XCjjHvRRVkhRxj3oooAKOMe9FFABRxj3oooAKOMe9FFABRxj3oooAKU7dowTnuMdKSigAo44x+NFFABRxxj8aKKACjjjH40UUAFHHGPxoooAKDjtRRQAUHHaiigCS3/ANYf9x//AEE1Gcdqkt/9Yf8Acf8A9BNR0upT+Ff12Cg47UUUyQoOM8dKKKAClbaG+Qkj1IxSUUAFBxnjpRRQAUHGeOlFFAAMZ56UUUUAAxnnpRRRQADGeelFFFAAMZ56UUUUAAx3ooooABjvRRRQADHeiiigBRt53EjjjAzzSUUUAHHOfwooooAOOc/hRRRQAcc5/CiiigA45z+FSN/x6x/77fyWo6kb/j1j/wB9v5LSZUdn/XUj45z+FFFFMkOMe9FFFAHL+OUjTxVKI3Lf6LaEdCObaMtz7HtXO12fibR5dTiGpWmHmgiWOeLI3Oqg4dR3wowR7A85OOMrzZxcZNM7ItNaC/LtOSd2RgY4x9fyr0bTwv8AwimhMH3M1pIW5zgi4mAH5AVwuk6Tc6xfLbWu1eheSRtqRjOMsew5rvgscNvb20G7ybaIRR7gMkdST9WJOOcZxWuHi+a5nVatYTjHvUGuj/ikL4kfwrglMZ/ep0Pf+lT1BrmP+ERv8dSqZw//AE1Tqv8AXpXZL4X6M51ujg4EtGQm5mmjbPAjhDjH1LCioKK8o7i/oP8AyMem/wDX3F/6GK9BnCbuD0A2hW3L/iPoea8+0H/kY9N/6+4v/QxXoN1u8759wbHIZcMPr6n37134f4Gc1X4kQ0cc5/CiitzIkkm89t1zb2ty/eS4tY5XP1ZlJP50v2hhAYo1jgjJGY4IViVvqFAB/GoqKVkO7CjjHvRRTEFHGPeiigAo4x70UUAFHGPeiigAo4x70UUAFHGPeiigAo4x70UUAFKdu0YJz3GOlJRQAUccY/GiigAo44x+NFFABRxxj8aKKACjjjH40UUAFBx2oooAKDjtRRQBJb/6w/7j/wDoJqM47VJb/wCsP+4//oJqOl1Kfwr+uwUHHaiimSFBxnjpRRQAUrbQ3yEkepGKSigAoOM8dKKKACg4zx0oooABjPPSiiigAGM89KKKKAAYzz0ooooABjPPSiiigAGO9FFFAAMd6KKKAAY70UUUAKNvO4kccYGeaSiigA45z+FFFFABxzn8KKKKADjnP4UUUUAHHOfwqRv+PWP/AH2/ktR1I3/HrH/vt/JaTKjs/wCupHxzn8KKKKZIcY96KKKAHI5jYOjMrqQVK8EH1zSvIkjbprOxlfu8llE7H6krkn3NMooaT3C7JDKfJ8pFjhi3bvKgiWNCf721QBntnGajooo2AOMe9Qa7n/hEb7JONi4BT/pqnRvT2qeoNc/5FG/xt+6mcPz/AK1Oq/1pS+F+jBbo4OBLRkJuZpo2zwI4Q4x9SwoqCivKO4v6D/yMem/9fcX/AKGK9Bn2E8EcAbQhJXr78ivPtB/5GPTf+vuL/wBDFeg3efO+bdkqD8wHP4j7317134f4Gc1X4kQ0cc5/CiitzIKOMe9FFABRxj3oooAKOMe9FFABRxj3oooAKOMe9FFABRxj3oooAKOMe9FFABRxj3oooAKU7dowTnuMdKSigAo44x+NFFABRxxj8aKKACjjjH40UUAFHHGPxoooAKDjtRRQAUHHaiigCS3/ANYf9x//AEE1Gcdqkt/9Yf8Acf8A9BNR0upT+Ff12Cg47UUUyQoOM8dKKKAClbaG+Qkj1IxSUUAFBxnjpRRQAUHGeOlFFAAMZ56UUUUAAxnnpRRRQADGeelFFFAAMZ56UUUUAAx3ooooABjvRRRQADHeiiigBRt53EjjjAzzSUUUAHHOfwooooAOOc/hRRRQAcc5/CiiigA45z+FSN/x6x/77fyWo6kb/j1j/wB9v5LSZUdn/XUj45z+FFFFMkOMe9FFFABxj3ooooAOMe9FFFABxj3qDXT/AMUhe5z9xcZUEf61O/UfT8anqvrmP+ERv8bc7Uzhjn/Wp1H9RSl8L9GC3RwkCWjITczTRtngRwhxj6lhRUFFeUdxf0H/AJGPTf8Ar7i/9DFegTCMMSu0cDATJU/nyK8/0H/kY9N/6+4v/QxXoN3u875y5OAcvgk/iOv1rvw/wM5qvxIho45z+FFFbmQVbtraOaFSVd5WZgAHVAAqgkkmqlX7L/VL/u3H/oqqXUuHVh9iT+4n/gdFS/Yotpyq57D7bFVKGF55RHEAWIJ5YAYAyeTx0FTfYJv79v8A+BMf/wAVS5mHO/6SJvsSf3E/8DoqlaHfCI5HLKv3UN/EVFVPsE39+3/8CY//AIqj7BN/ft//AAJj/wDiqOYfO+34Im+xJ/cT/wADoqX7FFtOVXPYfbYqg+wTf37f/wACY/8A4qlXTp3YKhgZmOABcRkk/wDfVHMHO+34Il+xLtYiINtUsQt5GxwBk8AelVbhEjkxHuwVVsMc4yoPX8aksP8Aj4f/AK4y/wDotqZdf65f+ucf/oAp7oG7xuQ0cY96KKkzCjjHvRRQAUcY96KKAClO3aME57jHSkooAKOOMfjRRQAUccY/GiigAo44x+NFFABRxxj8aKKACg47UUUAFBx2oooAkt/9Yf8Acf8A9BNRnHapLf8A1h/3H/8AQTUdLqU/hX9dgoOO1FFMkKDjPHSiigApW2hvkJI9SMUlFABQcZ46UUUAFBxnjpRRQADGeelT2SLJeRq4ypzkfhUFWdP/AOP+P8f5GlL4WaUlepFPujVeK1jx5iQrnpuAGaaba1uIzsWPH95McH8Ki1G3ln8vyl3bc55A9KXTreWBH80Y3EYGc1yfZ5r6nr71eRw07lWzSGK4kS625Xgb+laCQ20gzHHCw6ZCg1k35BvpMeoH6Vf0r/j1b/fP8hWlRPl5rnPh5JTdK21x8oslVgwhBGcgYzWdZWhuZPmyIx1IqK5/4+pf98/zrYtsQaejHkBN5/nQ7wjp1JjavV95WURPIs4PlZYx/vkZ/WobuxhNu0sI2kDcMHg1mSSNLIzv1Y5pwnlWIxBzsPaqVOS1uRLEU5JxcNOhGMd6KKK2OEBjvRRRQAo287iRxxgZ5pKKKADjnP4UUUUAHHOfwooooAOOc/hRRRQAcc5/CpG/49Y/99v5LUdSN/x6x/77fyWkyo7P+upHxzn8KKKKZIcY96KKKADjHvRRRQAcY96KKKADjHvUOu7v+EPvvv7dqYyMqD5qdD2Pt+NTVX1zH/CJ3/8AqydiDIzuH71OvY0pfC/Rgt0cJAloyE3M00bZ4EcIcY+pYUVBRXlHcX9B/wCRj03/AK+4v/QxXoExQsSAOQCCnH5jsfYV5/oP/Ix6b/19xf8AoYr0G6JMoLFiSOrMGz/wIda78P8AAzmq/EiGjjnP4UUVuZBWhZEfZ145xcYP/bKs+r9l/qV/3bj/ANFU1sy47P8ArqiGw/4+W/64y/8AotqrcY96s2H/AB8v/wBcZf8A0W1VJJEijLyuqIOrMcAUiB1HGPeqUOr2E9wYIrlDJnAHIBOccE8H8KtySJEheV1RR1ZjgCkmnsOzQ6rOm4/tS19fPT/0KsmLWNPmnMMd0hcHHOQDzjgng/hWtpv/ACFLT/rsn/oQounsFmtwsP8Aj5f/AK4y/wDotqZdY81fXy0/9AFPsP8Aj5f/AK4y/wDotqZdf65f+ucf/oAquhS+B+v+ZDRxj3oopEBRxj3oooAKOMe9FFABSnbtGAc9znrSUUAFHHGPxoooAKOOMfjRRQAUccY/GiigAo44x+NFFABQcdqKKACg47UUUASW/wDrD/uP/wCgmozjtUlv/rD/ALj/APoJqOl1Kfwr+uwUHHaiimSFBxnjpRRQAUrbS3yAgehOaSigAoOM8dKKKACg4zx0oooABjPPSrOn/wDH/H+P8jVarFgwW+jLHA5/lUy+FmlL+JH1RrTXKwSxK44kJG7PSi8eWO2Z4cbh147VS1ZgWiAOSAT/ACqfT7rzovLkPzr69xXNy2ipHre15qkqTfp9xkZy2WycnJ561raV/wAerf75/kKzbpEjunWP7oPFaGlMPs7rkZ35x+AraprC5xYVctez8zPuMfbJc9PMP861of3+mqq949v44xWROQ1zIRyC5I/OrWn3ghJjlPyHofQ0pxbirBh6kY1WpbMpEbWIYHI4IpK3Wgtrpt5CufVW/wAKiupLeK0eJSmcYCjnmhVb6WHLCcqcnJWMcY70UUVscIDHeiiigBRt53AnjjBxzSUUUAHHOfwooooAOOc/hRRRQAcc5/CiiigA45z+FSN/x6x/77fyWo6kb/j1j/32/ktJlR2f9dSPjnP4UUUUyQ4x70UUUAHGPeiiigA4x70UUUAHGPeodd3f8IffZD7dqYJ5X/Wp+R9u9TVX10D/AIRK++7u2J/CQ2PNT8CP1pS+F+jBbo4SB7RUIuYZpGzwY5ggx9CpoqCivKO4v6D/AMjHpv8A19xf+hivQbjYJGG35sDoCuD3yD3+nFefaD/yMem/9fcX/oYr0C5BEgHzY2jG5g2B7EdRXfh/gZzVfiRFRxzn8KKgkvbWFyktzCjDqGkANb3sZE9aFkR9nXjnFxg/9sqxv7Ssf+f23/7+r/jWlYXlq8KFLmFgUuuRID92HJ/IcmhSVmaQTs/T9UFh/wAfL/8AXGX/ANFtVjwtbW+q67PbXCM+yWCFEKApl1lkdmz1GyIgDBBbAIIzWfpt/aSXTiO6gYi3mbCyA8CJiT+ABNM0zxLY6BqEmoLcpIH8sOkLKzAq3DAZBJAZgRkZUsOM5Hn5gqs8NKNB2ka4fljUTqLQ3dJ8Wad4r12bRLTT71ZpIpBE93ds8UjICw3xEFYwSoO5ACvQYrK0fRINY8Ty29xLLKiz2pso5Y1VUWaJ5yZF5BZIkYKOVL4yCpNWLPxP4Ts76S6062sIb+dTEJ7P7R5jlzglUdVSEnJOQ77enzVmy6rdaTrM+tRbdk0kckqKqgQGPHl+WOPlAyhXOWRmXPOa+Zo4fEulUWHi4Rts97339Lf129SdSkpx9o03fp2NfTde03xlqkvhy2g1KGSdJAJry5M8cpXLYeAjbGDjOY9hU8KVqtB8PLfRfE1ve29/JIltcRXMUbRAfKCH2E569s/pUZ8V+GNMafUNHi0+21BlKm4s0nMsgY/NsjkURwsQSc7pNvTDda5W++IPiG+vZLg3ccO88RxwJtUdABkE8CuzKoKFWTlBqFtnvzd99rfd27Y4uTlBcsk5X/A6PTPAH9nXbz/2l5m+3ng2/Z8Y8yJo8/e7bs474xSX3w9jvWjcag0cixRxE+TkHYirnG7vtrH0Lxbrd5qEsVze70WzupQPKQfMlvI6ngdmUH8Kh1TxhrtvdokN9tU28DkeSh+ZolYn7vqTX0inhrP3X/V/M4lHE+ybUla6/W3T1NP/AIVj/wBRf/yW/wDs6cfhlH5YA1Rt+eW8jjH03f1rnv8AhN/EP/QQ/wDIMf8A8TR/wm/iH/oIf+QY/wD4mp58L/K/6+Zly1+/9fcb/wDwrH/qL/8Akt/9nTj8Mo/LAGqNvzy3kcY+m7+tc9/wm/iH/oIf+QY//iaP+E38Q/8AQQ/8gx//ABNHPhf5X/XzDlr9/wCvuN//AIVj/wBRf/yW/wDs6cfhlH5YA1Rt+eW8jjH03f1rnv8AhN/EP/QQ/wDIMf8A8TR/wm/iH/oIf+QY/wD4mjnwv8r/AK+Yctfv/X3HQD4dXNsrPZayRKB8o8kpk+hIY1FYai5k+wajG0F/Fw4Y8P7ioNC8fXkd6E1uQT278bwiqYz68AZFdVr2g2/iCzS5tZFS6UboZ1PX2PtW0YwlHmo/cZtyi7VPvM2jjjH41mafqEvntYaknk3kXBB/j9xWnTTTVwasFHHGPxoopiCjjjH40UUAFHHGPxoooAKDjtRRQAUHHaiigCS3/wBYf9x//QTUZx2qS3/1h/3H/wDQTUdLqU/hX9dgoOO1FFMkKDjPHSiigApW2lvkBA9Cc0lFABQcZ46UUUAFBxnjpRRQADGeelFFFAAMZ56UUUUAAxnnpRRRQADGeelFFFAAMd6KKKAAY70UUUAAx3ooooAUbedwJ44wcc0lFFABxzn8KKKKADjnP4UUUUAHHOfwooooAOOc/hUjf8esf++38lqOpG/49Y/99v5LSZUdn/XUj45z+FFFFMkOMe9FFFABxj3ooooAOMe9FFFABxj3qDXAf+EPv/vY2p3BH+tT8R/X8KnqvruP+ESvvu52IQNuG/1qd+4pS+F+jBbo4SB7RUIuYZpGzwY5ggx9CpoqCivKO4v6D/yMem/9fcX/AKGK9BuCvmEYO7GPu7MH3X/CvPtB/wCRj03/AK+4v/QxXoNycyDByNoxh9wx7dwPY8134f4Gc1X4kYGr6hcG5TTNNRnupuPl6jP9e9b2l+AtMhsl/tSM3VyeXbzGUA+gwRWDpX/JT4fx/wDRJrv9SuXs9Ju7qIKXhgeRQw4JCkjP5VvRjGfNOetjKpJxtGJlf8IR4e/6B/8A5Gk/+Kq1b+GtJtYwkFpsVRKAPMc8Spsfqe68e3bBrh/+Fkav/wA+1l/37f8A+KrZ0vxjqF7bRySw2wLJesdqt/yxtxKv8Xduvt6daPbYez0/A0hRxDT5X66+a/Wxt2vhPRLOVpLay2O0bxE+a5+V0KMOT3ViPxq7ofw/8OzXiyHTwWSRQgaR2XPLZIJIOAp4IIJxnjNcbo/jrU9QvpIZoLRVW1uJgURgd0cLyL/F0yoz7Uuk/FXU7O6/0i2tfJYglo423IR0YAtzjJyuRkEjIzkcGPnCeGnGgrTtpoXh4TjVi6j09Tq9F8VaP4u8UjRhp+oIr7+J7oyxTBQSVeFgUQHGcqAQeAQKl02y0/X9b1DTYbWCCziSxNpHcwCSRWlTzmYsdx3hFcAElSeGDDg4TeKvC+nfab/SYbC31B43Tz7NbjzpN3B2I6hIGOc7t0m3ph6iX4uyNfzTGHUIRcOhwb1ZI4NrhvlQRqSMDaV3DcpIyCdw+Tp0cTyVFh4uCa2fe+/pY9qU6d4+0abv07Glpuv6b4x1WXw5bQ6lDJPHIBPeXJnjlKgsQ8BG2MHGcx7Cp4UrWVf/AAttobuZYNTkCNteINAMqrDOD83J5xnjpVg+K/DGmNPqGjxafbagylTcWaTmWQMfm2RyKI4WIJOd0m3phutcrffEHxDfXslwbuOHeeI44E2qOgAyCeBXo5VBU60nKDULbPfm7+lvu7duXFtyglGScr/gdFpngD+zrt5/7S8zdbzwY8jGPMiaPP3u2/Pvikvvh7HeOjjUWjdYo4ifJyDsRUzjPfbmsjQvFut3moSxXN7vRbO6lA8pB8yW8jqeB2ZQfwqHVPGGu292iQ321TbwOR5KH5miVifu+pNfSKeGs/df9X8ziUMT7JvmVrr9bdPU0/8AhWP/AFF//Jb/AOzpx+GUflgDVG355byOMfTd/Wue/wCE38Q/9BD/AMgx/wDxNH/Cb+If+gh/5Bj/APiannwv8r/r5mXLX7/19xv/APCsf+ov/wCS3/2dOPwyj8sAao2/PLeRxj6bv61z3/Cb+If+gh/5Bj/+Jo/4TfxD/wBBD/yDH/8AE0c+F/lf9fMOWv3/AK+43/8AhWP/AFF//Jb/AOzpx+GUflgDVG355byOMfTd/Wue/wCE38Q/9BD/AMgx/wDxNH/Cb+If+gh/5Bj/APiaOfC/yv8Ar5hy1+/9fcW9c8C3ek2f2m2n+2Rr/rAI9rKPXGTkU7wr4uXSFis7pZDbsx3uz5CZ6FRjgevJ/Cum8K+LY9aiFpf7UvAPTAlHsPX2rE8YeDzbF9R0uPMR5liUfc9x7VUqailVoEqV/wB3UOm17QbfxBZpc2sipdKN0M6nr7H2rm7DUJPONhqKGG9i4YN/H7is/wAJ+LH0eUWl6xeyY/Uxn1HtXaa9oNv4gs0ubWRUulG6GdT19j7Vqmqq54b9URrTfLLYzaOOMfjWZp+oS+e1hqSeTeRcEH+P3FadCaauNqwUccY/GiimIKOOMfjRRQAUHHaiigAoOO1FFAElv/rD/uP/AOgmozjtUlv/AKw/7j/+gmo6XUp/Cv67BQcdqKKZIUHGeOlFFABSttLfICB6E5pKKACg4zx0oooAKDjPHSiigAGM89KKKKAAYzz0ooooABjPPSiiigAGM89KKKKAAY70UUUAAx3ooooABjvRRRQAo287gTxxg45pKKKADjnP4UUUUAHHOfwooooAOOc/hRRRQAcc5/CpG/49Y/8Afb+S1HUjf8esf++38lpMqOz/AK6kfHOfwooopkhxj3ooooAOMe9FFFABxj3ooooAOMe9Qa4W/wCEPv8Ahtu1Oj5XPmp1Xsfep6g10f8AFIXxI/hXBKYz+9Toe/8ASlL4X6MFujg4HtFQi5hmkbPBjmCDH0KmioKK8o7i/oP/ACMem/8AX3F/6GK9CuQFkIZSGwP4NmPqv+Fee6D/AMjHpv8A19xf+hivQLjAkG3G3HG19w/DuPoea78P8DOar8SMHSf+SoRfj/6JNd1r3/Iual/16S/+gGuF0n/kqEX4/wDok13Wvf8AIual/wBekv8A6Aa6KH8OfqzCr8cfRHildT4eKf2XGCPn2aoQfQfYuf6Vy1dP4f8A+PGH/rlqv/pEK8tfDL0/VHsUfhn6f+3Iz/DP/IWm/wCwfe/+kstL4U0aPxD4osNKldo1uJgHcMAQgBLYyD82Bx79jSeGf+QtN/2D73/0llqnpOp3Gi6va6lZFRPbSCRA2cHHY47HofY1z1VNwkqe9tPXoZQsmubY7TTrDwj4m1CbQ9L0i+sLvypWt72S73bnRSR5iEYVTjJxz2q38G0P9sNLGVVheRIzbgpZDb3RK+pG5UOB/dB/hyMdvHlnAlxPo/hmy0/U7mN4nvEldtocEMUTOFPJwfwpvw/8R2fh+8klvJEjdJ47iLzQ4Rysc0ZUsisV4m3Z2n7uOM5HjV6NeWFqw5XqlZN3d+vV6bf8A7Kc6aqwba07aeh6n4a+KVr4l8Vf2LFpc8AcP5czuDnaCfmXHy8D1PPFchrfgCHUNce9ub0q06QvIsMCoGYxrvPXALNk9OM96s6f4w8I6Vqz6np+n6LBeOGBkW7vOM9cA2uB+ArAt/Gt9rPj+xgtZnTTJ7y3t1ikRNzR5VDkgcEgZOOmeKeT4elQxM5ujKMeVb9769ScbUnUpRSmm7/h9xt23gfRbDzZbJ78XD280CtLOjKvmRtGSVCAnAYnGRWbffD2O8dHGotG6xRxE+TkHYipnGe+3NZGheLdbvNQliub3ei2d1KB5SD5kt5HU8Dsyg/hUOqeMNdt7tEhvtqm3gcjyUPzNErE/d9Sa+sU8NZ+6/6v5nnKOJ9k3zK11+tunqaf/Csf+ov/AOS3/wBnTj8Mo/LAGqNvzy3kcY+m7+tc9/wm/iH/AKCH/kGP/wCJo/4TfxD/ANBD/wAgx/8AxNLnwv8AK/6+Zly1+/8AX3G//wAKx/6i/wD5Lf8A2dOPwyj8sAao2/PLeRxj6bv61z3/AAm/iH/oIf8AkGP/AOJo/wCE38Q/9BD/AMgx/wDxNHPhf5X/AF8w5a/f+vuN/wD4Vj/1F/8AyW/+zpx+GUflgDVG355Pkcflu/rXPf8ACb+If+gh/wCQY/8A4mp7Px5rUN1G91cLcQg/PGY0XcPqAMGmp4X+V/18xctfv/X3Gfq+jXvhzUgshPB3QzpwG9x6H2ru/CfiyPWIRZ35VbxRjnpKP8fatSWHT/FegqSC8E4yjYwyEHH5givHY5HhlWSJijqcqwOCDTlfDTUo6xYK1aNpbo7Txh4PNsX1HS48xHmWJR9z3HtVPwn4sOkyx2d2WNm5wSxz5Zz1Ht7V1XhHxMNctmtbv/j8iXLccOvAz+tcZ420u30rxBstFKpNEJivZSWYED24oqJQtXpCg3L93M7XxZpFtqejvqMLhJ7aIzRzJzvUDOPpWJo18b/S45X/ANauUkPqQf8ADFdB/wA05/7hX/tKuS8J/wDIKk/67n/0Fa1qfxE11REPga7G5Rxxj8aKKYBRxxj8aKKACg47UUUAFBx2oooAkt/9Yf8Acf8A9BNRnHapLf8A1h/3H/8AQTUdLqU/hX9dgoOO1FFMkKDjPHSiigApW2lvkBA9Cc0lFABQcZ46UUUAFBxnjpRRQADGeelFFFAAMZ56UUUUAAxnnpRRRQADGeelFFFAAMd6KKKAAY70UUUAAx3ooooAUbedwJ44wcc0lFFABxzn8KKKKADjnP4UUUUAHHOfwooooAOOc/hUjf8AHrH/AL7fyWo6kb/j1j/32/ktJlR2f9dSPjnP4UUUUyQ4x70UUUAHGPeiiigA4x70UUUAHGPeoNcx/wAIjf46lUzh/wDpqnVf69KnqDXc/wDCI32ScbFwCn/TVOjentSl8L9GC3RwcD2ioRcwzSNngxzBBj6FTRUFFeUdxf0H/kY9N/6+4v8A0MV6JO6gyLLGxkYDazfKyeueOSfWvO9B/wCRj03/AK+4v/QxXoFxjeNu3bjjaxI/DPIruw/wM5qvxIwdJ/5KhF+P/ok13Wvf8i5qX/XpL/6Aa4XSf+SoRfj/AOiTXda9/wAi5qX/AF6S/wDoBrpofw5+rMKvxx9EeKV1fh10GjohTLsupkNnoBYnI/HI/KuUrp/D/wDx4w/9ctV/9IhXlr4Zen6o9ij8M/T/ANuRn+Gf+QtN/wBg+9/9JZaygyCFlKZcsCHz0GDkY98j8vetjwnDJca48MEbSyyWN4iIikszG1lAAA6k0z/hEvEf/Qv6p/4BSf4Vnpcw6GRTgyCFlKZcsCHz0GDkY98j8vetX/hEvEf/AEL+qf8AgFJ/hR/wiXiP/oX9U/8AAKT/AAp3QrMyK2vCLIPF2jKUy51K2IfPQeYMjHvkfl70z/hEvEf/AEL+qf8AgFJ/hWp4Y8Ma/b+LtHmuND1KKKO+gZ3e0kVVAkBJJI4FJtWGk7mX4Z/5C03/AGD73/0llqLW2Q3MChMOLa3JfPUeRHgY9sH8/apfDP8AyFpv+wfe/wDpLLVfWv8Aj/j/AOvS2/8ARCVovhfy/U3j/Al6r8mUKcWQwqoTDhiS+eowMDHtg/n7U2ipOcKcWQwqoTDhiS+eowMDHtg/n7U2igApxZDCqhMOGJL56jAwMe2D+ftTaKAPW/BH/Im2P/bT/wBGNXlk01s9jbRQ2nlTx7/On8wt52T8vy9FwOOOtep+CP8AkTbH/tp/6MavJK7cT/Dp+n+RzUfjmdb8N/8AkY5/+vRv/Q0o+JH/ACMcH/Xov/ob0fDf/kY5/wDr0b/0NKPiR/yMcH/Xov8A6G9H/ML8w/5fnXf805/7hX/tGuW8LujaMAibSsjBzn7x65/IgfhXU/8ANOf+4V/7RrkvCf8AyCpf+u5/9BWt6nxx9DKHwy9TcpSQQMDBA5560lFMApSQQMDBA5560lFABSsQT8owMDvSUUAFKxBPyjAwO9JRQBJb/wCsP+4//oJpjEE/KMDA70qO0bbl69ORmnfaH9I/+/a/4Utbl3jy2ZHSsQT8owMDvT/tD+kf/ftf8KPtD+kf/ftf8KNRe73/AK+8jpWILEqNoJ4Gc4p/2h/SP/v2v+FH2h/SP/v2v+FGoe73/r7yOnzPG8zNDH5SHom7OPxpftD+kf8A37X/AAo+0P6R/wDftf8ACjUPd7/195HSsQWJUbQTwM5xT/tD+kf/AH7X/Cj7Q/pH/wB+1/wo1D3e/wDX3kdKxBYlRtBPAznFP+0P6R/9+1/wo+0P6R/9+1/wo1D3e/8AX3jFIDAsNwB5GcZpKk+0P6R/9+1/wo+0P6R/9+1/wo1D3e/9feMUgMCw3AHkZxmkqT7Q/pH/AN+1/wAKPtD+kf8A37X/AAo1D3e/9feMUgMCw3AHkZxmkqT7Q/pH/wB+1/wo+0P6R/8Aftf8KNQ93v8A194xSAwLDcAeRnGaSpPtD+kf/ftf8KPtD+kf/ftf8KNQ93v/AF94xSAfmGRg96SpPtD+kf8A37X/AAo+0P6R/wDftf8ACjUPd7/194xSAfmGRg96SpPtD+kf/ftf8KPtD+kf/ftf8KNQ93v/AF94xSAfmGRg96SpPtD+kf8A37X/AAo+0P6R/wDftf8ACjUPd7/194kTxrv8yPzMqQvzY2n19/pTKk+0P6R/9+1/wo+0P6R/9+1/wo1D3e/9feMBAByMkjjnpSVJ9of0j/79r/hR9of0j/79r/hRqHu9/wCvvGAgA5GSRxz0pKk+0P6R/wDftf8ACj7Q/pH/AN+1/wAKNQ93v/X3jAQAcjJI456UlSfaH9I/+/a/4UfaH9I/+/a/4Uah7vf+vvGAgA5GSRxz0p7f8esf++38lo+0P6R/9+1/wpHlaQANtwOQAoH8qNR+6k7DQQAcjJI456UlFFMgXI2kY5yMHPSkoooAXI2kY5yMHPSkoooAXI2kY5yMHPSkoooAXI2kY5yMHPSq+uf8ijf42/dTOH5/1qdV/rU9Qa6f+KQvc5+4uMqCP9anfqPp+NKXwv0YLdHI6XqGkWlqyalon2+UuSJftbxYXA+XAHrk596KyqK8myO+5f0H/kY9N/6+4v8A0MV6HdEiQ7lJLKCGYc/mOG+vevPNB/5GPTf+vuL/ANDFd/OFEg2bMEfwEkfryPxr0MP8DOWr8SMLSf8AkqEX4/8Aok13Wvf8i5qX/XpL/wCgGuF0n/kqEX4/+iTXda9/yLmpf9ekv/oBroofw5+rMKvxx9EeKV1Ph2RBp9vC7wxGY6iiyTSrGoZrMKoLMQBksByfSuWrZ0XdfQNYG2sZUg33O+7eRAmQoblGH90dfevKUoxi+Z20PawsXOTpreSsvvT/AEG/8Izf/wDPfS//AAb2v/xyl/4Rq+2kedpWSQd39r2uR7f63/OKuCzie+t7SCz0SeSfdgxXE7Bdozzh+Paq2F+w/af7EsMfaPs3l77jfvxnp5lSpQf21+Js8DKO8l+Pk+3miP8A4Rm//wCe+l/+De1/+OUv/CNX20jztKySDu/te1yPb/W/5xV6bS54LV5pND0wFELvH5829VAyTjzadBpE89ssw0TTFLpvSNp5tzjGeB5n86n2lK1+dfiaf2ZWcuXr6S/+RM7/AIRm/wD+e+l/+De1/wDjlL/wjV8EI87Ss5B3f2va5Ht/rf8AOKt2Fl/aNo08Gk6WuJDGEkmnVmYDOB+8x+tJa2YuGull0rS7Q2pUS/aJp1AznHPmH0/UU3Kmr++tPUhZfUkotPfbf1/lK5ns9EgkhtP9J1OSN4pbkSBoYVdSrogH3ztJBfOOTtB4eq2tkfbIhtGRa2x3dz+4j4/z61px2cU161tFZ6I+yEzNKtxOUVQcHJ38Hv8ASqGv2l3DeJNdxwIkqKsRt3LIVRVUYySegHWmqlO3KpXb/wCCOphalLDye6utr6Wuu3mjKpcjaBtGQSd3c+3+fWkopnmhS5G0DaMgk7u59v8APrSUUAFLkbQNoyCTu7n2/wA+tJRQB634I/5E2x/7af8Aoxq8mZ1aNFEaqVzlwTlvrk449gK9Z8Ef8ibY/wDbT/0Y1eSV24n+HT9P8jmo/HM634b/APIxz/8AXo3/AKGlHxI/5GOD/r0X/wBDej4b/wDIxz/9ejf+hpR8SP8AkY4P+vRf/Q3o/wCYX5h/y/Ou/wCac/8AcK/9o1ynhZg2kMAoG2Ugkd+Ac/rXV/8ANOf+4V/7RrkvCf8AyCpf+u5/9BWt6nxx9DKHwy9Tco9OP/r0UUwCj04/+vRRQAUHntiiigAq8zx20cK+RG5dAzEj1qjVmO7URossIkKfdJOMVMk2bUpJXu7FlIoYpJ4x5O9WG3zemCOlM2bZZXnhjBiThVX5SSeDUC3EbMzTwea7NnO8jHtTje5uGkMY2Mmwpnt9az5ZG/tKdv6899Og63kW5vIleGJV5yFXGeKjKr/Zu/aA3m4z7bacLuNJo3itwmzOQG65HrTYriNLcxSw+YN2772O2Kqz6Ijmg1Zvvr93kWLW0EthIxXLt90+mP8A69MgZEs97Qo26bacrnAx2pp1BhJH5S+XGn8APWkjvERWDQbh5nmL82MGptLW5fNSVrPZWJBGkDXTbFfyyu0MMjBNOijjkubaTy1AlDZTHGRmq8d1hpfNQSLKcsM475pwvMXMcgjASMYVAf603GX9ego1Kenb/g/5Et6gS1XcsCyF8jyh1XFUDyemPapXnL26RsMlDw2e3pUVXFNLUwqyUpXQUHk9Me1FFUZAOD0z7UUUUAA4PTPtRRRQADg9M+1FFFAAOD0z7UUUUAA47ZooooABx2zRRRQADjtmiiigBQQM5UHIxz2pKKKAD14/+tRRRQAevH/1qKKKAD14/wDrUUUUAHrx/wDWooooAPXj/wCtRRRQAdun40UUUAHbp+NFFFAB26fjRRRQAdun41X1zH/CI3+NudqZwxz/AK1Oo/qKsVDru7/hD777+3amMjKg+anQ9j7fjSl8L9GC3RwUFxHEhWS0hmOc7pC4I9vlYCioKK8o7i/oP/Ix6b/19xf+hivQ7okSHeGYkAhn5P5j731rzzQf+Rj03/r7i/8AQxXfzkGTK7cEZ+UYz9R2PtXfh/gZzVfiRhaT/wAlQi/H/wBEmu617/kXNS/69Jf/AEA1wuk/8lQi/H/0Sa7rXv8AkXNS/wCvSX/0A10UP4c/VmFX44+iPFK3vCrL5uoI1v5+6zc7ASC44+UY9c1g1JBcz2she2mkhcjBaNipI9OK8WpHng4nsYWsqFaNR9Df0eB4vGFq/wBglsYpN+yKTcekZzgnrzz+NXrq9tpNHsNRQBBJqMcs6jorhSG/ln8a5ZtRvXlSR7y4aSPOxzKxK564OeKi86TyPJ8x/K3b/L3HbuxjOPXFYSoOUlJv+tT0IZjGlTlTgtG2+m75e3o9ujNnV9EvXv7y8giEtqxacTKw27Tk+taT2El94gt9WgVPsb+XO027/VhAMqfy/wA81yhnmMIhMrmIHITcdo/CnR3VxFE0cU8iRuCGRXIDA9ciqdKbVr+XyM44vDqbfI7NqW/VX8ttX5nRpbSazocwtF4l1VpCTwEUp1PpjNW/PhvLjxBJHbNfxMYAIo2IMmOOCOeCM/hXIx3dzFA0MVxKkTcsiuQp+opbe8ubXd9luJYd2N3luVz9cVDw7d9fT70/0NoZlBct4+u3aSVk9PtPfc3NEj+zalqRurKSGI2Ur/Z3yp2ZHGTz04zUXiZ4gLGKyUCxWItAQxbOTz19MdKyXvruR2eS6mZmTy2ZpCSV/u/T2qNppHiSN5HaOPOxCxIXPXA7VapP2im2c88ZD6u6EVv10vve3p5d0hlLkbQNoyCTu7n2/wA+tJRXSeaFLkbQNoyCTu7n2/z60lFABS5G0DaMgk7u59v8+tJRQB634I/5E2x/7af+jGryZnVo0URqpXOXBOW+uTjj2Ar1nwR/yJtj/wBtP/RjV5JXbif4dP0/yOaj8czrfhv/AMjHP/16N/6GlHxI/wCRjg/69F/9Dej4b/8AIxz/APXo3/oaUfEj/kY4P+vRf/Q3o/5hfmH/AC/Ou/5pz/3Cv/aNcp4WYNpDAKBtlIJHfgHP611f/NOf+4V/7RrkvCf/ACCpf+u5/wDQVrep8cfQyh8MvU3KPTj/AOvRRTAKPTj/AOvRRQAUHntiiigAq/HpFzMqNGjEOoK7YpGzkeymqFbNppaa94u0vRr9I5LGTSRcyqWIZgJGXAP1x05HXPFYYnEU8NRlWqbI6KFJ1HypXehC2gX6KWeCZQOpNvKP/Zaq3NlJbLukwOQCpVlPOefmA9DV3w7qng/xL4iTQrLwqLBwJDb3in94rKp5bGD0GeSefzpqWtxPp8MEvlmdRIh8kEhmjkuFOPqU/WuTBZgsTN0pwcJJXs+3fRvsb1sOoLmjZrXa+69TLq+uj3MjYjUvzt+SKRgSOOoX1qC4sLq2XdNBIq4BLFDgZGcZ9ecfWtm30u21DUbuTWIkuNMsdOe6lgXPmSHzpsAc9P3Z/HFdWKxVPC0HWnsjGhSdWXKkUW0C/RSzwTKB1Jt5R/7LVS7s5bTaZRt3k4G1lxj2YD1q74fuPDHjeS7t9M8NR6TfWFu1xbTxyBVY9PnIA7kdQeMkYxVTTzPq3hvR5XUCWaK4cBATna78AdeduK58Fj1inKE4cso7r122bNa2HUI80dV317+ZVpWILZChR6Cp7iwurZd00EirgEsUOBkZxn15x9ar16BwhQeT0x7UUUAFV31CySRle6t0YEgqZQMH061Yrz7Uv+Qrd/8AXd//AEI1lVqOCLhHmO4Gp2IP/H5bn280f40f2lY/8/tv/wB/V/xrz6isPrD7Gvsl3PQRqdiD/wAflufbzR/jR/aVj/z+2/8A39X/ABrz6ij6w+weyXc9BGp2IP8Ax+W59vNH+NH9pWP/AD+2/wD39X/GvPqKPrD7B7Jdz0EanYg/8flufbzR/jR/aVj/AM/tv/39X/GvPqKPrD7B7Jdz0EanYj/l8tz/ANtR/jR/aVj/AM/tv/39X/GvPqKPrD7B7Jdz0EanYj/l8tz/ANtR/jR/aVj/AM/tv/39X/GvPqKPrD7B7Jdz0EanYj/l8tz/ANtR/jR/aVj/AM/tv/39X/GvPqKPrD7B7Jdz0IanYDObu2ORjmUcfrSf2lY/8/tv/wB/V/xrz6ij6w+weyXc9B/tOx5/0y3/AO/o4/Wj+0rH/n9t/wDv6v8AjXn1FH1h9g9ku56D/adjz/plv/39HH60f2lY/wDP7b/9/V/xrz6ij6w+weyXc9B/tOx5/wBMt/8Av6OP1o/tKx/5/bf/AL+r/jXn1FH1h9g9ku56NDcwXG7yJo5cddjg4/KpK5rwh/y+f8A/9mrpa6YS5o3MZKzsHrx/9aiiirJDt0/GiiigA7dPxooooAO3T8aKKKADt0/Gq+uY/wCETv8A/Vk7EGRncP3qdexqxUOu7v8AhD77IfbtTBPK/wCtT8j7d6UvhfowW6OCguI4kKyWkMxzndIXBHt8rAUVBRXlHcX9B/5GPTf+vuL/ANDFehXTHzPmBYkDDM24/wDfQ4Nee6D/AMjHpv8A19xf+hivQLkATEAAccgKV59x2P04rvw/wM5qvxIwdJ/5KhF+P/ok13Wvf8i5qX/XpL/6Aa4XSf8AkqEX4/8Aok13Wvf8i5qX/XpL/wCgGuih/Dn6swq/HH0R4pS5G0jaMkg7u49v8+lJRXknoBS5G0jaMkg7u49v8+lJRQAUuRtI2jJIO7uPb/PpSUUAFLkbSNoySDu7j2/z6UlFABS5G0DaMgk7u59v8+tJRQAUuRtA2jIJO7ufb/PrSUUAFLkbQNoyCTu7n2/z60lFABS5G0DaMgk7u59v8+tJRQB634I/5E2x/wC2n/oxq8mZ1aNFEaqVzlwTlvrk449gK9Z8Ef8AIm2P/bT/ANGNXklduJ/h0/T/ACOaj8czrfhv/wAjHP8A9ejf+hpR8SP+Rjg/69F/9Dej4b/8jHP/ANejf+hpR8SP+Rjg/wCvRf8A0N6P+YX5h/y/Ou/5pz/3Cv8A2jXKeFmDaQwCgbZSCR34Bz+tdX/zTn/uFf8AtGuS8Kf8gqT/AK7H/wBBWt6nxx9DKHwy9Tco9OP/AK9FFMAo9OP/AK9FFABQee2KKKACtiG4GleJ9G1cec7R2aQPGgLZjLtuwo5J5Jx+QJrHrYsdNbV/iD4btpHVYLWwW/YZ5crIwUf99AH6ZrjzCdGGEnKsrxtr9zOvCxnKSUN7obofjDwgk0+qaH4bmtNYK7QzKWiDuSSAQ2BwGPRSQpA9KglVjo9qgLSNtXknJYl5ST9SST9TVnQtY8J+KdRuPCln4YGnR3UTxrdoF8z5PmG44yPug8k8gA03T7e9srOyt7hlkuoGaFm3bQdks6E5+ik5ryMmdGFWcORxna/vO75dba3fnodmKU2lK6cddlbWwkhshaG3lgmhlRFLN9nG5W4ySSw4OfbqKsx6oNG11bxraW8gntWtZ7RPm3p5znKqercnjvkjvVSUXKWs6SiOZDEMXAH3l3JwGx82Dgc9MVestNjuPEo1PUGZ7LRbJ78QR/ekk8+bacd8eWT9cetenmU6EMJN11ePb8vxOXCqcp2puzv/AJmfaeL/AAfaNf2fg/SZrS4vIPL+0MSN3KghcsSPlLNnjlehOKhe2W30PS7e2VtiiXYmSduZCcDv3q/o2o6H8QdO1rRNH0CPRLhrTekkGFSQq6ld4UAZDAYzk4LYxk1T0aW4m8P6BNx5rwzAkttwA7KSSeh4JzXn5M6UXUp8rU1a/M7vy11OnFqbjzXTjbSyt1RNOLEI8LRXETRgFT9nAZen3ju5zn9eOOKzGILZChR6CtKUXKWs6SiOZDEMXAH3l3JwGx82Dgc9MVmV9AjyQoPJ6Y9qKKACvPtS41W7wMDzn4/4Ea9Brz7Uv+Qrd/8AXZ//AEI1zYjZG1LdldSFYEqGAOSp6H24pKKK4zoFUhWBKhgDkqeh9uKSiigDufhzAlzbaxBMN0cqRow9QQ4NXtK0KTRfBmt/aVxPLHMCfVVVgv8AU/Q1leB55LXRvEFxCcSRW6uhxnBCyEVr2viSTxFoevMYPs8cFkQse/dyVkyc4HoOPavSpcnJG+9n+pxVObmdttCp8OLJIobrUpiq72FvGWOPQkfiSv5Vx2s2B0zWrqzIIEUhC57qeV/Qiu61DVrTwhp2maZJp6XjqglbLY2OD94ZB5zu+mKj8QeF7nxPd2+qaa9vFDNbISJmYMScnPAPYgfhUTpp01CO6/UqM7TcnsyO+YI3ggsoYbUGD7iLmtWy0cy+PdS1WUfJCEji92Ma5P4A4/Gs3XbZ7O98HWspUvBIkbFTwSDEDj8q0r3WDL4903Soj8kO+SX3YxNgfgDn8a2VlL3u6/Izd3HTs/zMfTf+Z3/7a/8AtWqHg/8A5FzxL/16f+ySVf0We1bXPE+nXVwkDX0skaFzjPzOD+PzdKfFpcfhLw1q6397A819CY4lQ8t8rAYB92rKKvaXRX/Utvdd7focCpAPKhuCMH+dd98PtbuZ3/sh0iFvbwPIrAHeSXHU5x/Ee3pXAV1vw3/5GOf/AK9G/wDQ0rnw0mqqsbVknB3Nfwr4ju/EHiP/AEyOFPItJNvkqRnc8ec5J9BWRp3/ACSvVf8Ar7X+cVHw3/5GOf8A69G/9DSjTv8Akleq/wDX2v8AOKt4ycoXfaX5IzaSlZeRoT3dp4I0m1ht7WKTVpo1eRpRnbnrz6ZGMAis/wCI/wDyMcH/AF6L/wChvWlc21n450u3ntrqOHVIYwjxSNgH1464z0I9azfiP/yMcH/Xov8A6G9Orf2bt8OlhQ+NX31ucmCAGyoORgE9vekoorzzrFBADZUHIwCe3vSUUUAdL4P/AOXzj+5+H3q6Wua8If8AL5/wD/2aulr0KP8ADRyVPiYevH/1qKKK1IDt0/GiiigA7dPxooooAO3T8aKKKADt0/Gq+ugf8Ilffd3bE/hIbHmp+BH61YqDXAf+EPv/AL2NqdwR/rU/Ef1/ClL4X6MFujg4LiOJCslpDMc53SFwR7fKwFFQUV5R3F/Qf+Rj03/r7i/9DFehXBKtt5wVGNxD4A9GHb6V57oP/Ix6b/19xf8AoYr0C54mxxnHOE2/mOx+ld+H+BnNV+JGDpP/ACVCL8f/AESa7rXv+Rc1L/r0l/8AQDXC6T/yVCL8f/RJrute/wCRc1L/AK9Jf/QDXRQ/hz9WYVfjj6I8Updx2leMEg9Ofz/GkoryT0Apdx2leMEg9Ofz/GkooAKXcdpXjBIPTn8/xpKKACl3HaV4wSD05/P8aSigApdx2heMAk9Ofz/CkooAKXcdoXjAJPTn8/wpKKACl3HaF4wCT05/P8KSigApdx2heMAk9Ofz/Ctk2NkLmG0h07ULy4aCKUiC4XktGrnC+WTgbvU1FdRWFjcNBe6RqdvMv3o5rpUYfgYs1TUVLlclc6ZYdRbUprT1/wAj0bwR/wAibY/9tP8A0Y1eTNKzRohC4TOCEAJz6kcn8a9d8HmFvClmbZHji+fasjh2Hzt1IAz+VcQ2l+Fgx2axEV7E3E4P5fZK7MQl7Ond9PPyOLD04yqTTklr567+RL8N/wDkY5/+vRv/AENKPiR/yMcH/Xov/ob1s+DrPR7fWJX0y+juJTAQVWWVsLuXnDQRjrjufp3C+MbLRbjVIZNTv0t5/JChHkkTK7m5+WCQdSe4+nqWX1bfr5h7OP1nl5ltvrb8r/gan/NOf+4V/wC0a5Twq5bSGBx8spAwPYH+tdssVp/wiIhaYCz+w7TMCxxH5eN2dmenP3M/7PaubsYtE0vS0ktrxZreaZwsnnlcsoXI+eJD0ZegPXrW1S3PG76GNOCcJarfz/yLFGc49qgbWNNaZYbSC6vZmGRHasJGx+C1JbX1rcX0tjLZ3tldpGz+VdDa3Ck9Me1L2lNy5OZXKVGTV1+v+Q+jOce1FFUYhQTmiigAq7cTXeka9o2vaZNal7e2WCeC4vIoQ6HcSMuw55z+R7VSrM8SalfWulxG2vLiE+aF/dysvG08cGscRThVoyp1Fozow81Gau7eh6NKmiaMJ9V8NaJbQay6ELJcTeXEhY8nJOB19BnpxXL2815Fodm13PG95GAZJIpklBJeY53KSCSDknPJJzzmrXjC4mtfCl5NbSyQyrs2vGxVh86jqK57Rry6u9Dt2u7iaZiWJMrlifmPrWFDAUMHWfs73a6tv8W9vI0eJdWleb1vskktvz+RfWaRInjVyEkxuXscHNa8epXGm+KrS4j+y3FrcwSWV5azXcUe5fNYjajEFj8x4AOc+prFqlf6lfJr2ixpeXCpLchZFErAONycEZ56mtsRRp16Tp1VdOxnh6nJK97f8MztdUj0/QtFvf8AhBdMgsb68TypLie6EJRcdUZ2xn2BHrzisMwNpmj6bAsiNLEJd5TaV3M5YjC8Y+bp6cVH4+vLmy0GGSzuJbdzcqpaJypI2txkfSopJpJsebI8mOm5icVlhsDQwcpKndt2u27v72VPEOrSTlv2tZdAWaRInjVyEkxuXscHNNZizZOM+wxSUV2HKFBOTmiigArz7Uv+Qrd9B++foMfxGvQa8+1L/kK3f/XZ/wD0I1zYjZG1LdkujaXc6vqcdrZ7BIQW3SD5VA554P06d6mtPD17e65NpUHlmeEsHYsQoCnBPTOOnbvXUeBdPnttHvdWt4POunBit4yQAcYzySOM/wDoNU/HUNxpfiIXtnNLALyLDNG5UkjAI47cKan2KVJTf9IftG5uKKOp+DNV0ezN7IbeeOMgsIiWxz1IIGRWdDolxPoFxq6PELe3kEbKSd5J29BjH8Q710Pw6lmk1u7R3Zo2tiWBOQTuUD9M1Fp3/JK9V/6+1/nFR7OElzLs/wABc8k7Py/ExrzQ7zTodOuGMUv29Q8CJlj/AAkAgjH8Q45p2o+Gr/TtYg0xvLnuZ1DIISSOSR3A9DXoMFvZpoOjarf8rYWSsi4zlmRMY9+MAeppXVG+JEZcAsumZTPY+YRx+BNa/Vo/fYj2z/MqeKtEvNS0C2umgibUrXDOI/mDL3AyOexwfcc1xujeEtS1y2NxbCKKHOA8zEBj3xgGrvgy6uX8cK7k77jzfP465BY/+PAV02kGz1m01LRHWWKO1u5CHibaCpkYjB/pTUYV5cz/AOHsF5U1ynNS+AtVtp7dGnsi1w5iX5mIB2M3OV9FP44qT/hXGr/8/Nl/38f/AOJq94qt7O08BWMGmTNPbJeEJIxBLf6zPIA75ql4w/5Fzw1/16f+yR1MqdON7ra3XuOM5ytqVtO8D6tfWMd5E1qiSpuRZWOSCOvQjpVDTfDd9qWrT6anlwXEClnExIHBA7A+tUVvLtpoSLiZnj2iL5yduOgHpiuq8eJbt4xslu3McDQxiZ16qu9snoecVio03HmS2saNzTs3uUNT8F6lo2nS31xNaPHEAGCMzE7iF4BXHes3RNEuNevXtbR4kdIzITKSBgEDsD61113DpUHw41NdDuZbm3M6FnlGCG3R5H3R2xWb8N/+Rjn/AOvRv/Q0q3Sh7WMVsyVOXI32KOgeF9Q1qylurGS0VNxhYTgk9Acj5TjqOetVtd8N3nh/yPtskD+fu2+UxONuM5yB61FokOlT3zrrlzLbW4jJV4hklsjA+6e2a6Lx2sCaToC2btJbrAwidurLtjwTwO3tUKEHScrarz/QpykppdDjQxAYDHzDByM//qqfT7GfU7+KztVDSynCgnA6ZJ/IVXq/ommf2zrMFh53k+du+fbuxhSemR6VhFXkkaN2VzT1PwZqukae91L9nmjAAcREsyjPXkDv6fyrOh0S4n0C41dHiFvbyCNlJO8k7egxj+Id663w3pz6L4xbTrS+W+t2t2a52jAQgkbSMnBBA/Ornh3TbTUNB1WxBxaHU2xg9UUoQPxAxmuxUFJ6K2/3mDqtLXyOJuNDvLLQINVmMQgumCKnJcdSDyMD7vUHofema3olxoN6lrdvE7vGJAYiSMEkdwPSuy8b3Nvd+DrWWyx5AvNiYHGFDrx7ccVk/Ej/AJGOD/r0X/0N6mpSjFO3S34jhUlJq/mV/B5x9s99n/s1dLXNeEP+Xz/gH/s1dLWtH+GiKnxMM4z70UUVqQGeMUUUUAGeMUUUUAGeMUUUUAGeMVX13H/CJX33c7EIG3Df61O/cVYqDXC3/CH3/DbdqdHyufNTqvY+9KXwv0YLdHBwXcluhWNYSCc/vIEc/mwJoqCivKO4v6D/AMjHpv8A19xf+hivQrhjuwB8pUY53DH+yTyB7V57oP8AyMem/wDX3F/6GK9BuhibBGDjkGPYfxHT8q78P8DOar8SMDSf+SoRfj/6JNeizwR3VtJBOu6KVCjrnGQRgjivOtJ/5KhF+P8A6JNd1r3/ACLmpf8AXpL/AOgGunD6Ql6swrfFH0Rn/wDCEeHv+gf/AORpP/iqePBmgCMoNOXB65kcn885ryGtCb/kWrM/9Pdx/wCgQ1zrEU2m/Zr+vkdUMNOSk+fZX/FLv5npf/CEeHv+gf8A+RpP/iqePBmgCMoNOXB65kcn885ryGtfTLXTv7FvdQ1OG6n8m4ggSO3uFi++srEklHz/AKsccdan6zT/AOfa/r5Eexn/ADnon/CEeHv+gf8A+RpP/iqePBmgCMoNOXB65kcn885rzj7T4c/6BWqf+DOP/wCMUfafDn/QK1T/AMGcf/xil9Zp/wDPtfh/kHsZfznon/CEeHv+gf8A+RpP/iqePBmgCMoNOXB65kcn885rzj7T4c/6BWqf+DOP/wCMVW1yyj0zxDqNhAWaK1upYULkFiquQM478UfWaf8Az7X9fIPYz/nPTf8AhCPD3/QP/wDI0n/xVPPgzQDGE/s5cA5z5j5/POa8hq/rX/H/AB/9elt/6ISr+sU7X9mv6+Raw03Bz59ml99/PyPTP+EI8Pf9A/8A8jSf/FU8+DNAMYT+zlwDnPmPn885ryGip+s0/wDn2v6+RHsZ/wA563/whHh7/oH/APkaT/4qn/8ACGaAYwn9nLgHOfMfP55zXkNFH1mn/wA+1/XyD2M/5z3TVWl8HeAW1LwnbIL2aeOKeTYZWSNV2qOc9AqDn+8T1NY1/dJ4p+ES6x42d4rm3u9ltc29uvmSqcDAUlQRy2cED5c9RXO6V4n8QaDOkmhahZG1ltoA0Fxdw7dwhRT8rOCpBBHbpVXxPrfiPxbIn9q3enCGM7o7eG9gWNT643kk/UmvmqmW15Yx1FFW5+bm62/l226b2sfQyjUUXG0paW2dn/e+e52XhNbZPC9otlLLNAN+15ohGx+ds5UMwHOe5/pXkDSs8aIQuEzghACc+p6n8a9d8Hwtb+FLOKQxll35Mciuv32/iUkGuIbw3oqsQNclYDuI7TB/O6r6bERk6dNJdP8AI8PD0ak6k1GLdn/mdB8PPsq6KuEjF3JcXGWES7jGqwYBfqACx+Xock9qb8TLCR9L02+htnZY5JkmnVCQg/d7Ax7clsZ96PB2k6fYaxLLZahJcyGAqUZYBgbl5/dzOew7Y9/V3jHR9PvtUhnvdSe2kEIQRqkJyAxOcyTIe/YHp1o5ZfVrW6h7Gp9Z5OV3tsb+kwR3XhOygnXdFLZRo65IyCgBHFNn8M6RcWFvZS2YNvas7RIHYbS+NxyDk52jr6UotIf+ETFo05WD7D5RmYLkL5eN2N23pz97HvjmuVHhzT7nQ7eyttQa5jguZpfM3wx8usYxhZHH/LMHqOvSumbtyrkvp/XQ56dGpJSkrpJnaX0R8DfD17/wbaIl3PdKtw+0ysi8jvn0Xg8fMfWqsd7deJfhW+t+INo1S1adbKdUEbSgxEegz1bp/cB7GsTRBr/hQuPDupw+TIcvbXciOhPrweD9Kt3V9r2takl54hv7by7dJDDa2rKI1ZkK56kk8989a+VjlWI+ve15Elz83N1t/Ltt032Pew+KhTUXK+lla2nqU6M5x7UUV9QfPhQTmirELeXayuqoW3qMsgbAw3r9Ka1KiruxXrD8WHOlRf8AXYf+gtXTwNPcPtRYAAMszRIFUepOKTU7FJbJPPu4YBvBEj2dtsfg/dM7xg/hz+HNZ1eXkd2a04wc1dv7v+CT+N/+RNvv+2f/AKMWud0Fy+h2xOOhHAx0Yiu01iCO50maKa5itUbbmWZInVfmB5ErKh9OSOvHOKzbGygisYlS/t5hz8yQWijqeySFfyJrSry+217fqZ0lF0bt637eXqZ1ZOpHPiPQf+vpf/Q0rrvs8f8Az8xf9+rf/wCLqpc2cLajYOb23VklBVDBZkvyOAWlDD/gAY+2cZifLbft+ZcIwvq393/BKnxI/wCRcg/6+1/9Aem/kPoMVua/aQ3mnok97b2aiUMHnigkBODwBM6rnnOQc8dOtMmsJVwIJImdvurJbom//dPIP51c+X2r+REFH2Sbffp/wTGpWYs2TjPsMVM9xMjlHSNWU4IMCZB/Km3ahbyZVAAEjAADpzTsrXQOKtdEVBOTmiipICvPtS/5Ct30H75+gx/Ea9BrPl0LTppXkkt8u7FmO9uSfxrGrTc0rGkJKL1MabxfeLpNjY6aGsRaph3R8+acDnoMc5OOetW7Hx5LbWUEV1YrezRbt080uWYMxPHHHYfhVv8A4R7S/wDn1/8AIjf40f8ACPaX/wA+v/kRv8alRrJ3Uh3ptWsQS+PVjtZk0zR4LKWUYMqsD+OAozWNb679n8K3ejfZ932iUSedvxtwV4245+76966D/hHtL/59f/Ijf40f8I9pf/Pr/wCRG/xocazd2/6+4E6a2Ri6l4olvtJ02xjg8lLJVyS+4SlQADjAx0PHPWpr3xhJc+JrXWIbURNbxeWYjJuDDLZ5wP71an/CPaX/AM+v/kRv8aP+Ee0v/n1/8iN/jS5K3ft+Ac1PsRnx/EhkltNDgguXUjzg4zk+uFBPPPWsiPxPLb+GTpNrAImdiZbkP8z5OemOOMDqeBW3/wAI9pf/AD6/+RG/xo/4R7S/+fX/AMiN/jVONZ7v+vuBOmuhgT6+ZvC1rowtgPs8pk84tndnfxtxx9/rntSaxrv9rabplr9n8r7BF5e7fu8zhRnGBj7vv1roP+Ee0v8A59f/ACI3+NH/AAj2l/8APr/5Eb/GodOq1a4+aF72Ken+NIbWztorrRoLqa2RUScsA2F4HVTyB71TXxOk3iGfU9T02G9WSPy0t5CNsYyMEZB9D27mtj/hHtL/AOfX/wAiN/jR/wAI9pf/AD6/+RG/xquWs0ldaf12Fen2IH8c2v2JrW30C2hheRXePcpRsMCcrsxkgYz24Pan23juxs5DJZ+G7e3cjaWikVSR6ZCe1Sf8I9pf/Pr/AORG/wAaP+Ee0v8A59f/ACI3+NV+/ve6/r5C/d7WMTRNesdLsnhu9Et9QdpC4llK5AwBt5U+mfxo8R+I016OzjisVs0tQyqqvuGDtwAMDGNtbf8Awj2l/wDPr/5Eb/Gj/hHtL/59f/Ijf41HJV5eS6t/XkVzQ5ua2pxIYgMBj5hg5Gf/ANVdNf8AjL7VrVjqcGnJBLa7g2ZN3mAjGCcDHf161of8I9pf/Pr/AORG/wAaP+Ee0v8A59f/ACI3+NKNKpFWTG5wluivc+PA1ncx2Gkw2c9yCHnRxkk9TwoyeTzmsvT/ABLJp3hm70qGH5rly3nh8bQQoIxj0B5z3rc/4R7S/wDn1/8AIjf40f8ACPaX/wA+v/kRv8apxrN3b/r7iU6aVrGDN4hMvhO30QWwURSFzMXzu+ZjjGOPvdc9vem+JNd/4SDUo7r7P9n2RCPbv35wSc5wPWug/wCEe0v/AJ9f/Ijf40f8I9pf/Pr/AORG/wAal06rVm/6Q1KCd7Gb4POPtnvs/wDZq6Wq1np1rYb/ALJF5e/G75ic46dT71ZrenFxikzOTu7hnGfeiiirJDPGKKKKADPGKKKKADPGKKKKADPGKr66R/wiV8ON2xDymDjzU79x9asVBrh/4o+/APG1MgScZ81P4fX36UpfC/Rgt0cHBdyW6FY1hIJz+8gRz+bAmioKK8o7i/oP/Ix6b/19xf8AoYr0GcgNtTbtIGMNuA+ncfTrXn2g/wDIx6b/ANfcX/oYr0G63ed85Ytjncm1vxHr7134f4Gc1X4kYGk/8lQi/H/0Sa7rXv8AkXNS/wCvSX/0A1wuk/8AJUIvx/8ARJrute/5FzUv+vSX/wBANdFD+HP1ZhV+OPojxStKWQjwraR8Ya9nY8eiRf41m1oTf8i1Z/8AX3cf+gQ15kdpen6o9ej8FT0/9uRn1r23/Ilal/2ELT/0Xc1kVr23/Ilal/2ELT/0Xc1mzBGRS7jtK8YJB6c/n+NJRTAK2PFrH/hMtcXjB1Gc9Of9Y3f8ax61/Fv/ACOut/8AYQn/APRjUuodDIrR1tj9siXjAtbY9Of9RH3/AArOq/rX/H/H/wBelt/6IStF8D9V+pvH+BL1X5SKFLuO0LxgEnpz+f4UlFQYBS7jtC8YBJ6c/n+FJRQAUu47QvGASenP5/hSUUAet+CP+ROsf+2n/oxq8maVnjRCFwmcEIATn1IGT+Nes+CP+RNsf+2n/oxq8krtxP8ADp+n+RzUfjmenaWI5PiDPdW8lo8M2mxqv2WZHVWRIVYYUkrhgQM9ccZqxrl9Z2uoyxXE0MF1MtutrNLysDEXS+aR1wpYcj7pKtzjaeW+G/8AyMc//Xo3/oaUfEj/AJGOD/r0X/0N6Vv9kt5jv/tFzr/+ac/9wr/2lXJ+FXLaQwOPllIGB7A/1rsbGA33geC2jYZm09YgT0BMeK4bQ5m0q5m0vUEME3mZG7pnAGP0roqaSg32MYaqSOioznHtRRTAKM5x7UUUAFTrzZS/9dE6D2aoKt2jpGgeZd0a3EZZcZyPmyKpFw3+/wDIty/6DAyrtZIZNio3IaXb87EHrjoPw/HDubQ+ItWgsriUsytHLJk8tmRIUXODjLSLk84AJwxwDq3kUsdntlbe8d1KrtnOSQvOffBqnoMiw+M2llVWQW9rncTgf6fBzwe3X8O/SvOzCpKnhpShvp+ZthoqdZKWx6Zq3gTTdX0uaxnuLpElAyyMuRggjqvqK8ti0y40ie70q4Xc+nymIuq4DJgFWx2yGX866LwV4I8U6P8AEFtS1aRjaAS75ftIfz85wCOvUhuQP6VFrlwkfxD1Zy2YvtMaSY7r5EQYflkV87k+Lrzx7pTre1XJe/Z32PRxlGnHDqUYcruY/lvz8rcAE8dAeh/UU7QtJtvEPi23S6aVfsVxCkaxELukZXlJYkH5QkB4AySQMjqL0d359veLPcRxCcDCFD1DKRyAeAAQBVLwvFLPrmowW8vkzTXcUcUgbG12sbsKc9uSK9zNqk6WDnKDs+5wYOMZVkpK6O/8T/D/AEzWdEkhnvp7XyiZlmZl2oQpGWyPu8nPI+tef6ddzWqvFdxcgtBcwnGNykq2PTBBwa0fCvgjxRo2meJv7URsXWlywwxLOJPOkK8HAzyOnPrWddsr6tqTIcq2oXRBHcec9eVkuIrTxFSnUre1Vt/u/wA/wOvG04RpxlGHKaGpIJo3k+9LCwy/eSJh8rn36A/UVSvTuv5yf+ejD9avTfJaMrn5ls0Qk+pkDAfXb/L2rPv3WO7uGchVWRiSe3NfVr4TzfsP5fqQswRSzkKoGSScAVnv4g0xXIa6XOedqMR+gqkq3fivUPsWn5SzQ/vZSOMf56Cu4s/DGj2dqkP9n28xUYMksSuzH1JNTFTqfDsRJxh8W5yX/CQ6X/z9f+Q2/wAKU+ItLJz9q/8AIbf4V2f9g6R/0CrL/wAB0/wo/sHSP+gVZf8AgOn+FX7Gr3RHtIdmcYPEWlg5+1f+Q2/wpP8AhIdL/wCfr/yG3+Fdp/YOkf8AQKsv/AdP8KP7B0j/AKBVl/4Dp/hR7Gr3Qe0h2Zxg8RaWDn7V/wCQ2/wpP+Eh0v8A5+v/ACG3+Fdp/YOkf9Aqy/8AAdP8KP7B0j/oFWX/AIDp/hR7Gr3Qe0h2Zxg8RaWDn7V/5Db/AApP+Eh0v/n6/wDIbf4V2n9g6R/0CrL/AMB0/wAKP7B0j/oFWX/gOn+FHsavdB7SHZnGDxFpYOftX/kNv8KT/hIdL/5+v/Ibf4V2n9g6R/0CrL/wHT/Cj+wdI/6BVl/4Dp/hR7Gr3Qe0h2Zxg8RaWP8Al6/8ht/hSf8ACQ6X/wA/X/kNv8K7T+wdI/6BVl/4Dp/hR/YOkf8AQKsv/AdP8KPY1e6D2kOzOMHiLSx/y9f+Q2/wpP8AhIdL/wCfr/yG3+Fdp/YOkf8AQKsv/AdP8KP7B0j/AKBVl/4Dp/hR7Gr3Qe0h2Zxg8RaWP+Xr/wAht/hSf8JDpf8Az9f+Q2/wrtP7B0j/AKBVl/4Dp/hR/YOkf9Aqy/8AAdP8KPY1e6D2kOzOMHiPTFzi5HIxzEx/pSf8JDpf/P1/5Db/AArtP7B0j/oFWX/gOn+FH9g6R/0CrL/wHT/Cj2NXug9pDszjP+Ei0sZ/0rr/ANM2/wAKT/hIdL/5+v8AyG3+Fdp/YOkf9Aqy/wDAdP8ACj+wdI/6BVl/4Dp/hR7Gr3Qe0h2Zxn/CRaWM/wCldf8Apm3+FJ/wkOl/8/X/AJDb/Cu0/sHSP+gVZf8AgOn+FH9g6R/0CrL/AMB0/wAKPY1e6D2kOzOM/wCEi0sZ/wBK6/8ATNv8KT/hIdL/AOfr/wAht/hXaf2DpH/QKsv/AAHT/Cj+wdI/6BVl/wCA6f4Uexq90HtIdmcZ/wAJFpYz/pXX/pm3+FJ/wkOl/wDP1/5Db/Cu0/sHSP8AoFWX/gOn+FH9g6R/0CrL/wAB0/wo9jV7oPaQ7M5O11WyvHKW1wrt/dIIJ+mat1qal4S0u9tz9nto7O4XmKaBAm1u2QOtc9BPPbXbafqi7LpPut2lHqKlqUHaRSal8JdzxiiiigAzxiiiigAzxiiiigAzxioNdz/wiF8Tn7i4yuP+WqdD3H+fWp6g1z/kUb/G37qZw/P+tTqv9aUvhfowW6ODgu5LdCsawkE5/eQI5/NgTRUFFeUdxf0H/kY9N/6+4v8A0MV6DO2G2rs2kD7p3D8M8j6V59oP/Ix6b/19xf8AoYr0G7z53zbslQfmA5/Efe+veu/D/Azmq/EjA0n/AJKhF+P/AKJNd1r3/Iual/16S/8AoBrhdJ/5KhF+P/ok13Wvf8i5qX/XpL/6Aa6KH8OfqzCr8cfRHilaUshHhW0j4w17Ox49Ei/xrNrQm/5Fqz/6+7j/ANAhrzI7S9P1R69H4Knp/wC3Iz617b/kStS/7CFp/wCi7msite2/5ErUv+whaf8Aou5rNmCMil3HaV4wSD05/P8AGkopgFbHi1j/AMJlri8YOoznpz/rG7/jWPWv4t/5HXW/+whP/wCjGpdQ6GRWjrbH7ZEvGBa2x6c/6iPv+FZ1X9a/4/4/+vS2/wDRCVovgfqv1N4/wJeq/KRQpdx2heMAk9Ofz/CkoqDAKXcdoXjAJPTn8/wpKKACl3HaF4wCT05/P8KSigD1vwR/yJtj/wBtP/RjV5M0rNGiELhM4IQAnPqRyfxr1nwR/wAibY/9tP8A0Y1eSV24n+HT9P8AI5qPxzOn8AXkFp4kP2iRYxNA0aFjgFtykD9K6Pxz4cl1ONdRtMtNbx7Xix95QSePcZNeagkEEHBHQ16H4P8AGAuQmnapJiUcRTMfv+x96KE4Sh7KYVYyjL2kTE8J+K5NGlFpelnsmPfrEfUe3tXZ+IPD9r4jsVnt2UXAXdDMvRh6H2rm/GfhJopJdU01C0bkvNEP4T3Ye1UPCfix9HlFpesXsmP1MZ9R7VcZezfsauxLjzr2lPcn03Upra6OmaspjuIztVm/irbznHtWj4g8P2viOxWe3ZRcBd0My9GHofauT03Upra6OmaspjuIztVm/iq2nTfLLboyU1NXRt0Zzj2ooqhBUynNjL/10T+TVDU0TxeQ8crOu5lYFVB6A+49apblw3+8vQTfbdwKl3kQJJGOrYHDr7jHTvz61l6hZ6rbSW97o9mupvE4ElvHH5yyR8Eq8fUjKqeRwQCORmpgtsOk03/fof8AxVV9e1dIdPRri5aT94AGm06G5PQ9pCfzrCvT56bi7WNaMJc6s/xR2uteINXg0mWXSLRrm9Qo0USNPIWO4Z+UPk8ZOOnHPGa460s9au45LzVLC6W+uZGln3W7IS2cD5cccAV0moJNPpsq2cqpKVBR2gSYdc/cf5Tkcc+ua5/T9SQ2caXcyQS/MIo5tMhiaUKxB2oucAdMnGSDjJBxmsHRw9bnpxSurfj5jjKpWo3b2fdL8B/9m33/AD53H/fpv8Kiit9VsNetJoNDkuYZpY/tLPbybYijqUlypG1lG7DE9CwOQa2Gt5EuGhd4gyzJD/x5RYJbOD9OP1qhbZ1We1uLN4jFDdSRyO2lwHBjCMRuPKE7wAV5yCewNaVqSqx5JWdwpQknzJrS/VG34n1zXE02J9D09726WdWCKs0+zAOH2biDg4I3AgHB64rIs9JaOQLcrvlOWEKgbix53PjAUE8noad4t1L+zNKik88Ql5gufscdzkYJ+5IcDp16/nUMt2JoyjXUoQ8lI4FRSfUgMAamnhKVCrL2aSvYnmqVKUXJ9+qF1K6jWMxCUOd5lnmOAGf29hz+ZrB1Gy1LxNr91aWqNDZQzsk0zDAJB/Wn3EVlq2t22kLfToxbc8YtxhgBuwW3ccD0Nd2qJGG2KqAks2BjJPU10xp+0TT2FUao07P4nZrVNW138yrpmmWuj2K21ogVFGWY9WPqa5TxL47S3zbaHIskoPzz7cqvsM9fr0ql4w8Xm4Z9O0uTEQO2WVT9/wBh7VxccbzSrHEpd2OFUDJJrGviLe5TM6dG/vTNz/hN/EP/AEEP/IMf/wATR/wm/iH/AKCH/kGP/wCJrsvCnhKPSIhd36q94y9DyIh/j71el8ZaBDK0b6gpZTg7Y3YfmBiiNKpy3nUt/XqDnC9oxuef/wDCb+If+gh/5Bj/APiaP+E38Q/9BD/yDH/8TXe/8Jv4e/6CH/kGT/4mj/hN/D3/AEEP/IMn/wATRyf9Pvx/4Ic3/Tv+vuOC/wCE38Q/9BD/AMgx/wDxNH/Cb+If+gh/5Bj/APia73/hN/D3/QQ/8gyf/E0f8Jv4e/6CH/kGT/4mjk/6ffj/AMEOb/p3/X3HBf8ACb+If+gh/wCQY/8A4mj/AITfxD/0EP8AyDH/APE13v8Awm/h7/oIf+QZP/iaP+E38Pf9BD/yDJ/8TRyf9Pvx/wCCHN/07/r7jgv+E38Q/wDQQ/8AIMf/AMTR/wAJv4h/6CH/AJBj/wDia73/AITfw9/0EP8AyDJ/8TR/wm/h7/oIf+QZP/iaOT/p9+P/AAQ5v+nf9fccF/wm/iH/AKCH/kGP/wCJo/4TfxD/ANBD/wAgx/8AxNd7/wAJv4e/6CH/AJBk/wDiaP8AhN/D3/QQ/wDIMn/xNHJ/0+/H/ghzf9O/6+44L/hN/EP/AEEP/IMf/wATR/wm/iH/AKCH/kGP/wCJrvf+E38Pf9BD/wAgyf8AxNH/AAm/h7/oIf8AkGT/AOJo5P8Ap9+P/BDm/wCnf9fccF/wm/iH/oIf+QY//iaP+E38Q/8AQQ/8gx//ABNd7/wm/h7/AKCH/kGT/wCJo/4Tfw9/0EP/ACDJ/wDE0cn/AE+/H/ghzf8ATv8Ar7jgv+E38Q/9BD/yDH/8TR/wm/iH/oIf+QY//ia73/hN/D3/AEEP/IMn/wATR/wm/h7/AKCH/kGT/wCJo5P+n34/8EOb/p3/AF9xwX/Cb+If+gh/5Bj/APiaP+E38Q/9BD/yDH/8TXe/8Jv4e/6CH/kGT/4mj/hN/D3/AEEP/IMn/wATRyf9Pvx/4Ic3/Tv+vuOC/wCE38Q/9BD/AMgx/wDxNH/Cb+If+gh/5Bj/APia73/hN/D3/QQ/8gyf/E0f8Jv4e/6CH/kGT/4mjk/6ffj/AMEOb/p3/X3HBf8ACb+If+gh/wCQY/8A4mj/AITfxD/0EP8AyDH/APE13v8Awm/h7/oIf+QZP/iaP+E38Pf9BD/yDJ/8TRyf9Pvx/wCCHN/07/r7jgv+E38Q/wDQQ/8AIMf/AMTVq38S+LLuB5re6MiIcNiKLP5YzXf2OtaRrolgtLiO5+X542QjI+jAZFcZrWh3HhW8N9poaSwc/Omc7PY/40pQnFcym2vIcZRbs42ZveFPFqaxGLW+Kx3qjr0EvuPf2rW1rRYNZtPLk/dzJzFMvVDXnd7ZJfRrqmjsVlU7mVeCD/jXXeE/Fa6vGLO9IS9Qf9/AO/1ralV5v3dT/hzOcLe/Ay4Lie2um07VFCXSfdbHEo9QauVva1osGs2nlyfu5k5imXqhrlIJ57a7bT9UXZdJ91u0o9RRKLg7PYIyUldF3PGKKKKQwzxiiiigAzxioNdJ/wCERvshvuLjKDB/ep0br+FT1BrhH/CI34BXO1MgMc/61Oo/qKUvhfowW6ODgu5LdCsawkE5/eQI5/NgTRUFFeUdxf0H/kY9N/6+4v8A0MV6BMVViIxHtYD7vI/DPIrz/Qf+Rj03/r7i/wDQxXoN3u875y5OAcvgk/iOv1rvw/wM5qvxIwNJ/wCSoRfj/wCiTXda9/yLmpf9ekv/AKAa4XSf+SoRfj/6JNd1r3/Iual/16S/+gGuih/Dn6swq/HH0R4pWlLIR4VtI+MNezsePRIv8aza6TQ4Em0+1aWNZFhOozKGGRvS1V0JHf5lBweDjmvMj8MvT9UexR+Gfp/7cilHosNtbR3Gu3bWSTKGhhhjWadgeQxjLrsXHQsQTkYBGSLcc3hyPRbnT/7R1Q+fcQz+Z/Z0fy+Wsi4x5/fzP096wJppLid5p3aSSRizuxyWJ6mmVnYwua/2bw5/0FdU/wDBZH/8fpfI8O7Sv9q6ngkH/kFxZ/Pz/eseiiwjX+zeHP8AoK6p/wCCyP8A+P1c1a48O6prF9ff2hqcQvLl59n9mxMU3MTjd5/v7ZrnKKLBc3bbSdFvjNFY6pftOlvNOqzaeiK3lxtIQSJiRkKRnBqnrbH7ZEvGBa2x6c/6iPv+FTeGf+QtN/2D73/0llqvrX/H/H/16W3/AKISrXwv5fqdC/gS9V+UihS7jtC8YBJ6c/n+FJRUnOFLuO0LxgEnpz+f4UlFABS7jtC8YBJ6c/n+FJRQB634I/5E2x/7af8Aoxq8maVmjRCFwmcEIATn1I5P416z4I/5E2x/7af+jGrySu3E/wAOn6f5HNR+OYUu8/L229CBjvmkoriOk9D8IeLxdKmm6s483G2KVv4/Y+9U/GXhJoM6hpkY8lR+8iRQCvJOeOvXv/KuIBIIIOCOhr0Pwf4uF2E0zVWzKflilb+P/ZPvXbTqRqx9nU+TOWcHTfPAxPCfix9HlFpesXsmP1MZ9R7V2fiDw/a+I7FZ7dlFwF3QzL0Yeh9q5rxh4PNsX1HS48xHmWJR9z3HtVHwn4sfR5RaXrF7Jj9TGfUe1XGTpv2NbYUo837ynuT6bqU1tdHTNWUx3EZ2qzfxVt5zj2rR8QeH7XxHYrPbsouAu6GZejD0PtXIWmsSadI1jratFLFwHKk5/L+dU06TtLboxJqautzeoJzWZ/wkOl/8/X/kNv8ACj/hIdL/AOfr/wAht/hT54dw5ZdjTqG8tYr62aGdcq3p2PrVWLXdNlkCJdLk9NylR+ZFaFO8ZIWqGeG9bfS500XV3kbc6pZyhchgc8E5z12gcd+2BWr4v1abR9EEsbNIsh8gQPhogWy28qQfmBGQRg5xnI4PL6j/AMjHoP8A19r/AOhpWx8SP+Rcg/6+1/8AQHpPWlKL1tsG1SLXUtah4peLWNNgiDkTarFIhIAJjDnIY88ncg9gDW1qeqxaZp/2rU7mR0iUIHkbc7nsPcn/ABrg9R/5GPQf+vtf/Q0rY+JH/IuQf9fa/wDoD0QtBTmt0Eve5Y9GYl6T4r1prtnm/syI4gSRdhI78AnHPfNbP4AewGBVXTBjSbTH/PFP/QRVqlBaXe7HJ9EY2k/8lQi/H/0Sa7rXSR4d1Ejgi1lx/wB8GuF0n/kqEX4/+iTXda9/yLmpf9ekv/oBrSh/Dn6szq/HH0R4pXpvg3w5FZ2keq3WyS5uEEiYGBGrDPA9a8yr135o/h5zkMul/kfKrlwkU5OT6G9duyS6nMeMPF5uGfTtLkxEDtllU/f9h7VxTMWYscZJycDA/IUlFc9SpKpLmkbQgoKyFVirBhjIORkZH5GkoorMsVWKsGGMg5GRkfkaSiigBVYqwYYyDkZGR+RpKKKAFVirBhjIORkZH5Gkrr/DfgdtUt/tWpvLbwMP3apgM3vyOBW7/wAK30j/AJ+b3/v4n/xNdMcLUkrpGMq0Iux5orFTkY6EcjNJXpn/AArfSP8An5vf+/if/E0f8K30j/n5vf8Av4n/AMTVfU6pP1iB5orFTkY6EcjNJXpn/Ct9I/5+b3/v4n/xNH/Ct9I/5+b3/v4n/wATR9Tqh9YgeaKxU5GOhHIzSV6Z/wAK30j/AJ+b3/v4n/xNH/Ct9I/5+b3/AL+J/wDE0fU6ofWIHmscrR7toU7lKncgbg+meh9xzTa9M/4VvpH/AD83v/fxP/iaP+Fb6R/z83v/AH8T/wCJo+p1Q+sQPNAxAYDHzDByM/8A6qSvTP8AhW+kf8/N7/38T/4msLxL4HbS7cXWlvLcQqP3qvgsvvwBkVMsLUirtFRrQk7HIhiAwGPmGDkZ/wD1UlFFcxsKGIDAY+YYORn/APVSUUUAT2V9cafcCe0fy5B0bAJHIP4dP6d69T8PeIbXxJYNDcKouAuJYT0Yeo9q8lqa0u57G6S4tZDHLGcqwrejWdJ+RlUpqa8z0S50WPwzHLc2FjLfCeUL5Ssf3annnGcAEdcd8VHPpOnfbLm5gtd88CmVSkzIcgA44PB961/DXiOHxBYkPtS6jGJYvUf3h7Vzxtk8PeNZ7ZJZJIr+2ZwGOSCSepPXG08+9XWoL4oN8svPVPy/yOrD4uKtGcVzR6tKzVnvpq/M6Lw34jTXLIyTRrbSCUxKpfO8hQeOBz149qta1osGs2nlyfu5k5imXqhrmZ4LdbOTckw2yi4VrcgOrjuM8VsaB4rsdXcWgeVLlF/5bgKZMd+OM+1dFCs23RrL0fdf5nLiaMOVV6MlruuqfXTsYcFxPbXTadqihLpPutjiUeoNXK3ta0WDWbTy5P3cycxTL1Q1ykE89tdtp+qLsuk+63aUeorSUXB2exyxkpK6LueMUUUUhhnjFQ66D/wh98cNjagztBH+tT+LqPp+NTVX1zH/AAid/wD6snYgyM7h+9Tr2NKXwv0YLdHCQXcluhWNYSCc/vIEc/mwJoqCivKO4v6D/wAjHpv/AF9xf+hivQJn+Y7CArAE7OM/Udj7V5/oP/Ix6b/19xf+hivQbokygsWJI6swbP8AwIda78P8DOar8SMDSf8AkqEX4/8Aok13Wvf8i5qX/XpL/wCgGuF0n/kqEX4/+iTXda9/yLmpf9ekv/oBroofw5+rMKvxx9EeKV1Hh52GmxID8rR6oSPcWXH8zXL10/h9T/Z8TY4EWqAn/tyFeWvhl6fqj2KPwz9P/bkcxS7mClQTtJBIzwT/AJJpKKk5wpdzBSoJ2kgkZ4J/yTSUUAFLuYKVBO0kEjPBP+SaSigDX8M/8hab/sH3v/pLLUOtsxvIlJO0WtsQM8A+RH/gKm8M/wDIWm/7B97/AOkstV9a/wCP+P8A69Lb/wBEJVL4X8v1OiP8CXqvykUKXcxUKSdoJIGeAf8AIFJRUnOFLuYqFJO0EkDPAP8AkCkooAKXcxUKSdoJIGeAf8gUldV8N/Bx8deNrXSHkaG22tNcyJ95Y1649ySB7ZzSbsrgtTsfBH/Im2P/AG0/9GNXkzSyPGkbyMyR52KWJC564Havs+x0Lwh4bsotHtNJg8mDKgvF5rDJJOXbLHkn6V4P8bvhnp/g2ez1fw8rR6bfOY2gLFhDJjI2k87SM8dsH1GF9fo4m1Om9Y6P+vkH1adK85dTyalLMwUMSQowAT0HX+tJRVAFLvf5fmb5Pu8/d5zx+NJXVeI/At5olu09rFqN1FFPNFNJLp5iVBHt/eAhmyjbjhjj7ppXSCzZ03g/xQmrWq2F6QLuJMZY/wCtUcZ+vrXP+OfDqaa8d/asFt5X8vys/cY5bj26/jWD4cSSTxPpiQqzObqPCqMk/MK9A+JlpcweGYHmt5Y1+1qMuhA+49drrQnS5ajXMtu5zezlCpeOzOU8KeLX0eQWt6WeyY/Uxn1Ht7V3U994aviJLu40uc44MzxsQPxryCKKSaVYoUaSRztVEGSx9AKuXuiarpsKy6jpl5aRscK89u6KT6AkVnDGezShKz7XLlQ53zI9N/4pH/qCf+Qqe8vhSUgyPozkDA3GI4ryOKKSaVYoUaSRztVEGSx9AKuXuiarpsKy6jpl5aRscK89u6KT6AkVbx0U7NK7J+rNq92em33hnRdZ0wiyhtYt3MdxaoowfqvUVyVvd3Wjaj/ZetHG0ARSseNvQc+nFV/BGs3lnrUFhCrTQ3coQxA9GJA3D3rqfibp0sOgW1zPaSRsLkIJHjI4KscZ/CrlWpSjzppS7dyIwnF8r1Riaj/yMeg/9fa/+hpWx8SP+Rcg/wCvtf8A0B6xrhWn8Q+HUhDSu1whAAyW+ZK3/iZaXMHhmB5reWNftajLoQPuPTdSCjOLauw5ZOUWkYOo/wDIx6D/ANfa/wDoaVsfEj/kXIP+vtf/AEB6yLsG58S+H1gBkZrlSqqMkjclbvxMtLmDwzA81vLGv2tRl0IH3HolUhGM4tq7Dlk5RaRnab/yCrT/AK4J/wCgirJJJyeSaz7a7jtdKslk3NI8SLHEilnc7RwAOpq9embTYbaXUrO6s47gfK88DIAcA7ST0PP6Gn7WnG0ZNJhySd2kZGk/8lQi/H/0Sa7rXv8AkXNS/wCvSX/0A1wujAyfFGARjcWyFC85/cnpXf8AiO1uIPDeomeCSMG1lALoR/AfWro1IRjKLau2yKkZOSaXY8Pr2DRZ4Nb8JxRebv32wgmweVbbtbNeQxRSTSrFCjSSOdqogyWPoBW/p7a/4QuEvLrTb22t5CFcXEDxpJ7ZI61xUMRClO0nudNWlKcbroZWqaXc6RfPa3ibWU8MOjD1FVGZnYs5LMxySTkk163cWen+NdEjeHlm/wBWwGWjb0/+tXm+s+G9X0GZl1OwuIIw21ZniYI/cYbGDxSrwjSklffYKUnNbbGYrMjBkJVlOQQcEGkp0UUk0qxQo0kjnaqIMlj6AVcvdE1XTYVl1HTLy0jY4V57d0Un0BIrBzinZvU2s2rlJWZGDISrKcgg4INJToopJpVihRpJHO1UQZLH0Aq5e6JqumwrLqOmXlpGxwrz27opPoCRQ5xTs3qFm1cpKzIwZCVZTkEHBBrtPB3hD7Xs1LU0/cZzFER9/wBz7VxsEEt1cRwW6NJLIwREUZLE8AAV7P4pkvdC8M3N7HbvE0YVUMkZABLAf1rqw/suZuo9uncwq89rRW5jeK/FkejQm0sSrXjLjjpEPX6+1efNr+sMxJ1W959Lhh/Wqf7+8uuA888rcAAszsf5mrV7omq6bCsuo6ZeWkbHCvPbuik+gJFRWxTnOzduyKp0VGOwv9vav/0Fb3/wIf8Axo/t7V/+gre/+BD/AONUKKjnl3K5V2L/APb2r/8AQVvf/Ah/8aP7e1f/AKCt7/4EP/jVCtTT/DWr6pbieysmkiJwHLKoP0yRmnFzk7K4morci/t7V/8AoK3v/gQ/+NH9vav/ANBW9/8AAh/8a7vT/Bus6r4iu0eHT9O0mJnk+0tZW7hIwTtGMZzj19DVq68N6PqmlXk3grWrXULuwiMs1vNpkA3r1+U+WO3159K5q+OoYer7KpJ3W+jsr7XfS56CwlGSbjK616LW3b3tTzr+3tX/AOgre/8AgQ/+NH9vav8A9BW9/wDAh/8AGpL+Y3WiWdxJFAkpuJkLQwJHkBYiAdoGfvH86zK6pOSdrnLVpRhKy12f3q5fGv6wDkare/jcP/jXbeFPGf28iw1h189uEmIAD+x9686rW8K6DP4o8V6doto/lyXkwTzP7i9Wb8ACfwqqdedN3uYSpRmrHS+KfBcqPJe6IhZH/wBbboefXIHcZA4rhOnWvtO10Hwz4asbfT2slu3ijVTJPGJXbAxlie568V5B8cPhjpWm6P8A8Jf4aj8iOSYfbIAxKHeeJFz0+bgjpyOmDXLLGYatVcKT1W6Nlh6tOCc9jwwMyhgpIDDBAPUdf6UlFFaEihmUMFJAYYIB6jr/AEpKK7KSyt5Ph7HK2mWwaGwE63qoyymVr1o9pIO1h5atwQTxweKTdgSuR/Df/kY7j/r0b/0NK0vE3/JQdP8A+vT/AOOVp6IthPrVvfaNDZRWUlpLCogikSUMhhLLKGJBI8wfMvXn6VJeny/HEck1nayWrWaRzXFwCWhBZziPB+8cehOAegzXoU3/ALOn5nLJfvn6FLJxjtWDrOjP5n27TspOh3ELxn3HvXWqc6XMssYVwqshMCjK7gMhuueR+GfWpkjtXubRjB5cDW0hcZ3Ekb+c9zwDVyipKzJi2noUfCfixdWjFnfEJeoMc8eZ/wDXrX1rRYNZtPLk/dzJzFMvVDXHeNdKhsvsl/psqK8MI3vEoXe3mN82QOTk4yewHpXX+G9Qm1Xw7a3lzt82QMG2jAOGIz+la0Zud6czOpHl9+JytndTx3Mum6iNl5BywHRx2YfnV6qviXj4g2OON1pz7/6yrVQtG12LeyYZOMdqh13d/wAIffZD7dqYJ5X/AFqfkfbvU1V9dA/4RK++7u2J/CQ2PNT8CP1ol8L9GJbo4SC+u7ZCltczRKTkrHIVBPrxRUFFeUdxf0H/AJGPTf8Ar7i/9DFeiTSvGZIkIVZAN6hNuSPUHofpxXneg/8AIx6b/wBfcX/oYr0C5BEgHzY2jG5g2B7EdRXfh/gZzVfiRg6T/wAlQi/H/wBEmu617/kXNS/69Jf/AEA1wuk/8lQi/H/0Sa7rXv8AkXNS/wCvSX/0A10UP4c/VmFX44+iPFK6vw7MV0dIR9111Nz9RYkf+zGuUrqPDyE6bE/YR6oD+Nl/9avLj8MvT9UezQ+Gfp+qOXpwlcQtED8jMGIx1IBA/wDQjTaKk5gpwlcQtED8jMGIx1IBA/8AQjTaKACnCVxC0QPyMwYjHUgED/0I02igDX8M/wDIWm/7B97/AOkstRa3K5uYIifkW2t2Ax0JgjB/9BFS+Gf+QtN/2D73/wBJZar61/x/x/8AXpbf+iEql8L+X6nRH+BL1X5MoU4yuYViJ+RWLAY6EgA/+gim0VJzhTjK5hWIn5FYsBjoSAD/AOgim0UAFeifBTxXaeGPiFbNq0iRWVzFJb+a+AsLPsIYn0JjUZ7Zz6153RSaurAnZ3PuG70Fru8NxBOnlSncSeevpjrXjf7R/iXTnsdL8L2kqT3dvN9onKkEwgIVVT7ncTj2HqK5fwbrurWnhKzhtdUvYYgHASO4dVHzt2BryrOeteTgsLQpVqkqas76/iezj6VWlQpSnK6mrry0X+YU55XkSNXORGu1BjoMk/zJptFeueMFad0Y5PDOmucmdJ7iEt6RgRsq/wDfTyHP+17Csyus8UaDZIZ7nR30uKGO5nVYYNTjkLQLt8uTDSsxZstkD+6OBnlPcaO48BRWUWvajd3cNvGtrPdtC8NtGhgSDywRkLk7hOO4I8vHIYiuu0Xxd4e+IK3mlwwSyoiBpI7qFdrrnqBk9DjrXNeHLnTdNkntr6O5nnvJb1zbW8W9mhlmEZbAO448gZABOGz0BxqeHtP8G+DIbzVtDv3vkYLHO8c6TCBeWJO3G0YUnnk4wuSQD8ZnC5sTVlUUudP3Lbb/AIaW/wCHPpPaT9o0mrXd7mN4H0+103Xrua+a1nW286aC6gtI4isUaRNn5Uz863KkgYI8vHIYiup0bxb4d+ISXmlwwSyoiB5I7qEYcZ6jk9DjrXN+ENc8Papq97b2P22dZ/OXyPs20JBIkKdd5JCiBR6ndnGATWxpOjeFPh6LvUrG5mm8wLHKTKsnkpycnGMD5GPPJK4XJIByzCFSU3Oupe005Wtr/LYxpVpStaSa1vfcwvA+n2um69dzXzWs623nTQXUFpHEVijSJs/KmfnW5UkDBHl45DEV1OjeLfDvxCS80uGCWVEQPJHdQjDjPUcnocda5bwz4l8NXWpXkNr9uuBcNLGtqtrhhDIkKEKA5Z9ogGcDdhicYUmt/SdD8LfDyO61SynuJllCxyuZFkES8knjGB8pPPJIwMkgEzCM5zlOspOqlHla2v8ALYdKtJ2tJOOt77mB4H0+103Xrua+a1nW286aC6gtI4isUaRNn5Uz863KkgYI8vHIYiuq0Xxd4e+IK3mlwwSyoiBpI7qFdrrnqBk9DjrXLeHvEfhuTULyOJdQmS4MiJai0w3kyrDH8oDlnCiFQcDcd2cYBNa3h+x8F+CobvV9G1Br1H2xTPHcJMIV5bnbjAwpPPJxhckgFZhGc5ylX5va6cr6X+WwUq0pJWknHW99zL8F2sVlrtzLqk8F0toJpbe6ito4ykUSRN2XPzLcqcDBHl45DEV0+i+LvD3xBW80uGCWVEQNJHdQrtdc9QMnocda5bw94j8NyaheRxLqEyXBkRLUWmG8mVYY/lAcs4UQqDgbjuzjAJrW8P2PgvwVDd6vo2oNeo+2KZ47hJhCvLc7cYGFJ55OMLkkAmYRnOpKVfmdXTlfS/6BSrSklaScdb33MvwXaxWWu3MuqTwXS2gmlt7qK2jjKRRJE3Zc/MtypwMEeXjkMRXT6L4u8PfEFbzS4YJZURA0kd1Cu11z1Ayehx1rlvD3iPw3JqF5HEuoTJcGREtRaYbyZVhj+UByzhRCoOBuO7OMAmtbw/Y+C/BUN3q+jag16j7YpnjuEmEK8tztxgYUnnk4wuSQCZhGc6kpV+Z1dOV9L/oFKtKSVpJx1vfcy/BdrFZa7cy6pPBdLaCaW3uoraOMpDEkTdlz8y3KnAwR5eOQxFdNo3i3w78QkvNLhgllREDyR3UIw4z1HJ6HHWuX8PeI/DcmoXkcS6hMlwZES1FphvJlWGP5QHLOFEKg4G47s4wCa2dE0vwh4EhudX0u8kuo5gscjrMsoiTls8YwMKTzycYGSQC8wjUnUlOvzOqkuV9L/LboFKtKSVpJx1vfcxfA2n2uma9dzXz20wtvOmgure0jiKxRpE2flXPzrcqSBgjy8chiK6zQvGWg+PBe6baxTOqxfvY7mIAOh445P61ynh7xH4bk1C8jiXUJkuDIiWotMN5Mqwx/KA5ZwohUHA3HdnGATWt4fsfBfgqG71fRtQa9R9sUzx3CTCFeW524wMKTzycYXJIBWYRnKpKVfm9rpytbX+WwqVWUrNSTWt77mR4H0+103Xrua+a1nW286aC6gtI4isUaRNn5Uz863KkgYI8vHIYiup0bxb4d+ISXmlwwSyoiB5I7qEYcZ6jk9DjrXL+HvEfhuTULyOJdQmS4MiJai0w3kyrDH8oDlnCiFQcDcd2cYBNdL4W8L+GvCfn6lotzJcLcIqtIZlkCp14wBwcZ5znoPSnj4VKlSUqyk6vu8ttr/LboFOs2k+Zcut77nO+BbaPStWvheXEMsEXmzRzW9tHCPLiSJgSFUE7hcqexBjxyGIrpdG8W+HfiEl5pcMEsqIgeSO6hGHGeo5PQ461zeja5oEGtajZvHqDu7yRGzFr8yxSrDHhcOxcKIVGRyd2QMAmtLw/Y+C/BUN3q+jag16j7YpnjuEmEK8tztxgYUnnk4wuSQCZjGpKpKVfm9qrcvZP9Ao1XKzUk11vuZHgfT7XTdeu5r5rWdbbzpoLqC0jiKxRpE2flTPzrcqSBgjy8chiK6nRvFvh34hJeaXDBLKiIHkjuoRhxnqOT0OOtcr4b8T+GZ9QvY7f7dcJO0kYtVtfmEMqwx4UByz7RCoOBu+bOMAmtjw/Y+C/BUN3q+jag16j7YpnjuEmEK8tztxgYUnnk4wuSQCswjOc5Sr83tdOVra/y2HSrSklaScdb33MjwPp9rpuvXc181rOtt500F1BaRxFYo0ibPypn51uVJAwR5eOQxFdTo3i3w78QkvNLhgllREDyR3UIw4z1HJ6HHWuV8N+J/DM+oXsdv9uuEnaSMWq2vzCGVYY8KA5Z9ohUHA3fNnGATWx4fsfBfgqG71fRtQa9R9sUzx3CTCFeW524wMKTzycYXJIBMwjOc5Sr83tdOVra/wAtgpVpSStJOOt77mR4H0+103Xrua+a1nW286aC6gtI4isUaRNn5Uz863KkgYI8vHIYiuq0Xxf4e+ICXulwQSyokYaSO6hXa656gZI4OOveuU8N+J/DM+oXsdv9uuEnaSMWq2vzCGVYY8KA5Z9ohUHA3fNnGATWx4fsfBfgqG71fRtQa9R9sUzx3CTCFeW524wMKTzycYXJIBMwjOc5Sr83tdOVra/y2ClWlK1pJx1vfcyPA+n2um69dzXzWs623nTQXUFpHEVijSJs/KmfnW5UkDBHl45DEV1OjeLfDvxCS80uGCWVEQPJHdQjDjPUcnocda5Xw34n8Mz6hex2/wBuuEnaSMWq2vzCGVYY8KA5Z9ohUHA3fNnGATWz4d0/wb4NS51PQr57/wA5kt3MU6z+XkkgfLjGdpODksQAuSQCZhGc6kpV+b2unK1tf5bBSrSklaScdb33PJ9cjfT01GBXtZTBcwpDcQWscReKWKQ/wqOqleD0NcvW/q2oDVLbVrtbf7Mj3tuqQ5J8tVjlVVyfQAVZ8J+E31iUXd6pSyU8DoZT6D2r7ylGpUhBS+K2p4+LqK0Zt9P/AG5j/BvhmfULgX1xmO0CsnIGZQQVIGe2Cefyrstc8RWPhm0jjWMSS4xHbxnbgep9BTPEPiG18N2CwwKpuCuIoV6KPU+1eV3d3PfXT3F1IZJXOWY13SnHDR5IfEeXGLrPmlsesaZ8RNNg1bVND1uF7S0mElq12jb8EErnGOBgn17U2GDw98LoNSnh11dU1C8tPKgtUQfKGAYFsE4ByDzjjpmuP8T6Volj4mvYr7VL8Tu4nZYdPR1XzAJAAxmBOAwGcCq+rXHh7Vb5bl9S1RGEEMJB06M5McSx7v8AX99ufbNeBjMDHEYic3JpStzLTW23S/3HtSmqEnTgvhbt5alOG/ubDw3F9kk8v7RLdQS/KDuRkhyOenTqOax63tf0+PTNMs7eCdriJpWmSR4wjESQW8gBUE4I3Y6npWDXpT3+S/IwxHxr0j/6ShySvGkiocCRdrjHUZB/mBXR/DvX7bwx8Q9H1e+XNtbz4lOM7VZShbHsGz+Fc1RWdrmB9w3enJrRj1DTbuGWGdAyurblYdiCOoxXm/x88Qado/w7XwwJkm1C9eMiMEbo0Vt5cjtkgAeuT6GvLfh7rGp2GgTR2Oo3dsn2lvlhnZB91ewNcNrksk/iDUJZnaSR7qRmdzksSxySe9eRhsLQhipzgve6ns4qlVhhKdSUrqXQppK8aSKhwJF2uMdRkH+YFNoor2DxhySvGkiocCRdrjHUZB/mBWjp0819s0+9urqSwgjlmW3WYhQVR3+UHIGSDzjuazKv6L/x/wAn/Xpc/wDoh6qCTmr9zbDpOtBPuvzPQ/CsGoXGqRtcG8mJtN8Mdxq0d06rJscYiUb0yADkgds9qvas17b+IoYDdXNtH5Q3QDVY7dWY7sZgb5mPT5h7elUtQv7rw/4ds77Q1VriXT7EzZTeGzCA+5ejbBFBjP3ftBP8a4q28x8XaDZzeKZAitcxoJRiIIgniQMv8K71luM4ADfZgf8Alm2eT+1JQjZxXKnbzv8A18ynRcqnPfVrsrW+7c3rfTL280t5LVTNYxkl2TUYWjQjk5PQY71TfMsaO8kkisknlyLdrKvC5I4/D865vW9b1e18RWcMFjE8jRwvFE9qHkV2ALRRMQXiCSF41WNlKFMcMCT2N5b28NjbyQSGVpJbzdIW3GQKoVJCe5dFSQt/EZC3RhXRhMfUq1YwqRVpaq36mc6MUrro1fRHIeInY+HrhCxK/Jgenzj/ABNdF4I/5E2x/wC2n/oxq5vxD/yAbj/gP/oQrpPBH/Im2P8A20/9GNXrUv4/y/U8yp/D+Zj+JZXHjqxiB+RrYMRjqR5oH/oRqeqvib/koOn/APXp/wDHKtUvty9Svsr0F3EKV7Egn/P41X1wH/hD7/72NqdwR/rU/Ef1/Cp6r67j/hEr77udiEDbhv8AWp37iiXwv0YlujktL8Tavotq1vpt35MTOXK+UjZbAGckHsBRWVRXk8q7Hfdl/Qf+Rj03/r7i/wDQxXoNwxWRlBwCBuAXZ+Y6Zrz7Qf8AkY9N/wCvuL/0MV6DcnMg642jb8xIx7Z7e1ehh/gZy1fiRgaT/wAlQi/H/wBEmu617/kXNS/69Jf/AEA1wuk/8lQi/H/0Sa7rXv8AkXNS/wCvSX/0A10UP4c/VmFX44+iPFK6nw9IRpccfZk1Rj+Fl/8AXrlq6jw8hOmxP2EeqA/jZf8A1q8uPwy9P1R7NC/LO3b9UUfCc0lvrrzQSNFLHY3jpIjEMpFtKQQR0Ipn/CW+I/8AoYNU/wDA2T/Gjwz/AMhab/sH3v8A6Sy1kVnbU5+hr/8ACW+I/wDoYNU/8DZP8aP+Et8R/wDQwap/4Gyf41kUU7IV2a//AAlviP8A6GDVP/A2T/GtHQ/EWtX95c219rF/cwPp95uimuXdWxbSEZBOOozXL1r+Gf8AkLTf9g+9/wDSWWk0rDTdw8M/8hab/sH3v/pLLUOtsxvIlJO0WtsQM8A+RH/gKm8M/wDIWm/7B97/AOkstV9a/wCP+P8A69Lb/wBEJWi+F/L9TeP8CXqvykUKntLa4vriO0tlaR2b5UHqep/QflUmmaZdaterbWabnbqeyj1Nen6Xomn+FbXcga4u5flDYG+Rv7qjsP8AJNbUaDqavY4alRQ06lXS/DGlaFpytqsMd3cSHB3QmUk4+6qgEnueB6ntV4/2EYxGdJYoDkL/AGTJjP08upZ7iLSLd9T1eQGdl2qi87R/cQdz796W1v8AV7u2SddLt4VcZVZrshse4EZxXpqMY+6l+Fzibk9WVvK8Pf8AQF/8pEn/AMbp/wDxIjGIzpLbAchf7JkwD9PLq752r/8APjZf+Br/APxqjztX/wCfGy/8DX/+NVVl/SFd/wBMfapZLbILa1EUXO1BbFMc/wB0gYqgz6I8KxPpcjRp91DpUpC/QeXWrG1yY1M0USv/ABKspYD8do/lVbztX/58bL/wNf8A+NVEYq70X/gLNqkpOMbt/wDgSf4dDP8AL8Pf9AX/AMpEn/xunt/YTqqvpLMq8KDpMhA+n7urvnav/wA+Nl/4Gv8A/GqPO1f/AJ8bL/wNf/41V2X9Ixu/6ZQWPw/uG3RsHPB/smQf+06272RWQ2V6GnidN5j2GWMgDvgFc8dOvT1FUxNq2ebKyA9rx/8A41VwmTyZSqqZQ+IlLfKy+pOOD14wfrRZdvwC7/plK6ksIruC5ntHkuIuYZUsnkaPDk8MFJX5sn8c980t9rNtqcAg1KK+u4g24R3FhPIoPrgp15NS3El8pH2W2t5BznzLhkxycdEPbB/Ejtkxedq//PjZf+Br/wDxqplTg5czSv6FOctdX95ROpWWlW0kmj6TcibgrHBp8kW49udg6Vw2sTeL/EE7HU4tVmRn3LC6SeWh6DCngfWvR/O1f/nxsv8AwNf/AONUedq//PjZf+Br/wDxqsauHhVacunkXTqygnbr5mD4b0CDwraNquqSKl0ozvz/AKoeg9/es691jUvGV6sl3PMNKt3zFA7kiQ+pB6/0/OrniHw94i1+YeZPYw26/dhWVyAfUnZzTLfw54jtreOGN9K2xqFGWkycfhWU6KlJLk0XlqXGo0m+bVjbi0iudhfcrxndHJGxVkPqCOlNltZrvYupalqGoRodyRXdy0iKfXBqx/YfiX+/pP5yf4Uf2H4l/v6T+cn+FOVGMpKUoXa8gVRpWUiG4tIrnYX3K8Z3RyRsVZD6gjpTZbWa72LqWpahqEaHckV3ctIin1wasf2H4l/v6T+cn+FH9h+Jf7+k/nJ/hRKjGUlKULteQKo0rKRDcWkVzsL7leM7o5I2Ksh9QR0pstrNd7F1LUtQ1CNDuSK7uWkRT64NWP7D8S/39J/OT/Cj+w/Ev9/Sfzk/wolRjKSlKF2vIFUaVlIhuLSK52F9yvGd0ciMVZD6gjpU1752pQ28WpXd1dx2/KJPOzgnGMkE8mj+w/Ev9/Sfzk/wqW20HXmnUXdxp0UX8TRJI7D8CVz+dEqUZS5nDX0BTaVlIqXFpFc7C+5XjO6OSNirIfUEdKbLazXexdS1LUNQjQ7kiu7lpEU+uDW7N4emjsbg29/FcXRQCFZLUxIrblySRIxI27hjHUiqEeg+IHjKGTS1lJ+Vi8m0+x44+uf8aHRjOSlKGq8g9o4qyloU7i0iudhfcrxndHJGxVkPqCOlFp4o1bRbsW2t31zf6bIw2SzSF2iP41O2g+J1YqzaUCDggmXj9KqajoniBdPla5g0+6iVctFCX3ke2e9KdJOSqcuq62HGenLfRm9ruhw69ax3VlN5V3GA9vcxtj3HIrgtTg8XXmbXU/7XvY424WRpZUyOMjORUvhnxdNoRNtdq81pn7g+8h9s/wAq6f8A4WRpH/Pte/8AftP/AIqpnHD12qktJDi6tL3VqjhItG1uGVZYdN1CORDuV0gcFT6g4q5er4t1KFYtRXWruNTlUnErqD6gGuv/AOFkaR/z7Xv/AH7T/wCKo/4WRpH/AD7Xv/ftP/iqh4fDNpuWpXtayVuU4SLRtbhlWWHTdQjkQ7ldIHBU+oOKuXq+LdShWLUV1q7jU5VJxK6g+oBrr/8AhZGkf8+17/37T/4qj/hZGkf8+17/AN+0/wDiqHh8M2m5ah7WslblOEi0bW4ZVlh03UI5EO5XSBwVPqDirl6vi3UoVi1Fdau41OVScSuoPqAa6/8A4WRpH/Pte/8AftP/AIqj/hZGkf8APte/9+0/+KoeHwzablqHtayVuU4SLRtbhlWWHTdQjkQ7ldIHBU+oOKu3g8XahGkd+NaukRgyrP5rhT6gHoa67/hZGkf8+17/AN+0/wDiqP8AhZGkf8+17/37T/4qh4fDN3ctQ9rWStyi6X4cbUdLgn1RXhluVSW6G3Y8kiSTqdw7EoyZOORz1JNXfEPiG18N2CwwKpuCuIoV6KPU+1Ral4xgt/D6ahawSs06K0SyADG55UGcH1hb8x748wu7ue+unuLqQySucsxrWVWFGCjT1fcqUatXldVWSWn3v9bi3N7cXl3Jc3ErNNJnc2fXjH0xUFS2sDXN5DAiyO0rhAsKb3OTj5V7n0Heuk17wTdeHdLF5eJcyrcIjwFYGQQqx/5b5GFftsBPJPPA3ec5a6myWmgnxDZl8d3hViD5MA4PY26A1y9dx4ns7a++J17FfCUwJZCdlhcIzeXZCQAMQQMlQM4Nc7d6ZaLozalZzs8bTRxpG7DfGSrl1cDuCq4YYBBzgHIFTfvs6MUv38/V/mafimaSKx0/ypGTfBErbWI3A2VpkH2rlq6fxZ/x46Z/1yh/9IrSuYpy3XovyDEfH8o/+koUMyhgpIDDBAPUdf6UlFFSc53fgf8A5Ak3/Xw3/oK1yGr865fY/wCfmT/0I11/gf8A5Ak3/Xw3/oK1p6J4RWLV7nVNSUNI07vBGeQo3Ehj7/yrkwdKVXFTSPoMzqRhleHucAmi6xtPl6bfbXGDtgfDDr6e1H9g6v8A9Aq9/wDAd/8ACvSdT8baTpd41tJ508ife8hQQp9Mkiqf/CyNI/59r3/v2n/xVey6FFOzmfKqrUevKcGND1lQwXTL4BhggW78jr6e1XdI0XVIr2RpNNvEBtbhQWgYcmFwB07kgV1//CyNI/59r3/v2n/xVTWvj7S7uZo44LwFY3kO5F6IhY/xeimnGlRUk1M1o1a0akXGF3dDLGLV7G9W5a4+x6f9itoyJFZW3i2gD7ZN8flYIQMTLHuKhfnI21ieM49RuoBcpcR3NkZI94iUly5DhGeTfJ5v3ZApEsm3BX5Pu11mv39v4q8MpoNi7R3eIbjDDcQrJ5q/u1zI2RKQdivtKc4B3VS02bRPCPh+z0zWA11qB1DzZFjQq8MQMUhOyTayqfJjHzhTjJGVwW+UqNxxTcFdp2S01Xf+teux6UdaXvOytv8Ap/X5jrI6lbWTWWoahbm+ld0dACoeTJRhJGJ4/tDbgQzCGbfjGZMbaLGKePSoWuzvmkadmlVdqSAwpsKDA2r5ezC4GBgYGKz/ABB4RsdXvoL7QtW+06fFawRS3K28jAeXEqH5gpROFyfNePGfm2rhj0V9qQ1S3t5diqVa6zsYMvzqJCAR97a0jJuHDbdw4IrryqyxK5db76W5f62IraxV9NVbzOQ8Qk/2Bcjt8v8A6EK6PwR/yJtj/wBtP/RjVzfiH/kA3H/Af/QhXSeCP+RNsf8Atp/6MavpqX8f5fqeNU/h/MxfEzMPH1goJ2m1yRngn95/ias1V8Tf8lB0/wD69P8A45Vql9qXqV9legZOMdqg1wt/wh9/w23anR8rnzU6r2PvU9V9dI/4RK+HG7Yh5TBx5qd+4+tEvhfoxLdHCQX13bIUtrmaJSclY5CoJ9eKKgoryjuL+g/8jHpv/X3F/wChivQrksshXlcgbl2bM/Velee6D/yMem/9fcX/AKGK9BnXLZQZRQOVbcoz/L6Hmu/D/Azmq/EjA0n/AJKhF+P/AKJNeg39r9t025td+zz4mj3Yzt3AjOPxrz7Sf+SoRfj/AOiTXod3cpZ2U91KGKQRtIwUckAZOPyrqw9uWV+7MK1+ZW7HEf8ACsf+ov8A+S3/ANnWrp3g5rCzFuNQ3qBc5/c4yZoPKB+9/Dkn36cdah/4WRpH/Pte/wDftP8A4qtCz8XWN9bCeGC6EZE5yyLx5UXmsPvd16e/p1qFHC2f/BNoSxST5e2u211+tjL0zwB/Z128/wDaXmbreeDHkYx5kTR5+923598V1Hg/4aWKaTL9oufNkadt8n2ZH3KFXauyXegwS5zt3fNgEDIOJY+OtM1C4aGGC7VlhlmO9FA2xxtI38XXCnHvXe/DvX7TXtCnmsxImy5ZSkuA3Cqc4BPHNfOcRTp0sC5UHaV19x3Zaqk66VTbUw7DRPBmq6tJpen6xFPeRhiY10yy7HBwfs2Dj2Nc/D4GsrnxUN0KbL15kt7NAywebBI8ckjAHIQbQwTIOZQoOATXYeG/hha+HPFj63FqMs/D+XA0YG0tnq2eeD6CqFhq0aeKbMREO1pJeytGvLTRXN1IwKerBY0cL1ZHJHTn5GOJalL6rNyVtXtrZ+S6pf59T2XTVl7WKTv/AJf8Eo2vh/wxq94dKstYsNRvY8me2/syCBZVXhgskcaspzyCGPuCM55dfBWo6P4iufskMp06azuRCZXQOGe3kRUdc/eDsUJ6dT0rpPDWk+FdD8Uf23pPiIarPKJPs2nwoPMJYEnPPAC55IAGOT2rPvfG91eeIJrOJbS4S1tbl5bmAMY3lWOWbC8/d3Dbn+IKSMZ493K5L6y4OUnCy3Vtdb6WXW347204MUn7NNJXv3v+pzWheEtbs9QllubLYjWd1ED5qH5nt5EUcHuzAfjTb/wfrl7fQBbfbCIYELNKuEIiRWOM56r264q9o/jrU9QvpIZoLRVW1uJgURgd0cLyL/F0yoz7U+bxjrrahBa2NlbSGaKJlzG/LNGrtzu6At+WK+sUcPbRvp+p5yliPZPRWuv1/wCCbVpaWfhPTUt7WP7Rez8BR96Vv6KKmaSPRLSTUtan826fOAOQuf4EH5fXvTB5WhWsmqa1MJr2QYJA6eiIPSoLGxlvrj+2/EOI1jG63t2Pywr/AHj7127WSX/A82cG+r/4cLGxlvrj+2/EOI1jG63t2Pywr/ePvVyCa/1iYzwTvY2AGIysamSY5+98wIC+nFMRH1+UT3AMemRnMcbcGcj+Jv8AZ9q5jxf4uaZW0/SWK2+Skk68byMZVT6dPzqZTjTjd/8ABY1Fzdl/wx2n9nXX/QZvf++IP/jdH9nXX/QZvf8AviD/AON14pnPWl2sFDYO0kgHHBI//WK5vri/l/E2+rvv+B7nHC6RqrXMshHV2C5P5KB+lVv7Ouv+gze/98Qf/G653whG0vhy0VBknf8A+htUq+EPDOmWsa6q0bSEcyz3Bj3nvgbgKWFxUq7naFkna7k/8jux+CjhadJuom5K9lFJrbfXX/gG7/Z11/0Gb3/viD/43R/Z11/0Gb3/AL4g/wDjdYH9jeB/+etl/wCB5/8Ai6c2g+C0VWc2iq3Kk3zAH6fPXZd+X/gT/wAjyrL+kbv9nXX/AEGb3/viD/43QNPuQedYvT7FIf8A43WB/Y3gf/nrZf8Agef/AIuum0iS3023sI9HujHARJ9ljjuCySgFi5AyQ+MtknOMdsDBzS8v/An/AJBZef3f8EbcWs0xHl39xBjORGsZzyT/ABIfXH4Dvk1F/Z11/wBBm9/74g/+N1yHj7RLiSayuLG3kljVGibZ8x3vM8nQDuZCBXI3uiarpsKy6jpl5aRscK89u6KT6AkVzVcYqc+SSs/U3VDnXNB3Xoeu/wBnXX/QZvf++IP/AI3R/Z11/wBBm9/74g/+N14tFFJNKsUKNJI52qiDJY+gFXL3RNV02FZdR0y8tI2OFee3dFJ9ASKzeOgnZrX1H9Wk1e/4Hrv9nXX/AEGb3/viD/43R/Z11/0Gb3/viD/43Xi0UUk0qxQo0kjnaqIMlj6AVcvdE1XTYVl1HTLy0jY4V57d0Un0BIoeOgnZrX1D6tJq9/wPXf7Ouv8AoM3v/fEH/wAbo/s66/6DN7/3xB/8brxaKKSaVYoUaSRztVEGSx9AKuXuiarpsKy6jpl5aRscK89u6KT6AkUPHQTs1r6h9Wk1e/4Hrv8AZ11/0Gb3/viD/wCN0f2ddf8AQZvf++IP/jdeLRRSTSrFCjSSOdqogyWPoBVy90TVdNhWXUdMvLSNjhXnt3RSfQEih46CdmtfUPq0mr3/AAPXf7Ouv+gze/8AfEH/AMbo/s66/wCgze/98Qf/ABuvFoopJpVihRpJHO1UQZLH0Aq5e6JqumwrLqOmXlpGxwrz27opPoCRQ8dBOzWvqH1aTV7/AIHrv9nXX/QZvf8AviD/AON0f2ddf9Bm9/74g/8AjdeLRRSTSrFCjSSOdqogyWPoBVy90TVdNhWXUdMvLSNjhXnt3RSfQEih46CdmtfUPq0mr3/A9gjsp0Vw2pXUhZcAssWUOc5GEHPbnPWmf2ddf9Bm9/74g/8AjdeLRRSTSrFCjSSOdqogyWPoBVy90TVdNhWXUdMvLSNjhXnt3RSfQEih46CaTWvqH1aTV7/gewPp93PAYxrl1FMBiOWSKFl9gwEeSP8A9XpjkbfxPq+heIG07xUQ6HA8wIo2g9GBUcrXI+H769sdat207c0ruIxGAT5mTjGB1r0P4k2LL4etpL61kiK3IjjnaIgoCrHGccqSOn4juDbxMZK6laS6X3M/Yyg7Wun5bGR4z8MRTWzazpYUjbvlROQ4P8Q/ma4JmZ2LOSzMckk5JNd54M1+4huodDvw0olwtqyfPnPRcjqP5dKr+PfB8mi7dThgeC2mkEbIyEbXIJ49uDWFd05e/FpN9OpvScl7r1Xc4tWZGDISrKcgg4INJToopJpVihRpJHO1UQZLH0Aq5e6JqumwrLqOmXlpGxwrz27opPoCRXI5xTs3qdFm1cpKzIwZCVZTkEHBBpKdFFJNKsUKNJI52qiDJY+gFXL3RNV02FZdR0y8tI2OFee3dFJ9ASKHOKdm9Qs2rlJWZGDISrKcgg4INavhzQZde1JYRuS3XmaUD7o9PqazbeCW6uI4LdGkllYIiKMlieAAK9hmso/BPhd5jZzJFCBkshBkcnHJPua3oqnKdpyS/UxqOSj7qMdPhzpSMrLdXyspyGEiDB/75pP+Fb6R/wA/N7/38T/4muLn8R65qV+TDeXSvK+Eht5GA56AAdalvX8W6bCs2otrVpGxwrzmVFJ9MmtpYjDKXLymapVmr3PQZvCdjPottphmuUitxhXRwHYbpGwTjH/LZ+3TH45v/Ct9I/5+b3/v4n/xNZV7d6q3gjTrqC7u/NeNCzpK25iJroHJHJ4Cfgo9BXL/ANu6wOuq33/gQ/8AjWjq0bL3P6ua1KNdRi3Pdafezv0+Helxtuju79DgjKyqOCMEfd9Dinw/D7R4op033MnnRhNzspMfzK25fl4Py4+hNefprOtytiLUr9zjOFnc/wBab/busD/mKX3/AIEP/jU+2ofyGXs6v8x6vNoFpP4mfXJC7XLReUY2CtFjyfK+6ynPy885Ge2OKp3ngnRbu3WNIGtmyGkkgIDStjqcghep4QKPbgY5jxzqGpweNLu3sLu7jjWKBhFBKwAzChJwPck1zf8AbusA/wDIUvs/9fD/AONOVWhzO8DWtSrxqSjKd3dnqN/4XtNQt7eJ7i6hWCEQ5hkCl1EaR4bjniJf1rL/AOFb6R/z83v/AH8T/wCJrL8ST6wmnadLZ3N4N8ETuYZHyQbW3JJx/tFzn1YnvXLf27rA66rff+BD/wCNDq0V9jsFajXjL3p30X5I74fDnSVDBbq+AYYIEicjr/d9qT/hW+kf8/N7/wB/E/8Aia4RNZ1uQMY9R1BgoyxWdzgfnTf7e1f/AKCt7/4EP/jS9tQ/kMvZ1f5j1bRvD1nokDRWrzSKXL/vWBwcAdgPSuf8ZeLvse/TdMf9+RiWUH7nsPem+D7+8utIle5u55nE5AaSQsQNq8c1i2Xhq41/xLeu+Y7RLlxJL64Y8D3rCni4znKjQhZ9T0sTgalPDU8TXqcyeiXYp+G/Dl1r1yxLvFadJpf73fA9TkCuv/4VvpHe5vf++0/+JrW1C/svC2i4iiwsSfu4kHJ5xk+2SOfeuLlPifUpGujqMlqJDlYVndAo7DC12ezpUlytczPK5pzd07I3x8OdJUMFur4BhggSJyOv932qW18A6XaTNJHPeEtG8Z3OvR0Kn+H0Y1zH2DxMc41qbjr/AKVLVzSrTXUvibzVppYvJmyouZDk+W23r05xz2604unzL92XTVTnVp21Oo1qxh8KaFHrlizi5dYbRneRkBQJsQFkw4wLfPyFSS5zlRiq+l30HibQC809vp1xcXgjEtxYpemTaYo2JaYlsEzw8MzbRG2Dg4Fua21dpLe2uHhu7N7G3nFrHFczNEWiVGY4gljbe8JP7xXPGRtPNYHjHTtSt9NuEU29nb2oWZ4RFOjyhJAgKkwRxYVpycRqoO/J3Hmvj604yxLjDR82/by/r021PVpwl7JSnqrbfr/X5knibxha6XrrWmh6akWlzwwzNDG7wAiSJWx5aERNlSM+YkmTkHK4Ub9zp62Nhb7GB3TXsZCoFUGICJiFHChmjZ9o4XftHAFVtL07W9S08391pdqb5GZyfNAO8sSxLmFzC2/cWVJYthyQIycmslr4igsRDdWUQmV7kLEsgKRILeJIAp3ZKnZtySTgAk85PXlcZfWI8qtbfW/N/T1uFVxdlJ7tWvpYw/EJP9gXI7fL/wChCuj8Ef8AIm2P/bT/ANGNXL3mk+LL60e3l0uFUfGSsqZ4Of7/ALV2HhaxuNN8NWtpex+XNHv3LuBxl2I5HHQivp6Kbq3s9v1PEqNeztfqc74mZh4+sFBO02uSM8E/vP8AE1Zqt4mVj4+sGAO0WuCccA/vP8DVml9uXqV9legZOMdqg1w/8UffgHjamQJOM+an8Pr79KnwcZxxUGu5/wCEQvic/cXGVx/y1Toe4/z60S+F+jEt0cHBfXdshS2uZolJyVjkKgn14oogsru5QvbW00qg4LRxlgD6cUV5Wh3aljQf+Rj03/r7i/8AQxXoN1u875yxbHO5NrfiPX3rz7Qf+Rj03/r7i/8AQxXoM6ktlVGxQOUYso/PkfQ134f4Gc1X4kYGk/8AJUIvx/8ARJrute/5FzUv+vSX/wBANcLpP/JUIvx/9Emu617/AJFzUv8Ar0l/9ANdFD+HP1ZhV+OPojxSuo8PMRpsS9jHqhP4WX/165eup8PRk6VHJ2VNUU/jZf8A1q8uPwy9P1R7FC/LO3b/ANuRm+Gf+QtN/wBg+9/9JZa6b4Z6Zc3F75kN3cwtcO0VvFDO8aOyKGd5ChB2oHT5cgsXAB61zPhn/kLTf9g+9/8ASWWuy+FWpiC5hVRl7OaZmjAy0yTJGDs/2lMIbb1ZS2Pu4PlZm5rDT5O3+f62/rUeGs6iudtBfDV7ltK0/wAcTalfxAma1MSwLMF4IWSNVZTnnIY+4IznitO8HJrPihp9QkutQi1DcNPa9kYtJsG2QzEHdiIgJjK7jjacc1t+GtJ8K6H4o/tvSfEQ1WeUSfZtPhQeYSwJOeeAFzyQAMcntTvDOvo3iK3topIrltHa4WSSAlhd/aX8yTyuOsbLgDkuoZhjGK+di50FP6te1t2uVrfsl1UV6N9rr0XafL7Tv3v29el/6302az8UNJoDeMTrUwObmxltkijnVCMgOiKykMMghj0GQRnPCL4Iv9H8SXD2tvI1hLZXKwNM6CQGSCRAjDP3g52k9OM9K6bw1pPhXQ/FH9t6T4iGqzyiT7Np8KDzCWBJzzwAueSABjk9qz73xvdXniCaziW0uEtbW5eW5gDGN5VjlmwvP3dw25/iCkjGePUylQhiHTTfJa+3LrrfSy8lf1WttOTF8zpqVlzet/19TnPD3hLWLTVmkvbTyoWtLmIt5qHl4HReh7lgK6oQad4Zshf3QH2jyI4SRyWKoFCr+VYOieN9V1TUHtntrXi2uJRsRs7khd1H3um5QD7VvR6abm6i1fXJExFAjJD0SFtoLnB9819pQVPX2Wr8+m55E5VfZNTaUbrTq9/6/pEFjYy31x/bfiHEaxjdb27H5YV/vH3qyiPr8onuAY9MjOY424M5H8Tf7PtSRhvEMguJ8x6XGcxxtx55H8Tf7PoO9cx4v8WtOrafpRK2+Skky8ByOqg+nIrSUo043e35nPGMpSsv+GE8X+L/ALRv03SX2wj5ZJV/i9h7VxVFLtYKGIO0kgHHBP8AkivMqVJVJc0jthBQVkJRRXbfCbwdb+NPH1rYakG+wRo9xOoJHmKmPkB9yy574JrJuyuWld2Oh8CAf8IlbHHO5+f+BGvNdW1W41jUHurtsseFUdFXsBX2lHe6ZoSrpmm6dFBaw/J5cKBFX1woFeF/tAeBNJ0Maf4i0G3is472Qw3EES7UL7dyuq9sgHOPb3rCjmNHEL2FN6xvc2rYecZe1l1S/BWPFKKKUqyhSwIDDIJHUdP6V0GIldlbyXVj4FiAm3JdwPMhF/bQvEA8ibQjqZWH3z8jAHzGHds8bXV6niz097KHXdHubWFHW2jeyMk3ll2baJfI65Lch8Ak4NSxo9X8I6laSC4mu2YSW0t3vcfwrA6+n+zKnv8AKeuSKu6X4p8N/Ei3vtHSCeSJYw0kdwm3cMjBUgnocelcb4KuYG1q9ikEN1FcXV0ojhu0Z5IbjZucIGz8nkrlepEhI+5g9b4f8P8AhTwRHdarp18XjlVUkmluUZY13eowAMg9epGBzxXx+dRlLFVJVeZ1PsW23/yt/wAOfRwpyhJqEUotu/n/AF/Whzvg3wZoegatJeS6rNeyW6faY5o4TCFRI0dv4iSCJ4zjg/KQcgkHp9L8U+G/iRb32jpBPJEsYaSO4TbuGRgqQT0OPSuR8I6ml9rdx9ptrcxzPLAtpb30crC3lSJMgKxJ8sW6AjqRISPuYPV6D4b8L+BFu9TsrtykgWOWWadWWJc+vAAzzzycYHOBWGPo1eeUq6l7Wy5bLS/y+X/DmdGMo2hFWWt09/xOb8G+DND0DVpLyXVZr2S3T7THNHCYQqJGjt/ESQRPGccH5SDkEg9Ppfinw38SLe+0dIJ5IljDSR3CbdwyMFSCehx6VyPhHU0vtbuPtNtbmOZ5YFtLe+jlYW8qRJkBWJPli3QEdSJCR9zB6vQfDfhfwIt3qdlduUkCxyyzTqyxLn14AGeeeTjA5wKMfRq88pV1L2tly2Wl/l8v+HCjGUbQirLW6e/4nN+DfBmh6Bq0l5Lqs17Jbp9pjmjhMIVEjR2/iJIInjOOD8pByCQen0vxT4b+JFvfaOkE8kSxhpI7hNu4ZGCpBPQ49K5PwfqH9oazdGe0haKRpIfsttepKy20qRJuwrEnyxboCOpEhI+7g9ToPhzwv4EF3qdjeMVkCxyyTXCssS59eABkZ55OMDnijH0KvPKVdT9rZctlpe/l/XzFRTi1TiklrdPf8Tm/BvgzQ9A1aS8l1Wa9kt0+0xzRwmEKiRo7fxEkETxnHB+Ug5BIPT6X4p8N/Ei3vtHSCeSJYw0kdwm3cMjBUgnocelc94LeS/v724v7IRWk4lihhgnEweGRYkwCpP3BbqD3O8njbzu6D4e8K+BWvNQsb0neqJM81wjCJCePTAJB69cYHPFPMKFVScq6l7Wy5bbX+WwqG/s4WtrddTnPBvgzQ9A1aS8l1Wa9kt0+0xzRwmEKiRo7fxEkETxnHB+Ug5BIPT6X4p8N/Ei3vtHSCeSJYw0kdwm3cMjBUgnocelcr4VvGudamN9YxpFcNJDHaw3iSMbeVIkyAGJOwW6AgckSEj7mD0ugeH/Cngc3mo6ffbgyokzy3KMIVJyM4xgHrz1xgdhU5hRqKbddS9rZctu/y+WxdGnOKUYxtHW97/qc74N8GaHoGrSXkuqzXslun2mOaOEwhUSNHb+IkgieM44PykHIJB6fS/FPhv4kW99o6QTyRLGGkjuE27hkYKkE9Dj0rlfCl411rMxvbKJYrlpIY7WG8SRjbTJEm5QGJbYLdAcckSEj7mD0nh/QfCfghr6/07UN/CRXDy3KMIAT3xjbkjPPJxgc8UZhRqKbddS9rZctu/y+W35hRpzilGMbRd73/wCCc94N8GaHoGrSXkuqzXslun2mOaOEwhUSNHb+IkgieM44PykHIJB6fS/FPhv4kW99o6QTyRLGGkjuE27hkYKkE9Dj0rlfCl411rMxvbKJYrlpIY7WG8SRjbTJEm5QGJbYLdAcckSEj7mD02g+HvCvgVr3ULG9J3BI53muEYQqTxnpgHGeeTjA54p4+jV55Oupe1suWy0v8vl/w4UoTilGMbRd73/4Jzvg3wbofh/VZbyXVZr2SBPtMcyQ+SFRER2/iJIInjOOD8pByCQen0jxX4c+IsN9pEcE0kYiBkjuIwu5T3XBPIOPQg1ynhS8a61mY3tlEsVy0kMdrDeJIxtpkiTcoDEtsFugOOSJCR9zB6Tw/oPhPwQ19f6dqG/hIrh5blGEAJ74xtyRnnk4wOeKWYUaim3XUva2XLbv8vlsFGnOKUYxtHW97/qYngjwnonhzU5ri41SW/8AJjNxHMtv5QiREjdiPmJ5E0fA252kHIJB6XTfE3h/4gRXukLBNLF5IZ1njAEiE43KQT0P0INcj4UvGutZmN7ZRLFctJDHaw3iSMbaZIk3KAxLbBboDjkiQkfcweo8O6P4V8DLf3mn6h50Z2x3AmukZYAT6jG3kdTycYGehePoz539YUva2XLZaX+Xy2JpU6kLKEbRd7p319LnNeDfBmh6Bq0l5Lqs17Jbp9pjmjhMIVEjR2/iJIInjOOD8pByCQen0vxT4b+JFvfaOkE8kSxhpI7hNu4ZGCpBPQ49K5fwxdefrUrXdpD5F0ZI4raC9SRvs0yRJkANk+WLdQR1YOWH3eel0Hw34X8CLd6nZXblJAscss06ssS59eABnnnk4wOcCjMKFVTlKupe105bLS/y+Ww6MZRSjFWi773v+Jzfg3wZoegatJeS6rNeyW6faY5o4TCFRI0dv4iSCJ4zjg/KQcgkHp9L8U+G/iRb32jpBPJEsYaSO4TbuGRgqQT0OPSuR8I6ml9rdx9ptrcpM8kC2lvfRysLeVIkyAGJPli3QEdSJCR9zB6vQfDfhfwIt3qdlduUkCxyyzTqyxLn14AGeeeTjA5wKMfRq88pV1L2unLZaX+Xy/4cdKEo2jFWi73vv+Jzng3wbofh/VZbyXVZr2SBPtMcyQ+SFRER2/iJIInjOOD8pByCQeo0jxZ4d+IkF/pEUM0kflAyx3EYUOpPUYJ6HHoQcVx/hHU0vtbuPtNtblJnkgW0t76OVhbypEmQAxJ8sW6AjqRISPuYPTeG9F8H+Dft2paVqazKQkU8j3SOsCk5GSMBc4zzycYHpSzCjU5266l7Wy5bbX+Xy2/MKMJxSjGNo63vf9TA8G+DND0DVpLyXVZr2S3T7THNHCYQqJGjt/ESQRPGccH5SDkEg9f4d8b6H42ku7GwSfMSBnWeMLuUnqME9Dj0rifCOppfa3cfaba3KTPJAtpb30crC3lSJMgBiT5Yt0BHUiQkfcwem8N6L4P8G/btS0rU1mUhIp5HukdYFJyMkYC5xnnk4wPSjH0qim3XUva6ctu/y+W35hSpzilGMbRd73/4Jh6nr2kaZcTJNfApEYsuInPEib07d159qz/+E38Pf9BD/wAgyf8AxNcX4guXvo9RuGW3jNxdQGG3guUmKRRxSKPusTgLtyTXMV+i08XXjSgqi96yv6nzuIwVOnJWur9Pm/0set/8Jv4e/wCgh/5Bk/8AiaP+E38Pf9BD/wAgyf8AxNeSqrMcKCxwTgDsOTSVX12p2Rz/AFaB6/ceMNCtbmW3nvtksTlHXyXOGBwRkLTrrxZolnMsVze7HaNJQPKc/K6B1PA7qwP41x2meE5fFvjnV4PtC2lrbzTTXFwwyEUOe3r/AIGuj1vwp4f8SLdan4N1j7Zd6dbxGWzljO10jjVRtyB2QeoJ44rjxGdQoVnSl030dlfa76XOueWpSko7JtLVXdi43jDQltkuGvv3TuyK3kvyygEj7v8AtD86j/4Tfw9/0EP/ACDJ/wDE1j2ltZ3+hWssdlCULu+3ygQpwgPb26+1N/s2x/58rf8A79L/AIV6zq1eljhnSpp2V+n5G1/wm/h7/oIf+QZP/iaP+E38Pf8AQQ/8gyf/ABNYv9mWRziyt+Ov7pf8KP7Nsf8Anyt/+/S/4Uva1fL8SPZ0/M6mx1zTtShaayuPNRW2k7GHOAe496zdR8baRZLMkMzT3UbFBCsbDLA4xkjFSaLbQQ2brDDHGvmE4VAOwrKuLG0F/LILWEP5jHd5YznPXNYxrYhzcXa3o/8AM7auHw0KEJx5uZ73at8tDNtba51G+/tXWDumbmKM9Ix24rVowTnA6daK0Sscjdwqa1/1zf8AXOT/ANANQ4JzgdOtSW7rHNmTO0qynaMkZUj+tWtxw0kjU1GT/hI/CH9n6PeQw6n9nh3JLMsTSRRS3CkAk8hcFsdhg96o2Zj0Dwva2muy2t3JY7p7mCKdZ3gi+22Z8sgMV+YBjg/40yz2WEryaff6haln8x1gdkVm9SofB/EVp3PiK7vbFrO9vnubZxhop7OKQNzkZ3Zzzz+FfOVMprzqN3XK5c3ntbserDExjFaa2t0t+Zzup6ZrV/4r0yQalEPLS3jSSa8EbSMgCtLGrENKruGdXQNu3/3sgd7q+qWJvP8Aj5t4wAy48xRjaN5U89URlyO30rm49Uure1lt7LUXtLSTO+3g022WM5GDkbec471mTWlvOqCS5kKRJMscUVpBAoMsexmPlqMnAHJz0FdmCwdbD1VOdmlorfmzGrKnVXK3ZN67HR/29pH/AEFbL/wIT/GrcE8NzCs1tKk0TfdeNgynt1FeWar4etbHS5bmF5mZNuNxBHJA9K7jwR/yJ1j/ANtP/RjV7NKtKU+WS6HlTpxjHmTMTxN/yUHT/wDr0/8AjlWqreJlY+PrBgDtFrgnHAP7z/A1ZrL7UvU1+yvQKg1z/kUb/G37qZw/P+tTqv8AWp8HGccVBrpP/CI32Q33FxlBg/vU6N1/CiXwv0Ylujzyip4LK7uUL21tNKoOC0cZYA+nFFeVc7ixoP8AyMem/wDX3F/6GK9BujmbncTgZ3AZ/Mdfr3rz7Qf+Rj03/r7i/wDQxXoM65OUAKKByhJUc+/I/Gu/D/Azmq/EjA0n/kqEX4/+iTXda9/yLmpf9ekv/oBrhdJ/5KhF+P8A6JNd1r3/ACLmpf8AXpL/AOgGuih/Dn6swq/HH0R4pXaeD7M3lrbxOzpE5v0ZkTcRvtlQHGQOrdyB7iuW0zTbjVr9LW0Tc7Hk9lHqa9d0jS4PD+i+QjMyRK0sr4znAyxx6YH5CuPD0faXvojv+sKipK121b8U/wBDlbLwe2mSzz241K4la1uIUR7eCNSZIXjBLfaCQAWz0PSpfB/w1ur26kbXLI7TlLeMyAoWwSXco27avy8AgkuoyBkiOX4mRrKwh0tnTPDNPtJH02n+ddb4F8frqcrxmw8poX5USF2lVxzs4ALDywdvUruI+7g+dmvJSwk5UJPm/ry72/4G6rCc86qVRK39eZdla18TiXQB4wOt3A+a4sJrdIo5whyQHRFZSGGQQx6DIIzni9D8BWl1r+Hjlns7zeNNScld2ziUy7SGAib5SPl3kjBxk10PhfR/C2i+K11jSPEP9rXE/mC20+BF81mYE4PPGF7sFA7kV0Wi63YS6xqElnpt5Ja6Z5xhu4lLpciWUPPt+UA7XXgAkkA4Br5FVKtCM44W/La705bPXokutlftftdezyxm4ura9+9/8+l/63os1n4oaTQG8YnWpgc3NjLbJFHOqEZAdEVlIYZBDHoMgjOeEXwRf6P4kuHtbeRrCWyuVgaZ0EgMkEiBGGfvBztJ6cZ6V03hrSfCuh+KP7b0nxENVnlEn2bT4UHmEsCTnngBc8kADHJ7Vn3vje6vPEE1nEtpcJa2ty8tzAGMbyrHLNhefu7htz/EFJGM8eplKhDEOmm+S19uXXW+ll5K/qtbacmL5nTUrLm9b/r6lTwh4Wk0N31LUn8q4VGUIr8IpGCSR14/Q1jeOfEMt1dDT7V9tqESQsuf3m5Qwz7YPSpdO8Y6nq97PBceVFF/Z94SsKkbiLaQjOSfSub1pHbUITg4e1tgDjgnyI/8R+dfWzqx9m4U9tPnuefTo3pyqz3ukvx/yPQPHd1LYeF1S2cx+bKsLFf7u0kj9K8tr0z4kf8AIu2//X2v/oD15ptYKGIO0kgHHBP+SKWMf72xhh/gEoopdrBQxB2kkA44J/yRXIdAldp8J/F1r4M+IFrqWo5FlIjW87qCTGrfxY74IBPtmuLpdrBQxB2kkA44J/yRSaurAnZ3PtxLXStcT+09O1OGe1l+cyQurqfX5geK8M/aA8daTrh0/wAPaDcR3kdjIZZ54m3IGxtVFYcHAJyfp714rTmikSNJHjZUkzsYqQGx1we9cdDBUaE5Tpqzlub1K86kVF9BtFFKVZQpYEBhkEjqOn9K7TASuqk8XWk/hufS003+z3kAImsXVF+VSNpXaGdWz825265HTbXK0pVlClgQGGQSOo6f0pNXC9hYpZIZVlhdo5EO5XQ4Kn1Bq5e63qupQrDqOp3l3EpyqT3DuoPqATVGtXRPD95rV4kUSMkRG5pWX5QucZHryCPwoVNTktLsHLlW+ha8F2N3c+JbW4tAyrayCR5QcbQPf1rs/iXf3Mvh2CCS4mkElyvys5OcK3b8q0ANN8J6JgfJGn/fUrf1NVNM0y41S+XWNcXDDm1tT0iHqR616X1WCjytJy79ji9tJvmvZfmZ3gvwmbF4tV1FSLlSGgj/AOeZ7Mff09Kk8Y6td+JriHw9p881yyyiSdmkJRMAjn86m1zXLi9vf7E8P/PcNxNOOkQ781oadp1h4V0h3kcDA3TTt1c/57VToUpe6ktN3/kL2k17ze/QXS9PsfCWkM5kCFRulnPBYj/PArIEmpePb1ZtSmnTRbd8wwu5/en1P+Pbt3qKGG68a34ubsNBo8Lfu4uhmPqa19e1638P2aW1rGr3TDbDAo6e59qqUKc7Sklyx2EpTjonqxdd8Qw+HLWOGyUG8IAt4U42+h46CueWLUdTjEniLUbu+Yv5ggmnZo0Psucf0qLT9Pl89r/Un868l5JP8HsK0yCOoxWcoRqzU5rbbyLi3CPLF+pBcWkVzsL7leM7o5I2Ksh9QR0ps1rNeKqanqWoahEh3LFd3TSIp9cGrNBBHUYolSpykpSim0CnJKyZXmsoZljG0xtCQYniOxoyOm0jpSTWs14FXU9S1DUIkOViu7ppEU+uDVmggg4IwRRKlTlJSlFXQKckrJleayhmWMbTG0JBieI7GjI6bSOlTX3m6nDBDqV3dXkUH3EnnZwTjGSCeT/9enUrKyNtcFSOxGKJUqcpczirgpySsmVprKGZYxtMbQkGJ4jsaMjptI6Uk1rNeBV1PUtQ1CJDlYru6aRFPrg1ZoIIOCMEUSpU5SUpRV0CnJKyZXmsoZljG0xtCQYniOxoyOm0jpTZ7Oa92LqWqajfxRtuWG7uWkQH6GrVBBBwRgiiVGnKSlKKuhqckrJmJ4hY2VtZ3NqfJkt518po/lKYBIxjp0FdJ8S766l8MRRy3EsiNdLlWckfdY1zviaMy2FvGvV7hVH4g1t/Ej/kXIP+vtf/AEB6iVOElOTSurApSTikzze2uJLS6iuIHaOWJw6OhwVIOQQa9P8AiXfXUvhiKOW4lkRrpcqzkj7rGvLoozLMka9XYKPxr0r4kf8AIuQf9fa/+gPWNKnCVOcmk2jWpKSnFJnmsUskMqywu0ciHcrocFT6g1cvdb1XUoVi1HU7y7jU5VJ7h3UH1AJqkqs7BUBZmOAAMkmkrlcYt3a1N7tKw6KWSGVZYXaORDuV0OCp9QauXut6rqUKxajqd5dxqcqk9w7qD6gE1SVWdgqAszHAAGSTSUOMW7tahdpWCilVWY4UFjgnAHYcmkqhBRSqrMcKCxwTgDsOTSUAeoeE7iXSfF/iC31XT75dO1YyxG5jtnbblmw3APBDHke1bdh4csvh3Y6leadfy6ze3sBtreCGHO0nuxBOBnHJxxXiqqzHCgk4JwB2HJr1Txt/x46b/wBhGL+TVz1MspYupOq5NJ25lpr21tdfLc7KuPippxhu3u9r9tF+JTtLCTTNCtLWVWDozk7h1JCk49s5paUKzZ2gnAycDoKSve06HjzlzO4UUYJzgdOtU5UvtT1mw0HRx/p2oyCNHJwEBOM/TqSewFRKSirsUYuTsjotKIFq2T/Gf5Cs25/4+pf98/zr2fQfhT4S8K6Wtpqks2oXUh8yWaaV1LMQASFQjA475+teffFDwCvgbb4i0W5mudGupttxbytvNuzZIZW/unpz7cnPHmUcbCdaUdPv1+aPVxFKP1eHLe6300+TucrRQPmGV5GM5FFeqeSFFGCc4HTrRQAUUYJzgdOtFABRRg4zjiigDM8Q/wDIBuP+A/8AoQrpPBH/ACJtj/20/wDRjVzniEH+wLk44+X/ANCFdH4I/wCRNsf+2n/oxqKX8f5fqKp/D+ZieJv+Sg6f/wBen/xyrVVvEysfH1gwB2i1wTjgH95/gas0vtS9Svsr0CoNcI/4RG/AK52pkBjn/Wp1H9RU+DjOOKh10H/hD744bG1BnaCP9an8XUfT8aJfC/RiW6PO6Kngsru5QvbW00qg4LRxlgD6cUV5VzuLGg/8jHpv/X3F/wChivQbvd53zlycA5fBJ/EdfrXn2g/8jHpv/X3F/wChivQJlLMWUIVAGWjBx/8AW/Su/D/Azmq/EjB0n/kqEX4/+iTXda9/yLmpf9ekv/oBrhdJ/wCSoRfj/wCiTXda9/yLmpf9ekv/AKAa6KH8OfqzCr8cfRHH/DJQX1Ju4EQ/Pd/hW/fXL3WufZGOIY4byN0GfnBtk689f3p5rC+GP/MT/wC2X/s9bQgeTxPcuo+UG5QknABNtBjJPA6Gphrhkv63Oukk51X2j+sSgiLGgSNVRQMAKMAVf0HSNO8R35stSgMwMjR2y7mCPIqZkZipBAQOnGRuLAdiaj+wT7WIMLbVLELOjHAGTwD6Vo+AUlttbeQ422000qoBua4SZEDBR2ZDADjqwY4+7z5WbzqwwknR3/4H+dv+BuXg4wlWXOb9je22s6rHpkfjNtWubaUTS2ctrGiSmNgwAZEUgh1DcMenIIzmFfCcN8l1bQpby6dcQtZ21sweMxbJdzPJ8wbCuoGF6kjoDWV4T0TwrpPi9NW0nxINTnn8wW1hCgMmWBJzz0C55IUDHJrS0rxVf/8ACSCznnNylk0gmdAxW9807yIsgcxEbQMksgcjGMV8Wp16EakcI3y2Td1y2evZLrZX7X6K69vlpzcXVte/e/6+v9bjNZ+KGk0BvGJ1qYHNzYy2yRRzqhGQHRFZSGGQQx6DIIznhF8EX+j+JLh7W3kawlsrlYGmdBIDJBIgRhn7wc7SenGeldN4a0nwrofij+29J8RDVZ5RJ9m0+FB5hLAk554AXPJAAxye1Z9743urzxBNZxLaXCWtrcvLcwBjG8qxyzYXn7u4bc/xBSRjPHqZSoQxDppvktfbl11vpZeSv6rW2nLi+Z01Ky5vW/6+pzWheEtbs9QllubLYjWd1ED5qH5nt5EUcHuzAfjWxp3g7dqsV9qygJDbwKIiwOXSJVOcdgR+P0qno/jrU9QvpIZoLRVW1uJgURgd0cLyL/F0yoz7VQ8SeK9UuIVtMRwwSwxOzRKQWLRK7Lknplv5V9fF4eCbV3t+p5v+0Oi1pa69ev8AwRfHPiOHVJUsLPDwwPuaXP3mxjj2GTXI0Uu1goYg7SSAccE/5IrkqTdSXMwjFRVkJRRS7WChiDtJIBxwT/kioLEoopdrBQxB2kkA44J/yRQAlFFOaKRI0keNlSTOxipAbHXB70ANoopSrKFLAgMMgkdR0/pQAlFFXdK0yXVNUtrND5ZuDwzDjAzk+/Q00m3ZCbsrk2haFc67fCG3G2NeZJSOEH+Nengab4S0TAGyNf8AvqVv6mlA03wloRIxHFGOf70jf1NZmgwP4inXXdTw0YYi0t/4YwDjJ9TxXqUqape6vif4HDObqavYn0zTLjVL1dX1xcEc21qekQ9T71BrmuXF7e/2J4f+e4biacdIh35qDxR4nYXS6LpDj7TM4ieXPCEnGB781saZptj4W0d3kcAqu+edurn/AD2q924RfqydrSl8kJp2nWHhXSHeRwMDdNO3Vz/ntWJDDdeNb8XN2Gg0eFv3cXQzH1NQWs0njvXHEzGLTLMhvJB5kznGfyre1/XYPD1jHBaxq1y42wQKOnufaleMo32gvxHZp2+0/wAA17Xrfw/ZpbWsavdMNsMCjp7n2rmtP0+Xz2v9SfzryXkk/wAHsKNP0+Xz2v8AUn868l5JP8HsK0yCOoxUNubu/ki0lFWQUUUEEdRimIKKKCCOoxQAUUUEEHBGCKACiilZWRtrgqR2IxQAlFFBBBwRgigAoooIIOCMEUAYnilimmwsvVbhSPyat34kf8i5B/19r/6A9YfidTLZ20K/ekuFA/Ij+tbnxI/5F23/AOvtf/QHqH8NT0RX2oHmiMUkVl6ggivS/iR/yLkH/X2v/oD1yPhfw3Prl6rtmO0iYGSTHX/ZHvXQ/EbU7d7eDTo5N0yy+bIo/hGCBn35rGmnGjNvqXN3qRS6HAUUqqzsFQFmY4AAySaSuM6QopVVnYKgLMxwABkk0lABRSqrMcKCxwTgDsOTSUAFFKqsxwoLHBOAOw5NJQAV6r42/wCPHTf+wjF/Jq8rVWY4UFjgnAHYcmvVPG3/AB46b/2EYv5NXZh/4c/kc1b44lSilCs2doJwMnA6CkrpMgpvh3U7fQ/i94d1K/k8q1STY8jcBN25ck+g3jPtTsE5wOnWs7XLSK60qZpFy0SF0buCBWVaPPBounLlkmfU+r6Tc3d59otgsiuo/iAxXA/Hi+t9J+EyaRPMpu72eNIoweWCtvY49BgDPuK4X4XeNPEVp4Tkt4tVnaKK5ZY1mxL5ahEwqlgcL7DjrXHz3V34g8b6tfazdzXs1vcusZmcsEG9sAA9AOwGAO1eFh8nhQrvExes/wDhz0amOc4eztsXrdDFaxI3VUAP4CpKME5wOnWivojywoowTnA6daKACijBOcDp1ooAKKlQQ+Q7SeZ5gOFCgbfxP/1u1RUAZniH/kA3H/Af/QhXSeCP+RNsf+2n/oxq5zxCD/YFyccfL/6EK6PwR/yJtj/20/8ARjUUv4/y/UVT+H8zE8Tf8lB0/wD69P8A45Vqq3iZWPj6wYA7Ra4JxwD+8/wNWaX2pepX2V6BVfXMHwnfn5NwRBwcEDzU6jv9f/rVYwcZxxUOu7v+EPvsh9u1ME8r/rU/I+3eiXwv0Ylujzuip4LK7uUL21tNKoOC0cZYA+nFFeVc7ixoP/Ix6b/19xf+hivQbrcZhv35x/GQSfxHX6159oP/ACMem/8AX3F/6GK9BuF/eMV2jaBuAG05/wB09/pxXfh/gZzVfiRgaT/yVCL8f/RJrute/wCRc1L/AK9Jf/QDXC6T/wAlQi/H/wBEmu617/kXNS/69Jf/AEA10UP4c/VmFX44+iOR+GP/ADE/+2X/ALPWuzEeKLlexF0f/JaD/Gsj4Y/8xP8A7Zf+z1tGPPiO7lyMKblcZ55tYe34VNP/AHdf11Oyl8Vb/D+sSOw/4+H/AOuMv/otql0DSrDxFfPY6hE0qu5igTcwSR1UNIzFCDtQPGcZG5mAHQkRWH/Hy/8A1xl/9FtWl4Bje21hpSyj7NPPKIur3CTJGG2DruQwg45yGOOnPlZvUrU8JJ0t/wDgP9bf8DcrBxhKslM2A1h4omfQJfGLaw4ybmxltkjSYKegeNFKkMAchj05BHXmYPDsGu63FC8Ly6fdFzpyTZXzWTIlaXbhlER+Ur8u5iMHHI0fCeieFdJ8Xpq2k+JBqc8/mC2sIUBkywJOeegXPJCgY5NT6FqZt/Elvbb4pm0lpwzRk5vRct5khhUgZ8pwowM7hkjHAPx0KlahCpHCt8trvTls9eyXXlV+1+117UowqSi6tr3737evS7/rW2zWfihpNAbxidamBzc2MtskUc6oRkB0RWUhhkEMegyCM54RfBF/o/iS4e1t5GsJbK5WBpnQSAyQSIEYZ+8HO0npxnpXTeGtJ8K6H4o/tvSfEQ1WeUSfZtPhQeYSwJOeeAFzyQAMcntVKXxtLfeKDYt9knhtred557diFkdY5ZiI8n7oYbd2DkDPGRXo5SoQxDppvksnty6630svJX9VrbTmxfM6alZc3rf9fU4PQ7Wey1+6t7qJopY7C9DKw6f6LLVLWv8Aj/j/AOvS2/8ARCV6jB/ZviG0bUbRQ8zW01uCTgr5kbIQ3XpurzfxPZyWWriKUAFbeBCNwJBWJAf1HWvqqlF04trVO1vxOOlUU8PJdbrT5Mx6KKXadobjBJHXn8vxrmIEoopdp2huMEkdefy/GgBKKKXadobjBJHXn8vxoASiinNEyxo5K4fOAHBIx6gcj8aAG0UUpUgKTj5hkYOf/wBVACVveCP+Rxsf+2n/AKLasW3gkurmK3gXdLK4RFzjLE4A5rpvB+lz2/iuzlkktSq78iO8idvuN/CrEmtqMZOcWl1CVOcqcpRTaVzU+JrEJpq9iZT+W3/Gt3wR/wAidY/9tP8A0Y1ZPxEspbz+zvKaBdnm5864SLrs6biM9O1bXg+FrfwpZxSGMsu/JjkV1++38Skg16ME/rMn5f5HFOE1hoza0vv955hojF/E2ns3Ja8jJ/77Fd78RmK+G4gP4rpQf++WP9K4/RdIuYte092ksyFuYyQt9Cx4YdAHyfoK7Xx9ayXegwpE0SkXKnMsyRj7rd2IH4Vz0oyVGasdFWjUjWhFxd35GV8Mf+Yn/wBsv/Z6j1ACX4j328A+VCpXPb5U/wATVz4d2Utn/aPmtA2/yseTcJL039dpOOvemXVjK3j7UJw8GxoVABuIw33U6rnI6dxWijL2MFbr/mQ6dT2048rul/kWKKm+yyf3ov8Av8n+NH2WT+9F/wB/k/xrTlfYj2c+xDRU32WT+9F/3+T/ABqOSNon2PgHAPBB6jPUUrNCcZJXaG0UUEYpEhRRQRg4oAKKKVlKtg4z7HNACUUUEYOKACoLy8isbdpp2wo6DuT6UXl5DY27TXDYUdAOpPoKTRfD8+rXn9p64oW3jJ8m33Aqcd8jgj370tW+WO49ErvYTQNDm1i6TV9ZUpbxndbwMcD/AHj7fzq1qanxlfx2VqCNOtJN01z/AH2xjav681Pc3M/iW6bT9LcxadEdtxcrxv8A9hag8Q+IbXwzYLpmkKouAuAB0iHqfeqtCMHfbq+5PvOXn+QeIfENr4ZsF0zSFUXAXAA6RD1PvXm0srzStJKxd2OWYnJJpWeS5nLSPukkblnbqT6k0yvPq1XUfkddOmoIKKVVLMFGMk4GTgfmaSsTQKKVVLMFGMk4GTgfmaSgAopVUscDHQnk4pKACilVSxwMdCeTikoAK9V8bf8AHjpv/YRi/k1eVqpY4GOhPJxXqnjb/jx03/sIxfyauzD/AMOfyOat8cSpRShS2cY4GeTikrpMgqtqX/IKu/8Arg//AKCas4zn2qK6iM9nNEOskbKPxGKT2Gtyf4b/APIuT/8AX23/AKAlY+nf8jJrw/6em/8AQ3rS+HNzHHYXdhIwS4Scv5bcHG0D+lVte0650DWZtWgUzWV02ZgBzGT/APXpf8uoS7B/y8ku5doqOCaO5hEsLBkYZBzUlUIKKMZz7UUAFXLWFJIkBSLczSEvKWAVVUN2P1qnjOfatbRTbrPA18wW1AuDcE54j8n5+n+zmlOfJCU+yNaSu7f1uh1vpsdzaxXCTackcwBjEpmjeTPTajAM2TwNoOTwM1XaCIA4WBxmVCUEqsjoOQQ+CCDjgisXW/8AhKP7an/sFLpLDI+1LDj7Os+B9oE+P3f+s3538bcfwba6i+NgbO0/ssKsIWcBR12eWphJ+sJhPPPr82a8nB4+rWrRhPlafbp6ndOjGKul1W6Rx3iH/kA3H/Af/QhXSeCP+RNsf+2n/oxq5zxCP+JBcn/d/wDQhXR+CP8AkTbH/tp/6MavXpfx/l+p5dT+H8zE8Tf8lB0//r0/+OVaqt4mU/8ACfWDcYFrjrz/AMtO34VZpfal6lfZXoFV9cKnwnf42bgiZ4Ib/Wp+BH61Yxxmoddz/wAIdffewFTuCB+9T8R/X8KJfC/RiW6PO6KngtJLhC0bQgA4/eToh/JiDRXlXO4saD/yMem/9fcX/oYr0G5GJFHONoxlg3H1HX/I7V59oP8AyMem/wDX3F/6GK9BuF/eMfl+UDOF2H/vn1+ld+H+BnNV+JGBpP8AyVCL8f8A0Sa7rXv+Rc1L/r0l/wDQDXC6T/yVCL8f/RJrute/5FzUv+vSX/0A10UP4c/VmFX44+iOR+GP/MT/AO2X/s9a7H/iqrkdsXX/AKTW9ZHwx/5if/bL/wBnraMefEd3LkYU3K4zzzaw9vwqaf8Au6/rqdlL4q3+H9Ykdh/x8v8A9cZf/RbUzSdIi8RamLS4HmxOzRQw7mCSSKoZ3kKnOxA6fKCpZmAzwSH2H/Hy/wD1xl/9FtVrwVcf2brRkcrm3nmm8rPz3EUyRhjGO7RtCpK9SG9gD5OcVKtPCSdLf+v1sv6uXgowlWSkbkrWvicS6CPGB1u4X5riwmt0ijnCHJAeNFZSGGQQxxgZBGc81Z+G01zVoILmN5rK53/2esuV81kyJWl24YCI/KQNu9jkHHTf8P8Ag/RvDmt/8JNYavJqIkjZ7a0ii+c7wSM4OQMHqQAOpIFVPDGvRt4ggt4p7a5k0Z51k8h8m8Fy/mSGFT18t1A77l5GOAfj41K2HhUjhm+W13dcuuvSy6pK/Zvtdey4wqSi6lr3737evS7/AK10maz8UNJoDeMTrUwObmxltkijnVCMgOiKykMMghj0GQRnPm8ugHw/4yuraNZRA+l3k0PnAB1Q20o2tjjKsCpxxkcV2fhnSfCuh+KBrWleIxqs8ok+z2EKDzCWBJzzwAueSFAxyav3t7ZeIPOa3ktbmaGKa2eaElkV5VbeFPGR+8xnocA16WUUm68qVFvltfVcuut9LLrZf5205cZNRpqc979Hf06vz/rfzHwLqD2euuhd/INtPLJGDgMY4ncf+g12Ot6JaeKtMS4gKpdCNWjfjIBAYK34H9a4XRbOew8QXVtdxtFKlhfAqR/06y8/Spxrl1oetQXEDBo2tLZZIi33h5CHp269a+ro1FGLhU2/4c4JUVKk6sfiTX43/wAjBu7Sexunt7qMxyocFTUNeo6lpun+NdHW7sXVblR8r9wf7rV5reWU9hcNDdJ5cisVKk88f0561nWoum7rVMzp1Of1IKKKXadobjBJHXn8vxrA1EooqW3tZ7ueKC1iaaaZ/LjijG5nY4wABzzmgCKivVLH9nbxteWK3ErabZuwz9nuLhvMHsdqsv61wXiTwtrHhLUvsOv2ZtLgjcql1bcucbgQTx7+x9KlST2G4tGRRRSlSApOPmGRg5//AFVQjT0m2ltPFmnwXC7ZFu4cgEEEFlIII4IIIII4INdr4XtNJ0fQtM17UNPjw1vLI1ysknmAiaSNm2htpA3QJt2gky53AKTXHaff3Op+MNOur1w8zXNuhIRVGFKqowoAGAAPwrrdHnt7/wAJ6Zpd5dWKWv2WSNzJfQxsha4kZkZWcMoO2Bw4VseWPlYMwOOKdSNGLh31tvbTbzOyKi6ST7y/KJqahd6T42tVGgaOmoXkLmOKO7aSIFz82zCyj7yJI+4tgeTtwCwzYg8U+FNAhg0x4XgG0MvlJI6BHyySZZ9yhlKvtw5Xfg5IIGSII/C9q3/CK6jprXjOJEkGr20ojcAqHLyeXnCNKgTy8fvSxYkKFWXQtB1SVL67vNMidFVBbvrMKlUQYSPapYShVCru8yIsFGdpyx8xVsSpuTc+Xp/NfTfy7CdOm48tlf8AD5eY610nTrHxA89x4fsoLS1CyqUuJzLG4L+YpbzCpMXkXBLbSH8ngfOtah1fwv4z26dDbeZtlQmS5SRFi3HYr7Uky/zuiY3LjzN3IBxjpf3NxrFwL2fTvsM0YTzDqFoz78v5kjIJx/rPPuAVDjb53yn5Fqe307RPDlncyaZLp95fTBRHs8QQOg2sHG/csWAHWNsLkkpgkDIZe0xSXK27u1rbfP8ArffSw+Wk3ey/rt/XpqWNIv8AQvCfh22vtW0o2M+pKJFW2SV0lQKGUDfLyQrqSwIHzgckNtc0dnqmrRazouj281vOF8+WWWQSShjshUfvAE3yo0P3W2eXvJKkE0LWy0zWfDttYa9LYQXdpgJKdfgSMgKEyCokIYqkYKldvyZBBJDTQSXVlPb6b4fltJ7SNcSGC/tZG+QmSEopk3SeXKXlyQm8sVKqoAFSqYt+5Bvmu/8ADbXbz7eYlCj8Ukrfj8/1IdM1rw5canLDa2ySYV2zIbnZGi8vIGMzGXYgZtpSMsFONrYU3YhYWNqdR8RWtvaG1RVuYovtEu+Uz3EOFHnhQAIM9Dkkn0qzdeE/DWiwy6lp28yldhhluQiFH+WRNz7Qm5CU3F2KhsgFgDVrRbTSde8NPaeLdasnnuB5stvbX0bMkouJ5GcBCV5Ey9PT2rLESxVJLl9pa67uW7vZdrF0lRndvlvZ+nzK2k2uk6paTalaRJJbS2V3Kh2zxtBLG0YCkNO6HIkzwB0/Csy6/wBcv/XNP/QBW9dXHhvw1oS6T4e1Jb0NDLG0CSebIzySRfOxUYXaFbrj0rBuv9cv/XOP/wBAFe1k86k6E3Pm+J/Fo7dDkxkYxso26bfMhoopcHOOpPpXsHnCUV2+mfCXxNqVmtw621kHGVS6kIcj6Kpx9Dg1z3iDwzqnhm8Fvq1v5e/JjkU7kkA7g/061EakJOyepbhJK7Rk0UUrKVbBxn2OasgSoLy8isbdpp2wo6DuT6UXl5DY27TXDYUdAOpPoKTRfD8+rXn9p64oW3jJ8m33Aqcd8jgj370tW+WO49ErvYTQtCm1q4XVdaXZbLzBbt0I9T7Vo3NzP4lum0/S3MWnRHbcXK8b/wDYWobzUZfEupnR9IZo7GP/AI+rpB2/uj69Ks69rNn4V0cWdgqrcFNsMa/w/wC0f881S5YxeunV9yXzNrv+RV8Q+IbXwzYLpmkKouAuAB0iHqfevNpZXmlaSVi7scsxOSTSvJJdXBeV98kjZZnbGSe5JplefVquo/I66dNQQUUqqWYKMZJwMnA/M0lYmgUUqqWYKMZJwMnA/M0lABRSqpZgoxknAycD8zSUAFFKqljgY6E8nFJQAUUqqWOBjoTycUlABXqvjb/jx03/ALCMX8mrytVLHAx0J5OK9U8bf8eOm/8AYRi/k1dmH/hz+RzVvjiVKKUKWzjHAzycUldJkFFGM59qKAM3UNPlM63+nOYr2LkEfx+xrpNA1+38QWb212ipdKNs0DD73uPas3Gc+1ZmoadI0632myGC9j5DKcbvY0k3B3j80NpSVmXbnwXqVrdyN4fvoobeQ58uYn5T7fKaj/4RnxX/ANBKy/z/AMAqv/wmPidNynTIGKDk+Q/PPs3P4Un/AAm3iX/oFRf+A0n/AMVUc1DzQ7VfIs/8Iz4r/wCglZf5/wCAUf8ACM+K/wDoJWX+f+AVXHjTxMQxGlQ/KMnNvIP/AGbmk/4TbxL/ANAqL/wGk/8AiqOaj3f4hap5Fn/hGfFf/QSsv8/8ArSsPDuuyQJaXV1bTNKl1EQpxuMsOyP+EdH5Pt69KxR408TEMRpUPyjJzbyD/wBm5rc0LxB4gvfJmXTl85ftThI4H3bo4Q8WASc5fj36DmpnKl7OTi3t1ubUo1XfRW67d1sTyan4f8JwvoXiCK4vdTVMWupOVke2jK4jKyFd0IH3wqh9mcgk8VJ9iuLTT4DclmZ5LoFnfc5dUUSBj/EwkDqW/iKlhwRTtS8Nwa/fR65q+l38vyp5ttHBcwSIuM+Vs8lvMZR8u8OgOBnb94xNqE82nxDUlktXje7by7qPypcmNHkdlPQszNJgcKJAAcAV85lMl9aXs/8At7/gfM9OtFuKTXVW/wCCc14h/wCQDcf8B/8AQhXSeCP+RNsf+2n/AKMauU12+tJtFnSK5hkc7cKsgJPzA9Pwrq/BH/Im2P8A20/9GNX1NFp1tO36niVP4fzMTxN/yUHT/wDr0/8AjlWqreJlP/CfWDcYFrjrz/y07fhVml9qXqP7K9Aqvru3/hEr77u7Yn8JDY81PwI/WrGOM1Brm7/hD7/7wXanAbIJ81OSOx/nRL4X6MS3R55RU8FpJcIWjaEAHH7ydEP5MQaK8q53FjQf+Rj03/r7i/8AQxXoNycyDrjaNvzEjHtnt7V59oP/ACMem/8AX3F/6GK9CuUPmE4A2gZBQIf++f8ACu/D/Azmq/Ejn9J/5KhF+P8A6JNd1r3/ACLmpf8AXpL/AOgGuF0n/kqEX4/+iTXda9/yLmpf9ekv/oBroofw5+rMKvxx9Ecj8Mf+Yn/2y/8AZ612P/FVXI7Yuv8A0mt6yPhj/wAxP/tl/wCz1tGPPiO7lyMKblcZ55tYe34VNP8A3df11Oyl8Vb/AA/rEjsP+Pl/+uMv/otq5vxUSulxMpIKzqQR24auksP+Pl/+uMv/AKLaszULBNRsmhkO3nKkHoe1TNc0Wjni7NM2NbvL7VfhiQ9xcTmS1hdlZy27BVjx+FeUAlGBBKsDwRwQa7TRfFU/hsHTtYikmgjH7powCwHpzjIrY/4WNpG0t9nvMAgfdTP5b/asXSoSSs1HysWpVIt6XOEuvEmuXts9veazqFxDIMPFLdOyt35BODWn4FvzZa8++RxAbWeSRFPDeXE0gz/3zXT/APCyNI/59r3/AL9p/wDFVasPHGm6hM8cMN0pSKSU71QcJG0h/iz0Q/jjOM1EMPRpfBNL0RUqtSfxRv8AMfa+JPD2rTb/AJGuBBKzCW3JZYxGxkGcYxsDZGeRkc5qKXUvCDXUcVxHZea6R7fMtP4SoKclcAbSPpS2PjrTNQuGhhgu1ZYZZjvRQNscbSN/F1wpx71ZTVtJ17dps68zRK4jm2gsGRXGMEkEBx+OcZwa71NS2km/T/gmPJJQb5Xb106+Rk6jpdz4Yvm1fQV3WjHNxaDoB6j/ADxVnUtN0/xro63di6rcqPlfuD/daltrmfwzdLp+puZtNlO23uW58v8A2GqtqWm3Phu+OsaEu+1fm5th0x6ipaST006rt6E3d1rr0fc88u7Sexunt7qMxyocFTUNeo6lpun+NdHW7sXVblR8r9wf7rV5reWU9hcNDdJ5cisVKk88f0561wVqLpu61TOqnU5/Ugr1D9nu1t7j4qxvcxh3gs5ZIc/wvwufyZq8vrV8N+IL/wAKa/aa3pTqtxbOSobkMMYZWHXBBx/LpXNJXVjZOzufYGp6jex6pIqSvGI2wqjgY+neuf8Ai74d0zXdA0y71a0WWeGbYp3MpUOpLDgjuoqvoXxo0XWNHtr/AFDR7mG5cHKxhJApBI4YkHt6V494y8Tan8U9QtNR1GGPT9Lt1dbS2ilEjAkgMSfUlR1A4AwO9eVleFqUsVOcp86vt23PRx6lGhDmjy3Wj77F/wD4Qjw9/wBA/wD8jSf/ABVH/CEeHv8AoH/+RpP/AIquS/4ROx/563H/AH0v+FH/AAidiMfvbjn/AG1/wr6vmX/Ptf18j5+z/nf9fM7G38H6Fa3MVxBY7ZYnDo3nOcMDkHlquf2FpH/QKsf/AAGT/CuC/wCETsf+etx/30v+FH/CJ2Ix+9uOf9tf8KpVGtoL7/8AgGsKlWCtCq16X/zO9/sLSP8AoFWP/gMn+FH9haR/0CrH/wABk/wrkdJ8NafbzzSMhuNsLEJOFZc5HOMVfudA0m5t3i+xRx5h85XjUKwPPcfyrOWJmpW9n+P/AADupqrOnzfWGn297/M3/wCwtI/6BVj/AOAyf4Uf2FpH/QKsf/AZP8K53/hG7G4vLNp1kl+z2aBEYja2OmeOetIugWE2qWsktgkJUMCgQBH4PVcY4/wpfWp/8+/x/wCAVy1f+giW6X2vLzOj/sLSP+gVY/8AgMn+FTW+nWNkxktbO2t2xgtFEqHH1Arl7bQNOLT3q2KSO7hfK2AoigDouOtEWg2NncXtvHayTW8+zfGGJC45/mf0o+tVNbU9fUmVKU4pVMQ3F/4vO332+RLql3q2p6sbG0gSKxVdztNGknm4bqA2R2GM896lkae2wjxQpxwBCmMflUtrYx2bNb2yssZtjtQjkZ7fWmXCNFYwJKMNkkD0FKFSd/ed2Z1FywcYXUVsr3/RakX2qT+7F/35T/Co5JGlfe+CcAcADoMdqbQRitrtnE5Sas2FdF4AghuPH2kx3KB08/dg+oUkfqBXO1NaXU9hew3VrIY54XWSNx2I5FRJXi0hRdmmfReu313DqHlRyvEgUEbTjPvWF8TVW7+GP2i7QGdJI3RiMEEtjP4gmqukfFjT9Q09W1rS5BcIdrGFVdWPHI3EEfTmuJ8e+PrjxVILKGD7LYW8pKoWy0jDIDN6cdh0z1NfNYLLsRSxs6s53T6dv62PZxFeDw6SicXUF5eRWNu007YUdB3J9KS8vYbG3M1w2FHQDqT6CpNK8NXOoXo1HxAqpCnzQ2wcMMdQSRxivptW+WO542iV3sRaFoU2tXC6rrS7LZeYLduhHqfarGpalc+JL5tH0JtlqnFzdDpj0FGpalc+JL5tH0JtlqnFzdDpj0FWNS1LT/BWjraWKK1yw+VO5P8Aean7qi0np1fcnVvz6LsGpalp/grR1tLFFa5YfKncn+81eaXd3PfXT3F1IZJXOWY0Xd3PfXT3F1IZJXOWY1EylWKnGQcHByPzFcNas6jstEjqp0+ReYlFKqlmCjGScDJwPzNJWBqFFKqlmCjGScDJwPzNJQAUUqqWYKMZJwMnA/M0lABRSqpZgoxknAycD8zSUAFFKqljgY6E8nFJQAUUqqWOBjoTycUlABXqvjb/AI8dN/7CMX8mrytVLHAx0J5OK9U8bf8AHjpv/YRi/k1dmH/hz+RzVvjiVKKUKWzjHAzycUldJkFFGM59qKACijGc+1FABRRjOfaigAooxnPtRQAVteGJvK8TaFzgPdzIff8AcNj9cVi4zn2qxGJlWzuLOWKO5s7rz4zIwwCNvUbhkEZB571jiKcqtCcI7tM6MPLlnf8ArdFPWLyXS9Y0fTNI0awm06a3tHBk0yNzdkp95iVJJOW5Bz15612uueH9NT7LZi0VLe2W5EMaMyhBLI4cDB6EY+nbFV4tct47FbY6fKi/KStvrksSKQMYRQ/yLzwoOAMDsKxNWgttSFuiq9tb2sdyI42vRMQZkw3zH5jk88knLHtgDxcBg8TTrwlUp2S81rvru9df+CejVnGatGTXyemvTYf/AMIR4e/6B/8A5Gk/+KrXsbG306zS1so/LgjztXcTjJJPJ56k15hqvh61sdMluYZJi6bcBmGOSB6V3Hgj/kTrH/tp/wCjGr6OjNOfLypaHh1ItRvzXMTxN/yUHT/+vT/45Vqq3iZT/wAJ9YNxgWuOvP8Ay07fhVms/tS9TT7K9AqvrpH/AAiV8ON2xDymDjzU79x9asY4zUGuH/ij78A8bUyBJxnzU/h9ffpRL4X6MS3R55RU8FpJcIWjaEAHH7ydEP5MQaK8q53FjQf+Rj03/r7i/wDQxXoFxgSDbjbjja+4fh3H0PNef6D/AMjHpv8A19xf+hivRJ4mYyO8g3IBlZF2O2fbue+a78P8DOar8SOd0n/kqEX4/wDok13Wvf8AIual/wBekv8A6Aa4XSf+SoRfj/6JNd1r3/Iual/16S/+gGuih/Dn6swq/HH0RyPwx/5if/bL/wBnrXY/8VVcjti6/wDSa3rI+GP/ADE/+2X/ALPW55IbXr2bcoKNcJtJ5ObWE5Ht8v6ipp/7uv66nZS+Kt/h/WJUt5zbzeYFV/lZSrZwQQQensal+1Q/8+Fv/wB9Sf8AxVVqXHylsjggY70zlJrpbVn8u70mzcr2fe2P/HqadNshGXbw/YhR1JjYf+zVJqH/AB/yfh/IVLfCVrh2Qtt2jgE9CP5VnvY6HBJy30dvzIP7Ls/LDjw/YFSMjCMf03VHEtjAzNbaTYwu0bx70RsgMpVurdwSKvpv+12uwkDyV3emOaoOAWdlI2hsAe3NKNnugqxUUmhum2FpHdOY7WBSbeZcrGBwYmBH4gkVjeI9Gb7Sl9puY5o4osqnHRFAI9MAdK6Gw/4+X/64y/8Aotqbdj50bI5jQY7/AHFq3CLi0RGTUH6r9SHw94htvEli2l6wqm4K7SG6SD1HvU9tcz+GbpdP1NjLpsp229y3Pl/7DVzOs6M/mfbtOyk6HcQvGfce9dLoOt2niTRxZas0P2liY2hZuZMAfMB+P6GnCbvyy+Lo+5lKKtdbfkV9S0258N3x1jQl32r83NsOmPUVamuvCniO3iuNQltd+OFln8p19jyDTbe6m8L3AsdUZpdMkOILlufL/wBl/b3psumeCbiVpZJbDc5ycXu3n6Bqu26VvNMn1v6oZ/Y3gf8A562X/gef/i6P7G8D/wDPWy/8Dz/8XR/Y3gf/AJ62X/gef/i6P7G8D/8APWy/8Dz/APF0uXyj/XyHzebNvT7TSIbGOPTmhNsM7Ck24dTnnJ75rE/sbwP/AM9bL/wPP/xdben2mkQ2McenNCbYZ2FJtw6nPOT3zWJ/Y3gf/nrZf+B5/wDi6xpwtKXuw+X66Hbial6dNc9R6fa2W3w67f8AAD+xvA//AD1sv/A8/wDxdH9jeB/+etl/4Hn/AOLo/sbwP/z1sv8AwPP/AMXR/Y3gf/nrZf8Agef/AIutuXyj/XyOLm82H9jeB/8AnrZf+B5/+Lo/sbwP/wA9bL/wPP8A8XR/Y3gf/nrZf+B5/wDi6P7G8D/89bL/AMDz/wDF0cvlH+vkHN5sfHpnguHd5VzaJvUo23UWG5T1B+fkU/7D4Q/5/bf/AFflf8hJvuf3fv8AT26VD/Y3gf8A562X/gef/i6P7G8D/wDPWy/8Dz/8XS5F/LH+vkUqkkrKUv6+ZMLHwgssci3tuJIl2RuNSbKL6A7+BzT2t/CjypI+owtJHnY51RyVz1wd/FVv7G8D/wDPWy/8Dz/8XR/Y3gf/AJ62X/gef/i6Xs4/yx/r5DVaaVlOX9fMmt7Hwha7vst7bw7sbvL1Jlz9cPSxWnhKGSR4b+CN5Dl2TU2Bc+p+fnqag/sbwP8A89bL/wADz/8AF0f2N4H/AOetl/4Hn/4uj2af2Y/18gVaataUtP67k62nhJbhp1v4BMww0g1NtxHoTvz2H5UyfTvB11IHubu1mcDAaTUWYgenL1H/AGN4H/562X/gef8A4uj+xvA//PWy/wDA8/8AxdP2a35Y/wBfIHVm1Zylb+vMP7G8D/8APWy/8Dz/APF0f2N4H/562X/gef8A4uj+xvA//PWy/wDA8/8AxdH9jeB/+etl/wCB5/8Ai6fL5R/r5Ec3mw/sbwP/AM9bL/wPP/xdH9jeB/8AnrZf+B5/+Lo/sbwP/wA9bL/wPP8A8XR/Y3gf/nrZf+B5/wDi6OXyj/XyDm82a2lWWhW9qyaS0BhLkt5dwXG7A75PbFZtzpHgx7qV7mSzEzOTJuvSDuzzxv45rS0qy0K3tWTSWgMJclvLuC43YHfJ7YrNudI8GPdSvcyWYmZyZN16Qd2eeN/HNYwh77fLD9fnodtapehBc9R+T+H5ajY9K8EwyLIklhuU5G693D8i2DUOo6pc+J706ToLbLReLi6HTHoPb+dSf2N4H/562X/gef8A4ulvvEOi+GtJ8jQzbyyNnYkMm8A+rHJ/WtnotbJeRxLV6Xb8yTUtS0/wVo62liitcsPlTuT/AHmrzS7u5766e4upDJK5yzGi7u5766e4upDJK5yzGo5Y/KmePer7GK7kOVbB6g+lefWrOo7LRI6qdPkV+o2iinSx+VM8e9X2MV3Icq2D1B9KxNRtFOij82ZI96pvYLuc4VcnqT6U2gAop0UfmzJHvVN7BdznCrk9SfSm0AFFOij82ZI96pvYLuc4VcnqT6U2gAop0UfmzJHvVN7BdznCrk9SfSm0AFFOij81yu9UwrNlzgcAnH1OMD3ptABRToo/NcrvVMKzZc4HAJx9TjA96bQAV6r42/48dN/7CMX8mryyKPzXK71TCs2XOBwCcfU4wPevU/G3/Hjpv/YRi/k1dmH/AIc/kc1b4olSinxRebv+dE2qW+c4zjsPemV0mQUUoGQTkDAzz3pKACilAyCcgYGee9JQAUUoGQTkDAzz3pKACilAyCcgYGee9JQAUUoGQTkDAzz3pKACilx8pbI4IGO9JQBmeIf+QDcf8B/9CFdJ4I/5E2x/7af+jGrnPEI/4p+5bI4KDHf7wro/BH/Im2P/AG0/9GNRS/j/AC/UVT+H8zE8Tf8AJQdP/wCvT/45VqoPEsefHVjJvUbbYLtJ+Y583kD0GP1FT0vty9Svsr0CoNdz/wAIhfE5+4uMrj/lqnQ9x/n1qxj5S2RwQMd6r65/yKN/jb91M4fn/Wp1X+tEvhfoxLdHnlFaul6F/alq039q6ZZ7XKeXd3HlseAcgY6c9fY0V5PMjvsyHQf+Rj03/r7i/wDQxXoFxjzBjbjHGxiR+GeR9DXn+g/8jHpv/X3F/wChivQrn5pWJbkAcMBk/iOD9e9ehh/gZy1fiRz+k/8AJUIvx/8ARJrute/5FzUv+vSX/wBANcLpP/JUIvx/9Emu617/AJFzUv8Ar0l/9ANdFD+HP1ZhV+OPojkfhj/zE/8Atl/7PWux/wCKquR2xdf+k1vWR8Mf+Yn/ANsv/Z62iinxFduWAZTcAL3INrDk/oPzqaf+7r+up2Uvirf4f1iUqdHG0sionVjim1asWmWYeTnaWUPhc8f0ok7IwpxUpJMcZ7yWVrffuJypAA59ajmeeBfIaYMpHIVs49qtW6MNSuNyHkMQCOvNOaBPOhJgUGQEOnoPX2rLmSdrHZ7Oco3u73t1KaT3Myrbq/ykYAwBx9aZNbPAFLFWVujKcirNzbZuI4beMBcZDevuTTbxPKhWFVYIh5ZgfmJ9KpSV1YylB8r59bdRlh/x8v8A9cZf/RbUy6/1y/8AXOP/ANAFPsP+Pl/+uMv/AKLamXX+tXn/AJZpx/wAVr0MF8D9f8yGsHWdGfzPt2nZSdDuIXjPuPet6jt1/ColFSVmSm0yHSPG+n3Wnm28RhVkXAYvEXWT6gA81Y/tnwP/AM8rL/wAP/xFZ9xo2n3UhkmtlLHqVJXP5VH/AMI7peP+Pb8PMf8Axo5qtrOz9Q5YeaNT+2fA/wDzysv/AAAP/wARR/bPgf8A55WX/gAf/iKyv+Ee0v8A59f/ACI3+NL/AMI7peP+Pb8PMf8Axo5qvaIcsO7Ov0+60iaxjk05YRbHOwJDtHU54wO+axP7Z8D/APPKy/8AAA//ABFaWkWNvb6XDFDHtRd2BuJ/iNcyfDulBQfs4JPUeY/H61hSqV3KV4x+Sf4no4unh40qThKbbWt2rdNvL/gGp/bPgf8A55WX/gAf/iKP7Z8D/wDPKy/8AD/8RWV/wj2l/wDPr/5Eb/Gl/wCEd0vj/Rv/ACI/H61vzVe0TzuWHdmp/bPgf/nlZf8AgAf/AIij+2fA/wDzysv/AAAP/wARWV/wj2l/8+v/AJEb/Gl/4R3S+P8ARv8AyI/H60c1XtEOWHdmp/bPgf8A55WX/gAf/iKP7Z8D/wDPKy/8AD/8RWV/wj2l/wDPr/5Eb/Gl/wCEd0vj/Rv/ACI/H60c1XtEOWHdmp/bPgf/AJ5WX/gAf/iKP7Z8D/8APKy/8AD/APEVlf8ACPaX/wA+v/kRv8aX/hHdL4/0b/yI/H60c1XtEOWHdmp/bPgf/nlZf+AB/wDiKP7Z8D/88rL/AMAD/wDEVlf8I9pf/Pr/AORG/wAaU+HdLH/Ltn/to/8AjRzVe0Q5Yd2an9s+B/8AnlZf+AB/+Io/tnwP/wA8rL/wAP8A8RWV/wAI9pf/AD6/+RG/xpT4d0sf8u2f+2j/AONHNV7RDlh3Zqf2z4H/AOeVl/4AH/4ij+2fA/8Azysv/AA//EVlf8I9pf8Az6/+RG/xpT4d0sf8u2f+2j/40c1XtEOWHdmp/bPgf/nlZf8AgAf/AIij+2fA/wDzysv/AAAP/wARWV/wj2l/8+v/AJEb/GlPh3Sx/wAu2f8Ato/+NHNV7RDlh3Z1WlXuhXFqz6SsAhDkN5duUG7A7YHbFZtzq/gxLqVLmOzMyuRJusiTuzzzs55qfQdOtbSxdLeLYpkJI3E84HqaxrzQdNlv7iR7bLNKxJ8xuTn61hCpXdSScY29Hc9CvTw6w1NxlO/W7VvkXv7Z8D/88rL/AMAD/wDEUf2z4H/55WX/AIAH/wCIrK/4R7S/+fX/AMiN/jSt4c0pWwLcMPUSP/jW/NV7RPP5Yd2an9s+B/8AnlZf+AB/+Io/tnwP/wA8rL/wAP8A8RWV/wAI9pf/AD6/+RG/xpT4d0sH/j2z7+Y/+NHNV7RDlh3Zqf2z4H/55WX/AIAH/wCIo/tnwP8A88rL/wAAD/8AEVlf8I9pf/Pr/wCRG/xpT4d0sH/j2z7+Y/8AjRzVe0Q5Yd2an9s+B/8AnlZf+AB/+Io/tnwP/wA8rL/wAP8A8RWWPDulk/8AHtj38x/8aT/hHtL/AOfX/wAiN/jRzVe0Q5Yd2bTaV4b8T6bKmkpbRuh4kgh8tkPbIwDg151qml3OkXz2t4m1l6Hsw9RWzJbXXhvUkvtMkKw7gG3ZIUejY6iuu/4lvjrQ+0dwg/4FE39RWcoqtpa0l+JSbp67xPLKKu6lpVxpWoG0vR5bA8Oc7SPX6VSrhaadmdSd1dBRSqAzAFgoJwWPQe/FJSGFFKoBPLBeCcn+VJQAUUqgE8sF4Jyf5UlABXqvjb/jx03/ALCMX8mrytQCeWC8E5P8q9U8bf8AHjpv/YRi/k1dmH/hz+RzVvjiVKKUAHOWAwM896SukyCij15/+vRQAUUevP8A9eigAoo9ef8A69FABRR68/8A16KAJrS0mv72G0tIzJPO4jjQd2JwK9bsPgzpcFmh17VpvtDDJFuyIin0G4En68VwPw+nitvH2lS3EixoJtuW7lgVA/MivbtdsbubUPNjieVCoA2jOPavCzjHVsHTUqUbno4HD06z99nkHjr4fT+EfLurec3WnzPsWQrho26gN68DqPTtXG17p8TXFn8Mfs10wE0kkaKpOSSG3Y/ACvC69LB1p1qKnNWZy4iEadRxiZniH/kA3H/Af/QhXSeCP+RNsf8Atp/6Mauc8Q/8gC55/u8f8CFdH4I/5E2x/wC2n/oxq66X8f5fqc9T+H8zE8Tf8lB0/wD69P8A45Vqq3iYD/hPrA7hkWuNvc/6zn/PrVml9qXqV9legVBrpP8AwiN9kN9xcZQYP71Ojdfwqft1/CoNcI/4RG/AK52pkBjn/Wp1H9RRL4X6MS3R55RU8FvHKhaS7hhOcbZA5J9/lUiivKO4saD/AMjHpv8A19xf+hivQLjG8bdu3HG1iR+GeRXn+g/8jHpv/X3F/wChivQ7oEyEuzZwCN/JP4jr9a78P8DOar8SOe0n/kqEX4/+iTXda9/yLmpf9ekv/oBrhdJ/5KhF+P8A6JNd1r3/ACLmpf8AXpL/AOgGuih/Dn6swq/HH0RyPwx/5if/AGy/9nrWY/8AFVXI/wBm6/8ASa3rJ+GP/MT/AO2X/s9bRRD4kumaREbfOih2xndbQdzx1wPxqaf+7r+up2Uvirf4f1iUqckjx58t2XPXacZqf7BN/ft//AmP/wCKpw0ycxl99vhSAf8ASE6n8fY/lT0OZXWxWaR2YMzsWHQk80eY+Sd7ZIwTnqKn+wTf37f/AMCY/wD4qlGny7SfNtgfT7SnP60aBdkCyyJjZIy46YbGKHmkkGJJGYdcFiam+wTf37f/AMCY/wD4qnNpk6Fg726upwVa4QEH86Wg7ytYbYf8fL/9cZf/AEW1Muv9cv8A1zj/APQBVm2tmgkeSWSAKIZBxOhOSjAcA+pqtdf61ef+Wacf8AFV0H9h+v8AmQ0UUduv4UiAooo7dfwoAKKKO3X8KANvT/8Ajwj/AB/maxK2tPZfsMY3DPPGfc1jEAKDuBJ7elYUmnKVjvxUWqVO66f5CUUUenP/ANatzgCiij05/wDrUAFFFHpz/wDWoAKKKPTn/wCtQAUUUHjvmgAoooPHfNABRRQeO+aACiig8d80Aa+lf8erf75/kKzLn/j6l/3z/OtPSv8Aj1b/AHz/ACFZlz/x9S/75/nWMPjZ3Vv4ECOiilYBWwGDD1FbHCJRRQeD1z70AFFFB4PXPvQAUUDk9ce9FADXRZEKuAysMEHvXNzwXfhrUV1DS2PlZ+ZewHofaumHJ6496a6LIhVwGVhgg96iUeb1KjKxb/4lvjrQ+0dwg/4FE39RXnGqaXc6RfPa3ibWXoezD1FbMttdeHNTS/0yTbCWw2ckKPRsdq7KWCw8b+HVlUCOTBCseWif0Pt/SplH26s9JL8RqXstfsv8DyiirN5Yy2GoyWd1iOSN9rEg4+v071Wrz2mnZnVuFFKoBPLBeCcn+VJQMKKVQCeWC8E5P8qSgAr1fxjHJcaDb3NvG0gt7iO4YIMnaAcn9a8pUAnlgvBOT/Ku08IeL/s2zTdVfdAflilY52f7J9q6sPOKvCXUwrRbtJdDRgnjuYVlhYOjDIIqSn6joJ0uSbUtNkQWW0yT25zx7pgHn2PHuOtQQTx3MKywOHRhkEV16p2e5ho1dElFHrz/APXqRLaeRQ0cMjKehVCQae40m9iOipvsdzz/AKPL/wB+zzR9juf+feX/AL4NPlfYr2c+xDRU32O55/0eX/v2ahIKkgjBHBB7UrNEuMlugoo9ef8A69FIQA4ORwa9G8PfE3xDbaaIpngvPLO1XuEJbGB1IIz9TzXnPrz/APXrX0r/AI9W/wB8/wAhWFeKlHVHZg9ath3ibxJqniPU3l1W4MgjdhFGo2pGM9AP6nnjk1jVLc/8fU3P/LQ8fjUVaxSUUkcs/iZmeIf+QDcf8B/9CFdJ4I/5E2x/7af+jGrnPEP/ACALnn+7x/wIV0fgj/kTbH/tp/6Maqpfx/l+pFT+H8zE8Tf8lB0//r0/+OVaqt4mA/4T6wO4ZFrjb3P+s5/z61Zpfal6lfZXoFQ66D/wh98cNjagztBH+tT+LqPp+NTduv4VX1zB8J35+TcEQcHBA81Oo7/X/wCtRL4X6MS3R57RU8FvHKhaS7hhOcbZA5J9/lUiivKO4saD/wAjHpv/AF9xf+hiu/nx5mR5ZyOseQD+Hb9K4DQf+Rj03/r7i/8AQxXoV0CZCXLA4GA5yW/EdfrXfh/gZzVfiRz+k/8AJUIvx/8ARJrute/5FzUv+vSX/wBANcLpP/JUIvx/9Emu617/AJFzUv8Ar0l/9ANdFD+HP1ZhV+OPojkfhj/zE/8Atl/7PV3UT/xUUw/6ay/+irWqXwx/5if/AGy/9nrR1ONY9YlkkdVZp3VFPVsxQdP++aVP/d1/XU7KSblW/wAP6xI6sx/8gqf/AK7R/wDoL1WqzH/yCp/+u0f/AKC9Bylaiijt1/CgAqzqX/IVu/8Ars//AKEarVZ1L/kKXfP/AC3fj/gRoArVNdf65f8ArnH/AOgCoamuv9avP/LNOP8AgAquha+B/L9SGiijt1/CpICiijt1/CgAqPTNKv8Axh4wtPDGkym3ab57i5xnyowMsfyx9SQKkrV+FGp22kfG1DfSiFb+3a2iLcAsVQrz05KkD3rDEScad0a0UnPU9f0z4aeCtB06HS5VluJIxjzZ533kkk5O3Cjk+leQ+P8AwLJ8Ntcsvsl3Nd6HqLMkRmwXgk4+UnHI9Dx344yfe9Q0S7m1GSSEKySNncWxt+teYftGajbJpfh3QhMDdm5FwVHVUClAx9MknH0PpXzGXYjEyxE1OHKk9H33+89rGWlRinNystu3p2PPaKKPTn/61fXHgBRRR6c//WoAKKKPTn/61ABRRR6c/wD1qACiig8d80AFFFB475oAKKKDx3zQAUUVoaHpEmua9Z6ZAwVrmQLvx90dSfwGfyobSV2CV3Yn0r/j1b/fP8hWZc/8fUv++f519B23h3wr4atkshpsVw+Ms80Yldj6kt0+grifiZ4H06y0tfEGgp5MTOBPCp+X5ujDPTngj37V5dDH0KteVOL1PSr0pqhG62PLqKKVgFbAYMPUV6h5olFFB4PXPvQAUUUHg9c+9ABRQOT1x70UAFFA5PXHvRQA10WRCrgMrDBB71hKbvwpqP23T8vaOcSxHpj0/wADW+OT1x7010WRCrgMrDBB71Mo31W407aMs6lp2n+NdHS7snC3Cr8j91P91q80u7Sexunt7qMxyocFTXTxfavC+qLd2L/6FIwEqNnao98ZP4gV0mpabp/jXR1u7F1W5UfK/cH+61Zzgq3lJfiVGXs/8J5dRU9zZzWV3JbXa+TJHnIYH/PNQVwbaM6gopVAJ5YLwTk/ypKBhRSqATywXgnJ/lSUAd74I8UAgaVqco6Yt3bv6qT/ACo1vTJvDWoNe2Me7TJiDJEv/LI+v0rhYlDFiZViKruBbPzH0GAefrx7133hTxXHfQjStaKszDYkj9JB/dPvXbSqqolTk9ejOacHF80fmSwTx3MKywsGRhkEVcaNphaRxjc7pgD1O9qz7rwrqmmX0j+HzHLayc+TM2Np9BWjFba5HorTfZIv7QFuIooQwwGaR95J3Yx5eB1zl/Y10Xkk00whytSaa2/VGffarJp+oLpem2MF5fyoW+0TPmKIDdkgHAwACxdjtAzxgZPRReC/E89kvl+JdPF9JCJl09rNVXB7lWUEHtkpnscVydpoXiebWVvJ9LjY/Z5LdoknRC0boyPhskKQrscngYyQelewWfh3TE8Qwau9ys2vLYqhLSbQ/wAu3zDH1Gf8818jneOxOEmuSTimu17vz7Lz+6/T1MDQpVou6v8A1+Z5mdI8R3MLWWo2cel6jExxeBo/KkHHDooOOM4ZBzxkc7h0DaYg0i4cQ2lxqEsxMZneVRGplfkhcAgJsIHXO7PpWY3jW91Dxn/ZV/pBtblrhIJSZ9/ljcqbuEAOAc54B9uKX/hIb7+0J7c6JN5UbMEn89FVwHK5y2AM7SQM549Oa+kw7hOhGU5u7Sbtte3TTY4Z+1ipKMFa/Xfrbr95zuoaN4rj1O6TTw01osziCRjCC6ZO0kH1GKr/ANk+Nf8Anj/4/D/jWw3ji9V3U+HbgFezOwJ5xj7n+cUz/hPL7/oXLj/v43/xFbctH+d/j/kc16n8q/r5mV/ZPjX/AJ4/+Pw/410fh601uDT5F1VNs3mkryh+XA/u++apDx1ekNnw7OMDIBkbn2+5WxpGvz6naPNLpslqVkKBGYkngHP3R61hXjh3D3qkkvK/+R35fLEqvejTjJ22drfi0cxfaZ4wfULhraHMLSsYzuh+7k46nPSoP7J8a/8APH/x+H/Gtu58a3lvczxLoM0gikKAh2+cAnkfJjt696h/4Ty+/wChcuP+/jf/ABFbRjQsrTl+P+Rx1HV53eKv/XmYtxoPjC7gaG4tt8bYyPMhGec9jXb+FrG403w1a2l7H5c0e/cu4HGXYjkcdCKwJPiBcwwmSbQZY1BA+eVh199mP/110+ian/bOjwX/AJPk+du+TduxhiOuB6VvQVLnvGTbt1/4YwqufL7ysjlPE3/JQdP/AOvT/wCOVaqt4mA/4T6wO4ZFrjb3P+s5/wA+tWaj7UvUv7K9AqHXd3/CH32Q+3amCeV/1qfkfbvU3br+FV9cKnwnf42bgiZ4Ib/Wp+BH60S+F+jEt0ee0VPBbxyoWku4YTnG2QOSff5VIoryjuLGg/8AIx6b/wBfcX/oYr0C5wJjgKOOQFK8+47H6cV5/oP/ACMem/8AX3F/6GK9CuFw3XGFGASGyPYjg/5Hau/D/Azmq/EjndMYJ8ToS3Qkj84iK9D1C3N3pt1bLgGaF4xn3BFeca9ZTw3UWq2GRLEQzY6jHQ12PhnxNDr1rtfbHdoP3kfr7j2rehJJypy6mNVNpTXQ4rw1qj+FNcnttThZElwkvqhB4PuOTXc+INGTxDpkTW82yaP97byqeDkfyNM8S+GoNetdy4ju0H7uT19j7Vh6Lq83hjSLe31ZTsN3JC+4ndGAqkEe3J/OnGLpJ05/CbQhKspVKe8Vd+l0v1Kun6hL57WGpJ5N5FwQf4/cVuR/8gqf/rtH/wCgvVnXtBt/EFmlzayKl0o3QzqevsfasvTJbg6DcxahH5VzDdxxPuONx2SEY9eAfyqZJwdpfJmaakrodRRR26/hTEFWdS/5Ct3/ANdn/wDQjVarOpf8hS75/wCW78f8CNAFaprr/XL/ANc4/wD0AVDU11/rV5/5Zpx/wAVXQtfA/l+pDRRR26/hUkBRRR26/hQAVk+INPa8sRJACZ4TuXHUjuP6/hWtR26/hSlFSVmNOzudD4I+Leu3VhHpdzqsq3US7Q0oV2lHruYE5/GjxJpaeJ4H/tGRpLknclwxyyt9fTgcVwGs6M/mfbtOyk6HcQvGfce9dX4X8TwatZw291PGuoAENHggtjvyMZ74FKjyJ+zmld/iFTmfvxf/AADnba5udNvTpesDbMv+rlPSQfWtWt3XNDttcsjDONsi8xyjqhrj4J7jTLxdM1dh52Pkk5wwyQBk9+/Hr605RdN2ewotTV1uadFFHpz/APWpgFFFHpz/APWoAKKKPTn/AOtQAUUUHjvmgAoooPHfNABRRQeO+aACtnwjqkOi+LtO1C6yIYZh5hHZSCCfwBzWNQeO+aUkpKzGnZ3Ppi90sas6XtjcxPHKgIbOVYdiCPauR+K2p2emeDYtCWYSXc7odgPKop3FiO3PA/8ArV5FZ61qmnRmPT9SvLVCclYJ2QH8AaqzSyTzPLNI0sjHLOzElj6knmvHw+UUcPiJYiO7O6rjZ1KapsZRRSsArYDBh6ivZOASiig8Hrn3oAKKKDweufegAooHJ6496KACigcnrj3ooAKKByeuPeigBrosiFXAZWGCD3rDVrvwpqP23T8vaOf3sR6Y/wA963hyeuPemuiyIVcBlYYIPeplG+q3Gnbcs6jpun+NdIS7snVLhR8r91P91q41vA/iAMQLAN7iaPn/AMerQgtNU0PUzPoTK0Tglo5D8vToRn8q0f8AhJvFf/QNsv8AP/A6iSp1NaiafkUueGkWreZzv/CEeIf+gf8A+Ro//iqP+EI8Q/8AQP8A/I0f/wAVXRr4l8Vk86dYrwTk5/L79J/wk3iv/oG2X+f+B1HsqH97+vkPnq+Rzv8AwhHiH/oH/wDkaP8A+Ko/4QjxD/0D/wDyNH/8VXRr4l8Vk86dYrwTk5/L79J/wk3iv/oG2X+f+B0eyof3v6+Qc9XyOd/4QjxD/wBA/wD8jR//ABVKPBPiIEEWGCOh8+P/AOKrpI/Efil926x0+PapI3bvmPoMOefrxTP+Em8V/wDQNsv8/wDA6PZUf739fIOer5EtnP45s7VIDp1vOEGA80ilvzDitf7V4lbSd62Nqt9tVtjN8h+Zwy8N1x5RHOPv5OcCsQeJfFRBzp1iMDIBzz7ffqefxF4ljhtGj060ZpIi0oJ+629hgfP/AHQp79a6IyjbeX9fIUYyak7R/prbU2fDVx4vvPEVrbX+nWdvayllmlUhiqFTnAEh5xwODg9jXdjw3pi+KTqq3K/26LUxoWbIC42hzHnJ4wDyAfauQ0PWbtNat70vai0WF3ktmjkNxG+0hU+TdvGcHcqnjqo6Uf2/pf8Awlza/wDYov7W2iBU8292kYIJI+ycN0GPc9xz8Tn1HFYjE2oqTjydut3p007/AJHs5fKlTp+/a9+/S2/qYN1bPH8W2uLmbzXnYGNmjCMwjlMBLAcZJiJyOCDnAzgaVy1wxW3ZBLBJOSv7vlSHbjIxnqf++qh1COaXxZHrepTriGL/AEdArb5N0rS7SGCsoXeUBYAkLnA6VdQJE0rpOwZmZmT7YsYJ8wjbt4P3ec5r6fLYzhg4RmrP+rHBiXFzk4vS/wDmY95n7dPufzD5jZfGN3PXHaoaknRUuZURwyq5CsDncM+tR13nEFa+lf8AHq3++f5CsuKJ55ViiUvI5CoijJYk4AHvXq2g/CO/GmhtT1CK0mc7vKSPzNvA4JyBn6ZrnrzjGOrOzB6VLnlVz/x9S/75/nUddP4y8Faj4Vu/Muyk1tPIfKnjzgnk4I7H2/LPOOYraElKKaOaompO5meIhnQbj22/+hCuk8Ef8ibY/wDbT/0Y1c54h/5AFzz/AHeP+BCuj8Ef8ibY/wDbT/0Y1VS/j/L9TOp/D+ZieJv+Sg6f/wBen/xyrVVvEwH/AAn1gdwyLXG3uf8AWc/59as0vtS9Svsr0Coddz/wh1997AVO4IH71PxH9fwqbt1/Cq+u7f8AhEr77u7Yn8JDY81PwI/WiXwv0Ylujz2ip4LeOVC0l3DCc42yByT7/KpFFeUdxY0H/kY9N/6+4v8A0MV6Bc8TY4zjnCbfzHY/SvP9B/5GPTf+vuL/ANDFehXH3vn3KQo2jdvX8D6fnXfh/gZzVfiRARkYNc5qWmz6ZdDUtJZkKHcyr/D/APWro6MAg59PTrWsoqSIjKxpeGfE0GvWu1sR3aD95H6+49qk8TaAmv6aIg+yeIlom7Z9D7GuG1LTZ9MuhqWksyFDuZV/h/8ArV2vhnxNBr1rtbEd2g/eR+vuPatKdTnXsqm/5kSi4e/DY5Hw94huvDN+2maurC3DYIPWI+o9q7bXdMGu6G0VpMsbO6XEUqgfMyg7eeuMMfzp2saFY6tCzXFpFNcKjCJnZlG7HGSpBIzim2kcPh6WbSnbzBDbtexNbROyPDuRONzswZT5hKjIwM+uZdqS9nVd4vY1UFUh7WEkpLprf8rfiYOmWWtSQsl7pV4rx8ecLdjHIM4yGAx1q7/Zt9/z53H/AH6b/CtXU9Mt9Ut1nSKKWdFDQOzso9RypB/HtmqX2UQvb2t4kJvmtluZxGknlIrSbOHLckblz7k80OLpu0nvsOMYzhzJ69Vr/kV/7Nvv+fO4/wC/Tf4Ual/yFbv/AK7P/wChGrU8EanbFaJuM8keHZuAu3rz7nJqRLS1l80wrAwACx5dgXfbk4BOTzwB/tD8Xp3Fyrv+f+RkVNdf65f+ucf/AKAKPPj/AOfWL83/APiqZLJ5r7yApwBtHQAAAfyp9BOyja4yiijjHvUkBRRRxj3oAKKKOMe9ABWDrOjP5n27TspOh3ELxn3HvW9WhpbhWlUSIglidHZ2C4ypwBk85OPyFTOKkrMqLaZX8J+K11dBZ3x2XyD6eYB3+tauuaHba5ZGGcbZF5jlHVDWJqEDp4fmls5RlbYMltG6j9+LlWMoXOd/lhl3YyVOM44rftvEOkapdSR2FxGr+fJGkBkBZgh+8PVSDkH69cZqqdXm/dVf+HInTt78Diba5udNvTpesDbMvEcp6SDtzWrW5ruh2+uWJhlAWVRmKXHKH/CuQ0e5kdZ7O5YvcWbmORsYB5IH16e1JxdOXK9ug1JTV0adFFHHGPxpgFFFHHGPxoAKKKDjtQAUUUHHagAp0cbTSBIxlj74x7/Sm1LbzCCcPjcpUqw6cMpB/maAC4gNvLsMkcnGd0bhh+YqKr9rdW1rI4R7kK2397GQj8Hkdehz69gadc6kssLQwCRIWjI8on5VYyb+B9OM0AZ1TLayNb+dwF5wCeWA6kfTP+cHFu5vYb9T9qmuARM7phA+FbGBywxjHSo1uoEjKjzHZI2jiJUKCGzkkZPPJ+vHTFAFee3lt9gmQozruCsMEDJH9KWG1knjd0xhSBycbickAe+Af/14qyLm1SSzdTK/2cYKsgXd8zN1ye5A/wA4p41CD7QztHIUSXzYRx8pxjb9OF/7596BlU2b/ZDcCSNguNyq+WXPTIpLa0e5V2DxxpHjc8jYAz0qZJbT7EIfNnjLENLiENuPbncOB/n2dbz2tq/yTXDIx+dTEuGA7FSSD169vQ54BFFhtYqcHBxwcinRRNNJsTrtLfkCf6Vdk1FW09rdBIgOVEe7KAb9wP1HT8akbVU/tZbstPMis7BJD90MD8o5PHv+lAGWBk4HJp80TQTyRPjdGxU49QcVfOp405bOOWVI0fgbFO4btwPqDVW+uftd202+Rt5zh/4Oc7Rz0H4fSgCvRQMZ56UUAFFAxnnpRQAUUDGeelFABRQMd6KACigY70UAFFAx3ooAKKUbedxI44wM80lABUq3MioqfIQowN0anHOe496i45z+FFO7Ww1JrYm+1Sf3Yv8Avyn+FTf2vfGHyjP+7xt2bVxj0xiqfHOfwop8zK9pPuTfapP7sX/flP8ACondpJGdzlmOSfU0nHOfwopXbE5Se7CijjnP4UUiTpPh6Yh8QNI88KV8/jd/e2nb+OcV7V4h+0fbxnf5W0bMdPf9a+dIpXhlWWJ2jkjIZGU4IIPBz2r0rTPjXqFtZrFqelxXsqjHnJN5Rb3I2kZ+mK8TN8unj6ahGVj0MFio4eV5I6j4lkj4Xn7aP35ki27+obd/Pbn9a8KrpPFvjbUfF8yfbNsFtC2YraP7oPqT3Pv/ACrlLy8isbdpp2wo6DuT6V6GEovD0VCT2OavUVWpzIo+JHCaHKpIBcqB7/MD/Suo8FoU8H2IPo5/N2Nczoehy+Kbg6jqheKzjYCGILxIOc8/l9fwrvv3NnbfwQwxL9FUCu2hFuTqPY5qrSjyHFeJv+Sg6f8A9en/AMcq1WS95Hr/AIvm1C3ZhDaqI4/l++MMM9cjkkjjp6VrVnF3cmu5bVkkFQa5u/4Q+/8AvBdqcBsgnzU5I7H+dT8Y96r66R/wiV8ON2xDymDjzU79x9acvhfoxLdHntFTwJaMhNzNNG2eBHCHGPqWFFeUdxY0H/kY9N/6+4v/AEMV6Bc487jrjB+TYc+46D8K8/0H/kY9N/6+4v8A0MV6FcbS/wB7oo2gNuGP5gex5rvw/wADOar8SIKKKOOc/hW5kBGRg1zmpabPpl0NS0lmQodzKv8AD/8AWro6MAqc/liplFSRUZWNLwz4mg1612tiO7QfvI/X3HtU+vaIdTjSeznktNQgB8i4icqw9sjtXB6lps+mXQ1LSWZCh3Mq/wAP/wBau18M+JoNetdrYju0H7yP19x7VpTqKovZVd/zM5RcHzwOY0nxbqekavJZ+JZZ7hWYB5J3Luh6ZyeSK6/XbGTXtCENlfSxDIli8uUhHPUZA4PQc9sVB4l8NQa9a7lxHdoP3cnr7H2rkvD3iC48M3j6brHmCIOBtK58vrkg56dOMd/bmP4X7uprF7Mr4/fhuXrHXb9r57bVJpYtRj+UsXILj2q+08ryb3ldnzu3FiTn1/QVf17QbfxBZpc2sipdKN0M6nr7H2rm9Pv5POew1JTDfRHBUjhx60NODs/kxpqSujSooo4x70xBRRRxj3oAKKKOMe9AF7TYYZS/nxeaBnA3FeiO3X/gIq/Db6Y7bZoEibYzFPtLMwwCecLx0+tReHP+Qgn/AF0H/oElV7KKSXVJYwGeZkmUJjJZijDA9TTlKyNrtJWX4eZJO+nQMB9g8xT0eO6JB/Tr7VF9q03/AKBr/wDgSf8ACtSLwh4kPhybzLC2+0meNorcXXzbfmDEkqAOqnr0B74rmrqRbTKSCQzrJ5RhC/P5mcBAOu4+lceHx+GxHN7KafLuaVKdanbnja/kjR+1ab/0DX/8CT/hWHq174d0q8iun8PSyzysWDx6i8RUjHPCn1re/wCEN8Ym0F2NNtFAUE2huczH6HG38M1x2t2V9rP2JdP069mnxKXgWBmdNrBTwM8AjGfXiso4/DYiMvYzTsaxjXoyTlG1+6X+R6hXH2d/or6xqkcWiSRzRzkTym9Y+c25vmC7fl5ycc9a9FHhzUPK37EDf3N4z/h+teT2cRh8Ta6jhlcXR3Ky42ne/H5YrqWOw2JklQmpW3s0zlhTrUIvnja/df5nR3sdsbOOa2g8neFOC5Y9XH/so7Vn1fuP+QRb/wC6v/octUOOMfjXQyZ62YUUUccY/GpICiig47UAFFFBx2oAKKKDjtQAVqxXc1ppAeBwpyg5UNjJlz1+g/KsqtTzTBpAaMIc+WPnQMOsvYg1S2ZpTbSbXb9SL+27/wD56p/35T/Cj+27/wD56p/35T/CoP7Rl81IkiiklkOEijtEd3+ihST+FJ9vuzL5aW1qZSdot2kshPu/uGLfvD542YznjGa56mJp0nac0vVlx9tNXjdlj+27/wD56p/35T/Cj+27/wD56p/35T/CoP7RmErxPFFHLH9+KSzRHT6qVBH40r6g6ZIFvsAzua2jH9K2U+ZXT0Ic5p2bZN/bd/8A89U/78p/hR/bd/8A89U/78p/hVUalc+XG7xafF5saSqs1xYxttZQykqzgjIIPI70ranLuO3+yyM8E3mnj/2pXN9ew/8Az8X3mvJX8yz/AG3f/wDPVP8Avyn+FTS3c13pBedgxy44UL0MWOn1P51nnUrny5HSGwl8uN5WWG4sZG2opZiFVyTgAngdqutcC50bzFMbKS+1o0VQRmLBwvFbUq9Oq3yST9GH72Px32f5GbRQMZ56UVZzBRQMZ56UUAFFAxnnpRQAUUDGeelFABRQMd6KACigY70UAFFAx3ooAKKUbedxI44wM80lABRRxzn8KKACijjnP4UUAFFHHOfwooAKKOOc/hRQAUUcc5/CoLy8isbdpp2wo6DuT6UN21YBeXkVjbtNO2FHQdyfSqGi6Lc+Kb0X2oho7CM/In9/2H+NP0PQ5fFNwdQ1QvFZxsBFCF4kHOefbjtz+Fd9+5s7X+GGGJfoFAop0/a+9L4fzCc+TRbh+5s7X+GGGJfoFArhNW1WfxTdm1sy0WmRt879DKaXVdVm8VXD21m7w6bE4Dvj/Xf544/+sDbggjtoVihUKijAApzn7T3Y7fmKMeTV7hBBHbQLFCoVFGABUlHGPeikMKg1w/8AFH34B42pkCTjPmp/D6+/Sp+Me9Qa6P8AikL4kfwrglMZ/ep0Pf8ApSl8L9GC3R55RU8CWjITczTRtngRwhxj6lhRXlHcWNB/5GPTf+vuL/0MV6DdAib5t27HO5drE+49a8+0H/kY9N/6+4v/AEMV6DPt3dQMAbQh3L+pyPpXfh/gZzVfiRDRRRxzn8K3MgoqxFp97PGJIbSeRG6MkRIP44ps9pcWoH2mCWEt93zEK59ev4UAQkZGDXOalps+mXQ1LSWZCh3Mq/w//Wro6MAqc/liplFSRUZWNLwz4mg1612tiO7QfvI/X3HtS+JfDUGvWu5cR3aD93J6+x9q4nUtNn0y6GpaSzIUO5lX+H/61dr4Z8TQa9a7WxHdoP3kfr7j2rSnUU17Krv+ZnKLg+eByPh7xDdeGb9tM1dWFuGwQesR9R7V12vaDb+ILNLm1kVLpRuhnU9fY+1L4l8NQa9a7lxHdoP3cnr7H2rkvD3iC48M3j6brHmCIOBtK58vrkg56dOMd/bmP4X7uprF7Mr4/fhuT6fqEvntYaknk3kXBB/j9xWnWlr2g2/iCzS5tZFS6UboZ1PX2PtXN2F/J5zWGpKYr6IkMpGAw9RQ04Oz+TGmpK6NKiijjHvTEFFFHGPegDQ0w7Y5iOoVj/5CkqfQ9a+xeJdNmuLWCRGmKS3BDB4wysN+dwXqeSQTg+tV9O/1M/8Aut/6Kkqhxj3rLE0Y16MqUtpKxvGbpyhNdP8AM6E+DfE3/C2l1zeDpv2ozCf7QOIsfd25z046Y4rnZbiK2+MDarMPL0gamDkn5NxXHm46Y3c5qv8AZHCmMX9+LctuNv8Aan8snOemfXmq3iML/YE5BJPy5GOnzCvBo5TUhGTrTT93lVlbTu9Xqdk8ZFtKC6317nZt4N8TD4tDXS4/swXRm8/7QMLFj7u3OenHTHFZmneKrOX40T3g1OC00wSmBVKn9/8AK4+UhSMGRi5yQCSDyRVl7u4f4atG00nltpeSm87f9Vnp9a8fyFKlCc9TkdDn/wDVWX9kzpfxpp3jyqytp3er1/ItYyM/gjazvq+v+R7WfBfij/hbi61uzp32syib7QOI8fd25z04xjHFczfzw3Pj/wASS22DGblQCDwSAQT+YNduJ5RF5QlcR/3Nxx+Veb6fj/hItcwST9qbOR0O967sJldTCVo1Ks1Ky5VZW0Wt3q9Tmq4yNaDjGNtbvW50lx/yCLf/AHV/9DlqhV+4/wCQRb/7q/8AoctUOOMfjXtvZHLLZen6sKKKOOMfjSICiig47UAFTxW6vD5ssvlpu2g7c5NQVdgZHt0XfEGjYkLLwCCP8amTaWhrSipS1I0tPN80wSbxGAR8uN1EVn5nkfPjzt3bpirDTJA900DIOU2BSMH1p/2iF7i1dWVQN5YE/dJFZ80v69DpVKl13/8AtrfkUpbdUhEsMvmpu2n5cYNWJSF0MFjgBoySe3MtRSBYLNo/MSR3YH5DkACpLqNZdBMZztcRqce/m5rWLbizGUUr200/U2/A+kxX2lyXWqBlt1i826/eYVxIN6wtjGUWIxuezGTB+6RWmfiXo6eCxPJ4bnOjOnlRqkCECPdsIYfdA/hC7vwHGa3w5uINW0Q6ROfJGpQrDu3DKyRxJbyR49Qsccg9RLjHyk1zP/ChdZ/t5bUXaGxMpU3e1flTyt4JXdnk5T8M9K/P4xp4jE1XjpWa0V3bve3dp6ddrdD27yp04+wXr+ht+ONIjsNLjutK3NbmIy2v7wFVCDe0IyCQhiEjLyAvl4H3gK5XUbK5OlXxiXPlWplds8KpXP8AI13HxAubXSNLi0kEyxaankyMuGYtJG9uiBcjc2ySSTHHEQBxvBHGyvYw2c6ySXCW8dibcMsSs+0Q+XuxuAz3xn8RXt5BUryw0lU2W1/67Wfne/U4cwjBVU4nf/DuVIPDNxLKwSONbVmY9gLC2JNRx/FrT5PDNxqn9i3Asop40E25fNYHcCdvTggD73epPA9nDfeGL+yjkcwXC28KyMoVirWFsMkZIBwemT9TXBWfwG8RTXTwXVzbQhXKiaNhJGflBB6huScfd4xXzOEoYOrXrPEvtbW3TW3mepVnWhCHszvfiJIk3hm3liYPHIt0ysO4NhckGuC0r/kT7X/rmf5Q13vxDsF0vwtaWCMXW2iuIgWGCdun3IrhdPCjwtCIySu1tpI7fua97haPLTmv63kcWZO80/J/kVqc0bLGrsMK+dpz1xREUWRTKu5O4zjNXJ5rc2cQWIHhto38pzX1jbTWh5EIRlFtvYjW1jEEck0/l784Gwmo3tmXydh3eaOOO/pV2CaRVi3T24jAGV3c4/xpsN1EsEhO0tGxMQPvWfNI6PZ02l0/r19SjNGIpmjDbtpxnGKZS53NlieTyaStkcbs3oFFAxnnpRQIKKBjvRQAUUDHeigAooGO9FABRSjbzuJHHGBnmkoAKKOOc/hRQAUUcc5/CigAoo45z+FFABRRxzn8KgvLyKxt2mnbCjoO5PpQ3bVgF5eRWNu007YUdB3J9KoaLotz4pvRfaiGjsIz8if3/Yf40aLotz4pvRfaiGjsIz8if3/Yf416D+5s7X+GGGJfoFAop0/a+9L4fzCc+TRbh+5s7X+GGGJfoFArhNW1WfxTdm1sy0WmRt879DKaXVdVm8VXD21m7w6bE4Dvj/Xf544/+sDbggjtoVihUKijAApzn7T3Y7fmKMeTV7hBDHbwrFCoVFGABUlHGPerKaZfyIrpZXDKwyrCJiCPXpS2GVqKkmt5rZtlzFJC5GQroVJHrzUdABUGuY/4RG/x1KpnD/8ATVOq/wBelT8Y96g13P8AwiF7nIGxcZXr+9TofT2pS+F+jBbo88oqeBLRkJuZpo2zwI4Q4x9SworyjuLGg/8AIx6b/wBfcX/oYr0G6JM3zBw2BkOoDD6+v17159oP/Ix6b/19xf8AoYr0GfYTwRwBtCElevvyK78P8DOar8SIar6rqceiaWJ3iWW5ugy2qN0TaRmQjuOcAcgnOemDYrG8foAmgsRtDaaxGF+832mbOfwx+lOtJxjoKnG71OUury5vrhp7yeSeVjkvI5Yn8TVvS9ev9IYi2mLQPjzLdyTHIPQj8eo5HUVnUvy7TkndkYGOMfX8q4btO502R6WJIbq1gvrQYt7lS6LnPlnOChPqP5EHjNNqh4act4PgB6Lezgf98RH+tX+Me9elCXNFM5JKzsBGRg1zmpabPpl0NS0lmQodzKv8P/1q6OjAKnP5YolFSQRlY0vDPiaDXrXa2I7tB+8j9fce1L4l8NQa9a7lxHdoP3cnr7H2ridS02fTLoalpLMhQ7mVf4f/AK1dr4Z8TQa9a7WxHdoP3kfr7j2rSnUU17Krv+ZnKLg+eByPh7xDdeGb9tM1dWFuGwQesR9R7V12vaDb+ILNLm1kVLpRuhnU9fY+1L4l8NQa9a7lxHdoP3cnr7H2rkfD/iG48MXh03VhJ5IchlI/1XoR6g88VH8L93U1i9mV8fvw3HQ619jZrXWla3uYjgkqSG9+Km/4SHS/+fr/AMht/hXVT33hq+Iku7jS5zjgzPGxA/Gof+KR/wCoJ/5Bp+zktpIXOuqZzf8AwkOl/wDP1/5Db/Cj/hIdL/5+v/Ibf4V0n/FI/wDUE/8AINH/ABSP/UE/8g0ck/5kHNHszGsvFej24fzLkncenkuQRtZSD07NUv8AwlPhz1T/AL8S/wDxdan/ABSP/UE/8g0f8Uj/ANQT/wAg0+WXdf18zT2sbWcTL/4Snw56p/34l/8Ai6r33iPw3cWbxNFHMGx8jW82Dg57Sr/MVuf8Uj/1BP8AyDSg+EQcg6KD6jyqTg2rNr8f8xxrQTT5PzLe+1Twp5ksH+iLZZaHaw+TZ93G/PTj7+f9rvXn51PwxnjSIcf9e8//AMl10Pi7xZZR6XJYadLHcyXCFGeNwVRTweR39q84OAF2kk45yOhrnxU4cyS1t6m+FkopuUE797/5nvNclbXmiPq2ppb6eEnScidysnztubJ/12OuegXr09OtrzrT9v8AwkWubSSftTZyOh3vXbXtzROOhJJSurnRXV1FLbrHEu0LgKoTAABY92J6tVKijjjH41mXKVwooo44x+NIkKKKDjtQAUUUHHagAoooOO1ABV9/+QIP96P+ctUK0Zsf2Mu3pmL/ANq0+jLjtL+upQhvL3TiZdL8gSFw7JMhKuQMA5GCrDsykEdjV8ePfE/liJv7SJ27SRqUQQ8Y+95PnY7Z8zfj+PPNZ1BxnjpXn4jL8NiZc9WN2a08TVpK0WDXF3eBH1EW+9DlFhU7U4xncxLMx7sxJOetRzxefbSxE4EiFc/UYqSlbaG+Qkj1IxXZCnGnBQitDGUnJ8zLWkePG8JabDbTrJCZAnmZs1uA7RxpFlWE8ZUFETgqTnPPYelr4rdOI5nRc5wIO/8A38rwTxf/AMuf/A//AGWvVa8inkWDqVZyku3b/I655hXjCKTMLxL41k8SC/01fMnktZJY0ItRAEZkeHLuZnLKEd+AoJOOnQ17OH7P4ajh3bvLDJuxjOPJGa57Tv8AkY9e/wCvtv8A0N66Zcf2K2Om6T+cVdmAwVLCRap9b/qTUrzqy97t+hn0UDGeelFdxyBRQMZ56UUAFFAxnnpRQAUUDGeelFABRQMd6KACigY70UAFFAx3ooAKKUbedxI44wM80lABRRxzn8KKACijjnP4UUAFFHHOfwqC8vIbG3aadsAdB3J9KG7asAvLyKxt2mnbCjoO5PpVDRdFufFN6L7UQ0dhGfkT+/7D/GjRdFufFN6L7UQ0dhGfkT+/7D/GvQf3Nna/wwwxL9AoFFOn7X3pfD+YTnyaLcP3Nna/wwwxL9AoFcJq2qz+Kbs2tmWi0yNvnfoZTT9S1STxVcyW1u8kGmQ9WVf9e2Rx14GM+v61ZggjtoVihUKijAApzn7T3Y7fmKMeTV7hBDHbwrFCoVFGABUlHGPeikMz9d1v+xbdYLXH9oTKJBLnm2XnH/AiORkcDaR144iWaWeVpJ5HlkY5Z3Ykn8TXQeO0iXxQ5jJw1paEDHBP2eMNyf8AaBrnK82pJylqdcYpI19D8QT6RcIsu65sS4Mtqzna3uPRvcfTpmu0IUxxSxNvhmjWSN8Y3KR/MdD7givNfl2nJO7IwMcY+v5V6PZbR4V0HH3/ALG+/r1+0S4/8d21tQk78pFWKtcdUGuEf8IjfgFc7UyAxz/rU6j+oqfjHvUOug/8IffHDY2oM7QR/rU/i6j6fjXXL4X6M51ujzuip4EtGQm5mmjbPAjhDjH1LCivKO4saD/yMem/9fcX/oYr0G7z53zbslQfmA5/Efe+vevPtB/5GPTf+vuL/wBDFegTbSxOUzgY8scN69en5V34f4Gc1X4kRVHqGkQ+INNW1kuEtru3LNaSScI+7G6Nj/COMg9Ad2fvZElHHOfwrWcVJWZnGTi7nnl5Y3Wn3DW99bS20y9UlQqR+BqfStHu9XmZbaM+VHgyzMDsiH+0ffHA6ntXpFrrGpWMfl2Wo3dun92KdkH5A+5qO6vru+YPfXU9y46GaQufzJ9hXMsPrq9Db2uhXSGKztIbG1bfBbAqr4x5hJJLke5P4AAdqKKOMe9daVlZGG4UUUcY96AAjIwa5zUtNn0y6GpaSzIUO5lX+H/61dHRgFTn8sVMoqSKjKxpeGfE0GvWu1sR3aD95H6+49qk8QeGbTX4h5h8m4X7syrkgehHeuH1DS7jTrxdR0csrodxROo+g7j2rbsviRai3VdStJ1mUYYwhWDH15Ix9KqNaLXJW/4ch02nzUyH/hWP/UX/APJb/wCzo/4Vj/1F/wDyW/8As6v/APCyNI/59r3/AL9p/wDFUf8ACyNI/wCfa9/79p/8VS5cJ/Vx81cof8Kx/wCov/5Lf/Z0f8Kx/wCov/5Lf/Z1f/4WRpH/AD7Xv/ftP/iqP+FkaR/z7Xv/AH7T/wCKo5cJ/Vw5q5Q/4Vj/ANRf/wAlv/s6P+FY/wDUX/8AJb/7Or//AAsjSP8An2vf+/af/FUf8LI0j/n2vf8Av2n/AMVRy4T+rhzVyh/wrH/qL/8Akt/9nR/wrH/qL/8Akt/9nV//AIWRpH/Pte/9+0/+Ko/4WRpH/Pte/wDftP8A4qjlwn9XDmrlD/hWP/UX/wDJb/7Oj/hWP/UX/wDJb/7Or/8AwsjSP+fa9/79p/8AFUf8LI0j/n2vf+/af/FUcuE/q4c1c66uOufAk8upXV1b61Jb/aZWkKpEeMknGQ4zjNP/AOFkaR/z7Xv/AH7T/wCKo/4WRpH/AD7Xv/ftP/iq1nUoT+JmcY1Y7Ih/4QO+/wChjuP+/bf/ABdH/CB33/Qx3H/ftv8A4upv+FkaR/z7Xv8A37T/AOKo/wCFkaR/z7Xv/ftP/iqzthu/4sv99/ViH/hA77/oY7j/AL9t/wDF0f8ACB33/Qx3H/ftv/i6m/4WRpH/AD7Xv/ftP/iqP+FkaR/z7Xv/AH7T/wCKothu/wCLD99/ViH/AIQO+/6GO4/79t/8XR/wgd9/0Mdx/wB+2/8Ai6m/4WRpH/Pte/8AftP/AIqj/hZGkf8APte/9+0/+Kothu/4sP339WIf+EDvv+hjuP8Av23/AMXR/wAIHff9DHcf9+2/+Lqb/hZGkf8APte/9+0/+Ko/4WRpH/Pte/8AftP/AIqi2G7/AIsP339WIf8AhA77/oY7j/v23/xdH/CB33/Qx3H/AH7b/wCLqb/hZGkf8+17/wB+0/8AiqP+FkaR/wA+17/37T/4qi2G7/iw/ff1Yh/4QO+/6GO4/wC/bf8Axdalr4f1C1tY4Bq+7y12+ZidGYbiedkwBxuPaqP/AAsjSP8An2vf+/af/FUf8LI0j/n2vf8Av2n/AMVTTwy2f5mkKmIg7x/JP80af9j6l/0Fm/7+XX/yRR/Y+pf9BZv+/l1/8kVmf8LI0j/n2vf+/af/ABVH/CyNI/59r3/v2n/xVVzYbyNPrGK7L/wGP+Rp/wBj6l/0Fm/7+XX/AMkUf2PqX/QWb/v5df8AyRWZ/wALI0j/AJ9r3/v2n/xVH/CyNI/59r3/AL9p/wDFUc2G8g+sYrsv/AY/5Gqml6rHny9ZkXPXbNdjP/kxWzXI/wDCyNI/59r3/v2n/wAVR/wsjSP+fa9/79p/8VVRqUI/C0Y1ZV61udbdkl+SRtXlhqVzOWXVXSPPyJ5lzhR6YWcD8gKpXXh/ULq1lhOr7TIoXzMTuyjcG+XfMQM7RVL/AIWRpH/Pte/9+0/+Ko/4WRpH/Pte/wDftP8A4qp5sP5Gv1jFWtZdvhj/AJEP/CB33/Qx3H/ftv8A4uj/AIQO+/6GO4/79t/8XU3/AAsjSP8An2vf+/af/FUf8LI0j/n2vf8Av2n/AMVUWw3f8WYfvv6sQ/8ACB33/Qx3H/ftv/i6P+EDvv8AoY7j/v23/wAXU3/CyNI/59r3/v2n/wAVR/wsjSP+fa9/79p/8VRbDd/xYfvv6sQ/8IHff9DHcf8Aftv/AIuj/hA77/oY7j/v23/xdTf8LI0j/n2vf+/af/FUf8LI0j/n2vf+/af/ABVFsN3/ABYfvv6sQ/8ACB33/Qx3H/ftv/i6P+EDvv8AoY7j/v23/wAXU3/CyNI/59r3/v2n/wAVR/wsjSP+fa9/79p/8VRbDd/xYfvv6sQ/8IHff9DHcf8Aftv/AIuj/hA77/oY7j/v23/xdTf8LI0j/n2vf+/af/FUf8LI0j/n2vf+/af/ABVFsN3/ABYfvv6sQ/8ACB33/Qx3H/ftv/i6P+EDvv8AoY7j/v23/wAXU3/CyNI/59r3/v2n/wAVR/wsjSP+fa9/79p/8VRbDd/xYfvv6sQ/8IHff9DHcf8Aftv/AIuj/hA77/oY7j/v23/xdTf8LI0j/n2vf+/af/FUf8LI0j/n2vf+/af/ABVFsN3/ABYfvv6sQ/8ACB33/Qx3H/ftv/i6P+EDvv8AoY7j/v23/wAXU3/CyNI/59r3/v2n/wAVR/wsjSP+fa9/79p/8VRbDd/xYfvv6sQ/8IHff9DHcf8Aftv/AIuj/hA77/oY7j/v23/xdTf8LI0j/n2vf+/af/FUf8LI0j/n2vf+/af/ABVFsN3/ABYfvv6sQ/8ACB33/Qx3H/ftv/i6P+EDvv8AoY7j/v23/wAXU3/CyNI/59r3/v2n/wAVR/wsjSP+fa9/79p/8VRbDd/xYfvv6sQ/8IHff9DHcf8Aftv/AIunQ/D8NdRyalqst7Ehz5bIRn8SxqT/AIWRpH/Pte/9+0/+Ko/4WRpH/Pte/wDfCf8AxVH+y9/xYfvzqf3Nna/wwwxL9AoFcJq2qz+Kbs2tmWi0yNvnfoZTTbrWLjxfI8SiSz06PnAGTM2ehPGOM+vT3q9BBHbQrFCoVFGABTnU9ppH4fzCMOTV7hBBHbQrFCoVFGABUlHHOfwopDCijjHvRQBS13RTr1qk0EijULaMRJEQc3KZJAB/vDOBnqMAdBnhp7ea1neG5ikhlQlWjkUqyn0IPSvRuMe9XoNc1a1iEVtql7DGOiR3DqB+ANc9ShzO6NY1LKzOA0Hw3caxMkk5az08NtlvHjJUf7K/3m56D6nA5rs3ZSkUca7IoY1ijTP3VUYH4nqT3JJ70+4up7yQy3c8s8v96VixPfqaiq6dJQ1JnNyCq+uBf+ETvyNu7YmcMc/61Oo/wqxxj3qHXd3/AAh999/btTGRlQfNToex9vxrSXwv0ZC3R53RU8CWjITczTRtngRwhxj6lhRXlHcWNB/5GPTf+vuL/wBDFeg3W4zDfvzj+Mgk/iOv1rzawuvsWpW11s3+RKsm3ON20g4z+FdavjbT3jkMulSrIQAqpMCCfXO0Y/AY9q7KFSMYtSZz1Iybuka1FYa+NrUMCdIZgBgg3Q5/8c/lTo/GlkAPM0uXK8jFwDu9j8o/xrb2tPv+ZnyT7G1RWEfGtrjjSHB3ZB+19PblOlSHxrYtE+dJkVyQVC3AwPXnb/j+FHtaff8AMOSfY2aKxD43tTn/AIk7crj/AI+u/r9zr+ntTk8a2IjBbSZC6jG37QMMPc7ePwo9rT7/AJhyT7GzRWEvjW2Xb/xKXO097oc+x+Sn/wDCaWRibOlyq+7KgXAI+mSv+NHtaff8w5J9jaorDbxraNvxo7LuxjF3936fJ/On/wDCa2LIxbSZA+3AAuOD/tZ28H8MUe1p9/zDkn2NmoJLK1mcvLbQux6lowTWWPG9qNudHZsDBzdfe+vyfypw8a2Kwp/xKpGkVuc3AwR6Z2/0z70e0pPr+Y+SfY0P7Nsf+fK3/wC/S/4Uf2bY/wDPlb/9+l/wrM/4TW1AYDSG5OR/pQ+X6fJ/PNOfxpYlGK6VKrsfui4GF9wduaXtKXf8A5Z9jR/s2x/58rf/AL9L/hR/Ztj/AM+Vv/36X/Csw+NrUtn+yGHy4IF0Off7nWn/APCa2Plof7JkZ1yCpuMBh2yQv9BR7Sl3Dln2ND+zbH/nyt/+/S/4Uf2bY/8APlb/APfpf8KzF8bWqqoOkMxBzk3Q5Hp9z/69O/4TSxEJxpUu8tkA3Awv0O3n8RR7Sl3/AADln2NH+zbH/nyt/wDv0v8AhR/Ztj/z5W//AH6X/Cs0+NrQs5/sdgGHT7V933HyU+Txpp/lqyaZMZCuHUzgKPcHb1/AUe0pd/wDlqdi/wD2bY/8+Vv/AN+l/wAKP7Nsf+fK3/79L/hWZ/wm1r8n/Eob5ev+lD5vr8n8qcfGliIkC6VK7A5bdcAfhkLyPyNHtKXcOWfY0f7Nsf8Anyt/+/S/4Uf2bY/8+Vv/AN+l/wAKzP8AhNbXaw/sh/mOQftX3fp8n8809vGtiASmkyEsOVNyAFPsdufzo9pS7hyz7Gh/Ztj/AM+Vv/36X/Cj+zbH/nyt/wDv0v8AhWb/AMJva+YGOjsRjBX7Vwff7n8sU5vGtiFiKaVIWUfODcAA/wDjv+FHtKXf8A5anY0P7Nsf+fK3/wC/S/4Uf2bY/wDPlb/9+l/wrMPjW1K4GkMPmyCLoce33OlOfxpY+WQmlSlmOTm4A2ewO3n8aPaUu4cs+xo/2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hWYPG1rls6Qx3DH/H0OD6j5P/AK1ObxpYo6mPSZHG3DB7kDJ/75P6Yo9pS7hyz7Gj/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hWYPG1r8n/Eob5ev+lD5vr8n8qdJ40ssARaVKec7jcAEe33TkfWj2lLuHLPsaP9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FZreNrRgw/sdl3HIxdfd+nyfzp0njWy3MY9KkYMvO64C7T6jC/8A1vaj2lLv+Acs+xof2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hWYvja1DAnSGYAYIN0Of/ABz+VObxpYqU8rSpW2/eL3AG72Pyn9MUe0pdw5Z9jR/s2x/58rf/AL9L/hR/Ztj/AM+Vv/36X/CsweNbTP8AyCHX5t2RdZwPTBTGKdL41sdziLSZNpPyk3OCv/jp/XNHtKXcOWfY0f7Nsf8Anyt/+/S/4Uf2bY/8+Vv/AN+l/wAKzf8AhNrQn5tHYArg4uu/r9zr+ntT5PGuno6iDSpZUAwTJOFLfUBT+lHtKXf8A5anYv8A9m2P/Plb/wDfpf8ACj+zbH/nyt/+/S/4VmJ41tQU3aVIQp5/0kHI9PuUsnjSyDEQ6VKRuyGNyAcemNpGKPaUu4cs+xpf2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hWYfGtmzNnR3UNj7t1936fJ/OnyeNrLe3l6Q7KVxk3AU59cBTg/p7Ue0pd/6+4OWfY0P7Nsf+fK3/AO/S/wCFH9m2P/Plb/8Afpf8Kz4/G1lvTzdIfaOGIuQSR9No/TFN/wCE1tFwBpLsFbIZrkAkeh+T/wCv70e0pd/wDln2NL+zbH/nyt/+/S/4Uf2bY/8APlb/APfpf8KzV8aWZbD6XKils5W5DFfYApz+NJJ41tC77NJkCt0/0ocf+OGj2lLuHLPsaf8AZtj/AM+Vv/36X/Cj+zbH/nyt/wDv0v8AhWcvjWyaQeZpEirjBK3OfxxtHP6Uh8a2YZcaQ7Be5usbvrhf5Ue0pd/6+4OWfY0v7Nsf+fK3/wC/S/4Uf2bY/wDPlb/9+l/wrOi8a2YKiTSZNoPLfaATj0xsH+PvSHxraDeE0lyCcqTcjK/+OfzzR7Sl3/AOWfY0v7Nsf+fK3/79L/hR/Ztj/wA+Vv8A9+l/wrPTxrYmR/M0mRFYcBbkHafb5R/OmN41tS2RpDj5cH/Shz7/AHOD9KPaUu4cs+xp/wBm2P8Az5W//fpf8KP7Nsf+fK3/AO/S/wCFZ0fjWxLL52kyAKDkrcg7vTPyj9P1pv8AwmtoFC/2Q7YbOTdYJHocJR7Sl3Dln2NP+zbH/nyt/wDv0v8AhR/Ztj/z5W//AH6X/Cs6PxpZEsJNKlUE5BW4DFfblen1zSHxtabnI0dsMMAfavu+4+Sj2lLv+Acs+xpf2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hVGLxrp5ctPpUqjZgBJwwY+/yjH4flUX/AAmtr8n/ABKHO3qftQ+b6/J/Kj2lLv8AgHLU7Gn/AGbY/wDPlb/9+l/wo/s2x/58rf8A79L/AIVnJ40svLw+mShl5Ui4Bz7H5en4GmnxralWH9kP8xyD9q+79Pk/nmj2lLuHLPsaf9m2P/Plb/8Afpf8KP7Nsf8Anyt/+/S/4Vnr41sT5m/SZF3D5VW4zg/Xbkfr9KYPG1oGB/sdiAMEG74b3+5/Kjnpd/6+4OWfY0/7Nsf+fK3/AO/S/wCFH9m2P/Plb/8Afpf8Kz08a2OAZNKkBQcYuAd/sflGPqKYfGtqVwNIYHdkEXQ49vudKPaUu4cs+xtqqooVFCqOAAMAUtYw8a2LLJv0mRCcFdlwDg/iv+NM/wCE2tctnSGO4Y/4+hwfUfJ/9an7Wn3/ADFyT7G5RWNH41sQcyaTIMLgKLjIY+/yjH4UweNbUbP+JQ529c3X3vr8n8qPa0+/5hyT7G5RWL/wmliYTnSpVcNlcXAIx6ZK/wBDTW8bWrBgNHZdxyMXX3fp8n86Pa0+/wCYck+xuUVjf8JtYmNi2kyByMAC4GP97O3g/himL42tQwJ0hmAGCDdDn/xz+VHtaff8w5J9jcorFTxpZCMFtLlLqeB9oBDD3O3+lMPjW1xxpDg7sg/a+ntynSj2tPv+Yck+xu1X1wqfCd/jZuCJnghv9an4EfrWafGlgySE6VKrnG1VueB+O3+hqrqvi9NR0iaxj09oBKqjcZ9wGGDZxtHXHYgUpVafK1fp5jUJ3WhzNFaul/8ACP8A2Vv7a/tPz952/ZPL27cDGd3Oc5/SivNudljKoooqhBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//9k=)

**Ilustración 12: Circuito impreso**

## 2.5. Gabinete Plástico

Para el diseño del gabinete plástico, se utilizó el software Fusion 360 de Autodesk. El gabinete consiste en dos partes, la base y una tapa. En la base se encuentra fija la placa principal, la fuente switching, el módulo relay, el tomacorriente y el módulo interlock. Mientras que, en la tapa, se encuentran el LCD, el encoder y el módulo de tarjeta microSD.

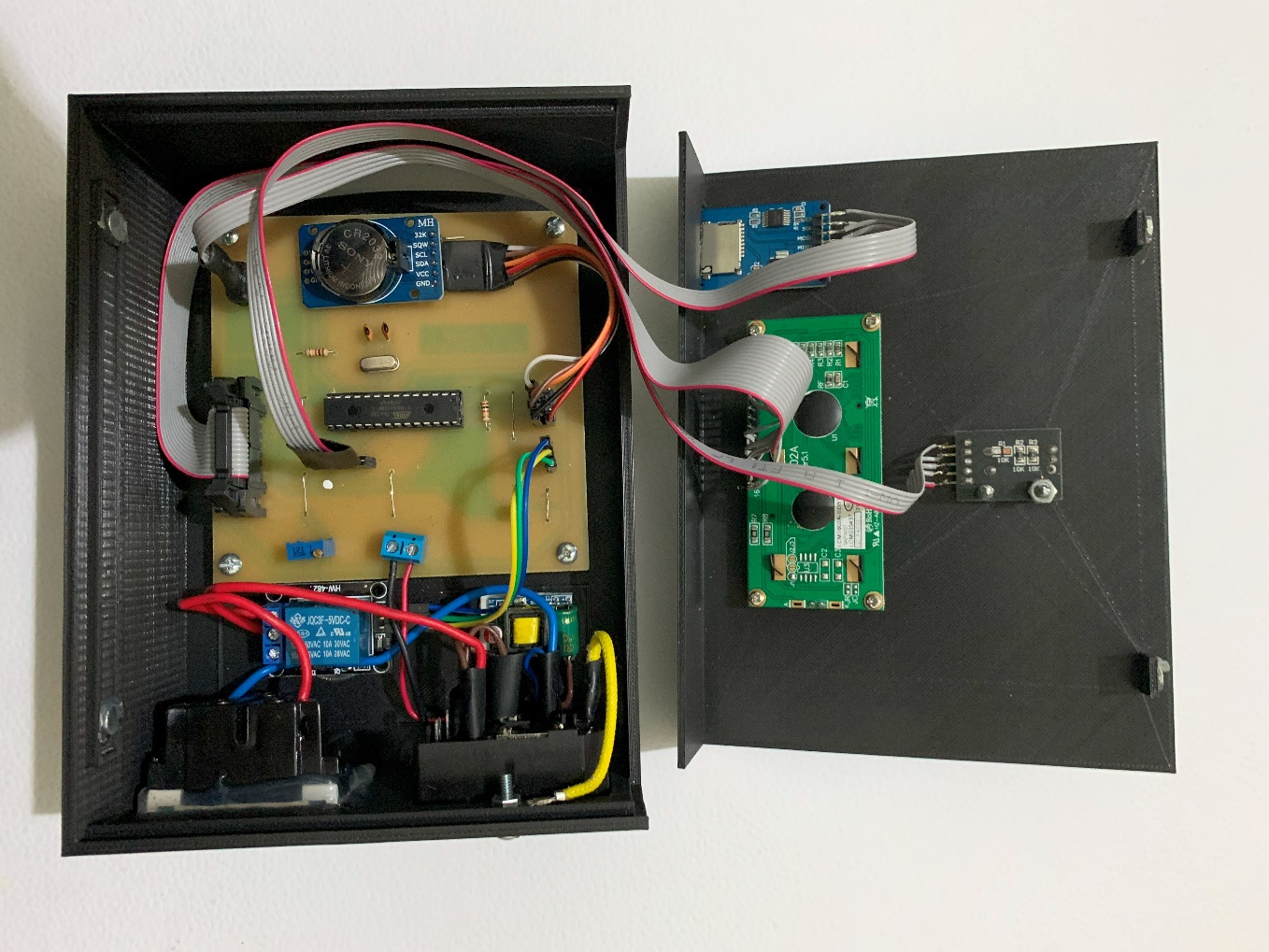
El gabinete posee unas guías sobre las cuales la tapa puede deslizarse libremente hasta cubrir la totalidad de la base. Además, se le incluyeron un juego de imanes de neodimio para el cierre de la tapa y evitar el uso de tornillos para el acoplamiento con la base.

Dibujo de ingeniería

Descripción generada automáticamente

**Ilustración 13: Diseño de Gabinete Plástico**

## 2.6. Montaje y ubicación de componentes



**TAPA**

**BASE**

**microSD**

**INTERLOCK**

**TOMACORRIENTE**

**FUENTE**

**RELAY**

**ENCODER**

**LCD 16X2**

**RTC**

**ATMEGA328P**

**Ilustración 14: Ubicación y montaje**

## 2.7. Costos

A continuación, se muestran los precios de todos los componentes que conforman al producto final, siendo los mismos registrados en junio de 2023.

|  |  |  |
| --- | --- | --- |
| **Componente** | **Precio** | |
| Fuente Switching 220 VAC 5 VDC 700 mA | AR$ | 1850,00 |
| ATMEGA328 con zócalo, cristal y capacitores | AR$ | 2829,00 |
| Display LCD 16x2 | AR$ | 1623,60 |
| Módulo Encoder Rotativo Ky-040 | AR$ | 565,50 |
| Módulo RTC DS3231 | AR$ | 1789,04 |
| Módulo Relay 5v 10A | AR$ | 798,94 |
| Módulo lector de memorias microSD | AR$ | 379,00 |
| Módulo Interlock con llave y fusible | AR$ | 2225,90 |
| Placa Pertinax 10cm x 10cm | AR$ | 408,27 |
| Transistor NPN BC548 | AR$ | 34,03 |
| Bornera azul de 2 contactos | AR$ | 126,42 |
| Conector hembra 1x40 | AR$ | 563,93 |
| Conector hembra 2x40 | AR$ | 781,96 |
| Conector hembra 2x7 | AR$ | 253,60 |
| Header macho 2x7 | AR$ | 157,08 |
| Resistencia 10K ohm 1/4 Watt | AR$ | 17,69 |
| Resistencia 1K ohm 1/4 Watt | AR$ | 17,69 |
| Preset Potenciómetro Multivuelta 10K ohm | AR$ | 419,00 |
| Cable plano de 14 conductores | AR$ | 607,93 |
| Plástico de base del gabinete | AR$ | 1900,00 |
| Plástico de tapa del gabinete | AR$ | 760,00 |
| **Total** | **AR$** | **18108,58** |

# 3. Consumos

A la hora de evaluar el consumo del dispositivo, era importante considerar que, si la carga a controlar era de hasta 10 A, limitada por el relay, el consumo del dispositivo debía ser mucho menor que este valor. Para este análisis, se plantearon 4 posibles condiciones de trabajo a las cuales puede estar sometido el equipo. Las mismas son:

* Que el backlight del LCD y el relay estén activados (**PEOR CONDICIÓN**)
* Que el backlight del LCD esté activado y el relay desactivado
* Que el backlight del LCD esté desactivado y el relay activado
* Que el backlight del LCD y el relay estén desactivados (**MEJOR CONDICIÓN**)

En función de lo planteado, se realizaron las mediciones correspondientes obteniendo los valores que se observan en la Ilustración 15.

Tabla

Descripción generada automáticamente

**Ilustración 15: Tabla de consumos**

De esta manera, se puede establecer que el equipo tiene un consumo que varía desde los 25mA hasta los 110mA aproximadamente, lo cual permite afirmar que el consumo del equipo no es representativo en comparación con el consumo del artefacto al cual busca controlar.

# 4. Firmware

## 4.1. Código

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Archivo: control\_de\_energia.ino*

*//*

*// Proyecto: CONTROL DE ENERGÍA POR TIEMPOS DE OPERACIÓN*

*//*

*// Materia: ELECTRÓNICA DE POTENCIA*

*//*

*// Fecha de creación: 1er CUATRIMESTRE 2023*

*//*

*// Profesores: Ing. Guillermo Luis Miquel*

*// Ing. Oscar Pugliese*

*//*

*// Autores: Nicolás Enrique Agostino*

*// Ignacio Luis Mehle*

*// Santiago Ruiz*

*// Federico Ladislao Sokolic*

*// Pablo Fabián Yujra Ventura*

*// Descripción:*

*// El dispositivo tiene como principal función la conexión y desconexión de*

*// forma automática de un artefacto o carga eléctrica monofásica siguiendo*

*// un programa semanal configurable, respetando los feriados según el*

*// calendario anual vigente. Este proyecto fue desarrollado a lo largo de*

*// la cursada de la asignatura “Electrónica de Potencia”, en el marco de*

*// impulsar una mejora en el consumo sustentable de la Universidad.*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

#include <LiquidCrystal.h>

#include <EEPROM.h>

#include <DS3231.h>

#include <Wire.h>

#include <SPI.h>

#include <SD.h>

#ifdef TIMER\_INTERNO

#include <TimerOne.h> *//Para simular el reloj del RTC*

#endif

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// ETIQUETAS*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*//Descomentar para habilitar el debug por puerto serie*

*//#define DEBUG\_SERIE*

*//Menu*

#define MENU\_PRINCIPAL 0

#define MENU\_AJUSTE\_FECHA\_HORA 1

#define MENU\_CONFIGURACION 2

#define MENU\_PROGRAMA\_SEMANAL 3

#define MENU\_FERIADOS 4

#define MENU\_CAMBIOS\_SD 5

*//Dias de la semana*

#define LUNES 1

#define MARTES 2

#define MIERCOLES 3

#define JUEVES 4

#define VIERNES 5

#define SABADO 6

#define DOMINGO 7

*//Rangos*

#define INICIO\_1 0

#define FIN\_1 1

#define INICIO\_2 2

#define FIN\_2 3

#define INICIO\_3 4

#define FIN\_3 5

#define INICIO\_4 6

#define FIN\_4 7

#define PRIMERA\_POSICION\_EVENTOS 0

#define ULTIMA\_POSICION\_EVENTOS PRIMERA\_POSICION\_EVENTOS+55

#define PRIMERA\_POSICION\_FERIADOS ULTIMA\_POSICION\_EVENTOS+1

#define ULTIMA\_POSICION\_FERIADOS PRIMERA\_POSICION\_FERIADOS+365

#define MAX\_VALOR\_HORARIOS 95 *//correspondiente a las 23:45*

#define CANT\_RANGOS 4

*//Encoder*

#define DERECHA 1

#define IZQUIERDA 2

#define ENTER 3

*//Para lectura de EEPROM*

#define HORA 0

#define MINUTOS 1

*//Para seteo de feriados*

#define ES\_FERIADO 1

#define NO\_ES\_FERIADO 0

#define TIEMPO\_PARA\_APAGAR\_LCD 60 *//en segundos*

*//Pin usado para disparar la interrupción, la salida SQW del RTC debe conectarse a el pin a usar*

#define CLINT 3

#define SSPin 10 *//CS de la uSD*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// VARIABLES*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**const** **int** Encoder\_OuputA PROGMEM = 9;

**const** **int** Encoder\_OuputB PROGMEM = 8;

**const** **int** Encoder\_Switch PROGMEM = 5;

**const** **int** Salida\_Rele PROGMEM = 0;

**const** **int** Salida\_Backlight PROGMEM = 4;

**int** Previous\_Output;

**int** diadelasemana=0;

**int** estado\_menu=0;

**int** estado\_menu\_anterior=0;

**int** estado\_ajuste=0;

**int** estado\_prog\_sem=0;

**int** estado\_prog\_rango=0;

**int** estado\_menu\_feriados=0;

**int** proximo\_menu=0;

**int** seg\_anterior=0;

**int** rango\_a\_programar=0;

**char** Rango\_num[CANT\_RANGOS\*2];

**char** Rangos\_hoy[CANT\_RANGOS\*2];

**char** codigo\_hora\_actual=0;

**char** codigo\_hora\_actual\_anterior=0;

**char** ultimo\_dia=0;

**bool** modificacion\_realizada=0;

**bool** cursor\_feriado=0;

**bool** feriado=0;

**int** tiempo\_sin\_pulsar=TIEMPO\_PARA\_APAGAR\_LCD;

*//ESTRUCTURA DE FECHA Y HORA*

**struct** RTC\_Time

{

*//Dia de la semana*

**char** DoW;

*//FECHA*

**char** dia; *//[1,31]*

**char** mes; *//[1,12]*

**char** anio; *//[0,255]*

*//HORA*

**char** seg; *//[0,59]*

**char** min; *//[0,59]*

**char** hora; *//[0,23]*

};

RTC\_Time ActualTime;

RTC\_Time AuxTime;

*// Setup clock*

DS3231 RTC;

*//Estructura para el archivo de la uSD*

*//File myFile;*

**volatile** byte tick = 1;

byte alarmBits = 0b00001110; *// Cada un min*

**bool** Century = false;

**bool** h12=0;*//Modo 24HS*

**bool** PM ;

*//Cantidad de dias por mes, usado para calcular la posicion en memoria*

**const** **int** cant\_dias\_mes[] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

*//Mention the pin number for LCD connection*

**const** **int** rs PROGMEM = 7;

**const** **int** en PROGMEM = 6;

**const** **int** d4 PROGMEM = 14;

**const** **int** d5 PROGMEM = 15;

**const** **int** d6 PROGMEM = 16;

**const** **int** d7 PROGMEM = 17;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

**int** min\_ant,seg\_ant;

**bool** estado\_uSD=0;

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: setup*

*//*

*// Descripción: Configuración inicial del programa*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** setup()

{

*/\*\*\*Letras seleccionadas\*\*\*/*

byte L[8] = {

0b01111,

0b01111,

0b01111,

0b01111,

0b01111,

0b01111,

0b00000,

0b11111

};

byte M[8] = {

0b01110,

0b00100,

0b01010,

0b01010,

0b01110,

0b01110,

0b01110,

0b11111

};

byte a[8] = {

0b11111,

0b11111,

0b10001,

0b11110,

0b10000,

0b01110,

0b10000,

0b11111

};

byte i[8] = {

0b11011,

0b11111,

0b10011,

0b11011,

0b11011,

0b11011,

0b10001,

0b11111

};

byte J[8] = {

0b10000,

0b11101,

0b11101,

0b11101,

0b11101,

0b01101,

0b10011,

0b11111

};

byte V[8] = {

0b01110,

0b01110,

0b01110,

0b01110,

0b01110,

0b10101,

0b11011,

0b11111

};

byte S[8] = {

0b10001,

0b01110,

0b01111,

0b10001,

0b11110,

0b01110,

0b10001,

0b11111

};

byte D[8] = {

0b00011,

0b01101,

0b01110,

0b01110,

0b01110,

0b01101,

0b00011,

0b11111

};

*// Begin I2C communication*

Wire.begin();

*//Interrupción de RTC*

configuraInterrupcionRTC();

*// put your setup code here, to run once:*

lcd.begin(16, 2); *//Initialise 16\*2 LCD*

lcd.createChar(0,D);

lcd.createChar(1,L);

lcd.createChar(2,M);

lcd.createChar(3,a);

lcd.createChar(4,i);

lcd.createChar(5,J);

lcd.createChar(6,V);

lcd.createChar(7,S);

#ifdef DEBUG\_SERIE

Serial.begin(9600);

#endif

pinMode (Encoder\_OuputA, INPUT);

pinMode (Encoder\_OuputB, INPUT);

pinMode (Encoder\_Switch, INPUT);

Previous\_Output = digitalRead(Encoder\_OuputA); *//Read the inital value of Output A*

pinMode (Salida\_Rele, OUTPUT);

pinMode (Salida\_Backlight, OUTPUT);

Leer\_Fecha\_Hora\_RTC(ActualTime);

#ifdef TIMER\_INTERNO

*//Para tener un timer de 1seg*

Timer1.initialize(1000000);

Timer1.attachInterrupt(ISR\_Segundo);

#endif

Rango\_num[INICIO\_1]=Rango\_num[FIN\_1]='1';

Rango\_num[INICIO\_2]=Rango\_num[FIN\_2]='2';

Rango\_num[INICIO\_3]=Rango\_num[FIN\_3]='3';

Rango\_num[INICIO\_4]=Rango\_num[FIN\_4]='4';

*//Cargo los rangos del día de hoy*

Cargar\_Rangos();

ultimo\_dia=ActualTime.DoW;

tiempo\_sin\_pulsar=TIEMPO\_PARA\_APAGAR\_LCD;

*//Lee si hay feriados en la uSD para agregar*

leerFeriadosDeSd();

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: leerFeriadosDeSd*

*//*

*// Descripción: Lee los feriados y eventos de la uSD y los agrega en la EEPROM*

*//*

*// Parámetros: void*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** leerFeriadosDeSd(**void**)

{

#ifdef DEBUG\_SERIE

**char** BufferAux[25];

#endif

*//Estructura para el archivo de la uSD*

File myFile;

**int** feriado\_anterior=PRIMERA\_POSICION\_FERIADOS;

**bool** hubo\_cambio=0;

#ifdef DEBUG\_SERIE

Serial.print(F("Inicializando tarjeta...\n"));

delay(50);

#endif

**if** (!SD.begin(SSPin))

{

estado\_uSD=0;

#ifdef DEBUG\_SERIE

Serial.print(F("Falló la inicialización de la uSD\n"));

delay(50);

#endif

}

**else**

{

*//Si entra acá es porque está la uSD presente*

*//La leo solo si no la leí previamente desde que está colocada*

**if**(!estado\_uSD)

{

*//Abre el archivo de feriados*

myFile = SD.open("feriados.txt");

**if**(myFile)

{

String feriados;

String eventos;

estado\_uSD = 1;

#ifdef DEBUG\_SERIE

Serial.print(F("if TRUE\n"));

delay(50);

#endif

*// while (myFile.available())*

*// {*

*//Feriados*

feriados = myFile.readStringUntil('|');

**int** feriados\_len = feriados.length() + 1;

String evento = "";

**int** eventos\_len;

**char** char\_feriados[feriados\_len];

feriados.toCharArray(char\_feriados, feriados\_len);

String feriado = "";

**for** (**int** i = 0; i < feriados.length(); i++)

{

**if** (feriados[i] == ',')

{

**if**((feriado.toInt()>=PRIMERA\_POSICION\_FERIADOS)&&

(feriado.toInt()<=ULTIMA\_POSICION\_FERIADOS))

{

**if**(EEPROM.read(feriado.toInt())!=1)

{

EEPROM.write(feriado.toInt(), 1);

hubo\_cambio=1;

#ifdef DEBUG\_SERIE

Serial.print(F("Lo guardo!"));

#endif

}

#ifdef DEBUG\_SERIE

sprintf(BufferAux,"feriado: %d \n",feriado.toInt());

Serial.print(BufferAux);

#endif

*//Borro todas las posiciones que no estén en la microSD*

**for**(**int** j=feriado\_anterior;j<feriado.toInt();j++)

{

**if**(EEPROM.read(j)!=0)

{

EEPROM.write(j, 0);

hubo\_cambio=1;

#ifdef DEBUG\_SERIE

Serial.print(F("Lo borro!"));

#endif

}

}

feriado\_anterior = feriado.toInt()+1;

}

feriado = "";

}

**else**

{

feriado += feriados[i];

#ifdef DEBUG\_SERIE

Serial.print(feriado);

Serial.print(F("\n"));

#endif

}

}

**if**((feriado\_anterior>PRIMERA\_POSICION\_FERIADOS)&&(feriado\_anterior<=ULTIMA\_POSICION\_FERIADOS))

{

*//Borro todas las posiciones que faltan de la uSD*

**for**(**int** i=feriado\_anterior;i<=ULTIMA\_POSICION\_FERIADOS;i++)

{

**if**(EEPROM.read(i)!=0)

EEPROM.write(i, 0);

}

}

**while** (myFile.available())

{

eventos = "";

evento = "";

*//Eventos*

eventos = myFile.readStringUntil(',');

*//if (eventos[i] != '.')*

**if** (eventos[0] != '.')

{

eventos += ',';

eventos\_len = eventos.length() + 1;

#ifdef DEBUG\_SERIE

Serial.print(eventos);

delay(100);

sprintf(BufferAux,"\n eventos\_len: %d\n",eventos\_len);

Serial.print(BufferAux);

#endif

*//char char\_eventos[eventos\_len];*

*//eventos.toCharArray(char\_eventos, eventos\_len);*

**for** (**int** i = 0; i < eventos.length(); i++)

{

#ifdef DEBUG\_SERIE

sprintf(BufferAux,"eventos[%d]: %c\n",i,eventos[i]);

Serial.print(BufferAux);

#endif

**if** (eventos[i] != ',')

{

*//Mientras no llegue la coma, armo el string*

evento += eventos[i];

#ifdef DEBUG\_SERIE

Serial.print(evento);

Serial.print(F("\n"));

#endif

}

**else**

{

String posicionEnMemoriaEvento = evento.substring(0,2); *//Agarro la posicion (antes del ':')*

String valorAGuardarEnPosicion = evento.substring(3,5); *//Agarro el codigo a guardar (despues del ':')*

#ifdef DEBUG\_SERIE

sprintf(BufferAux,"posicionEnMemoriaEvento: %d\n",posicionEnMemoriaEvento.toInt());

Serial.print(BufferAux);

sprintf(BufferAux,"valorAGuardarEnPosicion: %d\n",valorAGuardarEnPosicion.toInt());

Serial.print(BufferAux);

#endif

**if**((posicionEnMemoriaEvento.toInt()>=PRIMERA\_POSICION\_EVENTOS)&&(posicionEnMemoriaEvento.toInt()<=ULTIMA\_POSICION\_EVENTOS)&&(valorAGuardarEnPosicion.toInt()<= MAX\_VALOR\_HORARIOS))

{

*//Si es valido, me fijo si es diferente a lo que ya está guardado*

**if**(EEPROM.read(posicionEnMemoriaEvento.toInt())!=valorAGuardarEnPosicion.toInt())

{

*//Si es diferente, lo guardo*

EEPROM.write(posicionEnMemoriaEvento.toInt(), valorAGuardarEnPosicion.toInt());

hubo\_cambio = 1;

#ifdef DEBUG\_SERIE

Serial.print(F("Lo guardo!"));

#endif

}

}

evento = "";

}

}

}

}

**if**(hubo\_cambio)

estado\_menu = MENU\_CAMBIOS\_SD;

*// Cerrar el archivo*

myFile.close();

}

**else**

{

estado\_uSD=0;

#ifdef DEBUG\_SERIE

Serial.print(F("Error leyendo feriados.txt"));

#endif

}

}

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Leer\_Fecha\_Hora\_RTC*

*//*

*// Descripción: Carga en la estructura la fecha y hora guardada en el RTC*

*//*

*// Parámetros: RTC\_Time Time (estructura a cargar)*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Leer\_Fecha\_Hora\_RTC(RTC\_Time& Time)

{

*//Leo los datos del RTC*

Time.DoW=RTC.getDoW();

*//FECHA*

Time.dia=RTC.getDate();

Time.mes=RTC.getMonth(Century);

Time.anio=RTC.getYear();

*//HORA*

Time.hora=RTC.getHour(h12, PM);

Time.min=RTC.getMinute();

Time.seg=RTC.getSecond();

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Cargar\_Rangos*

*//*

*// Descripción: Carga los rangos de encendido del día de hoy*

*//*

*// Parámetros: void*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Cargar\_Rangos(**void**)

{

#ifdef DEBUG\_SERIE

**char** BufferAux[50];

#endif

**for**(**int** i=0; i<CANT\_RANGOS\*2; i++)

{

Rangos\_hoy[i]=EEPROM.read((ActualTime.DoW-1)\*8+i);

#ifdef DEBUG\_SERIE

**if**(i%2==0)

Serial.print(F("Inicio "));

**else**

Serial.print(F("Fin "));

sprintf(BufferAux,"%c: %d\n",Rango\_num[i],Rangos\_hoy[i]);

Serial.print(BufferAux);

#endif

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Leer\_Encoder*

*//*

*// Descripción: Devuelve la dirección del enconder o si se apretó el*

*// pulsador, teniendo prioridad el pulsador*

*//*

*// Parámetros: void*

*// Valor devuelto: int IZQUIERDA(antihorario), DERECHA(horario) o ENTER*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**int** Leer\_Encoder(**void**)

{

**int** giro\_encoder=0;

**if** (digitalRead(Encoder\_OuputB) != Previous\_Output)

{

**if** (digitalRead(Encoder\_OuputA) != Previous\_Output)

{

giro\_encoder = DERECHA;

tiempo\_sin\_pulsar=TIEMPO\_PARA\_APAGAR\_LCD;

}

**else**

{

giro\_encoder = IZQUIERDA;

tiempo\_sin\_pulsar=TIEMPO\_PARA\_APAGAR\_LCD;

}

}

Previous\_Output = digitalRead(Encoder\_OuputB);

*//El ENTER tiene prioridad ante los giros*

**if**(digitalRead(Encoder\_Switch)==0)

{

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

**if**(tiempo\_sin\_pulsar>0) *//Si está el backlight apagado, no realiza ninguna acción el enter, solo prende el backlight*

**return**(ENTER);

tiempo\_sin\_pulsar=TIEMPO\_PARA\_APAGAR\_LCD;

}

**return**(giro\_encoder);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Programacion\_Semanal*

*//*

*// Descripción: Se programan los rangos de encendido de cada día de la semana*

*//*

*// Parámetros: void*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Programacion\_Semanal(**void**)

{

**char** BufferAux[5];

**switch**(estado\_prog\_sem)

{

default:

estado\_prog\_sem=0;

**break**;

**case** 0:

lcd.setCursor(0, 0);

lcd.print(F("Seleccionar dia"));

lcd.setCursor(0, 1);

lcd.print(F("D L Ma Mi J V S"));

*//Selecciono DOMINGO*

*//lcd.setCursor(0, 1);*

*//lcd.write(byte(0));//D seleccionada*

*//diadelasemana=DOMINGO;*

*//delay(1000);*

diadelasemana=ActualTime.DoW;

estado\_prog\_sem++;

**break**;

**case** 1:

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** DERECHA:

diadelasemana ++;

**if**(diadelasemana > DOMINGO)

diadelasemana = LUNES;

#ifdef DEBUG\_SERIE

Serial.print(F("Variable: "));

Serial.print(diadelasemana);

Serial.print(F("\n"));

#endif

delay(50);

**break**;

**case** IZQUIERDA:

diadelasemana--;

**if**(diadelasemana < LUNES)

diadelasemana = DOMINGO;

#ifdef DEBUG\_SERIE

Serial.print(F("Variable: "));

Serial.print(diadelasemana);

Serial.print(F("\n"));

#endif

delay(50);

**break**;

**case** ENTER:

estado\_prog\_sem++;

**break**;

}

Seleccion\_dias();

delay(10);

**break**;

**case** 2:

*//Presento el día en pantalla*

lcd.clear();

lcd.setCursor(0, 0);

**switch**(diadelasemana)

{

default:

diadelasemana=DOMINGO;

lcd.print(F("DOMINGO"));

**break**;

**case** DOMINGO:

lcd.print(F("DOMINGO"));

**break**;

**case** LUNES:

lcd.print(F("LUNES"));

**break**;

**case** MARTES:

lcd.print(F("MARTES"));

**break**;

**case** MIERCOLES:

lcd.print(F("MIERCOLES"));

**break**;

**case** JUEVES:

lcd.print(F("JUEVES"));

**break**;

**case** VIERNES:

lcd.print(F("VIERNES"));

**break**;

**case** SABADO:

lcd.print(F("SABADO"));

**break**;

}

estado\_prog\_sem++;

rango\_a\_programar=0;

estado\_prog\_rango=0;

**break**;

**case** 3:

**if**(Programar\_Rango(rango\_a\_programar)==1)

rango\_a\_programar++;

**if**(rango\_a\_programar >= (CANT\_RANGOS\*2))

estado\_prog\_sem++;

**break**;

**case** 4:

*//Trato de guardar los cambios si hay uSD*

Guardar\_Cambios\_uSD();

lcd.setCursor(0, 0);

lcd.print(F("Config guardada "));

lcd.setCursor(0, 1);

lcd.print(F(" con exito "));

delay(3000);

estado\_menu=MENU\_PRINCIPAL;

estado\_prog\_sem=0;

diadelasemana=0;

Cargar\_Rangos();

modificacion\_realizada=1;

**break**;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Programar\_Rango*

*//*

*// Descripción: Se programan las horas y minutos de cada rango*

*//*

*// Parámetros: int rango: INICIO1,FIN1,INICIO2,FIN2...etc*

*// Valor devuelto: bool: devuelve un 1 cuando ya se programó ese rango, sino 0*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**bool** Programar\_Rango(**int** rango)

{

*//Por cada rango a programar va a entrar acá*

**char** BufferAux[5];

**switch**(estado\_prog\_rango)

{

default:

estado\_prog\_rango=0;

**break**;

**case** 0:

lcd.setCursor(10, 0);

lcd.print(F("Rango"));

lcd.setCursor(15, 0);

lcd.print(Rango\_num[rango]);

**if**(rango%2==0)

{

lcd.setCursor(0, 1);

lcd.print(F("Encendido:"));*//Rangos pares son encedido*

}

**else**

{

lcd.setCursor(0, 1);

lcd.print(F("Apagado: "));*//Rangos impares son apagado*

}

AuxTime.hora = Leer\_Rangos\_EEPROM(diadelasemana, rango, HORA);

AuxTime.min = Leer\_Rangos\_EEPROM(diadelasemana, rango, MINUTOS);

lcd.setCursor(11, 1);

**if**((AuxTime.hora<0 || AuxTime.hora>23)||(AuxTime.min<0 || AuxTime.min>45))

{

AuxTime.hora = 0;

AuxTime.min = 0;

}

**else**

{

sprintf(BufferAux,"%02d:%02d",AuxTime.hora,AuxTime.min);

lcd.print(BufferAux);

}

estado\_prog\_rango++;

**break**;

**case** 1:

*//Programo la hora*

lcd.setCursor(11, 1);

sprintf(BufferAux,"%02d:%02d",AuxTime.hora,AuxTime.min);

lcd.print(BufferAux);

Efecto\_Titilar(11,1,2,100);

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** IZQUIERDA:

AuxTime.hora--;

**if**(AuxTime.hora<0)

AuxTime.hora=23;

**break**;

**case** DERECHA:

AuxTime.hora++;

**if**(AuxTime.hora>23)

AuxTime.hora=0;

**break**;

**case** ENTER:

estado\_prog\_rango++;

**break**;

}

**break**;

**case** 2:

*//Programo los minutos*

lcd.setCursor(11, 1);

sprintf(BufferAux,"%02d:%02d",AuxTime.hora,AuxTime.min);

lcd.print(BufferAux);

Efecto\_Titilar(14,1,2,100);

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** IZQUIERDA:

AuxTime.min=AuxTime.min-15;

**if**(AuxTime.min<0)

AuxTime.min=45;

**break**;

**case** DERECHA:

AuxTime.min=AuxTime.min+15;

**if**(AuxTime.min>45)

AuxTime.min=0;

**break**;

**case** ENTER:

estado\_prog\_rango++;

**break**;

}

**break**;

**case** 3:

*//Guardo el rango en EEPROM*

#warning "Habria que hacer una logica que se fije que el inicio sea antes del fin en cada rango, y que no se superpongan"

Escribir\_Rangos\_EEPROM(diadelasemana,rango,AuxTime.hora,AuxTime.min);

estado\_prog\_rango=0;

**return**(1);

**break**;

}

**return**(0);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Leer\_Rangos\_EEPROM*

*//*

*// Descripción: Se leen las horas o minutos (según se solicite) guardado*

*// en la EEPROM del día solicitado en el rango solicitado.*

*//*

*// Parámetros: int dia (LUNES, MARTES... etc)*

*// int rango (INICIO1, FIN1, INICIO2... etc)*

*// Valor devuelto: bool hora\_min: aclaras si queres recibir la hora o los mins*

*// (HORA o MINUTOS)*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**char** Leer\_Rangos\_EEPROM(**int** dia, **int** rango, **bool** hora\_min)

{

**char** codigo\_hora=0;

**int** direccion\_eeprom=0;

*//Leo la hora y fecha programada*

direccion\_eeprom= (dia-1)\*8+rango;

codigo\_hora = EEPROM.read(direccion\_eeprom);

**if**(hora\_min == HORA)

**return**(codigo\_hora/4);

**else** *//if(hora\_min == MINUTOS)*

{

codigo\_hora=codigo\_hora%4; *//Agarro el resto (de 0 a 3)*

codigo\_hora=codigo\_hora\*15;*//Lo multiplico por 15*

**return**(codigo\_hora);

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Escribir\_Rangos\_EEPROM*

*//*

*// Descripción: Se escribe en la EEPROM el código generado por las horas y*

*// minutos en la dirección generada por el día y rango indicados*

*//*

*// Parámetros: int dia (LUNES, MARTES... etc)*

*// int rango (INICIO1, FIN1, INICIO2... etc)*

*// int hora*

*// int min (múltiplo de 15 preferentemente)*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Escribir\_Rangos\_EEPROM(**int** dia, **int** rango, **int** hora, **int** min)

{

**char** codigo\_hora=0;

**int** direccion\_eeprom=0;

direccion\_eeprom= (dia-1)\*8+rango;

codigo\_hora=((hora\*4)+(min/15));

EEPROM.write(direccion\_eeprom,codigo\_hora);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Seleccion\_dias*

*//*

*// Descripción: Se utiliza el encoder para seleccionar el día de la semana*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Seleccion\_dias(**void**)

{

**switch**(diadelasemana)

{

default:

diadelasemana=DOMINGO;

**break**;

**case** DOMINGO:

*//Selecciono DOMINGO*

lcd.setCursor(0, 1);

lcd.write(byte(0));*//D seleccionada*

*//Deselecciono resto de dias*

*//Deselecciono SABADO*

lcd.setCursor(14, 1);

lcd.print(F("S"));

*//Deseleccionar\_dias(diadelasemana);*

*//Deselecciono LUNES*

lcd.setCursor(2, 1);

lcd.print(F("L"));

**break**;

**case** LUNES:

*//Selecciono LUNES*

lcd.setCursor(2, 1);

lcd.write(byte(1));*//L seleccionada*

*//Deselecciono DOMINGO*

lcd.setCursor(0, 1);

lcd.print(F("D"));

*//Deselecciono MARTES*

lcd.setCursor(4, 1);

lcd.print(F("Ma"));

**break**;

**case** MARTES:

*//Selecciono MARTES*

lcd.setCursor(4, 1);

lcd.write(byte(2));*//M seleccionada*

lcd.write(byte(3));*//a seleccionada*

*//Deselecciono LUNES*

lcd.setCursor(2, 1);

lcd.print(F("L"));

*//Deselecciono MIERCOLES*

lcd.setCursor(7, 1);

lcd.print(F("Mi"));

**break**;

**case** MIERCOLES:

*//Selecciono MIERCOLES*

lcd.setCursor(7, 1);

lcd.write(byte(2));*//M seleccionada*

lcd.write(byte(4));*//i seleccionada*

*//Deselecciono MARTES*

lcd.setCursor(4, 1);

lcd.print(F("Ma"));

*//Deselecciono JUEVES*

lcd.setCursor(10, 1);

lcd.print(F("J"));

**break**;

**case** JUEVES:

*//Selecciono JUEVES*

lcd.setCursor(10, 1);

lcd.write(byte(5));*//J seleccionada*

*//Deselecciono MIERCOLES*

lcd.setCursor(7, 1);

lcd.print(F("Mi"));

*//Deselecciono VIERNES*

lcd.setCursor(12, 1);

lcd.print(F("V"));

**break**;

**case** VIERNES:

*//Selecciono VIERNES*

lcd.setCursor(12, 1);

lcd.write(byte(6));*//V seleccionada*

*//Deselecciono JUEVES*

lcd.setCursor(10, 1);

lcd.print(F("J"));

*//Deselecciono SABADO*

lcd.setCursor(14, 1);

lcd.print(F("S"));

**break**;

**case** SABADO:

*//Selecciono SABADO*

lcd.setCursor(14, 1);

lcd.write(byte(7));*//S seleccionada*

*//Deselecciono VIERNES*

lcd.setCursor(12, 1);

lcd.print(F("V"));

*//Deselecciono DOMINGO*

lcd.setCursor(0, 1);

lcd.print(F("D"));

**break**;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Menu\_Principal*

*//*

*// Descripción: Muestra la fecha y la hora y utiliza el encoder para elegir*

*// ajustar la hora/fecha o configurar la programación*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Menu\_Principal(**void**)

{

**char** BufferAux[16];

*//Muestro la fecha y hora actual*

lcd.setCursor(0, 0);

**switch**(ActualTime.DoW)

{

default:

*//Distingo si me trae basura*

lcd.print(F("?"));

**break**;

**case** DOMINGO:

lcd.print(F("D"));

**break**;

**case** LUNES:

lcd.print(F("L"));

**break**;

**case** MARTES:

lcd.print(F("M"));

**break**;

**case** MIERCOLES:

lcd.print(F("X"));

**break**;

**case** JUEVES:

lcd.print(F("J"));

**break**;

**case** VIERNES:

lcd.print(F("V"));

**break**;

**case** SABADO:

lcd.print(F("S"));

**break**;

}

lcd.setCursor(1, 0);

sprintf(BufferAux," %02d/%02d/%02d %02d:%02d",ActualTime.dia,ActualTime.mes,ActualTime.anio,ActualTime.hora,ActualTime.min);

lcd.print(BufferAux);

lcd.setCursor(1, 1);

lcd.print(F("Reloj"));

lcd.setCursor(10, 1);

lcd.print(F("Config"));

*//Para la primera vez que entra aca*

*//if((proximo\_menu != MENU\_CONFIGURACION)&&(proximo\_menu != MENU\_AJUSTE\_FECHA\_HORA))*

**if**(proximo\_menu != MENU\_CONFIGURACION)

{

proximo\_menu = MENU\_AJUSTE\_FECHA\_HORA;

lcd.setCursor(0, 1);

*//lcd.write(byte(8));//Flecha señalando Reloj*

lcd.print(F(">"));

}

*/\**

*#ifdef DEBUG\_SERIE*

*Serial.print(F("proximo\_menu:"));*

*Serial.println(proximo\_menu);*

*#endif*

*\*/*

Seleccion\_Principal();

*//Hago titilar el ':' para que se vea que está vivo*

Efecto\_Titilar(13,0,1,50);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Seleccion\_Principal*

*//*

*// Descripción: Dibuja una flecha en el menú seleccionado dentro del*

*// principal. Si se presiona ENTER, entra a ese menú.*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Seleccion\_Principal(**void**)

{

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** DERECHA:

lcd.setCursor(0, 1);

lcd.print(F(" "));

lcd.setCursor(9, 1);

*//lcd.write(byte(8));//Flecha señalando Config*

lcd.print(F(">"));

proximo\_menu = MENU\_CONFIGURACION;

delay(250);

**break**;

**case** IZQUIERDA:

lcd.setCursor(0, 1);

*//lcd.write(byte(8));//Flecha señalando Reloj*

lcd.print(F(">"));

lcd.setCursor(9, 1);

lcd.print(F(" "));

proximo\_menu = MENU\_AJUSTE\_FECHA\_HORA;

delay(250);

**break**;

**case** ENTER:

estado\_menu = proximo\_menu;

**break**;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Seleccion\_Configuracion*

*//*

*// Descripción: Muestra el menú Configuración y dibuja una flecha en el menu*

*// seleccionado. Si se presiona ENTER, entra a ese menu.*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Seleccion\_Configuracion(**void**)

{

lcd.setCursor(1, 0);

lcd.print(F("Prog. semanal"));

lcd.setCursor(1, 1);

lcd.print(F("Feriados"));

*//Para la primera vez que entra aca*

**if**((proximo\_menu != MENU\_PROGRAMA\_SEMANAL)&&(proximo\_menu != MENU\_FERIADOS))

{

lcd.setCursor(0, 0);

*//lcd.write(byte(8));//Flecha señalando Prog Semanal*

lcd.print(F(">"));

lcd.setCursor(0, 1);

lcd.print(F(" "));

proximo\_menu = MENU\_PROGRAMA\_SEMANAL;

delay(1000);

}

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** IZQUIERDA:

lcd.setCursor(0, 0);

*//lcd.write(byte(8));//Flecha señalando Prog Semanal*

lcd.print(F(">"));

lcd.setCursor(0, 1);

lcd.print(F(" "));

proximo\_menu = MENU\_PROGRAMA\_SEMANAL;

delay(250);

**break**;

**case** DERECHA:

lcd.setCursor(0, 0);

lcd.print(F(" "));

lcd.setCursor(0, 1);

*//lcd.write(byte(8));//Flecha señalando Feriados*

lcd.print(F(">"));

proximo\_menu = MENU\_FERIADOS;

delay(250);

**break**;

**case** ENTER:

estado\_menu = proximo\_menu;

**break**;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Reloj*

*//*

*// Descripción: Reloj en base al timer de 1seg para simular el RTC*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*/\**

*void Reloj(void){*

*if(ActualTime.seg >59)*

*{*

*ActualTime.seg=0;*

*ActualTime.min++;*

*if(ActualTime.min >59)*

*{*

*ActualTime.min=0;*

*ActualTime.hora++;*

*if(ActualTime.hora>23)*

*ActualTime.hora=0; //Faltaría cambiar de día*

*}*

*}*

*}*

*\*/*

#ifdef TIMER\_INTERNO

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: ISR\_Segundo*

*//*

*// Descripción: Timer de 1seg*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** ISR\_Segundo(**void**){

ActualTime.seg++;

**if**(tiempo\_sin\_pulsar>0)

tiempo\_sin\_pulsar--;

}

#endif

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Ajuste\_Reloj*

*//*

*// Descripción: Menú para modificar la hora y fecha actual usando el encoder*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Ajuste\_Reloj(**void**)

{

**char** BufferAux[16];

**switch**(estado\_ajuste)

{

default:

estado\_ajuste=0;

**break**;

**case** 0:

lcd.setCursor(0, 0);

lcd.print(F("Seleccione hora "));

lcd.setCursor(0, 1);

lcd.print(F("y fecha actual "));

delay(3000);

estado\_ajuste++;

**break**;

**case** 1:

*//Muestro la fecha y hora actual*

lcd.setCursor(0, 0);

lcd.print(F("Dia: "));

**switch**(ActualTime.DoW)

{

default:

*//Distingo si me trae basura*

lcd.print(F("?? "));

**break**;

**case** DOMINGO:

lcd.print(F("Do "));

**break**;

**case** LUNES:

lcd.print(F("Lu "));

**break**;

**case** MARTES:

lcd.print(F("Ma "));

**break**;

**case** MIERCOLES:

lcd.print(F("Mi "));

**break**;

**case** JUEVES:

lcd.print(F("Ju "));

**break**;

**case** VIERNES:

lcd.print(F("Vi "));

**break**;

**case** SABADO:

lcd.print(F("Sa "));

**break**;

}

sprintf(BufferAux,"%02d/%02d/%02d",ActualTime.dia,ActualTime.mes,ActualTime.anio);

lcd.print(BufferAux);

lcd.setCursor(0, 1);

sprintf(BufferAux,"Hora: %02d:%02d ",ActualTime.hora,ActualTime.min);

lcd.print(BufferAux);

AuxTime = ActualTime;*//Copio la estructura para modificarla*

estado\_ajuste++;

**break**;

**case** 2:

*//Modifico el día de la semana*

lcd.setCursor(5, 0);

**switch**(AuxTime.DoW)

{

default:

*//Distingo si me trae basura*

lcd.print(F("?? "));

**break**;

**case** DOMINGO:

lcd.print(F("Do "));

**break**;

**case** LUNES:

lcd.print(F("Lu "));

**break**;

**case** MARTES:

lcd.print(F("Ma "));

**break**;

**case** MIERCOLES:

lcd.print(F("Mi "));

**break**;

**case** JUEVES:

lcd.print(F("Ju "));

**break**;

**case** VIERNES:

lcd.print(F("Vi "));

**break**;

**case** SABADO:

lcd.print(F("Sa "));

**break**;

}

**if**(digitalRead(Encoder\_Switch)==0)*//ENTER*

{

estado\_ajuste++;

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

}

**else**

{

AuxTime.DoW=Modificar\_Variable(AuxTime.DoW,1,7);

Efecto\_Titilar(5,0,2,100);

}

**break**;

**case** 3:

*//Modifico el día*

lcd.setCursor(8, 0);

sprintf(BufferAux,"%02d",AuxTime.dia);

lcd.print(BufferAux);

**if**(digitalRead(Encoder\_Switch)==0)*//ENTER*

{

estado\_ajuste++;

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

}

**else**

{

AuxTime.dia=Modificar\_Variable(AuxTime.dia,1,31);

Efecto\_Titilar(8,0,2,100);

}

**break**;

**case** 4:

*//Modifico el mes*

lcd.setCursor(11, 0);

sprintf(BufferAux,"%02d",AuxTime.mes);

lcd.print(BufferAux);

**if**(digitalRead(Encoder\_Switch)==0)*//ENTER*

{

**if**(Validar\_Fecha(AuxTime.dia,AuxTime.mes)==false)

{

lcd.setCursor(0, 0);

lcd.print(F("La fecha elegida"));

lcd.setCursor(0, 1);

lcd.print(F(" no existe "));

delay(3000);

estado\_ajuste=0;

estado\_menu=MENU\_PRINCIPAL;

}

**else**

estado\_ajuste++;

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

}

**else**

{

AuxTime.mes=Modificar\_Variable(AuxTime.mes,1,12);

Efecto\_Titilar(11,0,2,100);

}

**break**;

**case** 5:

*//Modifico el año*

lcd.setCursor(14, 0);

sprintf(BufferAux,"%02d",AuxTime.anio);

lcd.print(BufferAux);

**if**(digitalRead(Encoder\_Switch)==0)*//ENTER*

{

**if**((AuxTime.dia==29)&&(AuxTime.mes==2)&&(AuxTime.anio%4!=0))

{

lcd.setCursor(0, 0);

lcd.print(F("La fecha elegida"));

lcd.setCursor(0, 1);

lcd.print(F(" no existe "));

delay(3000);

estado\_ajuste=0;

estado\_menu=MENU\_PRINCIPAL;

}

**else**

estado\_ajuste++;

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

}

**else**

{

AuxTime.anio=Modificar\_Variable(AuxTime.anio,0,99);*//De 2023 a 2099*

Efecto\_Titilar(14,0,2,100);

}

**break**;

**case** 6:

*//Modifico la hora*

lcd.setCursor(8, 1);

sprintf(BufferAux,"%02d",AuxTime.hora);

lcd.print(BufferAux);

**if**(digitalRead(Encoder\_Switch)==0)*//ENTER*

{

estado\_ajuste++;

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

}

**else**

{

AuxTime.hora=Modificar\_Variable(AuxTime.hora,0,23);

Efecto\_Titilar(8,1,2,100);

}

**break**;

**case** 7:

*//Modifico los minutos*

lcd.setCursor(11, 1);

sprintf(BufferAux,"%02d",AuxTime.min);

lcd.print(BufferAux);

**if**(digitalRead(Encoder\_Switch)==0)*//ENTER*

{

estado\_ajuste++;

**while**(digitalRead(Encoder\_Switch)==0)

{

*//Espero a que suelte el pulsador*

};

}

**else**

{

AuxTime.min=Modificar\_Variable(AuxTime.min,0,59);

Efecto\_Titilar(11,1,2,100);

}

**break**;

**case** 8:

lcd.setCursor(0, 0);

lcd.print(F(" Modificacion "));

lcd.setCursor(0, 1);

lcd.print(F(" exitosa "));

*//Copio la hora seleccionada en la actual*

AuxTime.seg = 0;

ActualTime = AuxTime;

*//Como la fecha es válida guardo fecha y hora en el RTC*

RTC.setSecond(0);

RTC.setMinute(ActualTime.min);

RTC.setHour(ActualTime.hora);

RTC.setDoW(ActualTime.DoW);

RTC.setDate(ActualTime.dia);

RTC.setMonth(ActualTime.mes);

RTC.setYear(ActualTime.anio);

RTC.setClockMode(h12);

delay(3000);

estado\_ajuste=0;

estado\_menu=MENU\_PRINCIPAL;

modificacion\_realizada=1;

**break**;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Validar\_Fecha*

*//*

*// Descripción: Verifica que la combinación dia/mes exista*

*//*

*// Parámetros: dia y mes*

*//*

*// Valor devuelto: bool: true (fecha valida) false (fecha inexistente)*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**bool** Validar\_Fecha(**int** dia,**int** mes)

{

**if**(dia>cant\_dias\_mes[mes-1])

{

#ifdef DEBUG\_SERIE

Serial.print("cant dias del mes:");

Serial.print(cant\_dias\_mes[mes-1]);

#endif

**return** false;

}

**else**

**return** true;

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Efecto\_Titilar*

*//*

*// Descripción: Efecto que hace titilar los digitos de pantalla cada 1 seg*

*//*

*// Parámetros: int col (columna inicial)*

*// int fila (fila inicial)*

*// int digitos (digitos a hacer titilar)*

*// int delay\_ms (para ajustar el tiempo en off)*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Efecto\_Titilar(**int** col,**int** fila,**int** digitos,**int** delay\_ms)

{

**if**(ActualTime.seg != seg\_anterior)

{

lcd.setCursor(col,fila);

**for**(**int** i=0; i<digitos;i++)

lcd.print(F(" "));

delay(delay\_ms);

seg\_anterior=ActualTime.seg;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Modificar\_Variable*

*//*

*// Descripción: Permite modificar una variable con el encoder entre un*

*// mínimo y un máximo*

*//*

*// Parámetros: int variable\_a\_modificar*

*// int minimo*

*// int maximo*

*//*

*// Valor devuelto: int: devuelve la variable modificada*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**int** Modificar\_Variable(**int** variable\_a\_modificar, **int** minimo, **int** maximo)

{

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** DERECHA:

variable\_a\_modificar++;

**if**(variable\_a\_modificar > maximo)

variable\_a\_modificar=minimo;

**break**;

**case** IZQUIERDA:

variable\_a\_modificar--;

**if**(variable\_a\_modificar < minimo)

variable\_a\_modificar=maximo;

**break**;

}

**return**(variable\_a\_modificar);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Menu\_Feriados*

*//*

*// Descripción: Muestra los feriados y permite agregar o quitar.*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Menu\_Feriados(**void**)

{

**char** BufferAux[30];

**switch**(estado\_menu\_feriados)

{

default:

estado\_menu\_feriados=0;

**break**;

**case** 0:

*//Parto desde la hora actual*

AuxTime = ActualTime;

estado\_menu\_feriados++;

**break**;

**case** 1:

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** DERECHA:

AuxTime.dia++;

**if**(AuxTime.dia > cant\_dias\_mes[AuxTime.mes-1])

{

AuxTime.dia=1;

AuxTime.mes++;

**if**(AuxTime.mes>12)

AuxTime.mes=1;

}

**break**;

**case** IZQUIERDA:

AuxTime.dia--;

**if**(AuxTime.dia < 1)

{

AuxTime.mes--;

**if**(AuxTime.mes<1)

AuxTime.mes=12;

AuxTime.dia=cant\_dias\_mes[AuxTime.mes-1];

}

**break**;

**case** ENTER:

estado\_menu\_feriados++;

lcd.setCursor(0, 1);

lcd.print(F(">"));

lcd.setCursor(0, 0);

sprintf(BufferAux,"%02d/%02d",AuxTime.dia,AuxTime.mes);

lcd.print(BufferAux);

**return**;

**break**;

}

lcd.setCursor(0, 0);

sprintf(BufferAux,"%02d/%02d",AuxTime.dia,AuxTime.mes);

lcd.print(BufferAux);

lcd.setCursor(6, 0);

**if**(verificaFeriado(AuxTime.dia,AuxTime.mes) == true)

{

lcd.print(F("es feriado"));

lcd.setCursor(1, 1);

lcd.print(F("Eliminar"));

feriado = ES\_FERIADO;

}

**else**

{

lcd.print(F(" "));

lcd.setCursor(1, 1);

lcd.print(F("Agregar "));

feriado = NO\_ES\_FERIADO;

}

cursor\_feriado=0;

lcd.setCursor(11, 1);

lcd.print(F("Salir"));

*//Hago titilar a los números*

Efecto\_Titilar(0,0,5,100);

lcd.setCursor(2, 0);

lcd.print(F("/"));

**break**;

**case** 2:

*//Elijo entre Agregar/Eliminar o Salir*

**switch**(Leer\_Encoder())

{

default:

**break**;

**case** DERECHA:

cursor\_feriado=1;

lcd.setCursor(0, 1);

lcd.print(F(" "));

lcd.setCursor(10, 1);

lcd.print(F(">"));

**break**;

**case** IZQUIERDA:

cursor\_feriado=0;

lcd.setCursor(0, 1);

lcd.print(F(">"));

lcd.setCursor(10, 1);

lcd.print(F(" "));

**break**;

**case** ENTER:

estado\_menu\_feriados++;

**break**;

}

**break**;

**case** 3:

**if**(cursor\_feriado == 0)

{

modificacion\_realizada=true;

lcd.setCursor(0, 0);

*// Modifico el estado actual*

**if**(feriado== ES\_FERIADO)

{

seteaFeriado(AuxTime.dia, AuxTime.mes, NO\_ES\_FERIADO);

sprintf(BufferAux,"%02d/%02d se quita ",AuxTime.dia,AuxTime.mes);

lcd.print(BufferAux);

lcd.setCursor(0, 1);

lcd.print(F("de los feriados "));

}

**else**

{

seteaFeriado(AuxTime.dia, AuxTime.mes, ES\_FERIADO);

sprintf(BufferAux,"%02d/%02d se agrega ",AuxTime.dia,AuxTime.mes);

lcd.print(BufferAux);

lcd.setCursor(0, 1);

lcd.print(F("a los feriados "));

}

delay(3000);

*//Me permite seguir eligiendo feriados*

estado\_menu\_feriados=1;

lcd.clear();

}

**else**

{

*//Presionó Salir*

estado\_menu\_feriados=0;

estado\_menu=MENU\_PRINCIPAL;

}

**break**;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: verificaFeriado*

*//*

*// Descripción: Chequea en la EEPROM si hoy es feriado*

*//*

*// Parámetros: int dia, int mes*

*//*

*// Valor devuelto: bool: true (es feriado), false (no es feriado)*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**bool** verificaFeriado(**int** dia, **int** mes)

{

*//Obtenemos el dia del año en formato "M|D"*

**int** indexMonthEeprom = mes - 1;

**int** daysAccum = 0;

**for**(**int** i = 0; i < indexMonthEeprom; i++)

{

daysAccum += cant\_dias\_mes[i];

}

daysAccum += ULTIMA\_POSICION\_EVENTOS + dia;

**if**(EEPROM.read(daysAccum) == ES\_FERIADO)

{

#ifdef DEBUG\_SERIE

Serial.print(F("Feriado detectado\n"));

#endif

**return** true;

}

**else**

{

**return** false;

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: seteaFeriado*

*//*

*// Descripción: Setea como feriado el conjunto dia/mes recibido guardandolo*

*// en EEPROM*

*//*

*// Parámetros: int dia, int mes, bool feriado (para indicar si es o no es)*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** seteaFeriado(**int** dia, **int** mes, **bool** feriado)

{

**int** indexMonthEeprom = mes - 1;

**int** daysAccum = 0;

**char** buff\_aux[4];

**for**(**int** i = 0; i < indexMonthEeprom; i++)

{

daysAccum += cant\_dias\_mes[i];

}

daysAccum += ULTIMA\_POSICION\_EVENTOS + dia;

EEPROM.write(daysAccum, feriado);

*//Trato de guardar los cambios si hay uSD*

Guardar\_Cambios\_uSD();

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: Guardar\_Cambios\_uSD*

*//*

*// Descripción: Si está colocada la microSD, guarda las modificaciones*

*// ya sea de feriados o de eventos*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** Guardar\_Cambios\_uSD(**void**)

{

**char** buff\_feriado[4];

**char** buff\_eventos[6];

**int** valor;

*//Estructura para el archivo de la uSD*

File myFile;

*//Borro el archivo actual para crear uno nuevo*

SD.remove("feriados.txt");

myFile = SD.open("feriados.txt",FILE\_WRITE);

**if**(myFile)

{

*//Guardo los feriados*

**for**(**int** i=PRIMERA\_POSICION\_FERIADOS; i<=ULTIMA\_POSICION\_FERIADOS;i++)

{

**if**(EEPROM.read(i)==1)

{

sprintf(buff\_feriado,"%d,",i);

myFile.print(buff\_feriado);

#ifdef DEBUG\_SERIE

Serial.print(buff\_feriado);

#endif

}

}

*//Ya terminé con los feriados*

myFile.print('|');

*//Guardo los eventos*

**for**(**int** i=PRIMERA\_POSICION\_EVENTOS; i<=ULTIMA\_POSICION\_EVENTOS;i++)

{

*//Guardo posicion:valor, asegurandome que tanto posicion como valor ocupen 2 dígitos*

valor = EEPROM.read(i);

**if**(valor > MAX\_VALOR\_HORARIOS)

valor=0;

sprintf(buff\_eventos, "%02d:%02d,", i, valor);

myFile.print(buff\_eventos);

#ifdef DEBUG\_SERIE

Serial.print(buff\_eventos);

#endif

}

*//Ya terminé con los eventos*

myFile.print('.');

myFile.close();

}

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: configuraInterrupcionRTC*

*//*

*// Descripción: Configura la Interrupción del RTC*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** configuraInterrupcionRTC (**void**)

{

*// Set alarm 1 to fire at one-second intervals*

RTC.turnOffAlarm(1);

RTC.setA1Time(0, 0, 0, 0, alarmBits, false, false, false);

*// enable Alarm 1 interrupts*

RTC.turnOnAlarm(1);

*// clear Alarm 1 flag*

RTC.checkIfAlarm(1);

pinMode(CLINT, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(CLINT), isr\_TickTock, FALLING);

*// Use builtin LED to blink*

*//pinMode(LED\_BUILTIN, OUTPUT);*

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: isr\_TickTock*

*//*

*// Descripción: Interrupción del RTC*

*//*

*// Parámetros: void*

*//*

*// Valor devuelto: void*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** isr\_TickTock(**void**) {

*// interrupt signals to loop*

tick = 1;

**return**;

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Función: loop*

*//*

*// Descripción: Bucle principal del programa*

*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**void** loop(){

#ifdef DEBUG\_SERIE

**char** BufferAux[50];

#endif

**int** min\_aux, seg\_aux;

*//Reloj();*

**if**(estado\_menu != estado\_menu\_anterior)

{

lcd.clear();

*//proximo\_menu = 0;*

estado\_menu\_anterior = estado\_menu;

delay(300);

}

**switch**(estado\_menu)

{

default:

**case** MENU\_PRINCIPAL:

Menu\_Principal();

**break**;

**case** MENU\_AJUSTE\_FECHA\_HORA:

Ajuste\_Reloj();

**break**;

**case** MENU\_CONFIGURACION:

Seleccion\_Configuracion();

**break**;

**case** MENU\_PROGRAMA\_SEMANAL:

Programacion\_Semanal();

**break**;

**case** MENU\_FERIADOS:

Menu\_Feriados();

**break**;

**case** MENU\_CAMBIOS\_SD:

lcd.clear();

digitalWrite(Salida\_Backlight,0);*//se enciende el backlight*

tiempo\_sin\_pulsar=TIEMPO\_PARA\_APAGAR\_LCD;

lcd.setCursor(0, 0);

lcd.print(F("Leyendo memoria"));

delay(2000);*//2seg*

lcd.setCursor(0, 0);

lcd.print(F(" Modificaciones "));

lcd.setCursor(0, 1);

lcd.print(F(" realizadas "));

delay(4000);*//4seg*

modificacion\_realizada=1;

estado\_menu = MENU\_PRINCIPAL;

**break**;

}

min\_aux=RTC.getMinute();

seg\_aux=RTC.getSecond();

*//El tick no lo estamos utilizando*

**if**(seg\_aux != seg\_ant)

{

*//Aca entra cada 1seg*

ActualTime.seg++;

*//Cada 5 segundos me fijo si metieron una uSD para cargar los feriados*

**if**(ActualTime.seg%10==0)

leerFeriadosDeSd();

**if**(tiempo\_sin\_pulsar>0)

tiempo\_sin\_pulsar--;

**if**((min\_aux != min\_ant)||(modificacion\_realizada))

{

*//Aca entra cada un minuto o cuando se haga una modificación en el Reloj o en los Rangos*

modificacion\_realizada=0;

Leer\_Fecha\_Hora\_RTC(ActualTime);

*//Si cambia el día, leo los rangos permitidos*

**if**(ultimo\_dia != ActualTime.DoW)

{

Cargar\_Rangos();

ultimo\_dia=ActualTime.DoW;

}

codigo\_hora\_actual = (ActualTime.hora\*4)+(ActualTime.min/15);

**if**(codigo\_hora\_actual != codigo\_hora\_actual\_anterior)

{

#ifdef DEBUG\_SERIE

sprintf(BufferAux,"Codigo actual: %d\n",codigo\_hora\_actual);

Serial.print(BufferAux);

#endif

codigo\_hora\_actual\_anterior=codigo\_hora\_actual;

}

*//Sólo con esto manejo la salida*

**if**((((codigo\_hora\_actual>=Rangos\_hoy[INICIO\_1])&&(codigo\_hora\_actual<Rangos\_hoy[FIN\_1]))|| ((codigo\_hora\_actual>=Rangos\_hoy[INICIO\_2])&&(codigo\_hora\_actual<Rangos\_hoy[FIN\_2]))||

((codigo\_hora\_actual>=Rangos\_hoy[INICIO\_3])&&(codigo\_hora\_actual<Rangos\_hoy[FIN\_3]))|| ((codigo\_hora\_actual>=Rangos\_hoy[INICIO\_4])&&(codigo\_hora\_actual<Rangos\_hoy[FIN\_4])))&& (!verificaFeriado(ActualTime.dia,ActualTime.mes)))

{

*//Si no es feriado y se encuentra dentro de alguno de los rangos programados, alimento la carga*

digitalWrite(Salida\_Rele,1);

}

**else**

{

*//Si se encuentra fuera de los rangos dejo de alimentar la carga*

digitalWrite(Salida\_Rele,0);

}

*// Clear Alarm 1 flag*

RTC.checkIfAlarm(1);

}

}

min\_ant = min\_aux;

seg\_ant = seg\_aux;

*//Controlo el backlight del lcd*

**if**(!tiempo\_sin\_pulsar)

digitalWrite(Salida\_Backlight,1);*//Si pasa cierto tiempo en el que no se tocó el encoder, se apaga el display*

**else**

digitalWrite(Salida\_Backlight,0);*//Si se toca el encoder, se enciende*

}

## 

## 4.2. Guardado de datos en memoria

Para guardar los datos relacionados a la programación del equipo se ha utilizado la memoria EEPROM interna del ATMEGA328P (1 KB de memoria). Las primeras 56 posiciones (0 a 55) se utilizaron para la programación de los eventos semanales, mientras que las siguientes 365 posiciones se utilizaron para indicar los días feriados.

* *Eventos*

Se establecieron los siguientes parámetros:

|  |  |
| --- | --- |
| LUNES | 0 |
| MARTES | 1 |
| MIÉRCOLES | 2 |
| JUEVES | 3 |
| VIERNES | 4 |
| SÁBADO | 5 |
| DOMINGO | 6 |
|  |  |
| ENCENDIDO\_1 | 0 |
| APAGADO\_1 | 1 |
| ENCENDIDO\_2 | 2 |
| APAGADO\_2 | 3 |
| ENCENDIDO\_3 | 4 |
| APAGADO\_3 | 5 |
| ENCDENDIDO\_4 | 6 |
| APAGADO\_4 | 7 |

La posición de memoria que representa cada combinación se obtiene con la siguiente fórmula:

Además, para cada horario del día se genera un código, el cual se guarda en esa posición de memoria. Este código puede ser un valor entre 0 y 95, y se obtiene de la siguiente fórmula:

Hay que recordar que los lapsos de horarios son cada 15 minutos.

**Ejemplo:**

Si se quiere guardar que el tercer encendido del jueves ocurra a las 19:45, el código realizará lo siguiente:

Se obtiene en la EEPROM:

|  |  |
| --- | --- |
| **Posición** | **Dato** |
| 28 [0x1C] | 79 [0x4F] |

* *Feriados*

Cada día del año, incluyendo al 29 de febrero, tiene asignada una posición de memoria, comenzando desde el 01/01 (posición 56 [0x38]) hasta el 31/12 (posición 421 [0x1A5]).

Si el día es considerado feriado, guardará un 1 en su respectiva posición. En cambio, si no lo es, guardará un 0.

# 5. Software

A continuación, se documenta el software para la creación del archivo *feriados.txt*, al cual se le puede cargar feriados y rangos para luego ser leídos por el dispositivo.

El Proyecto en Java 8 se compone de una clase ProyectoPotencia.java que contiene la clase **main** y se crean 3 clases (Dia, Evento y Mes) para crear instancias de estas y usarlas en la clase principal.

Para ejecutar el programa se requiere tener instalado java 8 o una versión superior y ejecutar en consola el comando:  
  
*java -jar Proyecto\_Potencia\_2023.jar*

Se adjunta el detalle de las clases descritas anteriormente:

**Clase ProyectoPotencia.java**

package proyecto**.**potencia**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**OptionalInt**;**

**import** java**.**util**.**Scanner**;**

**import** java**.**util**.**stream**.**Collectors**;**

**import** java**.**util**.**stream**.**IntStream**;**

**import** java**.**io**.**File**;**

**import** java**.**io**.**FileWriter**;**

**import** java**.**io**.**IOException**;**

**import** java**.**util**.**Collections**;**

**import** java**.**util**.**HashMap**;**

/\*\*

\*

\* **@author** ELECTRONICA DE POTENCIA - SOFTWARE PARA CARGA DE FERIADOS Y RANGOS EN

\* SD

\*

\* Clase ProyectoPotencia con metodo Main

\*

\*/

public class ProyectoPotencia **{**

//Listado de Meses

private static List**<**Mes**>** meses **=** **new** ArrayList**<>();**

//Se inicializan los meses con un id

private static Mes enero **=** **new** Mes**(**1**,** "Enero"**);**

private static Mes febrero **=** **new** Mes**(**2**,** "Febrero"**);**

private static Mes marzo **=** **new** Mes**(**3**,** "Marzo"**);**

private static Mes abril **=** **new** Mes**(**4**,** "Abril"**);**

private static Mes mayo **=** **new** Mes**(**5**,** "Mayo"**);**

private static Mes junio **=** **new** Mes**(**6**,** "Junio"**);**

private static Mes julio **=** **new** Mes**(**7**,** "Julio"**);**

private static Mes agosto **=** **new** Mes**(**8**,** "Agosto"**);**

private static Mes septiembre **=** **new** Mes**(**9**,** "Septiembre"**);**

private static Mes octubre **=** **new** Mes**(**10**,** "Octubre"**);**

private static Mes noviembre **=** **new** Mes**(**11**,** "Noviembre"**);**

private static Mes diciembre **=** **new** Mes**(**12**,** "Diciembre"**);**

//Listado de dias de la semana

private static List**<**Dia**>** dias **=** **new** ArrayList**<>();**

//Se inicializan los dias de la semana con un id

private static Dia lunes **=** **new** Dia**(**1**,** "Lunes"**);**

private static Dia martes **=** **new** Dia**(**2**,** "Martes"**);**

private static Dia miercoles **=** **new** Dia**(**3**,** "Miercoles"**);**

private static Dia jueves **=** **new** Dia**(**4**,** "Jueves"**);**

private static Dia viernes **=** **new** Dia**(**5**,** "Viernes"**);**

private static Dia sabado **=** **new** Dia**(**6**,** "Sábado"**);**

private static Dia domingo **=** **new** Dia**(**0**,** "Domingo"**);**

private static HashMap**<**Integer**,** Integer**>** decoMeses **=** **new** HashMap**<>();**

private static HashMap**<**Integer**,** Integer**>** decoDias **=** **new** HashMap**<>();**

public static void main**(**String**[]** args**)** **{**

String logo **=** " \n"

**+** " \n"

**+** " \n"

**+** " ..:::^^^^^^:::.. \n"

**+** " .:^~!77777?????????7777!~^:. \n"

**+** " .^!77??JJY55PPPPPPP55YJJ??77???77!^. \n"

**+** " :~77???JY555PPPPGGBB#&@@@@@&#GPY?777??77~: \n"

**+** " .^!???777777777777777777??J5PB&@@@@&#PY?77???7^. \n"

**+** " .~7??77?JY5PGB######BBGP5J?77777?JP#@@@@@&GJ777??7~. \n"

**+** " .~7????Y5PPGPPPPPPGBB#&@@@@@&#G5J77777JP&@@@@@BY777??7~. \n"

**+** " :7??77????7777777777777?JYPB&@@@@@#5?7777?5#@@@@@BJ777??7: \n"

**+** " ~??777777?JY5PGGGGP55J?777777?5B&@@@@&G?7777?5&@@@@@57777??~ \n"

**+** " ~??777?YPB&@@@@@@@@@@@@&#GY?77777JG&@@@@&577777?B@@@@@G7777??~ \n"

**+** " ~??77?P#@@@@@@@@@@@@@@@@@@@@&PJ77777JB@@@@@B?77777P@@@@@G77777?~ \n"

**+** " ^??7?P&@@@@@@@@@@@@@@@@@@@@@@@@&P?77777P@@@@@B?77777G@@@@@57777??^ \n"

**+** " .7?7J#@@@@@@@@@@@@@@@@@@@@@@@@@@@@B?77777P@@@@@G77777?&@@@@&?7777?7. \n"

**+** " ~?7?#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@B?7777?#@@@@@J77777G@@@@@577777?~ \n"

**+** " 7?7G@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@5777775@@@@@P777775@@@@@G77777?7. \n"

**+** " .77J@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#77777J@@@@@B77777J@@@@@B77777?7. \n"

**+** " :?7JGGGP#@@@@@@@@@@@@@@@@@GGGGGG@@@@@&PPPP5P@@@@@#55555G@@@@@&555PJ7?: \n"

**+** " .7?77777P@@@@@@@@@@@@@@@@&?7777?&@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@Y7?: \n"

**+** " .7?77777Y@@@@@@@@@@@@@@@@@Y77777P@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@B??7. \n"

**+** " ~?77777?#@@@@@@@@@@@@@@@@B77777?B@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@&J7?! \n"

**+** " :7?77777Y@@@@@@@@@@@@@@@@@577777?G@@@@@@@@@@@@@@@@@@@@@@@@@@@@#J7??: \n"

**+** " ~?777777P@@@@@@@@@@@@@@@@@Y777777Y#@@@@@@@@@@@@@@@@@@@@@@@@&P?77?~ \n"

**+** " !?777777P@@@@@@@@@@@@@@@@@P?777777YB&@@@@@@@@@@@@@@@@@@@#P?777?!. \n"

**+** " .!??777775&@@@@@@@@@@@@@@@@#Y?777777?YG#&@@@@@@@@@@@&BPJ?7777?!. \n"

**+** " ~??77777JB@@@@@@@@@@@@@@@@@#5J77777777?JY55PPP55YJ?777777??!. \n"

**+** " ^7??77777Y#@@@@@@@@@@@@@@@@@&BPY?77777777777777777??77??7^ \n"

**+** " .!???77777YB&@@@@@@@@@@@@@@@@@@&#BP5YJJJJJJJYY5GG5?7??!. \n"

**+** " :!???77777?5B&@@@@@@@@@@@@@@@@@@@@@@@@@@@@@&#PJ7??!: \n"

**+** " :~7???77777?YPB&@@@@@@@@@@@@@@@@@@@@@&BGY?7??7~: \n"

**+** " .^!7????77777?JYPGBB##&&&&&&##BGPYJ?77??7!^. \n"

**+** " .:~!7?????7777777?????????7777???7!~:. \n"

**+** " .:^~!77????????????????77!~^:. \n"

**+** " .::^^^~~~~~~^^^::. \n"

**+** " \n"

**+** " ------------------------------------------------------------------------------------------------ \n"

**+** " UNLAM - Electronica de Potencia - 1° Cuatrimestre 2023 \n"

**+** " ------------------------------------------------------------------------------------------------ \n"

**+** " "**;**

System**.**out**.**println**(**logo**);**

System**.**out**.**println**(**"MENU: "**);**

System**.**out**.**println**(**""**);**

//INICIALIZACIONES

meses**.**add**(**enero**);**

meses**.**add**(**febrero**);**

meses**.**add**(**marzo**);**

meses**.**add**(**abril**);**

meses**.**add**(**mayo**);**

meses**.**add**(**junio**);**

meses**.**add**(**julio**);**

meses**.**add**(**agosto**);**

meses**.**add**(**septiembre**);**

meses**.**add**(**octubre**);**

meses**.**add**(**noviembre**);**

meses**.**add**(**diciembre**);**

decoMeses**.**put**(**1**,** 31**);**

decoMeses**.**put**(**2**,** 29**);**

decoMeses**.**put**(**3**,** 31**);**

decoMeses**.**put**(**4**,** 30**);**

decoMeses**.**put**(**5**,** 31**);**

decoMeses**.**put**(**6**,** 30**);**

decoMeses**.**put**(**7**,** 31**);**

decoMeses**.**put**(**8**,** 31**);**

decoMeses**.**put**(**9**,** 30**);**

decoMeses**.**put**(**10**,** 31**);**

decoMeses**.**put**(**11**,** 30**);**

decoMeses**.**put**(**12**,** 31**);**

dias**.**add**(**lunes**);**

dias**.**add**(**martes**);**

dias**.**add**(**miercoles**);**

dias**.**add**(**jueves**);**

dias**.**add**(**viernes**);**

dias**.**add**(**sabado**);**

dias**.**add**(**domingo**);**

decoDias**.**put**(**0**,** 0**);**

decoDias**.**put**(**1**,** 8**);**

decoDias**.**put**(**2**,** 16**);**

decoDias**.**put**(**3**,** 24**);**

decoDias**.**put**(**4**,** 32**);**

decoDias**.**put**(**5**,** 40**);**

decoDias**.**put**(**6**,** 48**);**

//Se cargan en una lista los items del menu

List**<**String**>** lista **=** **new** ArrayList**<>();**

lista**.**add**(**"Cargar Feriados"**);**

lista**.**add**(**"Ver Feriados Cargados"**);**

lista**.**add**(**"Eliminar todos los feriados"**);**

lista**.**add**(**"Eliminar un Feriado"**);**

lista**.**add**(**"Cargar Rangos"**);**

lista**.**add**(**"Ver Rangos Cargados"**);**

lista**.**add**(**"Eliminar todos los rangos"**);**

lista**.**add**(**"Eliminar un Rango"**);**

lista**.**add**(**"Guardar Feriados en SD y Dias"**);**

lista**.**add**(**"Leer SD"**);**

lista**.**add**(**"Salir"**);**

Scanner scanner **=** **new** Scanner**(**System**.**in**);**

//Se despliega el menu en consola

**while** **(true)** **{**

System**.**out**.**println**(**"Selecciona un elemento de la lista (ingresa el índice):"**);**

**for** **(**int i **=** 0**;** i **<** lista**.**size**();** i**++)** **{**

System**.**out**.**println**(**i **+** ": " **+** lista**.**get**(**i**));**

**}**

int indiceSeleccionado **=** scanner**.**nextInt**();**

**if** **(**indiceSeleccionado **>=** 0 **&&** indiceSeleccionado **<** lista**.**size**())** **{**

**switch** **(**indiceSeleccionado**)** **{**

**case** 0**:**

cargarFeriados**(**scanner**);**

**break;**

**case** 1**:**

verFeriadosCargados**();**

**break;**

**case** 2**:**

eliminarTodoLosFeriados**();**

**break;**

**case** 3**:**

eliminarUnFeriado**(**scanner**);**

**break;**

**case** 4**:**

cargarEventos**(**scanner**);**

**break;**

**case** 5**:**

verEentosCargados**();**

**break;**

**case** 6**:**

eliminarTodoLosEventos**();**

**break;**

**case** 7**:**

eliminarUnEvento**(**scanner**);**

**break;**

**case** 8**:**

guardarCalendarioEnSD**(**scanner**);**

**break;**

**case** 9**:**

leersd**(**scanner**);**

**break;**

**case** 10**:**

System**.**exit**(**0**);**

**}**

**}** **else** **{**

System**.**out**.**println**(**"Índice inválido. No se seleccionó ningún elemento."**);**

**}**

**}**

**}**

//Este Metodo permite cargar feriados en el formato especificado

private static void cargarFeriados**(**Scanner scanner**)** **{**

**while** **(true)** **{**

clearScreen**();**

System**.**out**.**println**(**"Escriba el feriado en formato dd/mm: "**);**

String feriado **=** scanner**.**next**();**

int mes **=** Integer**.**parseInt**(**feriado**.**split**(**"/"**)[**1**]);**

int dia **=** Integer**.**parseInt**(**feriado**.**split**(**"/"**)[**0**]);**

//Busco el mes en el array

Mes mesBuscado **=** meses**.**stream**()**

**.**filter**(**m **->** m**.**indice **==** mes**)**

**.**collect**(**Collectors**.**toList**()).**get**(**0**);**

//Cargo el feriado al array de dias en el mes

mesBuscado**.**feriados**.**add**(**dia**);**

//Obtengo el indice del mes del array

OptionalInt indexMeses **=** IntStream**.**range**(**0**,** meses**.**size**())**

**.**filter**(**i **->** mes **==** meses**.**get**(**i**).**indice**)**

**.**findFirst**();**

meses**.**set**(**indexMeses**.**getAsInt**(),** mesBuscado**);**

System**.**out**.**println**(**"Se agregó el feriado: " **+** dia **+** " de " **+** mesBuscado**.**name**);**

System**.**out**.**println**(**"¿Desea agregar otro feriado? (s/n)"**);**

String agregarOtroferiado **=** scanner**.**next**();**

**if** **(!**agregarOtroferiado**.**equalsIgnoreCase**(**"s"**))** **{**

**break;**

**}**

**}**

**}**

//Permite visualizar el listado de feriados

private static void verFeriadosCargados**()** **{**

clearScreen**();**

List**<**String**>** listadoFeriados **=** **new** ArrayList**<>();**

meses**.**stream**().**filter**((**mes**)** **->** **(!**mes**.**feriados**.**isEmpty**())).**forEachOrdered**((**mes**)** **->** **{**

mes**.**feriados**.**forEach**((**feriado**)** **->** **{**

listadoFeriados**.**add**(**feriado **+** "/" **+** mes**.**indice**);**

**});**

**});**

System**.**out**.**println**(**"Feriados Cargados:"**);**

listadoFeriados**.**forEach**((**feriado**)** **->** **{**

System**.**out**.**println**(**feriado**);**

**});**

System**.**out**.**println**(**""**);**

**}**

//Permite vizualisar los rangos cargados en formato de lista

private static void verEentosCargados**()** **{**

clearScreen**();**

dias**.**stream**().**filter**((**dia**)** **->** **(!**dia**.**eventos**.**isEmpty**())).**forEachOrdered**((**dia**)** **->** **{**

System**.**out**.**println**(**"[" **+** dia**.**name **+** "]"**);**

System**.**out**.**println**(**"|"**);**

dia**.**eventos**.**forEach**((**evento**)** **->** **{**

System**.**out**.**println**(**"|-> Inicio: " **+** evento**.**horaInicio **+** " Fin: " **+** evento**.**horaFin**);**

**});**

System**.**out**.**println**(**""**);**

**});**

System**.**out**.**println**(**""**);**

**}**

//Elimina todos los feriados de la variable meses

private static void eliminarTodoLosFeriados**()** **{**

System**.**out**.**print**(**"\033[H\033[2J"**);**

System**.**out**.**flush**();**

**for** **(**int i **=** 0**;** i **<** meses**.**size**();** i**++)** **{**

meses**.**get**(**i**).**feriados**.**clear**();**

**}**

System**.**out**.**println**(**"Se eliminaron todos los feriados."**);**

System**.**out**.**println**(**""**);**

System**.**out**.**println**(**""**);**

**}**

//Elimina todos los Rangos de la variable dias

private static void eliminarTodoLosEventos**()** **{**

System**.**out**.**print**(**"\033[H\033[2J"**);**

System**.**out**.**flush**();**

**for** **(**int i **=** 0**;** i **<** dias**.**size**();** i**++)** **{**

dias**.**get**(**i**).**eventos**.**clear**();**

**}**

System**.**out**.**println**(**"Se eliminaron todos los rangos."**);**

System**.**out**.**println**(**""**);**

System**.**out**.**println**(**""**);**

**}**

//Permite eliminar un feriado en especifico de la variable meses

private static void eliminarUnFeriado**(**Scanner scanner**)** **{**

clearScreen**();**

System**.**out**.**println**(**"Escriba el feriado a eliminar en formato dd/mm: "**);**

String feriado **=** scanner**.**next**();**

int mes **=** Integer**.**parseInt**(**feriado**.**split**(**"/"**)[**1**]);**

int dia **=** Integer**.**parseInt**(**feriado**.**split**(**"/"**)[**0**]);**

//Obtengo el indice del mes del array

OptionalInt indexMeses **=** IntStream**.**range**(**0**,** meses**.**size**())**

**.**filter**(**i **->** mes **==** meses**.**get**(**i**).**indice**)**

**.**findFirst**();**

//Obtengo el indice del mes del array

OptionalInt indexDia **=** IntStream**.**range**(**0**,** meses**.**get**(**indexMeses**.**getAsInt**()).**feriados**.**size**())**

**.**filter**(**i **->** dia **==** meses**.**get**(**indexMeses**.**getAsInt**()).**feriados**.**get**(**i**))**

**.**findFirst**();**

meses**.**get**(**indexMeses**.**getAsInt**()).**feriados**.**remove**(**indexDia**.**getAsInt**());**

System**.**out**.**println**(**"Se eliminó el feriado " **+** dia **+** "/" **+** mes**);**

System**.**out**.**println**(**""**);**

System**.**out**.**println**(**""**);**

**}**

//Metodo eliminar un rango en especifico de la variable dias

private static void eliminarUnEvento**(**Scanner scanner**)** **{**

clearScreen**();**

System**.**out**.**println**(**"Seleccione el dia del rango que desea eliminar: "**);**

System**.**out**.**println**(**"0: Lunes"**);**

System**.**out**.**println**(**"1: Martes"**);**

System**.**out**.**println**(**"2: Miercoles"**);**

System**.**out**.**println**(**"3: Jueves"**);**

System**.**out**.**println**(**"4: Viernes"**);**

System**.**out**.**println**(**"5: Sabado"**);**

System**.**out**.**println**(**"6: Domingo"**);**

String diaDelEvento **=** scanner**.**next**();**

int diaDelEventoInt **=** Integer**.**parseInt**(**diaDelEvento**);**

//Busco el dia en el array

Dia diabuscado **=** dias**.**stream**()**

**.**filter**(**d **->** d**.**indice **==** diaDelEventoInt**)**

**.**collect**(**Collectors**.**toList**()).**get**(**0**);**

//Obtengo el indice del mes del array

OptionalInt indexDia **=** IntStream**.**range**(**0**,** dias**.**size**())**

**.**filter**(**i **->** diaDelEventoInt **==** dias**.**get**(**i**).**indice**)**

**.**findFirst**();**

dias**.**get**(**indexDia**.**getAsInt**()).**eventos**.**remove**(**indexDia**.**getAsInt**());**

System**.**out**.**println**(**"Se eliminó el Rango del dia " **+** diabuscado**.**name**);**

System**.**out**.**println**(**""**);**

System**.**out**.**println**(**""**);**

**}**

//Metodo que permite cargar rangos y los almacena en la variable dias

private static void cargarEventos**(**Scanner scanner**)** **{**

**while** **(true)** **{**

clearScreen**();**

System**.**out**.**println**(**"Seleccione un día de la semana: "**);**

System**.**out**.**println**(**"0: Domingo"**);**

System**.**out**.**println**(**"1: Lunes"**);**

System**.**out**.**println**(**"2: Martes"**);**

System**.**out**.**println**(**"3: Miercoles"**);**

System**.**out**.**println**(**"4: Jueves"**);**

System**.**out**.**println**(**"5: Viernes"**);**

System**.**out**.**println**(**"6: Sabado"**);**

String diaDelEvento **=** scanner**.**next**();**

int diaDelEventoInt **=** Integer**.**parseInt**(**diaDelEvento**);**

//Busco el dia en el array

Dia diabuscado **=** dias**.**stream**()**

**.**filter**(**d **->** d**.**indice **==** diaDelEventoInt**)**

**.**collect**(**Collectors**.**toList**()).**get**(**0**);**

int eventoInt **=** **-**1**;**

**if** **(**diabuscado**.**eventos**.**size**()** **==** 4**)** **{**

System**.**out**.**println**(**"Los Eventos de este día estan completos, por favor seleccione un rango a sobrescribir:"**);**

System**.**out**.**println**(**"0: Rango 1"**);**

System**.**out**.**println**(**"1: Rango 2"**);**

System**.**out**.**println**(**"2: Rango 3"**);**

System**.**out**.**println**(**"3: Rango 4"**);**

String eventoSeleccionado **=** scanner**.**next**();**

eventoInt **=** Integer**.**parseInt**(**eventoSeleccionado**);**

**}**

System**.**out**.**println**(**"Seleccione el horario inicial del rango en formato hh:mm"**);**

System**.**out**.**println**(**"(Los minutos deben ser múltiplo de 15)"**);**

String horarioInicial **=** scanner**.**next**();**

System**.**out**.**println**(**"Seleccione el horario final del rango en formato hh:mm"**);**

System**.**out**.**println**(**"(Los minutos deben ser múltiplo de 15)"**);**

String horarioFinal **=** scanner**.**next**();**

Evento evento **=** **new** Evento**(**horarioInicial**,** horarioFinal**);**

**if** **(**eventoInt **<** 0**)** **{**

diabuscado**.**eventos**.**add**(**evento**);**

**}** **else** **{**

diabuscado**.**eventos**.**get**(**eventoInt**).**horaInicio **=** evento**.**horaInicio**;**

diabuscado**.**eventos**.**get**(**eventoInt**).**horaFin **=** evento**.**horaFin**;**

diabuscado**.**eventos**.**get**(**eventoInt**).**valorCalculadoInicio **=** evento**.**valorCalculadoInicio**;**

diabuscado**.**eventos**.**get**(**eventoInt**).**valorCalculadoFinal **=** evento**.**valorCalculadoFinal**;**

**}**

//Obtengo el indice del mes del array

OptionalInt indexDias **=** IntStream**.**range**(**0**,** dias**.**size**())**

**.**filter**(**i **->** diaDelEventoInt **==** dias**.**get**(**i**).**indice**)**

**.**findFirst**();**

dias**.**set**(**indexDias**.**getAsInt**(),** diabuscado**);**

System**.**out**.**println**(**"Se agregó el rango al día " **+** diabuscado**.**name **+** " [Hora Inicial: " **+** horarioInicial **+** " | Hora Final: " **+** horarioFinal **+** "]"**);**

System**.**out**.**println**(**"¿Desea agregar otro rango? (s/n)"**);**

String agregarOtroferiado **=** scanner**.**next**();**

**if** **(!**agregarOtroferiado**.**equalsIgnoreCase**(**"s"**))** **{**

**break;**

**}**

**}**

**}**

//Hace una lectura de los feriados y los rangos cargados en la SD y los mergea con los preexistentes

//durante la ejecución del programa

private static void leersd**(**Scanner scanner**)** **{**

clearScreen**();**

List**<**String**>** lista **=** **new** ArrayList**<>();**

**for** **(**File sysDrive **:** File**.**listRoots**())** **{**

lista**.**add**(**sysDrive**.**toString**());**

**}**

System**.**out**.**println**(**"Seleccione la unidad donde se encuentra la SD (ingresa el índice):"**);**

**for** **(**int i **=** 0**;** i **<** lista**.**size**();** i**++)** **{**

System**.**out**.**println**(**i **+** ": " **+** lista**.**get**(**i**));**

**}**

int indiceSeleccionado **=** scanner**.**nextInt**();**

String elementoSeleccionado **=** lista**.**get**(**indiceSeleccionado**);**

/\*

GUARDA EN SD

\*/

String nombreArchivo **=** "feriados.txt"**;**

// Tarjeta SD accesible

File directorioSD **=** **new** File**(**elementoSeleccionado**);**

**if** **(!**directorioSD**.**exists**()** **||** **!**directorioSD**.**isDirectory**())** **{**

System**.**out**.**println**(**"La tarjeta SD no está disponible."**);**

**return;**

**}**

int mes**;**

int dia**;**

**try** **{**

File archivo **=** **new** File**(**directorioSD**,** nombreArchivo**);**

Scanner myReader **=** **new** Scanner**(**archivo**);**

String data **=** ""**;**

**while** **(**myReader**.**hasNextLine**())** **{**

data **=** myReader**.**nextLine**();**

**}**

myReader**.**close**();**

//Carga de feriados

String feriadosLeidos **=** data**.**split**(**"\\|"**)[**0**];**

String eventosLeidos **=** data**.**split**(**"\\|"**)[**1**];**

String**[]** feriadosLeidosArray **=** feriadosLeidos**.**split**(**","**);**

String**[]** eventosLeidosArray **=** eventosLeidos**.**split**(**"[.]"**)[**0**].**substring**(**0**,** eventosLeidos**.**length**()** **-** 1**).**split**(**","**);**

//Obtengo Feriados Leidos Array

**for** **(**int i **=** 0**;** i **<** feriadosLeidos**.**split**(**","**).**length**;** i**++)** **{**

int accum **=** 55**;**

mes **=** 0**;**

dia **=** 0**;**

**for** **(**int j **=** 1**;** j **<=** decoMeses**.**size**();** j**++)** **{**

accum **=** accum **+** decoMeses**.**get**(**j**);**

**if** **(**Integer**.**parseInt**(**feriadosLeidosArray**[**i**])** **<** accum**)** **{**

mes **=** j**;**

dia **=** **(**accum **-** decoMeses**.**get**(**j**)** **-** Integer**.**parseInt**(**feriadosLeidosArray**[**i**]))** **\*** **-**1**;**

**break;**

**}**

**}**

cargarFeriadosForSdMethod**(**mes**,** dia**);**

**}**

//Carga de Eventos

ArrayList**<**ArrayList**<**Integer**>>** eventosHorarios **=** **new** ArrayList**<>();**

**for** **(**String posicionHorario **:** eventosLeidosArray**)** **{**

ArrayList**<**Integer**>** e **=** **new** ArrayList**();**

e**.**add**(**Integer**.**parseInt**(**posicionHorario**.**split**(**":"**)[**0**]));**

e**.**add**(**Integer**.**parseInt**(**posicionHorario**.**split**(**":"**)[**1**]));**

eventosHorarios**.**add**(**e**);**

**}**

Collections**.**sort**(**eventosHorarios**,** **(**ArrayList**<**Integer**>** o1**,** ArrayList**<**Integer**>** o2**)** **->** o1**.**get**(**0**).**compareTo**(**o2**.**get**(**0**)));**

//Eventos ordenados

**for** **(**int i **=** 0**;** i **<** eventosHorarios**.**size**();** i **=** i **+** 2**)** **{**

ArrayList**<**Integer**>** ev **=** **new** ArrayList**<>();**

ArrayList**<**Integer**>** evSiguiente **=** **new** ArrayList**<>();**

ev **=** eventosHorarios**.**get**(**i**);**

evSiguiente **=** eventosHorarios**.**get**(**i **+** 1**);**

//determina dia

int diaEvento **=** 0**;**

**for** **(**int j **=** 0**;** j **<** decoDias**.**size**();** j**++)** **{**

**if** **(**decoDias**.**get**(**j**)** **>** ev**.**get**(**0**))** **{**

diaEvento **=** j**;**

**break;**

**}**

**}**

**for** **(**int z **=** 0**;** z **<** decoDias**.**size**();** z**++)** **{**

**if** **(**decoDias**.**get**(**z**)** **>** ev**.**get**(**0**))** **{**

diaEvento **=** z **-** 1**;**

**break;**

**}**

**}**

Dia diabuscado **=** diaBuscado**(**diaEvento**);**

int eventoAOcupar **=** diabuscado**.**eventos**.**size**();**

**if** **(**eventoAOcupar **<=** 4**)** **{**

//Determina Horario Inicial y Final

String horarioInicial **=** formatNumber**(**ev**.**get**(**1**)** **/** 4**)** **+** ":" **+** formatNumber**((**ev**.**get**(**1**)** **%** 4**)** **\*** 15**);**

String horarioFinal **=** formatNumber**(**evSiguiente**.**get**(**1**)** **/** 4**)** **+** ":" **+** formatNumber**((**evSiguiente**.**get**(**1**)** **%** 4**)** **\*** 15**);**

Evento evento **=** **new** Evento**(**horarioInicial**,** horarioFinal**);**

diabuscado**.**eventos**.**add**(**evento**);**

//Obtengo el indice del mes del array

OptionalInt indexDias **=** indexDias**(**diaEvento**);**

dias**.**set**(**indexDias**.**getAsInt**(),** diabuscado**);**

System**.**out**.**println**(**"Feriados y Rangos Leidos correctamente de SD"**);**

**}** **else** **{**

System**.**out**.**println**(**"Todos los rangos estan ocupados para el dia " **+** diabuscado**.**name**);**

**}**

**}**

**}** **catch** **(**IOException e**)** **{**

e**.**printStackTrace**();**

System**.**out**.**println**(**"Error al leer el archivo en la tarjeta SD."**);**

**}**

System**.**out**.**println**(**"Feriados y Rangos Leidos correctamente de SD"**);**

**}**

/\*

Este Metodo guarda los feriados y los rangos en la SD:

Codifica los feriados a la posición de memoria correspondiente a la EEPROM del

Atmega328, luego el firmware del dispositivo se encarga de setear un 1 en

dichas posiciones de memoria indicando que corresponde un feriado para esas fechas.

Para los Rangos se utiliza el formato XX:YY, siendo XX la posición de memoria

de dentro de la EEPROM e YY el horario a alamacenenar en dicha posicion de memoria.

Los feriados y los rangos estan separados por un PIPE (|) para facilitar el parseo

y se detecta el fin de trama con un punto.

\*/

private static void guardarCalendarioEnSD**(**Scanner scanner**)** **{**

clearScreen**();**

List**<**String**>** lista **=** **new** ArrayList**<>();**

**for** **(**File sysDrive **:** File**.**listRoots**())** **{**

lista**.**add**(**sysDrive**.**toString**());**

**}**

System**.**out**.**println**(**"Seleccione la unidad donde se encuentra la SD (ingresa el índice):"**);**

**for** **(**int i **=** 0**;** i **<** lista**.**size**();** i**++)** **{**

System**.**out**.**println**(**i **+** ": " **+** lista**.**get**(**i**));**

**}**

int indiceSeleccionado **=** scanner**.**nextInt**();**

String elementoSeleccionado **=** lista**.**get**(**indiceSeleccionado**);**

/\*

GUARDA EN SD

\*/

String nombreArchivo **=** "feriados.txt"**;**

// Tarjeta SD accesible

File directorioSD **=** **new** File**(**elementoSeleccionado**);**

**if** **(!**directorioSD**.**exists**()** **||** **!**directorioSD**.**isDirectory**())** **{**

System**.**out**.**println**(**"La tarjeta SD no está disponible."**);**

**return;**

**}**

File archivo **=** **new** File**(**directorioSD**,** nombreArchivo**);**

**try** **{**

String fileToSd **=** ""**;**

**for** **(**Mes mes **:** meses**)** **{**

**if** **(!**mes**.**feriados**.**isEmpty**())** **{**

**for** **(**Integer feriado **:** mes**.**feriados**)** **{**

Integer posicionEnMemoria **=** 55 **+** feriado**;**

**for** **(**int i **=** 1**;** i **<** mes**.**indice**;** i**++)** **{**

posicionEnMemoria **=** posicionEnMemoria **+** decoMeses**.**get**(**i**);**

**}**

fileToSd **+=** posicionEnMemoria **+** ","**;**

**}**

**}**

**}**

fileToSd **+=** "|"**;**

//Dias

**for** **(**Dia dia **:** dias**)** **{**

**if** **(!**dia**.**eventos**.**isEmpty**())** **{**

int posicionInicial **=** decoDias**.**get**(**dia**.**indice**);**

**for** **(**int i **=** 0**;** i **<** dia**.**eventos**.**size**();** i**++)** **{**

fileToSd **+=** formatNumber**(**posicionInicial**)** **+** ":" **+** formatNumber**(**dia**.**eventos**.**get**(**i**).**valorCalculadoInicio**)** **+** ","**;**

posicionInicial **=** posicionInicial **+** 1**;**

fileToSd **+=** formatNumber**(**posicionInicial**)** **+** ":" **+** formatNumber**(**dia**.**eventos**.**get**(**i**).**valorCalculadoFinal**)** **+** ","**;**

posicionInicial **=** posicionInicial **+** 1**;**

**}**

**}**

**}**

fileToSd **+=** "."**;**

// Escribir el contenido en el archivo

FileWriter escritor **=** **new** FileWriter**(**archivo**);**

escritor**.**write**(**fileToSd**);**

escritor**.**close**();**

System**.**out**.**println**(**"Los feriados se guardaron correctamente en la tarjeta SD."**);**

**}** **catch** **(**IOException e**)** **{**

e**.**printStackTrace**();**

System**.**out**.**println**(**"Error al guardar el archivo en la tarjeta SD."**);**

**}**

**}**

//Metodo para obtener el Objeto Dia por un indice

private static Dia diaBuscado**(**int diaEvento**)** **{**

**return** dias**.**stream**()**

**.**filter**(**d **->** d**.**indice **==** diaEvento**)**

**.**collect**(**Collectors**.**toList**()).**get**(**0**);**

**}**

//Metodo para obtener indice en la variable dias

private static OptionalInt indexDias**(**int diaEvento**)** **{**

**return** IntStream**.**range**(**0**,** dias**.**size**())**

**.**filter**(**d **->** diaEvento **==** dias**.**get**(**d**).**indice**)**

**.**findFirst**();**

**}**

//Metodo para cargar feriados dentro de la variable meses

private static void cargarFeriadosForSdMethod**(**int mes**,** int dia**)** **{**

//Busco el mes en el array

Mes mesBuscado **=** meses**.**stream**()**

**.**filter**(**m **->** m**.**indice **==** mes**)**

**.**collect**(**Collectors**.**toList**()).**get**(**0**);**

//Cargo el feriado al array de dias en el mes

mesBuscado**.**feriados**.**add**(**dia**);**

//Obtengo el indice del mes del array

OptionalInt indexMeses **=** IntStream**.**range**(**0**,** meses**.**size**())**

**.**filter**(**i **->** mes **==** meses**.**get**(**i**).**indice**)**

**.**findFirst**();**

meses**.**set**(**indexMeses**.**getAsInt**(),** mesBuscado**);**

**}**

//Metodo para borrar pantalla

private static void clearScreen**()** **{**

System**.**out**.**print**(**"\033[H\033[2J"**);**

System**.**out**.**flush**();**

**}**

//Formatea numero en string para siempre manejar 2 decimales

private static String formatNumber**(**int valor**)** **{**

**if** **(**valor **<** 10**)** **{**

**return** "0" **+** valor**;**

**}** **else** **{**

**return** "" **+** valor**;**

**}**

**}**

**}**

**Clase Dia.java**

package proyecto**.**potencia**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**List**;**

public class Dia **{**

public Dia**(**int indice**,** String name**)** **{**

**this.**indice **=** indice**;**

**this.**name **=** name**;**

**}**

int indice**;**

String name**;**

List**<**Evento**>** eventos **=** **new** ArrayList**<>();**

**}**

**Clase Evento.java**

package proyecto**.**potencia**;**

public class Evento **{**

String horaInicio**;**

String horaFin**;**

int valorCalculadoInicio**;**

int valorCalculadoFinal**;**

public Evento**(**String horaInicio**,** String horaFin**)** **{**

**this.**horaInicio **=** horaInicio**;**

**this.**horaFin **=** horaFin**;**

**this.**valorCalculadoInicio **=** **(**Integer**.**parseInt**(**horaInicio**.**split**(**":"**)[**0**])** **\*** 4**)** **+** **(**Integer**.**parseInt**(**horaInicio**.**split**(**":"**)[**1**])** **/** 15**);**

**this.**valorCalculadoFinal **=** **(**Integer**.**parseInt**(**horaFin**.**split**(**":"**)[**0**])** **\*** 4**)** **+** **(**Integer**.**parseInt**(**horaFin**.**split**(**":"**)[**1**])** **/** 15**);**

**}**

**}**

**Clase Mes.java**

package proyecto**.**potencia**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**List**;**

public class Mes **{**

public Mes**(**int indice**,** String name**)** **{**

**this.**indice **=** indice**;**

**this.**name **=** name**;**

**}**

int indice**;**

String name**;**

List**<**Integer**>** feriados **=** **new** ArrayList**<>();**

**}**

# 6. Foto producto

Una bolsa de papel

Descripción generada automáticamente con confianza baja

**Ilustración 16: Vista superior del equipo**

Imagen que contiene tabla, cámara

Descripción generada automáticamente

**Ilustración 17: Vista Lateral del equipo**

# 7. Video de Funcionamiento

INCLUIR LINK