**Herramienta de Software de Visión por Computadora para Aplicaciones de Robótica de Enjambre en una Mesa de Prueba**

**Computer Vision Software Tool for Swarm Robotics Applications on a Testbed**

José Ignacio Ramírez Soto (ram17787@uvg.edu.gt)

Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica, Facultad de Ingeniería, Universidad Del Valle de Guatemala

**Resumen**

La robótica de enjambre es un área que ha encontrado amplio campo dentro de distintas aplicaciones en últimos años, sin embargo, aún tiene grandes retos que se deben abarcar. Dentro de estos retos está una implementación eficiente de algoritmos para su aplicación en visión por computadora, que es una herramienta que puede llegar a ser de gran utilidad en esta área. El objetivo principal de esta herramienta es permitir al usuario reconocer la pose de los agentes (o robots) en un área de trabajo y los obstáculos dentro de la misma. Todo esto es posible hacerlo en tiempo real. La herramienta cuenta con una interfaz gráfica que facilita el uso de las funciones de generación de identificadores, calibración de cámara, abstracción de datos y comunicación externa. Este trabajo aporta una herramienta de mucha utilidad para proyectos profesionales, trabajos de investigación o cualquier persona interesada en la robótica de enjambre.

**Palabras clave:** Visión por computadora, Robótica de enjambre, Pose.

**Abstract**

Swarm robotics is a field that has found its way into many different applications in recent years, however, it still has big challenges that must be addressed. Within these challenges, there’s an efficient implementation of algorithms to work together with computer vision, which is a tool that can be of great use in this field. The main objective of this tool is allowing the user to know the pose of the agents (or robots) in a working area and the obstacles that lie within it. All of this is possible to be done in real-time. The tool includes a graphical user interface which simplifies the use of code generation, camera calibration, data abstraction, and external communication. This work contributes with a useful tool to professional projects, research works, or for anyone interested in swarm robotics.

**Keywords:** Computer Vision, Swarm Robotics, Pose.

**Introducción**

En el artículo redactado por Lizarazo y Ramos (2016), se detalla el proceso de diseño tanto de los robots en sí, como del programa de comunicación, detección y abstracción de su posición y orientación (combinación a la que se le conoce como “pose”). Aquí, utilizan *OpenCV* para el procesamiento de imágenes en Python.

Un ejemplo a mayor escala es la plataforma *HeRo* (Rezeck et al., 2017). En esta se utiliza un sistema de detección. En este se utiliza un tipo de código de barras conocidas como etiquetas AR. Estas son detectadas con la librería Alvar que cuenta con umbrales adaptables para tratar con distintas condiciones de luz y un método de detección que no disminuye significativamente la velocidad conforme se aumenta el número de agentes.

Otro ejemplo similar es *ARDebug* (Millard et al., 2018). En este se utilizan etiquetas ArUco en conjunto con la librería de detección de etiquetas fiducial ArUco (Garrido-Jurado et al., 2014). Esta librería fue diseñada principalmente para aplicaciones de realidad aumentada solucionando problemas de ese ámbito, pero es lo suficientemente eficiente como para detectar hasta 50 etiquetas en tiempo real por lo que es una excelente alternativa para aplicaciones de robótica de enjambre.

Sin embargo, esta es una de las pocas aplicaciones en robótica de enjambre de la visión por computadora resultando en un campo poco explorado. Existen otras aplicaciones del procesamiento de imágenes en la robótica. Por ejemplo, Guangrui y Geng desarrollaron un Sistema de detección de zonas de carga y descarga para vehículos en bodegas.

Por esta razón, Rodas (2019) comenzó el desarrollo de una herramienta para abstracción de pose con el objetivo de ser utilizado en aplicaciones de robótica de enjambre. En esta se fabricó una mesa de prueba con la que se estuvo trabajando para el desarrollo de los algoritmos. También se estableció un modelo estándar de identificador que se colocará sobre cada robot (o como se le denomina en la robótica de enjambre, agente). Estos darán un número único a cada agente con lo que se diferenciará a cada uno. Estos identificadores se diseñaron para ser fáciles en su creación y en el procesamiento para ser identificados en los programas.

La herramienta contaba con: un generador de códigos, calibrador de la cámara, y abstracción de poses. El primero se utiliza para generar las imágenes de los códigos que luego pueden ser impresas y pegadas sobre los agentes. El segundo identifica figuras circulares que se colocarán en donde se desean las esquinas de la mesa. Este recortará y rotará la imagen de manera que únicamente el área deseada aparezca en la imagen. Y el último obtiene las coordenadas y orientación relativas a un marco de referencia global. Todo esto se realizó con ayuda de *OpenCV* y se creó una interfaz gráfica con la librería QT5 en el lenguaje C++.

Este proyecto fue continuado por Guerra (2020). En esta segunda fase se adaptaron los mismos tres programas descritos con anterioridad al lenguaje Python. En este caso, se mantuvo el uso de *OpenCV* para el procesamiento de imágenes y QT5 para la interfaz gráfica. Otro objetivo era implementar programación en multihilos para intentar mejorar el rendimiento de los programas. Este no resultó siendo el caso y se concluyó que, ya que los hilos se ejecutaban de igual manera en un mismo núcleo del procesador, el tiempo que el procesador tarda en cambiar, crear y destruir hilos (también conocido como *overhead*) era mayor al tiempo ahorrado en procesamiento.

Por último, también se comenzó el desarrollo de la adaptación a Matlab, ya que es un lenguaje muy utilizado para procesamiento y análisis de datos. Sin embargo, únicamente se terminó el código de creación de identificadores y se comenzó a hacer pruebas de la calibración. Todo esto fue desarrollado y validado en una mesa de en área de trabajo. Por lo que una de las recomendaciones es validar en la mesa de prueba de la UVG o similar en dimensiones.

Luego de esta segunda fase, aún se tenían las siguientes limitaciones. No se llegó a un producto final en Matlab. Únicamente se detectan los identificadores, pero no otros objetos que puedan obstruir su movimiento. El desarrollo de Python no se validó en un área de mayor tamaño. La calibración presenta errores al tener un marcador circular afuera del área de trabajo; esta puede ser identificada como una esquina y resultar en una calibración errónea.

Tomando en consideración todo lo anterior, el objetivo de este trabajo es agregar funcionalidad y variedad a la herramienta de visión por computadora desarrollada en fases previas y validar el sistema completo en aplicaciones simples de robótica de enjambre bajo condiciones controladas.

**Materiales y métodos**

Se comenzó a desarrollar un procesamiento de imágenes similar en el lenguaje Matlab. Teniendo estos funcionales, se creó una interfaz gráfica para facilitar y agilizar el uso. Esta junto con la herramienta existente en Python se validaron en la mesa de prueba UVG. Y, por último, se realizaron pruebas en conjunto con otros trabajos en desarrollo para validar la aplicabilidad en robótica de enjambre.

**Materiales**

Para la verificación final de todas las pruebas, se utilizó una mesa de prueba ubicada en la UVG mostrada en la figura 1. Esta tiene m de dimensiones en el área de trabajo. A los costados, tiene tubos de acero de altura ajustable hasta 1.2 m. Sobre los tubos se tiene una tabla de madera en la cual se encuentra la cámara modelo HD Webcam C270 y una lámpara led. Ya que no se contaba con esta mesa en todo momento, para el desarrollo de los algoritmos, se también se utilizó una versión más simple. Esta únicamente consistía en un soporte hecho de madera, para la cámara, colocado en la pared y pliegos de papel, una tabla de melamina, o alguna superficie sin texturas sobre el suelo.

![Diagrama

Descripción generada automáticamente](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4REaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAcAAAISodpAAQAAAABAAAIZpydAAEAAAA0AAAQ3uocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEpvc8OpIElnbmFjaW8gUmFtw61yZXogU290bwAABZADAAIAAAAUAAAQtJAEAAIAAAAUAAAQyJKRAAIAAAADMzgAAJKSAAIAAAADMzgAAOocAAcAAAgMAAAIqAAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADIwMjE6MDQ6MTUgMDI6NDg6NTkAMjAyMTowNDoxNSAwMjo0ODo1OQAAAEoAbwBzAOkAIABJAGcAbgBhAGMAaQBvACAAUgBhAG0A7QByAGUAegAgAFMAbwB0AG8AAAD/4QsuaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyIvPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvIj48eG1wOkNyZWF0ZURhdGU+MjAyMS0wNC0xNVQwMjo0ODo1OS4zODQ8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIj48ZGM6Y3JlYXRvcj48cmRmOlNlcSB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6bGk+Sm9zw6kgSWduYWNpbyBSYW3DrXJleiBTb3RvPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAs8CHgMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpDaKNopaKAE2ijaKWigBuyjZTqKAG7KNn0p2a+OPiz8TtV1z4jXsmhatd2lhZt9mg+zTsgcKcFuDzk5P0xSegH2Lso2V8Kw/FHxzaR7YPFWqBTxhrlm/nnFbWh/Hzx9oswMmr/ANpRZ5ivoxID+Iw360XA+0NlGyvAvDv7VGmT7YvFGiT2jdDPZOJU+u04I/M16t4e+JXhDxSFGja9aSyt0hkby5PptbBP4UwOn2UbKXOehpaAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBuyjZTqKAG7KNlOooAbso2U6igBNoo2ilooAKKKKACiiigAooooA4T4yeK38I/DW/urYst3cj7LAw/hZwct+Az+OK+JmO3O7rX394m8M6Z4t0GbSNbhM1rNgkA4ZSOQQexrxPxB+yxZTq8nhzXpoG/hhvIw6/99Lgj8jUtMZ80u6lfemZzWl4m8OXvhXxFeaNqWw3NpJ5bmM5Un1B9Ky0piHUoYqwKkgjuKMUUCOu8O/FPxn4X2rpOv3QhXpBO3nR49Nr5A/DFereHf2qLyLZH4p0OKdehnsXKN9djZB/MV894pcZoA+2PDvxw8B+I9iRa1HYzt/yxvx5JB9Nx+U/nXfQzRXEKywSJLGwyro2QR7EV+d0NnNcK3kDeyjJQH5seoHf8K1/D3jPxF4Vm36Fq11Z85KJIdjfVTwaYH37RXyt4f8A2nvElkVTXtPtNTiHBZMwyfmMj9K9U8O/tEeCNa2R309xpE54K3ceUz7Ouf1xQM9Voqnp2radq9uJ9Kvre8iPR4JQ4/SrlABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB8bfH6yEPxf1YgYEyxSD8Y1/wrZ/Zy8PaF4mv9e0zxDpdvfr5MckZlX5k5YHBHI6jpS/tMW32f4n28uOLjTo3z7hmX/2Wm/sxXXk/E68gJ4n05xj1IdD/AI0gPSNf/Zl8Kajuk0W5utKkPRd3mxj8Dz+teYa/+zV4w0ve+lPa6tEOQIn2Pj/db+hr64oosM/P3WfCmv8Ah6Ux61pF5ZEHGZoWUH6HvWRnHWv0Vnt4bmForiKOWNhgo6hgfwNfF/jq4stV8VX6w6PFbGOdkK28QRU2nHb6Utgtc8+RsMCDtI6EU/yjIc53E1vHQ7KYYjkMb9wajbw3dopaBhIBzxVJicTDaFl6qRSbSOlaWy5g4lhJx14o3W8v+sTafUcVaRBBp2ralo9yJ9LvbizlHR4JSh/SvTPDv7Q/jbRtkd9cQ6vAvG26j+b/AL6XB/PNecGxRx+6kH41A9pIn8OfpT5Quz6j8O/tMeGtQCx6/ZXOly93X97H+nP6V6fofjDw94kjDaHrFneEjOyOUbx9V6j8q+CSpHBFSQzSwSB4JHjdTkMhwRU8o+Y/Qyivizw/8ZfHPh4ItvrMl3Av/LG8Hmrj0yeR+Br1Hw7+1BA7JF4p0Rov71xYvuH/AHw3+NKzHdH0FRXJeH/ih4O8TBRpmt24lb/ljOfKf6Ybr+FdYCGUFSCD0IpDFooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5Xxr4kutHl0fS9L2rf61efZopXXcIUA3O+O5AHAPGTVt9O1Wxv7O6TXbmeyiLG9huI4iHXYcMCqAgg44HFV/GnhefxBFp15pc8dvquk3QurR5gTGxxhkbHIDA4yOlW4pPEd9ZypcW1lpc3lMEeO4NxmQqQDyi4APPcn0pa8r7/APA/4cfVdv8Ag/8ADGTpnjS2kvdRuLu5vDA1oL+0t2tVAFsBguu3LMSQSQ2CMjgU+H4laNcSQxw2mqPJc2v2q1QWL5uE4zs9SMjPQc9a5zT/AAT4oS++13ltpYuJtFl0+4lF7I7yysciVmMeTn07D1rX0vwnrdjqPhi4lWwZdI0uSznC3L5Z2CYK/u+nyd8daf8AX/pX+S+8X9fl/wAE1D8QNHfS7W/tUurqO5tXuwkUXzJEmN7MGIxjOMdfQGg+PtJbVILC1gv7uae3jukMFqzL5TnAcnsPWuRh8B+K4NBs9Il/su6tFs7mCWBrqQRpLIxKS48v58A4wQMHkGtfwv4U1/S9WtLu8GnxLDocWnExXDuyyIzENgxgEfN69qP6/P8AyX3/AHH9fl/wTf0nxlpusajDZ2yXEb3MDXNs80e1Z41baWXnI5I4YA89K36838NeEfEtl4q0nVtYh015be0mtr25S6d5Z2ZlIk5Qddv3c4HY9q9Ip9F/XUXVhRRRSGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHzR+1Pa7df0G7A/1ltJET/utn/wBmri/2fbr7N8ZtNTOPPjmj/wDHC3/stenftUWu/RfD12B/q7iWMn/eVT/7LXjfwkuPsfxd8Py5xm6Cf99KV/rSA+4aKKKYBXzB8RvDdzo/jbUvPg8qC8nae3kA4dWOevtnpX0/WN4o8M2XirRpLC+XB6xSgfNE3Yj/AAqZK6GnY+QLzSCZPNiYkk81o6FouoX15HBZxvLIxwqqOTXX6h4J1rRLuWLUNNnaKNiBPFGXjcf3gR0/Guz+FGjRya298uNtupHPXcfb6ZqY7lvY87vvDV/akx6npsiHuJYzWHdeGLCf/lk0Te3/ANevsCSGKZSsqK6nqGGRWDqXgXw/qeTNYJG5/ji+U/pW3MZnyNc+C2GTazAnsCcVl3Gh6nacmJmA7gZFfUOofBu1fJ02/eM9llXP6iuV1D4X6/Y5MUC3K+sTZ/Q07isfPbEqcXEBz7io/s1tL9xth969a1Dw20chj1LTzGx674yprm7jwbbGRtshUH7uB0qri5Thjp0w+4Q30NV3idDh1IPuK6648I3sOTbShxnsazp7K/tTi4t2I+lPRisYQJUgqcH2rp9A+Ivi7wwynSdaukjH/LJ38yP/AL5bIrNKWknEsZjb2GKjbTu9vKD7GjlTEe0eHv2ndQh2x+JdIhuV7zWx8tvyOQf0r0/QPjl4G14qh1T+z5m/5Z3qeWP++vu/rXyC9nKo+eM4/vKKiNnIFLqNw9R2qXAfMfoDa3VvewLPZ3EVxE33ZInDKfxFTV8D6R4k1vw/OJdH1O6snH/PKQrn6ivTPDv7SPirTdkesw22qwjqzr5cn/fQ4/MVPKyro+rKK8l8PftF+ENX2x6oLjSJj185d8f/AH0vP6V6Vpeu6VrcAm0jUba8QjOYZA2Pw7VIzQooooAKKKKACiiigAooooAKKKKACiimSypDE0krBEUZZj0AoDcJZUgiaWZwiIMsx6AVFZXLXcHnGFokY/Ju6svrjt9KpRRyavMtxcqyWaHMMLDmQ/3mH8hWpUpt6m04xguV7/kLRRRVGIUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUYoxQAUUYoxQB43+03a+b8M7a4xkwagn4ZVhXzZ4Lufsvj7Q5ycBL6Ek+28V9W/H+0+1fBvVCBkwvFIPwcD+tfH2nS/ZtVtJ+nlzI35EGkB+hSnKg+1LUNpIJrKGT+/GrfpU2KYBRRijFABVWbTLGeTzJbSIyf89AoDD8RzVrFGKAKi2Plf8AHvcTIP7rNvH/AI9k08fakHIjl+mVP9asYoxQBXFzt/1sMif8B3fyzUiypJwjqx9AakxTWiR/vorfUUART2lvdJsuYY5V9HUGuf1D4f8Ah3UMlrEQuf4oTt/TpXReQB9x3T6HP86Nsq9GVvqMUAeZ6j8HozubTL8j0SVf6j/CuV1H4a6/ZA/6ILlB3iIb9Ote77yPvIw9xzQHU9+fSqUmB8sah4XRXKX2nNE/uhU1g3Xg23bJt5WiPoa+wZ7O3u0K3MEcqns6g1z2ofDzw/qGSbT7Ox/ihOP06VXMB8mTeHNUtOYSJl9BzVMvcWr/AOkWrKe5C19Laj8HxydN1D6LMv8AUf4VyuqfDXXbVTvsluUHePD8fTrVcxNjwuZbO6ZmwFc+oxVVtNDD92c/Q5r0vUfB8IZkurB4HHX5SKwbnwfsJNpcMh9Gp3QrHET6bcQDLRNt/vAcUlpf3+mXCzWF1NbSqeGicqR+VdVJp2s2I+Ueao7rWdd+RcHF/bvBJ/z0Rf5ihxTFsdT4e+PnjbQykdxepqcC9UvE3E/8CHzfrXqfh39pvQ73bH4h02fT3PBkhPmJ+XBH618+nSLB0H2e7aZvRl2/1NQSaUYwSbSVl/vK4P8ASpdMfMfbOheO/DHiRV/sbW7S4dukXmBZP++Tg/pXQV+fYgMb7oJijDoDlSK6zQPit438LMi2msTywL0guT5qEfj0/CocWh8yPtiivnjQf2olwkfiTQuf4prOT/2Rv8a9P8PfGDwR4j2raa1FbzN/yxu/3TfrwfwNSUdxRUX2m38oS+fH5ZGQ+8YI+tVrfWtKu7o2trqVnNcAZMUc6swH0BzQBeooxXPeMNYfSdPgMB/eyTA49VXk/wBKmUlFXZtQoyr1FTjuzfkkSKNpJWCIoyzE8AVmRo+sTLPcKUskOYomGDKf7ze3oKgsp4/EhW43qbFDlIgwzIfVh2HoK3MAdKS97XoXJOh7v2vy/wCD+QUUYoxVnMFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFGKMUAFFFFABRRRQAUUUUAFFFFABRRRQBx/wAWLT7b8J/EUWM4smk/75w39K+HRkFD6GvvzxXbfbfB2sW3XzbGZMfVDXwLJwD7Gl1A+9/CN19t8GaPc5yZbKJj9dgrZrjvhNdfbPhR4flzn/RFQ/8AAeP6V2NMAooooAKKKKACiiigAooooAKKKKACkIB6gGlooAZ5ajpx9DRtYdGB+tJPcQ20ZkuJUiQdWdgB+tUrDXtP1S8kt9Pm88xLud1U7R+NWqc5RcktEOzauXssOq/kaNw78fWn0VAitcWdteJtubeKZT2dAa53UPh34ev8n7J5DH+KFsfp0rqdq9uPpRg9m/OndoDy3Ufg6CC2mah9FmX+orj9W+GGuW6nzdOW7Qd4sPn8OtfQXPp+VJkY54+tPmYHyTqXgyKORlns5baQf7JGPzrGl8OXdvn7HeNj+69fZNxZ2t2hS5t45lPZ1BrnNQ+HXh3UMn7H9nY/xQtt/TpVqYrHyTd2OoqMXdosyj+JRmqCpbKcMskJz91hkV9Oah8GlOW03UeeyzJj9R/hXHav8KdagDGXT1uk9YsN/wDXquZMXKeLyQGWPZbw2cin0XDfnmqs2kTOm+OBonXqhHX3B/pXd6h4IWCUiSCe0kB9CKy5fD2owf8AHtd7x2V6ejJszi5Z76ONYZpZti/dRmOB9BTtK1e/0i+F3pt1Jbzg8Ohrprlb1EK6lp6zp/eAyfzrnr2OATZtYjGh7N1FQ46Bc9x+F/x8mjuZbP4gaiDbbB5Fx5GWDejFe34V6nJLpfj3UYrnSPENlLaxx7VSFw0mTySVPT8R2r43jthJbyFlcOMFSOh56U1GurGUPG0sMgOQykqfzrGdPmVmdNDEToT56e591aX4U0zSpRNDG0k4/wCWsjEn8un6VtV8aeG/jV418ObUTU2voF/5Y3o8wfgeo/OvUvD/AO07p8+2PxHo0ts3ea0fev8A3ycEfnQoqKskKpWqVpc1SV2e80Vymg/E3wh4jCjTdctvNbpDM3lv9MNiuqDBlBUgg9CKZmLRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADJoxLC8b8q6lSPYivz31CJra9uYG6xysp+oNfoXXwT46tfsPj/AF+1xjytQmX8nNID6r/Z/u/tXwd0xc5MDyxn2+cn+tel14x+zFd+d8M7qAnmDUHAHsVU/wBTXs9MAooooAKKKKACiiigAooooAKKKKACiis27u5bi4NjpxxKP9dN1EI/q3tVxi5OyGlc5nxP4fh1K6lbTXnku4VLyl5C8a4GduDn5j6Cr/gGx8jRHu3Xa93IW5HRRwP6/nXRWlnFZW4hgXA6kk5LHuSe5qSGFIIVihXaiDCqOwrsqYyUqHsOhrKq3DkH0UUVwGIUUUUAFFFFACbQe1G3HQ0tFADcH60fUU6igCtcWVrdxlLqCKZT/DIoYVz2ofDvw7f5P2PyGP8AFC239OldTjPUUm30yKd2gPLdQ+DcbtmwvyU7pKuM/iK8X8Y+BrnSdQmt5LLymU8NuLbvcE19dYP1qnfaTY6kuL+0jnwMDcucVan3FY+GJLK6tpCAXUg+tOF1eAYlAkHowr691L4WeGtSJP2bySf7mKy1+B/hgSb3NwccgAr/AIU+ZC5T5ZaCJ4Q89hJGGGQ8fSqz6dbSLm3uwD/dkXBr6e1P4JxOGOl6hjjiOZP6j/CuG1r4N6zb7i2mpcr/AHoDk/40rphZniL2F1DyqFh2KHP8q3NC+IXi3wwVGla1eQov/LFnLp/3y2RWxf8Agq5sJCAtxaOP4XU8VlT6XqEQxJHFdqPUc0WQao9J8P8A7TusW22PxFpVveqPvSwHyn/Lkfyr0/QPj54H1vak99JpczfwXke0f99DI/PFfKVzaQ8ia1ltm9V5FU2sQ/8AqJ0f2Pymp5Quz77sdRstTtxPpt5Bdwt0kgkDqfxFWa+ALDUdY0K5E+mXl1Zyr0eCQr+or0Pw/wDtEeNdG2x380GrQjgi6TD4/wB5cH881Ooz68orxTw/+014av8AbHr1hdaXIeC6Hzox+IwR+Rr0/Q/GnhzxJGG0TWbS7JGdiSgOPqp5/SgZuUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV8SfGa1+yfF/wAQIBjfceb/AN9AN/Wvtuvj79om1+zfGC7kAwLi3hkH/fAX+lAHoH7Kt3u0fX7Qn/VzxyY+qkf+y19A18x/sr3ezxJr1pn/AFltHJj/AHWI/wDZq+nKACiiigAooooAbJIkUbSSsqIoyzMcAD61W0/VdP1aAz6VfW17EDgyW0yyLn0ypNWmxtP0rhfg+Ui+GFmzFUVZrgkngAec9C6gdpJe20V3Fay3EKXEwLRwtIA7gdSB1OMip682vPEd7B43s7iKS3uo59VOlM8UIMcUWxmC7zhvMDLk7cr26iqj+LfFa25vVv7IpH4ibSTC1pw6GQoGJ3ZyODgdfWiOtrf1t/mgel/67/5M9Uory698XeJtPj1Gy+3Ws11Za5a2QuGtcB4pghwVB6jeRmrB8U+INP1jVNB1HUbWSa3ubXyLxbfbJLHMGJjVBkGQbTgnjHJ6ULVX/rp/mD0/r1/yO3vLya4uDY6ccSj/AF03UQj+re1W7O0isrcQwLgDkknJY9yT3NeaR+MPEVxpWniCeC2upvEEulTSTWyszIvmYYhTt3YQZxxmrWseKdd0W/WP+0Yr8W1/aWcwht1CYlKhjITyHy2QEJAGMjmtOa8VGO2nzvb/ADQ27K3b9L/5M9IooHSisxBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAlJtH0p1FADcH1oyfSnUUAVbmztbyPZd28Uy+kiAj9a5rUvhp4a1IEmxFu5/igO3H4dK6+k2j6fSgDyDVfgdG4Y6bqCn0SdP6j/CuA1z4K6zbbmOmeeg6PbsD+g5r6d2+h/Ojn0p3A+J7/wAGX1hIy/v7dh/DKhrBvNKvY8+ZbpL7pwa+7bqws71Cl5axTL3EiA1yWr/CfwtqwY/ZGtHP8Vu+39DkUXCx8TyxvGx3RunP8QpiTvC4aN2RgcgqcEV9YeDfhsmmePdQj8iLUdBSEws15ErbnODgcYOCDXmH7R/hLQ/DGuaUdA0+GxF1C7zJCMKxDYBx2qQH/Az4ieJpfiHpmhXusXV3p1yHQwzuZNuEJGCckcjtX1fXnvwX0mxtvhXoFylnAty9sHaYRgOSe5PWvQqEAUUUUwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAr5X/AGobXyvH2m3OP9dYAZ+jsK+qK+cf2q7XE/h27A6rNGT9CpH8zQBy/wCzPd+T8UpoCcCexkH4hlP+NfW1fFvwGu/svxm0jJx5okjPvlD/AIV9pUAFFFFABRRRQA2SMSxsjFgGGCVYqfzHIrnbLwFoenaWunWS30Vmrl/IGo3GwktuORv5BPUHg10lFAHLP8OvDzXxuvJulc3v29VS8lVEm5yyqGwM5OcdanPgTQTam3NvceW19/aBH2yb/X7t2/O715x0roq5CTWriy+Jt9bXl+y6XDoyXZjcALE3mOGbIGei980LRpf1or/p+QPVN/1rp+pR8Y+B0ubNm0OzuJLq91S2u71hdsCVjYZI3N8p2jA246Vs3XgXRb+2K3cVw0zXKXRuftLrOJVGFPmA5GBxgHGKVPHOii4u4LuSWzktbYXbLcJgvCTgOuCc88Y65wMVm6R4gvdQ+KV7YObyGyTSop0tbmNV2u0jDcNuc5AHU8e1C/l/rRf8D7wff+tX/wAEvQ/D3w9AEEcFyBHfG/QfbZvlnOct973PHTk0mofDzw/qV9d3dzDciS7mjuJVju5UUyx42yBQwAb5RyOa6iij+v6+5AIq7FCjOAMDJzS0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUx4ldSGH3hg44p9FADIokhjEcShEUYAA6V8yftUvnxRoielmx/Nz/hX09Xy3+1E27xxpi9k08frI9AHvHwtj8r4V+HV6f6BEfzUV1lc78P4/J+HPh+P+7p0I/8cFdFQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXhf7U1rv8G6Nd4/1V6Y8/7yE/8Aste6V5L+0lafaPhK0mM/Z72KT6cMv/s1AHzd8L7r7F8VvDs2cD7dGhP+8dv9a+66/Pvw5dfYvFek3WceTeRPn6ODX6BJzGp9QKBDqKKKBhRRRQAUUUUAFcR4i8G6hr3iPV5vMghsr/Rf7OWQSEyJJuZtxXbjHzDv2rt6KVr/ANd1Yadv6+Z5pL4M8Q614Pu9N1Cy0XSb5YkWG4ssuLiRHVwzfKCqkoPl56n0rX0HSvEb+OW13XLOxtY30uOzaO3u2lIdZGbIyg4O78Peu0oqr63/AK2sTbS39f1oFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFfK/7TL7vH1uv92wjH5u1fVFfKH7Rz+Z8Syn921hX+Z/rQB9MeE4/K8HaRHj7tnEP/HRWvVLR4/K0Oyj/ALsCD/x0VdoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArz/45Wv2v4M68oGWjSOQfhKuf0zXoFcz8R7X7b8M/EMGM50+Vv++VLf0oA+DkYpIjKcFSCDX6GaXci70i0uB0lhRx+IBr88mGBX3r8Pbv7b8N/D9xnJk0+En67BQB0dFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXyT+0Cxm+Lc0Y5OyBf0H+NfW1fJPxl/0j45yxnndc2qfotAH1hajZZwqO0aj9KmpkXEKD/ZH8qfQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVHW7b7X4f1C2PImtZI/zQir1Iyh1KsOCMGgD87bhdksin+FiK+1vgndfa/g7oLZz5cBj/AO+WI/pXxr4gtzZ+ItStmGDDcyIfwYivrD9nC7+0fCG2jzk29zLH/wCPbv8A2agD1eiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAr5J+JP+lftECId9Ttl/LZX1tXyR4nP2n9pyNeudbhH5MtID61QYRfpTqQUtMAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD4R+Jtp9j+J/iKHGMX8px9WJ/rXvn7LV35ngXVLbPMN9nH1Rf8K8e+Otr9l+M2tgDAkaOQe+6NSf1zXpP7Kd0PL8RWhPOYZAP++gf6UgPoqiiimAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFUdR01dQaEyXN1CkJLFbed4t5IwMlSCQPSkwL1FeV+E9UmsPhjH4t1O91K+lsXuHmR72RhKgkZcFWJBwMY+la/jXxMJtB8RaVCk9tcwaKb+O4ilKnDBgORgggqaJOy/r1Gld2/rex3tFcFY/ES30q3jtvEdhc6fs037ZDPI6uLlFwGxg8Nkjg+o5q/d+PoNJmnh1uwks5VsTfQhXEgljBAYZwMMCRkdADnJqmrO39f1oTF3V/6/rU66isrQtYfWbaSZreOONWAjlhnEscylQdysAMjnH1BrjPiB4lmvPDd8NGim8mx1K3tpryK4MZEnnIGAUfeUZ2nJ69j1pdbD6XPSKK4+4+INjbahFbRwNcQ/bBYySROC6SZxny8Z2BuC2ePTHNbeuay2kRQGO0a4adym4vsiiAUndI+DtHGM46kUdLh1satFcnp3ju31nTdOl0q0aa71CKWWO2aQKFWM7XJYZ43EAYHOagu/iNbW1gJf7OmS6S0F3PZXDCKZFJI2gHO5/lOBxn15o2A7OiuPj8frc6w9naaRcvDFaw3k1zLIsaxwybjuIPORtPFTWfjiCe70lbiykt7TWgTYXJcHeQNwVx/CSvI5P4U7CudVRWRq+uLY39ppltD9pv71XaKHzNgCJjczNg4HIHQ8muI8F+MP7P8Kxw3paXULi8vmjiu7rBCRzMMGQ56ZAHWpuM9OorM8Pa3D4i0G21S3hmgSdc+VOu10IOCCPqK06pqzsAUUUUgCiiigAr5HkIuv2oo+4/ts/8Ajp/+tX1xXyH4aTzv2m4Qfm/4m9w35bzSA+vKKKKYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHyP+0pafZ/iuJsY+0WUUn1xlf/Za1P2W7rZ431e1z/rbEPj/AHXA/wDZql/aotdvi3RLvH+ssTHn/dcn/wBmrn/2cLv7P8XI484+0Wcsf16N/wCy0hH1/RRRTGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTZA5jYRMquR8pZcgH6U6ijcDidN8ATWnwxvPCN3qcc32hJlW5jtym3zGLfdLHOCfWq7eAdW1CbUZdX1i1Zr/AEcaY4t7RlCYLYcZc/3uh/Su9ZgqlmOABkk9qwvEXieLSvB+pa3pZt9R+wwtIUScbTtGSMjPPtSdtW/66fqNXbSX9ap/oYmofDs+IbAQeIryN2h09rG3NrEUCbipMnJOTlF47Y96st4S1S5uLfULzUbV9SsrJrS2kW2Pl/Myl3dS3OQoGARjmte18VaTPfNYPewJeR28c80ZkA8sPnAPPXg/pWl9ttdxH2mHITzCPMH3f73096p3vr/W/wDmyY2tp/W3/AMPwp4WHhuXUZEaGNb6YS/ZbZCkMRCgHapJwSRk1zuofDnV5ItY0/TNatodL1O+W/8AKntGeSGTzFdgGDgFSVzyM813LaxpqwtK2oWojQgM5nXCkjIyc9xVpHWRFeNgysMhlOQRS63/AK/rRD6WON0zwhrWka5dNZa3CNHvLk3ctu1rmVJCcuEfdwrHnkEjJx61peJ9Av8AWbvSbnTb+K3On3BleK4hMscwKleQGHIzkHNbCaja3CzizureaSEHeqyg7D/tY6VHY6gstnAbue0Fy8XmOsEwZcdypPJX3o6egdThrH4cazpFrpU2la7bR6npjzoskloWimglfcUdQ4OQcHIIrRuvB+vRa9DrOka9Cl5LbLb34urTzI5gGJDqoYbSNxAGSMYrrJdUsIIUmmvbeOKQ4R3lUKx9jnmq8HiHS7nWLrS4LyJ7u0RXmQOPkDZx364GfxFAGPB4PmHiLUr68v0ubXUNOjsZITCQ+F3ZbfuxzvPaq+leB7i2h0O01O/jurXQXL2m2Eq74Uom85I4UnoOTzV/xJ4sj0jQBqemfZ9RQXUNu+ycYXfIqZyM5I3dOKl8Q+JU0rwfqWt6aIL82MLSGNZhtJUcgkZ59qL2V/67/qFru3f/AIb9Ct4k8MX+oeINL13Q9Qis9QsFkhIuITLHNE+NykBlIOVBBBrnrX4ca3piWd3pmu2qapa3Fy++SzLQyxTyb2Rl354OCCCK7yPU7YQWrXM8MElyoMcbyAFiR0GetXKLWFfmVyrp1vcWthHFeXP2mcDMkoTaGY8nA7D0FWqKKBhRRRQAUUUUAFfJHw+/0r9pSF+v+m3L/o1fWzNtUn0Ga+S/g6vn/tAxyH+E3L/zH9aQH1rRRRTAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAMUYoxRigAxRijFGKADFGKMUYoA+e/wBqu0zY+HbsD7sk8bH8EI/rXlXwRuhZ/GbQWJwHkeM++6Nh/PFe3ftQWvm/DuwuMf6m/Az/ALyN/hXzr8PLr7H8TPD0+cY1CEZ+rAf1pCPvXFGKQcilxTGGKMUYoxQAYoxRijFABijFGKMUAGKMUYoxQAYoxRijFABijFGKMUAGKMUYoxQAYoxRijFAGD43cxeB9WZdPfUT9nYfZU3ZkB4I+Xn3wOeK8zmW4uNM8epBbajcC+0uA23/ABL5I/NbyypCrt9cDB54/GvasUmKVtyk7NPseWvZvJ4m16IWFx5mo+G4o7ZjbOAzhZQylsYU8rwcHkVkRyHUE0tV03UHWHwrc2ku+wlGJsJhMFeTkH29DXtOKMY7UNXv5/8AB/zJjpbyt+n+R5NDa22l6D4NjTRJ7d5rZlnuo7F2khk8oLtaPb95uRucEDB9c10Xw+aeH4Q2aT2ly9xb2jo1tKjRyOyg/LyAeemRXb4oxVS97m8wj7tvI8c8Lkz+IbZ2sryG1m8OPA0UljJHFC6tnyssuTgd2Jz2NSeH2On2Pw61Oa3uZLRNLmgkeCB5drOsZQEKCRnaeTxXqWqaZ/adukX2q4tgrZJgfaXGCCp9Qc1NZWNvp1hBZ2cSxW9ugjjQdFUDAFH9f+lf/JB/X5f5Hi9ha3aeC7B7dLu31KC2ulNjeafI8N3G0rExHjKscDDA9+9at+0lpqfi+W58P3M8t5pVq8dqkMhV8IQ671HYnkA5OOK9awPQUYpdLD63/rp/keG39tcX6eJVjtNQuoLhtMuF3afJGHRHXzNq7R0A6fewO9a2t6Jix8a3fh6wmj0690iOBIILZx59wA2WWMDJ4KjOOfwr1zHtRjHSjo1/W1gi7Nf15nkGoRC41q/03xHba0bPUra2NgbS1Zlfai5jLbSY2Dgnkgc5r1yBNlvGpz8qgfMcnp60/FLinclK1gxRijFGKQwxRijFGKADFGKMUYoAiumCWcznjbGx/SvlX4CqLj43ySddtpO4P/AlH9a+otYby9Dvn/u20h/8dNfMf7OSeZ8WL2T+5YSfq60uoH1TijFGKMUwDFGKMUYoAMUYoxRigAxRijFGKADFGKMUYoAMUYoxRigAxRijFGKACiiigAooooAKKKKACiiigAooooAKKKKAPLv2h7X7T8Hb58ZME8Mv/j23/wBmr5D0ec2mv2FwDzDcxuPwYGvtf4xWv2z4Q+IY8ZK23mD/AICwb+lfDynbIrDgg5FID9E4XElvG4PDKD+lSVneHrkXnhnTbkHImtY3B+qg1o0wCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAI5vM2ZhI3A5we/tRDMsy5HBHDKeoNSVBNC27zYOJB1HZh6God07ozldPmRPRUcMyzLkZBHDKeoNSVSaaui001dBRRRTGZfiaXyfCWryA4KWMzflGa+dP2aE3/ELV5D/DZfzcf4V9BeOJPL8Aa83/AFDpx/5DIrwX9lyPd4o8Qy/3baIfmzf4UAfS9FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGF42tftngLXbfGfM0+df/HDXwKRgke9fojdwC6sp4H+7LGyH6EYr88rqMxXMsbjDK5BHpzSA+6PhjdfbPhd4emznNhEv5KB/Surrzv4EXX2r4NaKScmMSRn8JGA/TFeiULYb3CiiimIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAgmhbd5sHEg6jsw9DT4ZVmXIyCOGU9QakqCaFt/mw8SDqOzD0NQ04u6M2nF80Seio4ZhMuRkEcMp6g09WDD5eecVSaexakmro5v4iv5fw219v+nGUf+O4rxb9leP/AE/xPL/s26/q/wDjXsHxUlEPwr8QO3T7IR+ZA/rXlf7LEQFl4ilHeaJfyB/xo6jPoKiiimAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV8AeMrX7F411q1xjyb6ZMemHNff8AXw18X7X7J8XPEceMbr15B/wL5v60mB9Dfs0XX2j4UmLOfs99LH9MgN/7NXr9eC/srXW7wprdpn/V3qyY/wB5AP8A2WveqEAUUUUwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDN1C8itZA6bhN0xt4Ye9M0S4LiWKQ5bO8fj1rTeNZFw6hh6EVBFp8ENx50K7GxggdDXK6VT2qmnocTo1fbKonp2OT+MLhPhF4gJ/wCfdR+bqK8+/Zah2+FNbm/vXwT8kU/1rtPjpN5Pwb1v/bES/wDkVf8ACuV/ZeTHw91N/wC9qbf+i0rp6nae2UUUUwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvjL9oW1+y/GXUyBgTRwyD3zGP6ivs2vkz9qC18n4l2dwBgT6en44ZhQBv/so3WL7xFak/ejhkA/Fwf6V9KV8p/stT7PH2pxf89LAnHrh1/wAa+rKSAKKKKYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAeZ/tAvt+EF+P780S/8Aj2f6Vk/szxeX8L5m/wCel/If0Uf0q7+0VJs+FEij+O7iH8zSfs5R7PhHbsf+WlzM3/j2P6Uuo+h6tRRRTEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAGKMUUUAGKMUUUAGKMUUUAGK+Zv2r7Tbq/h27A/1kEsRP8AusD/AOzV9M14D+1bab/DWg3YH+rupIyf95Qf/ZaAPNv2cr9rP4vWsIOFu7eWJvy3f+y19j4r4d+DExtfjB4ffON1wUP4qwr7ipIAxRiiimAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQAYoxRRQB5B+0rJ5fwxgH97UIx/449an7P8ez4N6U3995m/8iNWD+09Jt+H2np/e1AH8kb/ABrqfgdF5Pwb0JfWJm/N2P8AWl1GegYoxRRTEGKMUUUAGKMUUUAGKMUUUAGKMUUUAGKMUUUAGKMUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeOftOWv2j4VwzAc2+oxPn2KuP6ivY683+P1p9q+C+snGTAYpR/38UfyNAHyd4EvfsPxC0G6zgR38JP03jNffAOVB9a/OyxuDa39vcL1ikVx+BzX6IW7iS1idTkMgIP4UkHQkooopgFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB4Z+1FJjwlo0f967c/kv/ANeu/wDhDH5Xwk8Or0zZo35815t+1NJjRdBj9ZpT+i16r8NIvK+GPh5PSwi/9BFLqPodRRRRTEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFcl8VLT7b8KfEUOM/6C7/98/N/SutrO8Q2ov8Awzqlo/3bizljP/AkI/rQB+eq9a/QPwjdfbvBmj3Wc+dZRPn6oK/P512Ssv8AdYivub4R3f2z4SeHJM522SR/98jb/Skg6HZ0UUUwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA+eP2qZcQ+H4/Xzz/6DXtPgiLyfAWhx/3bCEf+OCvDf2qXJvtBj9IZT+ZH+Fe/eHY/J8M6bH/ctYx/46KXUDSooopgFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFMlQSwuh6MpB/Gn0UAfnfq8H2bW7yBhgxzupH0Y19hfs93f2r4NaYCcmGSWM+2JDXyr8RLP7D8SPEEA4C6hNge28mvpD9mC6874Z3UBOTBfuMegKqaSA9oooopgFFFFABRXLa/42j8O+LtH0e+sj9n1ViiXok4ifsrLjueAc0vi7xrF4WvtHsls2vLnVbtbdFEmwRgnG9jg8Z7d+fSha2t10Da/3nUUVA13bx3EdvNcQpPIMrEXAZvoOppEvrSS4e3juoWmjGXjEgLL9R1FAFiioLa8tbxWazuYZwpwxicNg+hxWd4p1+Pw34dvdRJhea3geaOCWYR+btXJA/Kk2krsaV3ZGxRWfZ6tbzaZY3V1NDbNeRq6I8gGSQDgZ69afq+rWWh6XNqGqXCQW0IyzsQM+gHqT6VTVnZkp3V0XaKqjUrEtCv2y3DTjMSmVcv8AQZ5rH0jxSl9r2sabepDaNp90tvETMCZ90avkZxz83QUuthnRUVg6f4ie78V61pE9ukEemRwyCbzM7xIpOTwMYxWquo2T2rXSXlu1upw0wlUoPqc4oAs0VHDNFcRLLbypLG3R0YMD+IrB1nxUmmeJNG0qBYLhtQuWglxMN8GI2cEr77aOtg6XOioqGO7tpbh7eK4ieaP78auCy/UdRU1ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAfJ37SWuzXnxH/ALLkjQQ6dboEK9W3qHOfzr6Y8H3kmoeC9Hu5wqyTWcTsF6AlRXyb8fW8z4w6wfRYV/KNa+sPBkfleB9ET+7Ywj/xwUuo+ht0UUUxBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB8RfHC0+x/GTX0xgPMsv/fSK39a9b/ZSu92ieILTP8Aq7iOTH+8pH/stcB+0raC2+LTygY+02cUn1wCv/stdL+ynd7de160z/rLeOTH0Yj/ANmpAfTlFFFMAooooA4TxxoqeJ9UOklZklk02UwXAibbFMHRozvxgHK561yupRazf6T4Z1jXtOuU1SXWbQzQxQPJ5EMQYEnaDgFiW/4F7V7LRRHS3k7/AI3/AOAD1v8A10seTXMS3nizVdK8Qx6v9ok1NLvTzb258uVBtKEShfk24wQSOPrVdmutP+3i0srjXbB7G9ZYpLd4rm2JILRFwBvVj078cGvYMDOcc0YA6ClbS39bWHfW55p4AmY+O9SdVuvs9zpdo0TPZvDHlS4YDIHTgc8n3qh4xmkj1TxvZazYXVx9v0sDSXjtnlRgIjlAVBCsHyecda9aAA6CggHqM0SXN+P4ij7rv6fhY8buQJLw6f4jh1ddO1LSbaO0NraM/wAyrh4z8pMbbuQTj68V2PxBsJJvhNf20VvNPLHbxkR48yQ7WUnpyTgGuqtdQsr+WeOzuIp3tZPLmEbBvLfAOD6HBFWqqTvd99RRVreX9foeKeLL1bx9T/s/Tr63ZG0+WJo7KVjcoroSwJX5AoyNowc5zmrOs6NBqNx42jm02aTUr54X0uQ27Bi3koFZXxgYcc88Y5r2HA9KiW5tpLqS3SaJriIAvGHBdAehI6jNL+vy/wAh7HlWp/arG88Yvd6bdX+bKwQgQyBZSvDkED5tuckA5OKp2yNcajrqG81GzMmo2tzZ3h05vL3fZwMsm3GzIII69M+tez9aTaD1FHW/9dwWit/XY5nwdJfy+Cf31lHZXgaYKsalUdt7YkCtyAx+bB6ZrzjTp3uYPBcDadexa5p95Ml+8lnJ8kjQyAuX24IZiDkE9a9upMDOcc0PVsOlvX8TyzwFFFe6hpJ1CLV49d0uOSG5SaAxxpkYYl9oEgJAI5J716pSYGc4pabdxWCiiikMKKKKACiiigAooooAKKKKACiiigAooooAKKKKAPi744Pv+Lmun0kRf/HBX194cTy/DGmJ/dtYx/46K+OvjG5l+LXiAD/n72/kAK+zNJXZo1mvpAg/QUuo+hcooopiCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD5Y/aqtNnjbR7roJNP2H6rIx/8AZqz/ANmG68j4oXEJPFxp8i49w6H+hrqv2r7T5PDl2B1M8ZP02H+tedfs/wB19l+M2krnAnWWP/yGx/pUjPtKiiiqEFFFFABRRRQAVzHjjWNS0Wz0uTSXhR7rUoLWTzk3Da5x6109Zus6FY69HbR6isjLa3CXMXlysmJF5U8HnFHVeqDozz+98W+KtOs9emlvLOUaDqMUL4tsG5jk8s46/LgSdR6VqyeKNf1PULxvD9srx6dqAtZopNgRkGPMJYncG5JGOOBW1deBdEvIdTiuEuWTVZkmuwLqQb3XG09ePur09Ka3gPRT4gbV0W5jnk2mdI7l1jnK/dLoDhiMDk0R6X/rb/g/eD62OWstRn0zV/FjWc0NvJPr8MRLLuYhoI8iNccuccZ46k0ReNfEE+k6cY2gjuJNfl0qVpoRkovmYYgHAPyDOK6e88AaHeyXErrdJNcXqXzSx3UissyrtDKc/L8vHHakg+H2hW4QIt1iO/OoIDdyYWY5y3X3PHvRHon0t+HL/kwl1t5/jf8A4Byr+LvFFnZX1zcXdnKmk60mnzAW+Dco7oM9flIEg6elSat4k1PRx45urBbRLjT5rZo3MOS4dE4bnkgHH4V1U3gTRJ7a9glS5Md9drezj7VJ80qkEN145UcD0pbvwLol8uqLcpcsNWMZu/8ASpBv2ABe/GMDp6ULbX+tv1TDr/Xd/oc/L4p1yw1DXNPvr2xMlvDaT287RFQgmcqU2jJY/KdvqSM1s+CtevtYk1q21INv06+NujyIqOylFcbgvGfm7U/UvAGh6rNczXaXJmuY4Y3kF1ICPKbdGRzwQecir2h+GNP8P3V7caebgyXziSczTtJuYKF3cnqQBk96F5/1sJ+X9aa/ibNFFFAwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD4i+Jx874va8PXVHX/wAfxX2rYLt0+3HpEv8AKvijxyfP+MWsdw2tyY/7+19tW422sQ9EH8qlbj6EtFFFUIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAPDf2prTzfAml3OM+Tfbc+gZD/AIV4L8Jrv7H8WPDs2cf6Yqf99Ar/AFr6V/aQtPtHwfuJMZNvdRSfqV/9mr5Q8JXX2Lxlo1znAivoXJ9g4qWM/QbNFIpyin1FLVCCiiigAooooAKKKKACiiigAooooAKr3l7FYxpJcbgjOELAcKT0J9BVio7i3jureSCdd8cilWU9xVRtf3tgJM0Vk6RPJBLJpd45aaAZic/8tY+x+o6GtanODhKwBRRRUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAfD2un7R8Ybrv5muH9Zq+34+Ik/3RXw9/x8fGCHv5muL+swr7iUYQfSpW4dBaKKKoAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4L422hvPg5r6AZKQrJ/3y6mviS2k8m6ikH8DhvyNfevxCtftvw38Q24GS+nTYHuEJH6ivgbo30NJ7gfolps32jSrWYHIkhVs/UVarn/Ad39t+Hug3GcmSwhJPvsGa6CmAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBnavZSTxJc2fF5bHfEf73qp9iKsWF7HqFklxFkBh8ynqp7g+4NWaxp/+JNqv2peLK7YLOO0cnQP9D0P4VvH95Hk6rb/L/L/gi21NmijNFYDCiiigAooooAKKa8ixqWkYKo6kmsW+8Swwgra/O394jilcDbpa8/u9Turli8krDkcA1JBrl/a8LMWHo3NFwO8ormLbxaeBdQD6qa1rfXbC46S7D6PxTA0aKakiuMoysPY5p1ABRRRQAUUUUAFI3Cn6UtNc/u2+hoA+HtGBm+MWnZ6nW4z/AORQa+4h0r4g8Ij7R8ZtJx/Fqynn/fzX3BUxGFFFFUIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKup232zSbu26+dA8f5qRX54SrtuJFPZiP1r9Ga/PjxTZ/wBn+MNWtAMCC8lQD6MaTA+zfgxdfbPg/wCH5M5225j/AO+WK/0ruq8s/Z1u/tHwfs48/wConmj+nzlv616nTAKKKKACiiigAooooAKKKKACiiigAooooAKq6jJZpYyDUpI0t3G1jIcCrVRzQRXELRTxrJGwwVYZBqotKSbA88vvGv2UWulWVyty0d3GHuUOQ0O4Y59ex+lejA5ANcFr3w2jkDz+H5Bbvnd5D/cJznj0ruoN/wBnj8wYfaNw9DXpY14aVODoed+/QiPNfUkopCcdTisy+1y3tMqh8x/boK8ss02YKpLEADqTWPf+Ire2ysH7x/XtXP32tXN4SC+1fQVmFt3U0hl+81a4vW/eOcelUutNUFjgc1dgsyRul+UenelYDH1O8SyhWSU7V3DJP1qWK4WaIOpyCMitqRLdo/LMalfcZzVdraIjCfKPSmTrcobxSh6mksj/AA/pVZoJEP8AjRYZaivZ4DmGVl+hrUtvE95FgSlZR/tCudJdfvLSiQZ60WA7e38U20mBMhjPqOa04NRtbj/VTKT6HivORJUqTMvQkfSgZ6XRXBW+sXdv9yZiPQmtW28UuOLiMH3FAjqKiuG2WsreiE/pWfB4hspfvOYz/tCpr29gk0q7aGZGIgcjB/2TTA+MPhyPO+M+iDrnUc/lk19vV8OfDW/tdO+Lmk3upTCG2huneSRgTt+VvT3xX23ZXtvqNnHdWcolhkGVYdxUoZYoooqhBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV8I/FW1Np8WvEkJGAb+VwPZmyP5193V8XfHy0+y/GrVsDHmiKT841pMD2T9ly68z4e6jbscmHUWIHoDGn9c17bXzz+yrdZtfENnn7rwyY+oYf0r6GoQBRRRTAKKKKACiiigAooooAKKKKACiiigAooqnd6nb2oO5tzDsDQBczWfe6zbWYI3b39B2rn9Q1+W5ykZ2J6A1jPIzsSzZNLcZq3+vXF1kI21fQVkNIXOWJJpM1JDbSTN8g47k0xEXJNWYbJ5OWG1fU1ajt4rflvnf8ASiS4J+lMByiK3XEYyfU1G0xY8moXfvmozJSAmL0gkqAt70ZoGWBJTg4I5FVt9PD0ASmGJx0xUD6er8rg1IGp6ybaBFB7F0PGfxqIrInUZ+lbAl9aCIn+8ooAxhLzzxUqyd81eezjfofwNV5LAqPlB/CgBgkzT1lK/dJFV2glT3pvmFeGBFICneeFvD+pFmvNItmdusiJtY/iK6vRtXTR9OisrWCNYYxgADH8qwhKDThJRYDu7XXrW4Vd/wC7Y9RnIH41opPHJjY6tn0NebCTHQ4qaK/mi+7IfzoA9Gori7bxJcw4DHcK1rbxPBJgTDB9RRcDeoqpDqdpNjbKB9atKyuMowYeoNMBaKKKACiiigAooooAKKKKACiiigAooooAKKKKADFGKKKADFGKKKADFGKKKADFGKKKADFfI37Tlr5HxShnA/19jGx+oLL/AEFfXNfMP7Vlpt8R6Fd4/wBZavHn12tn/wBmpMCP9le72eLNZtSf9daK/wBdrf8A2VfUWK+Qf2aLvyPit5JP+vspVHuQQf6Gvr6hAGKMUUUwDFGKKKADFGKKKADFGKKKADFGKKr3N9BaKTLIAf7o60AWMVUudRt7XIdwXH8K9a57UvEUkuUt2Maei9T+NYb3Dv1PH1oGb9/4gaQFIuB6Kf61hy3Dyn5j+FQZpQCxwBk0CEOTTkhaRsKCTVuLT3JDSkKvcd6tgpCu2IYFMRXisEjG6c5P90VI82BtQADHamu+agZqABpCetMJppbmmlqBgxzTDS5pKACkoJoFIYop601RUgoAULTlFA6U4CgAxSgGlAxS0CHLUgPFRrxTx0oAeEV/vAGmPZROOOKcGpwegDPm0s9U5+nFUpLOWP1/Gt3fzSFgRyM0Ac8TIn3lNIJxn0rbkton7YPtVOXTg3TB+tICn5o9acJBSS2DpyMj9arskids0AXVnZPusR9DVuDVrmFgVkPHvWP52Oop6yigDqbXxVNHgTfOPeteDxJaS43ArXBB809WoA9MivLeYDy5VOe2anrzOO5lj+45H41oW+uXduMK5x9aYHeYoxXLW3iqQYE6q4+mDWrb6/aTfeJQ+/IoA1MUYqOKeKYZikV/oakoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvnz9q213aP4eu8fcmljz9Qp/pX0HXgX7VWpRReHND02SJi09y8wlHRQq4I/Hf+lJgeRfAq7+yfGfRGzjzGkjP4xtX21XwN4C1ZdE8faLfQruaO8jBL9AC2Dx9Ca++FOVB9RmhALRRRTAKKKKACiiobi6htY99xIEHuetAE1Q3F1Daxlpnx7dzWBf+KAAVtRtH95up/CueudRmuGJZzz3JpDNzUfEsj5S3/dL0yOWP+FYEt1JKxJY8nnPeoNxNHWgQuaBz0qSC3e4bCD8T0rRitYrflvnb3pgVrexkm5b5F9TV+OOK3X5Blu7GmtNnvUTSUwJXlz3qBpKY0lRliaAHF80wnNITim5oAM03NKxptAC0lLSgUANpQKUClxSGKKfikUc08CgBVpwpAKcBQAUtAFLigBRS5pvSlzQAZo3Uhpp60hDt1BaoyaN1ADy9JvqMtSFqAJd9NZI5PvKKj3GjcaAGSWEb/dP51Tk0x15UZ+lXxIR1p4lpgYrQSp1/WgOy/eU1uZVvvAGmNZwydPlPtTAy0kBqZWqy+l5+4Qahaylj7GgBRz0pupana+H9Kkvr0/dHyrnlj6CpA0dlbSXV46pFGu4k9hXi/jHxXL4j1NipK2kR2xR+3qfc1nOVkUlc9e+D+tah4h1DV769lJiXYscY+6mSTXqleS/AGLHhzUpf71wB+SivWqIfCKW4UUUVYgooooAKKKKACiiigAooooAKKKKACiiigArxL9qO1hk+HNjcvGDLFfqqP3AKtkfoK9trzj49aXBqfwf1Zp/vWmy4jI7MGA/kTQB8VQyPDOksfDxsGU+4Nfonp1wt3pltcJ92WJXH4jNfnWn3hX3z8PLz7f8ADfw/c5yZNPhJ+uwZpAdJRRTZJUiQvK4VR1JNMB1MlmjgQvK4RR3JrB1DxVDEClmN5/vt0rl73Vri8kLSSFvqaQzptU8TrEpSy+903kZx9BXLXOoTXMheR2Zj3Y1ULknJPNJuoESbiTnqaMmkUFmwoyT6Veg088NOcD+73oArRRPK2EUk1oQ2CxkNO27/AGRUwZIlCxgKKjeUk0wJjIFG1AFUelRNJk1EZCaZv96YEpemM1N3ZpueaAFJ9aQmkJpDTAM0lFFABijFFLSAB9KWgU6gBopcUtKBSGOUU8UAelOUUAOA4oApGkSNcyOqL3LHFLFLFPGJIJFkQ9GQgg0ALiinGkoAQ0lLSfjSASmHqaexwKjJoASkNBpCeOKBDTmjOKM5pCaACikyKU0AJQM0UopgOBqZDUIHNSpQBYjanSS7V/zxUW/aK89+InjH7FA2k6fJ/pEoxM6n7gPb60pSsrjSuYXxC8YHUrg6Xp8n+ixN+8ZT/rG/wFcRaWdxqF5FaWUTSzzMERFHLE1Fkk88kmvffg78P/7Ls18QatF/pk6/6PGw5iQ9/qa5tZyNXaKOj+FfhW/8J+FGtNVVFuZZ2kKo2QBgAc/hXbUUV0pWVjHcKKKKYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXLfEvSZNc+Geu2EBxJJaMy+5X5sfpXU1Q1z/kX9R/69Zev+6aAPzxA2vg9jX238D70XPwb0R3YARRvEST02uRXxLJ/rn/3jX0H8GNcuJPBP2EyN5dpM2xQeBu5z+ppAe/ah4mtrXcsH71x1PYVyeoa5c3zfO5I9OwrKaVnOWOabmgCQyFupzSZplWILWSc/KMAdSelAEYyelXLewkk5f5F96sw28VsuTh39TSvOW70wJEWK3XEa8+tI0xNVy9N35oAmaTNMLnFRlqQtTAeGpc5qMGnCgB+aD0popaACiloxTASilxigCgBKdikxTsUgAU4CkAxThSABUMkxSdFKYVuN3vU9DIrrhxkUASRj5eapa5FqUmjzjQ5o4b3bmJpFyufSrcbbCFb8D61OMY4q9APl688SeLrLxgg1u7kW7hl2mO5/wBV6dOmK9A8M6t4lg1Z7iykGpCd1M1pbw7YQO5DdAf516bqfhjR9ZuobnU7CG4lh+4zrnFWZrjTdDtA00kFlAo43EIKVgLSsTGpKlSRyp7UtcxD4/0fUb57DQ5P7Su0XcUjYKv5n+ma09H1211hJFjzFcwnbNbufnjP9R70mhmlTTSmkxUgNNNPWnUw9aAEJph5pxptAhKSlpDQAUtJ1pRQAuKcFoxT1WgBFWpQAq5NKFAHNZmva1baHpcl5dtwowi93bsBRew7XMrxp4pj8PaafLYNeTDESf3f9o14jNPJdTvNO5eR23Mx7mresatca3qkt5dtl3PA7KPQVp+DPCd14u8QRWFuCsIO6eXHEad/x9K5ZScnZGqVkdR8JPAB8RaqNW1KI/2batlVYcTOO30FfRaqFUKowAMADtVTStLtdF0uCwsIhFBAgVVH86uV0Qjyqxk3dhRRRViCiiigAooooAKKKKACiiigAooooAKKKKACkJwMnisjUvEllYZRW82X+6vQfjXI6l4purslQ2xP7q0AdZqviOCwG2ICWT2PAri9a8U3T2d08kmV8hwVHTBUisyS4eU/Mc1VvF820lT+8hH6UgPk+T/XP/vGvY/gjNmy1GH0dG/SvHblfLu5VP8AC5H616p8DmZ9Vv7dASWiVsD2P/16APZhzU0NvJM2EXPvVy305UAa4PP90VZaYIu2MBR7U7ARQ2UUB3SnefTsKlef5cLgD2qs8pJ61Gz0wJmlPrTd9Qb6TdSAm30obmoQaeDQMeTSZpM0UxDhUgpi1IKAFxS4opaAExSjpRRQAUUUooAXFKBRilxSASnUlOFABSikpaBgV3DB6GmGVoHAl+4ej/41KKVlWRSrYINO4Dlkz1rD8XeFLHxhorWV58rjmGZesbev0q3IZrB+8kHp3WrKzxsgcOu0+pp3EfPWj+HvEfg/xjJYLpMl2W4EsSfw/wB4N/Su70b4f+I38TLrlxqsmnYxthU72K91PbFekG/tkbJlTP15qZLoSD5Aze+MUaAKCVAB5PrQTUeJXbLFVHoKdjHfNSwEJNN5pxOKSkA05ppzTjTSPegBKSloFABinAUAU9RQAqipVWkVaV2Cr1wO59KBkV3dxWdtJPcuI4Y1LMxPQV4X4w8UTeJNWLKStpESsMft6n3NbHxC8Y/2pcNpmnyf6JE37xgf9Y3+Arhl5xmuac76I0ii1p9jcalfQ2dnGZZ5nCIgHUmvqTwF4Ot/B3h9LZVVruXD3EuOWb0+grkfg/4A/sizGvarFi8uF/cRsOYkPf6mvVq0pwsrsmUr6BRRRWxAUUUUAFFFFABRRRQAUUUUAFFFFABRVG/1iz05CbiYBv7o5NcZrPjV7hGitQY09QeTSuM63UdfsdOUh5Q8n9xTXIar4yuLgslufLTsBXKy3LyE5J/Oos0AWpLmSViWPJpgOaiB9KkQE9KBEg56U7YXBUDORVq206SXDP8AIvqa0UjhtlOwZPqaYHx3r1ubXxDfwMMGO4dcEe5r0H4F3Rt/Gk0Y/wCWlq36EVyXxBi8nx/rC9P9JZvz5rZ+EM/k+PrX/bR1/TP9KAPpYyk96iZ81FvppemA5mphNN3UhNADs0opmacKAHg4p6mmDrT1FIY8c0uKFFOA5oAVactIBThQIcKWkApcUALijFFOAoAbilFLS0wCjrRRSAXFLTaDQAtLSUDmkMdmnDNMHFOzQA9kEiENWbJoMMsu+RiB/dBrR3GjJNAFeHTrW3H7qFc+pGasYxRmkJoAXNNNIaM0ANNJmnE03NAhKbS0Y5oATFKFpacAaAALUirQoqT7ooAM7R71518RfGIsoW0jTpP9IkGJ3U/cH9361u+NfFUfh3TGWNg19MMRL/d/2jXhs08l1O80zF5HbczHqTWFSdtEaRiR9TzzXpvwj8AHxDqY1bU4v+Jdat8qsOJnHb6CuW8FeErrxf4gisYAVhU7p5ccInf8T0r6o0rTLXRtLgsLCMRwQIFVQP1qKUOZ3ZUpW0LYUKoCjAHAAFLRRXWYhRRRQAUUUUAFFFFABRRRQAUUVzfxB1Z9D8CahqMTOrQeUcocHmVAf0NJuyuBtXeo2tjHuuZlT2zzXF6742eTMWnN5ad27muHs9di12HzobppGxghjyPwp5hK8nnPepvcq1iae8muGLO5JPcmoKcq+1OC0xEdOAqaKB5W2ouTWlDp0cXM53H0HSmBStrOSc/KMD1NasFpDbcn539TStMAuEGB7VCZOetUItPPnpUDyZqMv71Gz0AfN3xVi8r4iah/t7H/ADUVB8OZ/I8d6Y2cZl2/mCK1fjJD5fjov/z1t0b+Y/pXN+EZfI8V6a/TFyn86QH1QDmkNNQ5UUppgFJRS4oAAaeKTFOVaAHrUgqNRzUgpAPFPHSmqKdtoAcOacOtIopwFACgU4CgU7FACYpRSgUEUAJRRRTAKKSikAuaSjiigAFLSUmaQx4OaAMUwU4GgB+aAcU2jr0oAcTSZpMikoAUmjNJScUAOOTTMU4GkxQITFLQBzTsUAIFqRVpFWpVWgAAwDmszXtbt9B0qS9uiPl4RM8u3YVdvLuKytZLm6cRwxLuZj2rwvxh4pn8S6mTkraREiGP0Hqfes5y5UXFXMvWNXutc1OW9vHLO54HZR2AqPT7C41K/hs7KJpZ5nCIijqTVdV5wK+gPg/4A/sixGu6rFi9uF/cIw5iT1+prljFzkaN2R1vgPwdb+DvD8dsgV7uQBriUD7zen0FdPRijFdySSsjDcKKMUYpgFFGKMUAFFGKMUAFFFFABRRRQAVw3xnfy/hFrTf9cP8A0fHXc1yHxV0e5174Y6xp1iMzyJGyDHXZKjn9FNJ6oa3PlW01G4sLoT2cjRkeh616JoHjW31Dbb6hthnPG7+Fq8yKGJyjjDKcEe9KODkcVyRk0b7nuqqrAFcEHuK0LbTg6h5WG09ga8h8P+MrrS2WG6zPb9OTyv0r0rSdagv7cS2UwdT1XPIrojJMycbG8WSFdsYCioHlJNRiYSd8H0prE5rQkcZM0m+mE0lMQ8tSE0go7UAeIfG6Hb4hsZf79tj8mP8AjXn+kS+Tq1rIP4ZkP616f8coPn0qf2dP5GvKbV9twjejA0gPrWA7oUb1UGpCaqaXJ5ul2z/3olP6VbxmmAmKcKAKWgBwpw600U5aQD1609RTV4qQH0pAOFPpi81IKAFFSAUwU8HNMBwp1JS5p2AXGaDxRmuQ+Ii+JjoDP4VmVZFz5qAfOy/7JquUVzR8QeMdF8MwF9UvI0fHESnLt+FZPhD4k6T4uuJbe33W9yhJWKU8uvqP8K+dbeaO41lm8SSXD8neWJLZ98113gi08N3OqXcsb6mt0n/HqluOee4I7/WgD6LLUZyKwvCg1o6OB4gCecGwhB+Yp23ds/Stv7vJPFJoB1Lmm5yOKKgYtFANJmgBaXOKTNGaAFBpQaaDS96BhS0lLmgAopM0uaBCUoo704c9KAAU8CkAqRRQAqLQ7hQcnAHU0pOOBXnPxG8ZCzibR9Nk/fOP37qfuj0qJSSQ0rmD8QvGB1a6bTNPkP2OI4dlP+sb/CuHVaAMmuk8GeErrxd4gjsYAUhU7p5cf6tP8fSuRtyZtsjqvhJ4B/t/Uhq+pxf8S+1fKKw4mcf0FfQwAVQFAAHAAqrpem2uj6ZBYWEYiggQKqirddkI8qsYt3YUUUVYgooooAKKKKACiiigAooooAKKKKACkdd8bL6jFLRQB8leP/B174W16fz0Y28rlo5MccmuTr7O1vw/p3iGxa11S3WZCCBnqPpXg/xA+D91oryX2go09nnLJ3QVzyp9Uaxn0Z5XjNWtP1K60y4E1pKyMOozwfrVdkZHKuCrDgg0YzWWxoeneHvGVrqe2G7IguOnJ4auuU5Hr714IoKkFeDXYeHPG1xYFbfUczQdA2fmWtY1OjM5RPTNtJtptjd22o2wms5VkQjqD0qxsrdMzZCBS08pmk2EUxHlfxwh3aFp8o/gnI/Mf/WrxaLhxXvfxkt/M8EB8f6q4U/nkV4Gh+cUgPqjwtJ53hXTZP71un8q2BXNfD+bzvAumN1xDt/I4rpcUwAU7FJS0AGKcPaminCgB6ipBTFp60hki1JTBSltqknoKAJB0pGkVASTjHWowS3sKSa3juLeSCVdySKVYeoNWkI5LxL8UdC8PbovP+2XS/8ALGA5wfc9BXnMnxX8W+IdTRNChjt442DGNRnIz/ETWN8Rfh/c+Fr9ru0DTabM2VfqYyex/wAaq+AZrK01E3MrTPcoflt0UkSj0wOtMR734Q8XR+IrZoblPsupQcT2zdR/tD1BrpTjFcENG1HxBq+n6rDaLoS2pBD9ZpV/ukdAPrXchsKATVJgcV4n+FuieJNWjv5Va2kz++8njzR7+/vXQ6L4e0nw9aiHTLOKBQOSB8zfU960iXbhOB6mhIwvJyzepqW0gFDE9BgUtFJUN3GLS0nakpAOpKbmigB9LTBmnUDFFKKbRQA6ikzRQIWlxSYp2KAEAzTwKAKeq8UAKq5p/wB0UDCisvX9cttB0qW9uz0GETPLt6Um7AZPjbxZH4c0wpCwa9mGI1/u/wC0a8QllkuZ3mmYvI5yzE8k1Z1fVbnW9Tlvbx9zueB2UegqqorknLmZtFWLWnWNzqOoQWljEZbiZwqIB1NfUfgTwfb+D/D6WygPdy/Pcy45ZvT6CuR+EHgH+yLFde1WLF7cJ+4RhzEh7/U16pW9OFtWRKV9AooorYgKKKKACiiigAooooAKKKKACiiigAooooAKKKKACmuiyRlJFDKwwQe9OooA8p+IHwgttYWS+0FEt7nGWjA4c14PqujX+iXz22o27wupx8wxmvs6uc8V+CtK8V2bR30IE2PklHUGspU1LYpSaPkkCnAZrtPE/wAONX8MSyvJbma0/hlTnFcgq4Nc7TibJ3NDR9YvNHuBJaSkD+JD0avT9A8V2WsxrG5ENxjlGPX6V5IoqaJmjcPGxVgcgg9KcZtCaTPdDFxxTCnFcP4c8cvDtttW+dOgl7j613sMsN3CJbd1dGGQwNdMZqRk42OE+K9v53w9vsDJQo3/AI8K+cF4YV9S/EK287wHqy4zi3LflzXyyv3qok+kPhZJ5ngGz/2Wdf8Ax412IrgPg7Nv8FlD/wAs52H8j/WvQB0pgFLTeKcKAAU4UlOA4oAcOtSrUaipKQyQdacDTBTgcUAIP3R/2D+lSD2puQRg9KaCY+Ccr61aYiK+sbbUrOW0vYVlhlUqyMMgis7RvCuh+HY8abYwwHvJjLH8avtcTzOUtY8KOsj9PwqWK0VCHmcyv6t0/KncRMGLL8n5mk2YOT8x9TTyRTTUt3GLmkzSZpMg1IC5ozSZFGRQA7NJnmkzRmgBaAaTvRQA7NLmmjrTjQAZozRQKAFzRmilWgBy08c00c09RQMcBUgGBSKKGI5ycADmgRBeXcNnayXNy4jiiUszHsK8J8YeKJvEurFwStrGSIY/b1+tbfxE8YnU7ptL05/9EiOJGB/1jf4VwqLXNUl0RrGNhyLXpvwl8BHxBqY1bUos6dav8isOJnHb6CuW8F+Errxd4gisbcFYQd1xLjhE/wAT0FfUul6Za6PpkFhYRrHBAgVVFFKF3dhKXQtABVAAwB0ApaKK6jIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAIbu0hvbZ7e5RZI3GCpFeIePfhFJZF9Q8PI0sf3niHVa91pCAQQRkHqKmUVJajTsfHDwvBK0cylXU4II6Uqivojxj8LdO15XubBRbXeOoHDGvC9X0W70S/ktb2JkZTgEjg1yyg4mqkmZwFbGi+IL3RZgYXLRE/NG3Q1lgZ7UtSm0Vuej32t2XiLwfqUUTBZmtJA0THn7pr5bIwx+tetTl1tpfKcqShGQcdq8kb/WEe9dMJXMpKzPdvglJ5nh29jz92fOPqo/wr00LgV5B8D7vZDqMPujfzr2JCHXIrQkjxmngcU7GKKBCAU4LxQBzTwKBgBinCkpaAHilzTAaXNADutO60wUuaAHg4HFGajJ6Uu6gB+aTNNzSE0xDqSkzSbqQDqM0maAaYDqSkzRmgB2aWmilpAOFFNpwoGKKXFJSigAxTwOKB1pyigQ5RUqrTVFSZ2igYE9q88+IvjIWFu2k6dJ/pEg/fOp+4PT61ueNPFcXhrSz5ZDXkwIiT09zXhU88t3cvPcOXkkbLMe5rCcraIuMeo1QSSTyTV7TtPuNSvobOyiMs8zBEQDqTVVBX0D8IfAP9jWI1zVYv9NuF/cow/1SH+prKMeZ2Kk7I6zwL4Pt/B/h+O1QK11IA9xLjlm9PoK6aiiuxKysjEKKKKYBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFYniPwrp/iWxaG8iUP/DIByK26KAPnHxf8O9Q8LnzQfPtj0dR0rkBX1vdWkF7btDdRLLG3VWGRXj/jX4USxO97oA3R9TAByPpXPOn1RpGXc8mkXKEe1eS3K7LqVfRyK9huIJLV2juUMbr1DcYryPUtp1O52HK+Y2CO/NOluKZ1Hw/8VjwvfSyyLvikADr3PNfQ+gazY65ZrPYzKwIyVzyK+WdGs/tEzNID5QGCfeu60zWbvT75ZLOTytijG3vj1qnKzJR9AmPmk8uuO8N+P7fUttvqG2G46BiflauxjnB+hqlIdhwTFG2pF2uMrzSFaYhpFJilxzSYpgApaMU1QQvzcmgBc0tJRigApc0mKM0ALmkxmg0lAhc0lJRmmAtKKbSigY6lNNzR1oEOBoFJRigY+nVGvWnjrSAeKXFIBzTwKYCgZqVVpqiplAC5NIAwAOazdb1q30LS5b27bAUfKvdj6Crdxcx21vJcXDhI4wWYnsK8L8Z+K5fEmqERkraQkrEnr71nOVkUlcy9b1i517VZby7YkuflXso9BVNUpFWuo8F+ELnxdr8VlBlYFw9xKBwi/wCJrm1kzTY6n4R+ATr2pLrOpxZ0+1fMasOJnH9BX0IAAAAMAdBVXTNNttI02CxsYhFBAgRFHoKt12Rjyqxi3dhRRRVCCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArL8R+ILDwt4fu9Y1aTy7W2Tc3qT0AHuTxWpXO+O/CVv448H3mhXUrQrOAVkX+Bgcg0AfLvxD8a23j6Nr3R7E6ZMM78yf65O3HrXEaVoMMqb7li7f3Rxium8ffDPWvh1d28F/IbzTpVJS5gjO1TnofQ1y1x4gUaWLe3j8udWx53cr71m0wNWYW9p5VsgCc/Kop9tLvu5MnjGBXI291Kb1JnYu4PG412GnxJNbpP9yR13bc1nJWKiWVJDZBwa7Lw141udP22+oEzW/QMeq1yQjA61NGvpUXsanuen6hBe26z2cquh9D0rQSQPweDXiek6vd6RcCS0kIHdT0NelaD4os9YURsRDc45Qnr9K1jO5DidKVppWkRiOG5FSDBHFWSR9KQnFSEUxqYDSaTNONNJpiDNFJmigAzSbqM0UAGaTNLSZpgLml6Cm0uaBC5pc03NOzQMUGlzxxSClzQADinimipFFADlqZRTVFSoKAHKKaz5PsKHbnA/GuD+Ifi8aTaHTLB/wDS5lw7D+Bf8aiTsNK5z/xH8ZG+mbSdOk/0eM4ldT98+n0rgEFN5ZizHJJyc1Mg7Y5rlk7s1tYuaZp9xqmoQ2VjEZZ53CIo7mvqTwN4Qt/B3h+Ozjw9zJ89xLjl2/wFcp8IfAP9iaeut6rFi/uV/dIw5hQ/1NeoV0U4WV2ZylcKKKK1ICiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAK19HZyWr/2gkTwqNzCUArx9a+B/GslpN411aXTZVktZbuR4mVcDaWJHFe7fH3xZqcficeH2kltdP8lXDocbiw5PuO1eFjRYbO4WS4mjuEZvl2Hrz3qWwMa2jYyK3O0HJNei+F/DGt65D9q0jTprmCIfO6LwOCcfpXb/AAi+HNj4s1V7zUrVZNMsuCmPllkPb8K+krDTbPS7VbbTrWK2hXokShRU8vNuNaHyBNAd+xkMcicOpGCPqKVI8CvpLxn8NNL8Uq9zAos9RxxMg4c/7Q714N4g8Oan4Zvza6rbtGc/JIOVceoNYyg4msZJmUF4pyM8TB42KsDkEHpQDS1mUdp4c8cNEUttXyy9BL3H1rvoJ4riFZbdw6MMgg14UTzWtonia90SYeUxeE/ejY8VrGfclxPZM5pDWRo3iKy1qANbuFkH3oyeRWrvrZO5AjUz60rNTM1RIppKaaN1ADqCaaTSZoAdSZpuaSmBJmjNMzS5zQA8U4VGDS7qAJKUZpgOaeDQA9amUVCvWp0oAkQU9m28DrTC4UfyrP1XVbfSdPlvLx9qRjP1PpSbGkUPFviWLw3pLS5DXMgxCnv614VdXc2oXklzdOXlkYlmNXPEXiC48QatJdzk7c4jTPCr6VnJzXLOV2apWQ9Vr1P4Q+ADrmoLreqRf6BbN+6RhxK4/oK5PwL4RufGPiCOziBW2TD3MuPuJ/ia+ptO0+20rTobKxiEUEKBUUdhVU4Xd2TJ9CyAAMDgCloorpMgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA8z+Lfwl/wCFj/ZLi1vxZXdqhQFlyrqTnB/WuO8Mfs5Nbaxby+JryC6sYBkwRAjzW7ZPpXvtFKyAoaLomneHtNSw0e1jtbZCSEQdz3q/RRTAKoaxoun69YPZ6pbJPE394cqfUHtV+igD598bfCu/8Pl7zSA97p45IAy8Q9x3HvXnxOK+wiAQQQCD1Brzbxx8JbPWhJfaFstL08tHjCSf4GsJU+qLUu54GxpjNVrU9Nu9J1CWx1GFobiI4ZGqkTWJpclt7ya0mEtvI0bqcgqa9A8O+PIrrbbaoRHL0EnZvrXmxbmoy+DxVRk0D1Pf1kDoGQhlPQilzmvI/D3jW60hlhuSZ7b0J5X6V6Xpur2mrWwms5Q47jPIraMkzNqxoHrSZpgf1p3WtCRCaTNBpmaYDs0ZpmaKAJM0opuRSg0APBpQabkUvFADwcDmnKaYAD1p60ASr1qbcFGTUK8cmoZJtzYB4FK4Ek90kaNLKwREGSSegrxPx14wfxBqRt7ZiLKEkKP7x9a1viN403ltH02T5Qf37qev+zXm6HvWE5dEaxRaQ1o6Vp91q2pQWNhEZbidwiKPWqVlaz31wkFnC80rnCoi5Jr6M+D/AMOz4dsP7Y1iHbqVwMJGw5hT/E1EYuTBysdd4G8IW3g7w7FZQhXuHG+4l7u/+ArpKKK6kraGIUUUUwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAglsbSeTfPawyOf4njBP60z+ytP/AOfG2/78r/hVqigCp/ZWn/8APha/9+V/wo/snTv+fC1/78r/AIVbooAqf2Tp3/Pha/8Aflf8KfHYWcX+qtIE/wB2MCrFFAEX2aD/AJ4x/wDfApfs0H/PGP8A74FSUUARfZoP+eMf/fAo+zQf88I/++BUtFAEX2aD/nhH/wB8Cj7NB/zwj/74FS0UARfZoP8AnjH/AN8Cj7NB/wA8Y/8AvgVLRQBH9mg/54x/98Cj7NB/zxj/AO+BUlFAEf2eD/njH/3yKPs8P/PGP/vkVJRQBH9nh/55R/8AfIo+zQf88Y/++BUlFAFM6RprMS2n2pJ6kwLz+lH9j6b/ANA60/78L/hVyigCtFptjBIHhs7eNx0ZIlBH44qzRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH/2Q==)

**Figura 1**. Mesa de prueba ubicada en la UVG.

Los programas se ejecutaron en una computadora con las siguientes especificaciones: procesador Intel i7-10700, 16 GB de memoria RAM y tarjeta gráfica Quadro P400.

**Métodos**

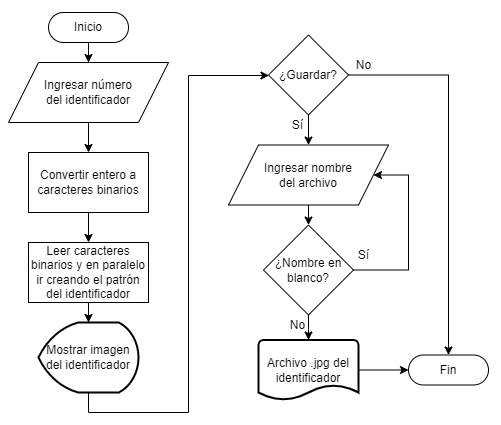
En la figura 2, se muestra el diseño de los identificadores utilizados para este trabajo y la manera de calcular el número de identificador. Estos se colocarán sobre los agentes que se tengan funcionando con un número diferente cada uno. Los valores pueden ir de 0 hasta 255.

Interfaz de usuario gráfica

Descripción generada automáticamente

**Figura 2**. Definición de diseño para identificadores de robots por Rodas (2019).

En la figura 3 se muestra el diagrama de flujo que se sigue para la generación de códigos.



**Figura 3**. Diagrama de flujo del algoritmo para generación de identificadores.

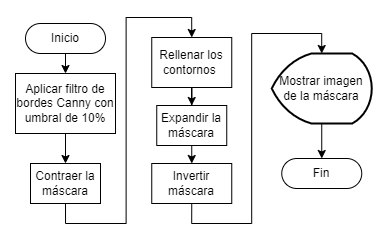
En la figura 4 se muestra el diagrama de flujo que se sigue en el programa para la calibración de la cámara. Cabe mencionar que en este procedimiento se incluye el “detectar obstáculos” ya que se ejecuta dentro de la calibración, pero se creó un diagrama de flujo separado para mantener mejor el orden y que sea más fácil de interpretar.

Imagen que contiene Diagrama

Descripción generada automáticamente

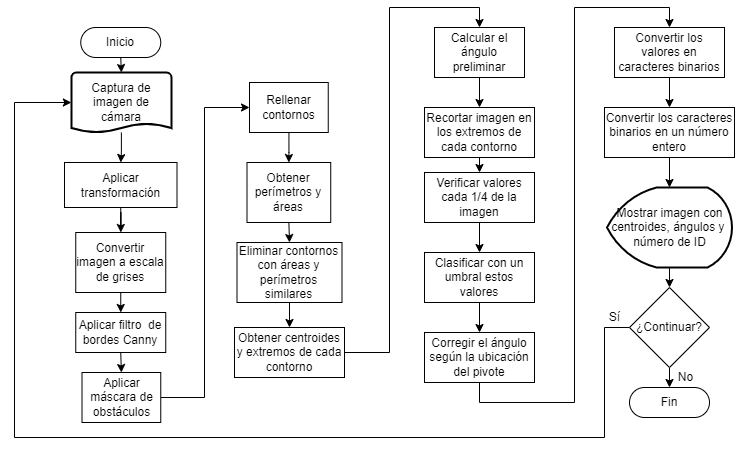
**Figura 4**. Diagrama de flujo del algoritmo para calibrar la cámara.

En la figura 5 se muestra el diagrama de flujo utilizado para detectar obstáculos durante la calibración.



**Figura 5**. Diagrama de flujo del algoritmo para detectar obstáculos.

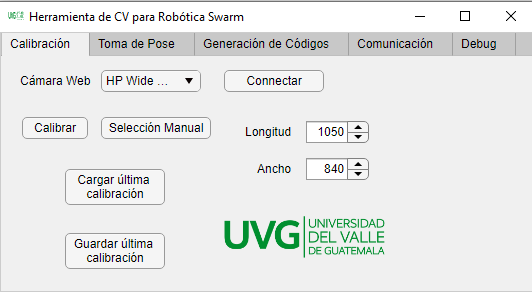
Por último, la figura 6 muestra el diagrama de flujo para tomar la pose de los identificadores. Este se realizó en un ciclo para que funcionase continuamente en lugar de tener que solicitar actualización manualmente.



**Figura 6**. Diagrama de flujo del algoritmo para obtener la pose de los identificadores.

Para facilitar el uso de la herramienta tanto para el usuario como para el desarrollo, se diseñó una interfaz de usuario (o GUI por sus siglas en inglés). Para mantener ordenado y dividido todo en sus respectivos algoritmos, se agregaron pestañas por las que el usuario puede navegar dependiendo de la tarea que desee realizar. En la figura 7 se muestran todas pestañas creadas.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente 

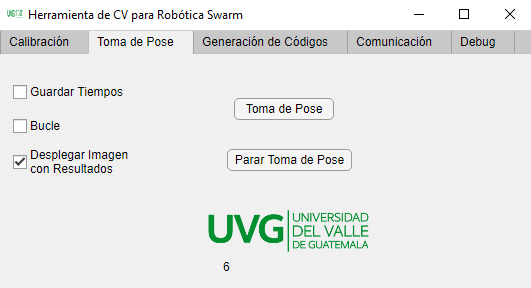
a)

b)

c)

d)

e)

 Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 7**. Capturas de las pestañas de la interfaz gráfica diseñada

En la figura 7a, se encuentra la pestaña de generación de códigos en el que se puede ingresar un número de identificador, generar la imagen y dar un nombre para guardar la imagen en formato jpg. En la figura 7b está la pestaña de calibración. Aquí tiene la opción de conectarse a la cámara deseada (de tener varias), calibrar automáticamente o realizar una selección manual de esquinas. También ingresar longitud y ancho del área de trabajo para realizar la conversión (estas pueden ser cualquier unidad deseada) y opciones de guardar y retribuir datos de calibración en caso no se cambie nada de la mesa para ahorrar tiempo.

La figura 7c contiene la pestaña de captura de datos. En esta se tienen opciones como funcionamiento continuo, despliegue de datos para visualización, guardar tiempos de funcionamiento y botones para comenzar y detener la captura de datos. La pestaña mostrada en la figura 7d fue agregada debido a la verificación de concepto. En esta se pueden agregar y eliminar direcciones IP y números de identificación conforme sea necesario. Luego de agregar el puerto, se puede habilitar comunicación UDP o TCP según se necesite, pero no ambas simultáneamente. Y, en la figura 7e, se muestra una pestaña de depuración que se implementó para facilitar el desarrollo de la aplicación. En esta se pueden cambiar parámetros sin necesidad de modificar directamente el código.

Por último, se agregó una funcionalidad para detectar marcadores circulares dentro del área de trabajo como nodos de movimiento para implementación en un algoritmo ACO. Este detecta todos los círculos en forma de matriz y realiza las conexiones entre ellos tomando en consideración si existen entre ellos obstáculos que eviten esta conexión. Esta detección es realizada durante la calibración.

**Resultados y discusión**

**Generación de identificadores**

En la figura 8 se puede ver la imagen resultante de ejecutar el código con el parámetro de identificador número 155. Esta luego puede ser guardada como una imagen para ser impresa y utilizada.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 8**. Imagen resultante de generación del código con número 155.

**Calibración de cámara**

En la figura 9 se puede ver el ejemplo de una calibración exitosa. Al lado izquierdo se ve la imagen cortada y rotada de manera que únicamente el área de trabajo se observe luego de esta transformación, y al lado derecho, se ve la máscara de obstáculos generada. En el caso de la máscara, se tiene en color negro los píxeles en donde se encontraron obstáculos. Esta puede luego utilizarse en algoritmos de planificación de trayectorias para evitar dichos obstáculos como los que se describen en el libro *Modern Robotics: Mechanics, Planning, and Control* por Lynch y Park (2017).

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

**Figura 9**. Imagen resultante de la calibración en Matlab.

**Toma de pose**

En la figura 10 se puede ver la imagen resultante de la toma de pose exitosa. Esta misma figura es modificada en tiempo real si es que se activa la opción en la interfaz. Sobre cada indicador encontrado, se coloca un círculo para denotar el centroide encontrado, y tres textos distintos mostrando el número del identificador encontrado, el ángulo y la posición al que se encuentra. La dimensión del ángulo se da en grados y respecto al eje horizontal (derecho). Las dimensiones de las coordenadas se encuentran en centímetros y respecto a la esquina superior izquierda. Esto ya que este se reconoce este como el origen o por defecto. Sin embargo, esta referencia se puede desplazar a cualquier lugar del área de trabajo.

Texto

Descripción generada automáticamente

**Figura 10**. Imagen resultante de la toma de pose en Matlab.

**Tiempo de ejecución**

Se midieron tiempos con distintas cantidades de identificadores. En la figura 11 se muestran los resultados.

Gráfico, Gráfico de líneas

Descripción generada automáticamente**Figura 11**. Gráfica de estadísticas de tiempos de ejecución.

Se puede notar que, conforme al aumentar la cantidad de identificadores, el tiempo de ejecución aumenta 0.0155 seg/identificador aproximadamente. Es importante hacer nota que este aumento puede no ser notable con una baja cantidad de identificadores. Sin embargo, en aplicaciones de robótica de enjambre, la cantidad de agentes no resulta ser el caso, por lo que esta herramienta no tendría un tiempo de respuesta bajo.

También se puede hacer la mención que la desviación estándar se mantiene en un valor bajo incluso al aumentar la cantidad de identificadores. Esto implica que los tiempos de ejecución son precisos. Por lo que con mayor cantidad de identificadores, esta herramienta puede ser utilizada en aplicaciones que no dependan de respuesta rápida, pero si constante.

**Precisión**

En la tabla 1 se pueden ver las medidas teóricas (tomadas con cinta métrica y transportador físicamente) y las medidas tomadas por la herramienta en Matlab. De estas dos se calculó el porcentaje de error que se muestra en la última columna.

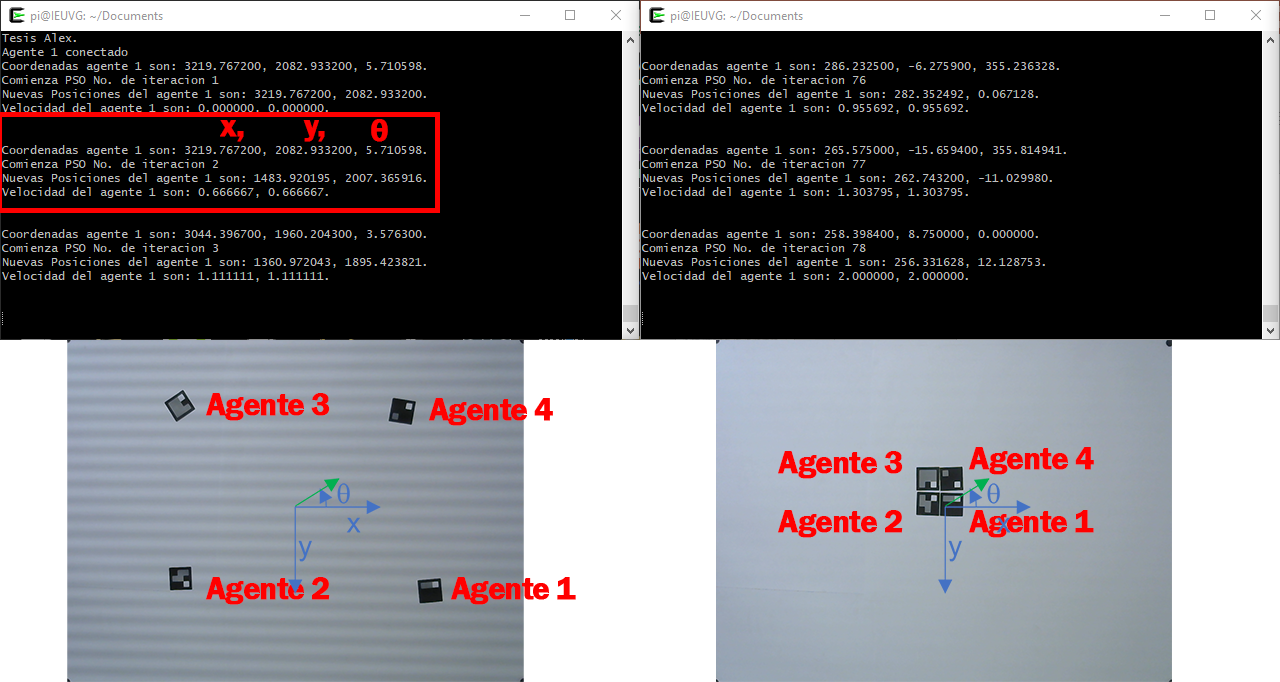
**Tabla 1**. Tabla de resultados de pruebas de precisión en Matlab.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | **Herramienta** | | | **Medidas teóricas** | | | **Porcentaje de error (%)** | | |
|  |  |  |  |  |  |  |  |  |
| 0 | 50.0799 | 43.8914 | 203.2723 | 50.5 | 43.3 | 201.4 | 0.832 | 1.366 | 0.930 |
| 30 | 82.4219 | 17.8128 | 64.7133 | 82.6 | 17.7 | 65 | 0.216 | 0.637 | 0.441 |
| 40 | 84.8310 | 36.2762 | 63.4974 | 84.9 | 36.5 | 63.1 | 0.081 | 0.613 | 0.630 |
| 50 | 26.2968 | 57.0546 | 337.4734 | 26.5 | 56.5 | 335 | 0.767 | 0.982 | 0.738 |
| 255 | 20.5425 | 10.5684 | 45.0343 | 20.7 | 10.7 | 45.3 | 0.761 | 1.230 | 0.587 |

Se puede ver que ningún error sobrepasa el 1.5% por lo que se puede considerar muy confiable el sistema. Es de notar que el error también puede venir de las medidas teóricas debido a los instrumentos de medición o al proceso de medición en sí.

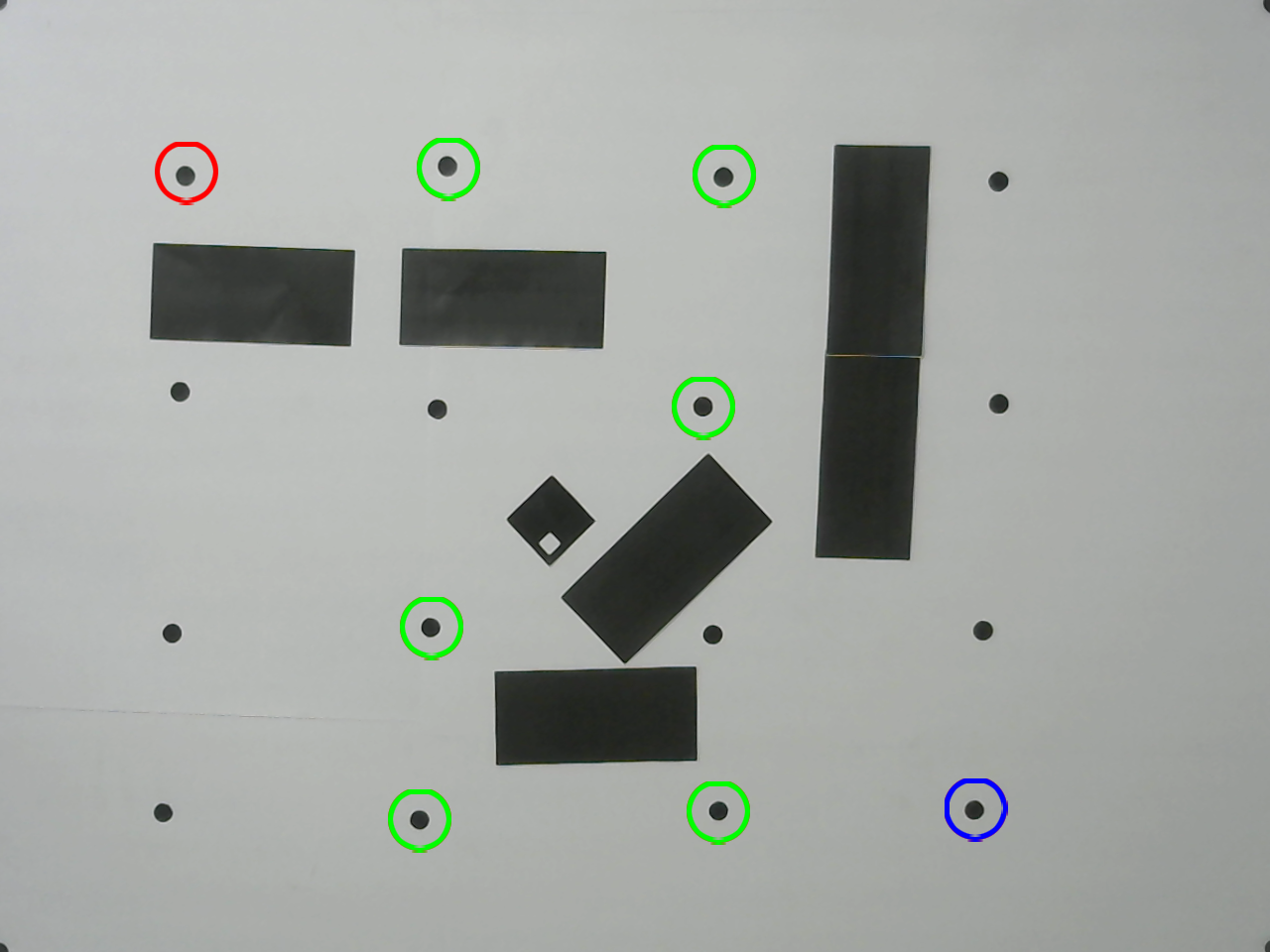
**Validación de concepto**

Como una validación de la herramienta en aplicaciones simples de robótica de enjambre se colaboró con dos trabajos desarrollados en paralelo. El primero es una implementación del algoritmo PSO por Alex Maas. En la figura 12 se muestran capturas de la consola del agente 1 e imágenes del área de trabajo. En cada bloque de código se muestra primero el valor recibido por la herramienta de visión por computadora, luego el número de iteración, seguido de la dirección a seguir y por último las velocidades de las llantas para efectuar dicho movimiento. En este caso, la herramienta se encarga de actualizar al algoritmo de la posición actual para que este realice una nueva iteración. En el lado izquierdo se muestra el estado inicial de los identificadores y del lado derecho se muestra el estado final de los identificadores luego de 78 iteraciones.



**Figura 12**. Consolas de agentes desplegando resultados de comunicación y algoritmos de PSO.

Se puede ver en el marco de referencia que el origen (0,0) se cambió al centro del área de trabajo. Esto es importante ya que en algunos casos de estas aplicaciones se requiere el uso de valores negativos. El segundo trabajo con el que se colaboró es una implementación del algoritmo ACO por Walter Sierra. En la figura 13 se muestra un ejemplo de las pruebas realizadas. El nodo inicial se denota con color rojo, el nodo meta se denota con el color azul y los nodos de trayectoria se denotan con color verde. Se utilizaron rectángulos negros como obstáculos entre los nodos para limitar la movilidad.



**Figura 13**. Ruta óptima seleccionada por el algoritmo.

En este caso, se provee al algoritmo de la posición (x, y) de los nodos respecto a un sistema de referencia global y las conexiones entre ellos. Esto se utiliza para calcular la trayectoria óptima y una vez se cuenta con esta, se provee de la posición del agente para que el algoritmo actualice el controlador del agente en cada iteración.

**Escalamiento de área en Python**

En esta sección se presentarán resultados de las distintas pruebas que se realizaron como validación de la herramienta en Python con la mesa de prueba de la UVG. Esto con el objetivo de validar la herramienta con un área de trabajo mayor. Cabe mencionar que no se realizó ninguna modificación a esta para las pruebas a continuación. En la figura 13 se puede ver el resultado de una calibración con la herramienta de Python. Se puede ver que se logra tener una calibración correcta con los marcadores circulares en las esquinas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 13**. Imagen resultante de calibración.

Se tomaron mediciones de posición y orientación similar a lo hecho en Matlab. En la tabla 3 se presentan los resultados. En estos se ve un error máximo de 2.751% para dimensiones y 1.655% para ángulos. Esto es un poco mayor a los resultados de Matlab, pero, de nuevo, puede deberse a errores ajenos a la herramienta.

**Tabla 3**. Tabla de resultados de pruebas de precisión en Python.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | **Herramienta** | | | **Medidas teóricas** | | | **Porcentaje de error (%)** | | |
|  |  |  |  |  |  |  |  |  |
| 0 | 35.147 | 42.93 | 244 | 34.6 | 42.8 | 244.8 | 1.581 | 0.311 | 0.327 |
| 30 | 67.797 | 17.213 | 25 | 67.5 | 17.5 | 25.3 | 0.440 | 2.751 | 1.186 |
| 40 | 70.293 | 35.316 | 27 | 70.8 | 35.7 | 27.4 | 0.716 | 1.076 | 1.460 |
| 50 | 19.071 | 51.441 | 87 | 19 | 52.2 | 86.5 | 0.374 | 1.454 | 0.578 |
| 255 | 20.669 | 10.09 | 43 | 20.8 | 10.8 | 42.3 | 0.630 | 6.574 | 1.655 |

**Conclusiones**

La herramienta de Python funciona de manera correcta aumentando el área de trabajo en un factor de 3.75 aproximadamente, tanto la calibración como abstracción de pose de los identificadores. Esta presenta una ligera pérdida de precisión, pero manteniéndose en porcentajes pequeños, incluso tomando en consideración la aplicación.

La herramienta desarrollada en Matlab es capaz de realizar las mismas tareas que se tienen disponibles en Python y C++. Esta puede tener una precisión igual o mayor a la herramienta de Python. Y tiene funcionalidades extra que se considera son útiles para aplicaciones de robótica y robótica de enjambre. Por ejemplo, se tiene la posibilidad de cartografiar el terreno en búsqueda de obstáculos que evitar, comunicación UDP y TCP para transferencia de datos y múltiples opciones para configurar el sistema. Todas estas funcionalidades se validaron en aplicaciones simples de robótica de enjambre (algoritmos PSO y ACO) bajo condiciones controladas.

**Agradecimiento**

Agradezco al MSc. Miguel Zea, de la UVG, por el apoyo para los códigos para el procesamiento de imágenes.

**Bibliografía**

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., Marín Jiménez, F. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition. 47, 2280–2292. doi: 10.1016/j.patcog.2014.01.005.

Guangrui, F., Geng, W. (2018). Vision-based autonomous docking and re-charging system for mobile robot in warehouse environment from 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), 79-83.

Guerra, J. (2020) Algoritmos de Visión por Computadora para el Reconocimiento de la Pose de Agentes Empleando Programación Orientada a Objetos y Multihilos.

Tesis de licenciatura en Ingeniería Mecatrónica, Departamento de Electrónica, Mecatrónica y Biomédica, Universidad del Valle de Guatemala.

Lizarazo, J., Ramos, M. (2016). Visión artificial y comunicación en robots cooperativos omnidireccionales. Ingeciencia, 1(1), 5-11.

Millard, A., Redpath, R., Jewers, A., Arndt, C., Joyce, R., Hilder, J., McDaid, L., Halliday, D. (2018). ARDebug: An Augmented Reality Tool for Analysing and Debugging Swarm Robotic Systems. doi: https://doi.org/10.3389/frobt.2018.00087.

Park, F., Lynch, K. (2017). Modern Robotics: Mechanics, Planning, and Control. Cambridge, UK: Cambridge University Press, OCLC: ocn983881868, isbn: 978-1-107-15630-2.

Rezeck, P., Azpurua, H., Chaimowicz, L. (2017). HeRo: An Open Platform for Robotics Research and Education. doi: 10.1109/SBR-LARS-R.2017.8215317.

Rodas, A. (2019) Desarrollo e implementación de algoritmo de visión por computadora en una mesa de pruebas para la experimentación con micro-robots móviles en robótica de enjambre.

Tesis de licenciatura en Ingeniería Mecatrónica, Departamento de Electrónica, Mecatrónica y Biomédica, Universidad del Valle de Guatemala.