

## SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING

A MAJOR PROJECT REPORT ON

**“SIGN LANGUAGE DETECTION USING CONVOLUTION NEURAL NETWORK”**

Submitted in fulfillment of the requirements for the award of the Degree of

# BACHELOR OF TECHNOLOGY IN

## ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

AYUSH RAI R17EC031

SHOURYA JHA R17EC371

T MUSKAN R18EC832

VARSHITHA L R18EC835

Under the guidance of

Prof. MADAN H T

Assistance Professor

School of ECE Reva University

June 2022

Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Bengaluru-560064 [www.reva.edu.in](http://www.reva.edu.in/)

# DECLARATION

We, **Ayush Rai** (R17EC031), students of B.Tech, belongs to School of Electronics and Communication Engineering, REVA University, declare that this Project Report / Dissertation entitled **“Sign Language recognition using Convolution Neural Network”** is the result the of project / dissertation work done by me under the supervision of Prof. Madan H T, Asst. Prof., School of ECE REVA University.

I am submitting this Project Report / Dissertation in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering by the REVA University, Bengaluru during the academic year 2021-2022.

I declare that this project report has been tested for plagiarism and has passed the plagiarism test with the similarity score less than 25% and it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

I further declare that this project / dissertation report or any part of it has not been submitted for award of any other Degree / Diploma of this University or any other University/ Institution.

*(Signature of the Student)*

*Signed by me on 16 May 2022*

*Certified that this project work submitted by Ayush Rai (R17EC031) has been carried out under my / our guidance and the declaration made by the candidate is true to the best of my knowledge.*

*Signature of Guide Signature of Co-Guide, (if any)*

Date:17 May 2022 Date:17 May 2022.

*Signature of Director Date:17 May 2022*

*Official Seal of the School*



**SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING CERTIFICATE**

Certified that the project work entitled “**Sign Language Recognition using Convolution Neural Network”** carried out under my / our guidance by ***Ayush Rai (R17EC031)***, a bonafide student of REVA University during the academic year 2021-2022, is submitting the project report in partial fulfillment for the award of Bachelor **of Technology i**n **Electronics and Communication Engineering** during the academic year **2021**-**2022**. The project report has been tested for plagiarism and has passed the plagiarism test with the similarity score less than 25%. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

**Signature with date Signature with date**

**Prof. Madan H T (Guide)**

**Dr. R.C. Biradar Director**

**Signature with date**

**External Examiner**

**Dr. M Dhanamjaya Vice Chancellor**

**Name of the Examiner with affiliation Signature with Date**

1.

2.

# ACKNOWLEDGEMENTS

We, students at REVA UNIVERSITY, would like to express our gratitude towards our Hon’ble Chancellor, Dr P. Shyama Raju, Hon’ble Vice-Chancellor, Dr S Y Kulkarni and our respected Director, Dr Rajshekhar C. Biradar for providing us with the opportunity to enhance our knowledge in an environment that promotes growth and innovation. We would also like to thank our Project Guide, Prof. Madan H T, for his constant support and guidance and for mentoring us throughout the project. We are also thankful to all faculty members for their kind contributions.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** |  |
| **Acknowledgements** | **i** |
| [**Table of Contents**](#_bookmark0) | [**ii**](#_bookmark0) |
| **List of Figures** | **iii** |
| **List of Tables** | **iv** |
| Chapter 1  **Introduction** | 5 |
| Chapter 2 |  |
| **Literature Survey** | 12 |
| Chapter 3 |  |
| **Proposed Work** | 17 |
| Chapter 4 |  |
| **Result Analysis** | 26 |
| Chapter 5 |  |
| **Conclusion & Future Scope** | 35 |
| 5.1 Conclusion  5.2 Future Scope | 35  36 |
| **References** | **37** |
| **Appendices (if any)** | **39** |
| Conference/ Article papers published | 40 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure no | Figure Name | Page No |
| 1 | Anatomy of Human Ear | 6 |
| 2 | CNN Layers | 9 |
| 3 | Flowchart for sign language recognition | 18 |
| 4 | Architecture of CCN | 19 |
| 5 | Layer of Convolution | 20 |
| 6 | Gestures of each alphabet | 22 |
| 7 | Accuracy graph of the model while training  (LR=0.01, epoch=25) | 23 |
| 8 | Loss graph of the model while training  (LR=0.01, epoch=25) | 24 |
| 9 | Loss graph of the model while training  (LR=0.1, epoch=25) | 24 |
| 10 | Accuracy graph of the model while training  (LR=0.1, epoch=25) | 25 |
| 11 | Loss graph of the model while training  (LR=0.1, epoch=50) | 25 |
| 12 | Accuracy graph of the model while training  (LR=0.1, epoch=50) | 26 |
| 13 | Capturing of dataset | 27 |
| 14 | Predicting the action | 29 |
| 15 | Convolution Method Structure | 30 |
| 16 | Testing the model (C) | 31 |
| 17 | Testing the model (G) | 31 |
| 18 | Testing the model (L) | 32 |
| 19 | Testing the model (P) | 32 |
| 20 | Testing the model (R) | 33 |
| 21 | Testing the model (U) | 33 |
| 22 | Testing the model (V) | 34 |
| 23 | Testing the model (W) | 34 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table no. | Table Name | Page no |
| 1. | Accuracy Comparison Table | 35 |

**Chapter 1**

**INTRODUCTION**

The whole world comprises different types of human beings, depending on different ways to communicate with each other where sign language is an exquisite language used by individuals with hearing loss. How exactly does this ensue?

Humans have five senses in their bodies, one of which is auditory perception (hearing). Hearing allows us to perceive sounds and noises and allows us to interact. A sound source generates sound waves that travel through the air and are caught by the outer ear. The sound waves then travel through a slender passage called the ear canal and reach the eardrum (Tympanic membrane); a flap of skin stretched tight like a drum. The sound waves from the source strike the eardrum, which causes vibrations in the bones of the middle ear, specifically the malleus, incus, and stapes. The anatomy of the human ear is depicted in Figure 1.

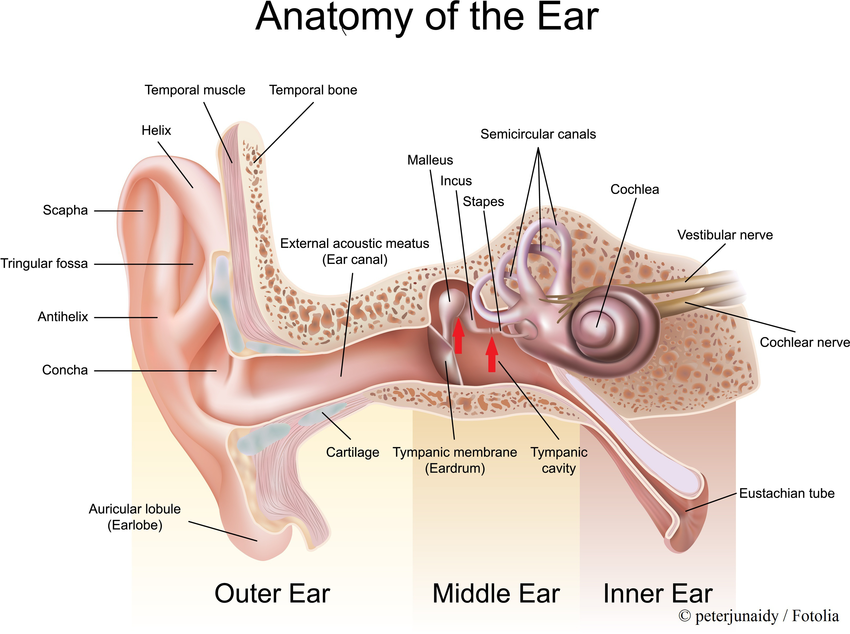


Fig 1: Anatomy of Human Ear

**STRUCTURE OF EAR**

The vibrations are amplified and released into the cochlea by these bones. The cochlea is shaped like a snail and is made of fluid; vibration disturbs the fluid and causes ripples, which affects the hair-like structures called stereocilia that form in the group. Each hair cell on the cochlea's baseline generates an electrical signal in response to vibration effects from stereocilia, which is then carried by auditory nerves to the brain, where the brain predicts, recognizes, and understands the sounds and rhymes. However, a single person can lose their ability to hear sound partially or completely during the transmission of sound waves from ear to brain, which is known as hearing impairment. Conductive hearing loss occurs when sound waves are blocked from reaching the inner ear.

**HEARING LOSS**

Sensorineural hearing loss is caused by an injury to the inner ear or the nerves that transmit sound to the brain and is more likely to be permanent. Hearing loss can occur at any age and can be caused by a variety of factors. People can lose their hearing power unexpectedly because of a virus

or gradually lose their auditory perception due to disease, nerve damage, or injury caused by noise.

Due to a variety of factors, both partial and total hearing loss can occur. Ear infections, fluid build-up in the back of the eardrum, holes in the eardrum, and problems with the middle ear bones can all cause conductive hearing loss. Tumours can also cause conductive hearing loss in rare cases, as they block sound from entering the ear.

**SIGN LANGUAGE**

Hearing loss is the third most common physical disability, following heart disease and arthritis. In this case, an Otoscope, which is a small handheld instrument with a light, is used to check for the presence of any blockage or ear infection, and the results are then sent to otologists. Even though we have the technology to detect hearing loss, sign language is the only way to communicate. In full communication, sign language is the movement of hands to convey meaning. Different countries have their sign language with their alphabets and word sets, such as America's American Sign Language (ASL) and Germany's German Sign Language (GSL). The similarity is that sign languages make use of hand movements. They can also communicate with their facial expressions, eyes, and heads. Because sign language is not universal, it is uncommon among healthy people and takes time to understand and interact with deaf or hard of hearing people. There is a need for hand gesture recognition, so our proposed system suggests a convolutional neural network that uses a live camera feed as input and could eventually achieve the goal of recognizing sign language.

**MOTIVATION**

The Convolution Neural Network method can achieve high accuracy for sign language recognition. The motivation is to achieve comparable results for the machine learning model and to ease the communication gap with the deaf community.

**CONVOLUTION NEURAL NETWORK**

A convolutional neural network (CNN) is a type of artificial neural network that analyses data using the perceptron learning rule and supervised learning. CNN is used in image processing, natural language processing, and other cognitive tasks. A convolutional neural network, like other types of artificial neural networks, has an input layer, an output layer, and several hidden layers. Some of these layers are convolutional, meaning they use a mathematical model to pass results on to subsequent layers. This mimics some of the activities of the human visual cortex. CNN is a basic example of a deep learning algorithm.

**STRUCTURE OF CNN**

1. Input Layers: This is the layer where our model receives input. This layer has the same number of neurons as our data's total number of features

2. Hidden Layer: The input from the Input Layer is routed through this layer. Depending on our model and data size, there could be many hidden layers. Each hidden layer can have a different number of neurons, which should be greater than the number of features. The output of each layer is computed by matrix multiplication of the previous layer's output with learnable weights of that layer, followed by the addition of learnable biases and an activation function, which makes the network nonlinear.

3. Output Layer: The hidden layer's output is then fed into a logistic function, such as a sigmoidal or softmax layer, which converts the output of the hidden layer.

Graphical user interface, diagram, text

Description automatically generated

Fig:2 CNN

**Data Preprocessing**

The first step in developing a working deep learning model is data preprocessing. It is used to convert raw data into a usable and efficient format.

Raw images are hand sign images captured with a camera for use in implementing the proposed system.

The photographs were taken in the following settings:

(i) From various angles

(ii) With varying lighting conditions

(iii) With good quality and focus

(iv) With varying object size and distance

The goal of creating raw images is to build a dataset for training and testing.

Gestures are physical expressions used to convey information. They include a finger, hand, face, and head movements, but hand gestures are the most common. Nowadays, digital cameras are built into desktops, laptops, mobile phones, and other handheld devices. Because these devices have enough memory and computing power, gestures can be used to interact with them. Hands have a high degree of freedom (DOF), making it difficult to develop a system that can provide real-time performance and adequate recognition accuracy.

Sign Language has a well-defined set of signs that can be used to train computers to understand what mute and hearing-impaired people want to say, and that message can then be converted to text or speech so that others can understand it as a result, sign language Hand gesture recognition is one of the best applications for which it can be used. Human-computer interaction (HCI) is also a growing field. It is friendly interaction between a computer and a human in which the computer understands human language. Hand gestures are important in this situation because they are intuitive, natural, and user-friendly. Simultaneously, the system should be able to handle complex backgrounds as well as changes in hand appearance. HCI is used in a variety of applications, including video games, multimedia application control, and machinery controls.

**COMPONENTS OF CNN ARCHITECTURE**

The goal of CNN is to learn higher-order features in data using convolutions. The CNN architecture is effective for object recognition, including image recognition. They are capable of recognizing individuals, faces, street signs, and other aspects of visual data. There are several CNN variants, but each one is based on the pattern of layers present.

CNN architecture is made up of various components such as different types of layers and activation functions. The following section discusses the purpose and operation of some commonly used layers.

The layer of convolution the convolutional layer is the foundation of CNN architecture.

Convolutional layers (Conv) are used to modify the input data. Convolutional layers (Conv) use a patch of neurons connected locally to the previous layer to modify the input data. The layer between the region of neurons in the input layer and the weights to which they are locally connected in the output layer will compute the dot product.

The convolution operation takes input, applies a convolution filter or kernel, and returns a feature map as output. This operation depicts the sliding of the kernel across the input data, resulting in the convoluted output data.

Data Collection The process of capturing visual images, such as those of a physical scene. Preprocessing: The goal of preprocessing is to optimise the input image data for further processing by suppressing unnecessary noise or enhancing essential image features.

Feature Learning: Feature learning builds its derived features from the original evaluated data to be descriptive and to facilitate subsequent learning.

Generalization and the Recent and Innovative Trends in Computing and Communication is an international journal that focuses on the most recent and innovative trends in computing and communication.

Classification is a process related to categorization, which is the process of distinguishing between concepts and objects.

Finding trends and regularities in results is the goal of recognition. Pattern recognition is the process of categorising input data into objects or classes based on key features.

**LAYERS OF CONVOLUTION**

The Convolution layer is frequently the first layer in a neural network. It receives the picture (a matrix of pixel values). Assume the input matrix is read from the top-left corner of the image. Following that, the programme chooses a smaller matrix, known as a filter, to place there (or neuron, or core). The filter then generates convolution as it passes through the input image. The filter's job is to multiply its values by the pixel values from which they are derived. Several of these multiplications are merged. Following that, a single sum is gathered. After running the filter over all locations, a matrix is produced, but it is smaller than the input matrix.

Convolution is a special operation that extracts multiple characteristics from data. In the first step, it extracts low-level features like edges and corners. Then, at a higher level, upper-level layers extract functionality. The input has dimensions of N x N x D and is convolved with the H kernels, each of which has dimensions of k x k x D. One output feature is produced when one input is convoluted with one kernel, and H features are produced when H kernels are convoluted independently. Starting at the top-left corner of the input, each kernel is shifted from left to right.

If a kernel enters the top-right corner, it is shifted downward one element before being moved one element at a time from left to right. Repeat the procedure until the kernel reaches the bottom-right corner of the screen. Convolution is a mathematical procedure that takes two inputs, such as an image matrix and a filter or kernel. The image matrix is a digital representation of image pixels, and the filter matrix is another matrix used to process the image matrix. Because the kernel is much smaller than the image, we can process any aspect of it. In this layer, apply a filter to the image matrix. Depending on the features to be removed, any number of convolution layers can be added.

The convolution function requires four arguments:

one for the number of filters,

one for the structure of each filter,

one for the input shape,

and one for the image form and resolution.

The fourth argument specifies the triggering function to be used. Which neuron can fire next is determined by the activation mechanism.

The layer of Pooling: A pooling layer is another CNN building block. The primary goal of pooling is to reduce the size of the image matrix. Its purpose is to make it easier to compute the large image matrix obtained from the convolution operation. The pooling layer operates independently on each feature map. To reduce the complexity of processing these matrices, the convolution operation must be followed by a pooling operation on the resultant matrix. A pooling operation's primary goal is to reduce the size of the matrix by downsampling it.

The layer of Flattening: The process of flattening a pooled function map into a single column that can be transferred to a completely connected layer. The process of transforming data into a one-dimensional sequence so that it can be passed on to the next layer is known as flattening. Flatten the contribution of the convolutional layers to create a single long function vector. It is connected to the final classification model, which is a completely connected sheet.

**Chapter 2**

**LITERATURE SURVEY**

Izzah [1] proposed a technique where 380 images were converted from RGB to YCbCr to reserve luminance. Further to pull out the features Generic Fourier Descriptor is applied and filtering using a mask is done. Ultimately for testing purposes, the pre-processed dataset is fed to Support Vector Machine, Principal component analysis with a performance percentage of 81.39 and 61.36.

The work presented in [2] uses 2D wavelet transform to pull out the features and then uses multiclass SVM for classification. Their methodology involves image acquisition, processing of the image, feature extraction and then classification. The image was recorded and altered to black and white images and rescaled. Wavelet transformation was applied to these pre-processed images. The feature vector obtained was given as input to SVM. They achieved an accuracy of 94% after training and testing over 350 samples.

The work presented in [3] proposes a model where initially pre-processing is done with 720 gesture images by resizing and global thresholding method and maintain the structural properties by canny edge detection, further usage of the histogram was seen to extract the features and eventually the results were observed using support vector machine with an accuracy of 93.75%.

In the work presented in [4], the photographs are initially captured, and image pre-processing is done to eliminate the unwanted noise and regulate the brightness. Image segmentation is carried out to spot the hand objects and edges of images. Then apply image analysis and convexity algorithm to bring out the outline position of the hand. This method can only be used to detect numbers and hence becomes a drawback.

Sign in the form of gestures is given as an input to the system proposed in [5] segmentation phase is performed based on the skin colour to identify the shape of the sign. The region identified is then modified into a binary image on which the Euclidean distance transformation was applied at a later stage. On the distance transformed image, Row and column projection is applied. For extracting the features central moments together with HU’s moments are used. For classification, neural networks and SVM are used. The average accuracy for 13 feature sets obtained is 92.12%.

In the work presented in [6], Identification of Sign language was done using a gyroscope, accelerometer, and a hardware module to control the sensors. An improved K- mean algorithm has achieved an average accuracy of 96.55 %. The static gestures were recognized effectively as compared to continuous gestures.

In the work presented in [7], recognition is done by converting the hand moves into an oral language. Flex sensors are placed inside the gloves on every finger which helps to get data employing various methods, the K-Nearest Neighbours (KNN) is used as a classifier. They obtained an average accuracy of 87.8% in drop3.

The work presented in [8] uses high-level CNN to identify the static sign images and processes them. RGB dataset is used. It classifies ten digits, 23 alphabets of English and 67 commonly used words. Different optimizers are used compared. 99.17% of training accuracy and 98.80 % of validation accuracy were achieved.

Shreyashi Narayan Sawant [9] suggests a technique using MATLAB. Otsu algorithm has been used as a part of prepossessing of the image where the resolution of the image taken is 380 x 420 pixels. Principle component analysis has been implemented to find out more descriptive data from the image by using Eigenvectors and ultimately recognizing the hand gestures.

The work presented in [10] describes the usage of simple CNN architecture to recognize the sign language. The datasets used were SIBI (Indonesian Language Signal System) and ASL (American Sign Language) to test and train the model. Using the above method, they attained an accuracy of 96.82% and this was compared with AlexNet.

Alina Kuznetsova [11] has proposed a method using a publicly available dataset on which an ensemble of shape function (ESF) descriptor is applied and applied to real-time data. And eventually using the multi-layered random forest to differentiate between clusters using features decided by the algorithm.

The work presented in [12] focuses on vision-based techniques to identify using principal component analysis. At first, the hand region is divided by using the skin colour model in the YCbCr colour space. Next, we apply thresholding to split the background and at last template-based matching is developed using principal component analysis. They achieved an accuracy of 91.25% after testing with 20 images for each gesture.

In the work presented in [13], the objective is from sign language videos to generate spoken language translations, considering grammar and the order of words different word orders and grammar. To evaluate the approach, the first continuous sign language translation dataset, PHOENIX14T has been collected.

An identifying method using the K Nearest Neighbour classifier was proposed in [14]. According to the study they made, there was a change in the accuracy when the pattern is represented by full dimensions compared to when it was represented by PCA reduced dimension feature. From their study, we were able to conclude that the KNN classifier is more preferred with full dimension in place of reduced dimension.

In the work presented in [15], SubUNets a unique approach to unravelling simultaneous alignment and recognising problems named “Sequence-to-sequence” learning are being employed. A framework for modelling the subunits of learning. The matter has been decomposed into a specialized unit called subunits.

The work presented in [16] collected the dataset using Microsoft Kinect. The depth from the dataset along with the segmented static model gave an accuracy of 98.81 % and with the dynamic model, 99.08 % accuracy was recorded. The model could classify the data into 36 different classes.

Adithya V, Vinod P. R, and Usha Gopalakrishnan [17] presented in their work, Artificial Neural Network-Based Method for Indian Sign Language Recognition. For segmentation, RGB colour spaces are transformed into YCbCr colour space, and the pixel of skin colour in the input images are identified by applying a thresholding technique based on the distribution of the skin colour in the YCbCr colour space. The result of segmentation produces a binary image in which the skin pixels are white and the background is black colour. For feature extraction distance transformation, row and column projection applied on distance transformed image, Fourier descriptor is applied on row and column projected image. Central moments are calculated.

Anchal Sood and Anju Mishra [18] have presented in their work, AAWAAZ: A Communication System for Deaf and Dumb. For segmentation, they have used Hue-Saturation-Value (HSV) histogram. For the extraction of the features, the Harris algorithm is used. For Feature matching and recognition, the dataset already has the feature extracted from the standard image and is stored as an N\*2 matrix mat file. The matrix value of this image query is then matched with each of those in the data set of every image and the minimum distance between the matched features is calculated to get the desired result.

Shreyashi Narayan Sawant and M. S. Kumbhar [19] have presented in their work, Real-Time Sign Language Recognition using PCA. Data acquisition: 260 images are used 10 images of each 26 signs. The algorithm used for segmentation purposes is Otsu’s method. Noise is removed from the images using morphological filtering techniques to get the contour. Here the main feature used is the principal component. In the phase of recognition, normalization is done for the subject gesture concerning the average gesture and then it is projected onto the gesture space using the eigenvector matrix. At last, Euclidean distance is calculated between this projection and all the other known projections. The one being the minimum value of these comparisons is chosen for recognition during the training phase. The recognized sign is converted to appropriate text and voice.

Suriya M, Sathyapriya N, Srinithi M, and Yesodha V [20] presented in their work, Survey on Real-Time Sign Language Recognition System: An LDA Approach. The algorithm used for segmentation purposes is Otsu’s method. Here the main feature used is the principal component. KNN classifiers are used for classification and Similarity measures like Euclidean distance, City Block Metric, Cosine Similarity and Correlation are made used to evaluate the performance of classifiers.

Madhuri Sharma, Ranjna Pal and Ashok Kumar Sahoo [21] presented in their work, Indian Sign Language Recognition Using Neural Networks and KNN Classifiers. In their work first derivative Sobel edge detector method is used as it can compute gradient using the discrete difference between rows and columns of 3×3 neighbours. Feature extraction techniques used are direct pixel value and hierarchical centroid. For classification 2 classifiers are used that are: K-Nearest Neighbour (KNN), a neural network pattern recognition tool.

In Jing-hao Sun[22] The human hand was separated from the complex context, and the camshaft algorithm was used to detect real-time hand gestures. Then, using a convolutional neural network, the region of hand movements that were observed in real-time is recognised, resulting in the identification of 10 common digits. The proposed system has a dataset of a total of 1600 pictures for the training dataset, 4000 hand gestures, and 400 images for each type. This experi- ment shan ows and accuracy is about 98.3 per cent.

Hasan[23] used scaled normalisation to recognise gestures usingbrightnesss factor matching. With a black background, thresholding techniques are used for segmenting the input im- ages. At the X and Y axis origins, the coordinates of any segmented image are shifted to match the centroid of the hand unit. and the image’s centre mass is determined.

Using a boundary histogram, Wysocki et al[24] provided rotation invariant postures. The input image was captured with a camera, a filter for skin colour detection was applied, and then a clustering procedure was used to find the borderline of each category in the pooling image using a standard contour tracking algorithm. Grids were created from the picture, and the boundaries were normalised.

Geethu Nath and Arun C.S. [25] developed an ASL sym- bol recognition system based on the ARM CORTEX A8 processor. The machine recognizes numbers using the Jarvis algorithm and alphabets using the template matching algorithm.

Using Principal Component Analysis (PCA) and various distance classifiers, Kumud Tripathi [26] developed a framework for recognising continuous ISLgesturess. The features from the keyframes are extracted from the own data set using Orientation Histogram and proper-videos input to the device.

Noor Tubaiz [27] proposed using the Modified k-Nearest Neighbor (MKNN) approach to classify sequential data. Data gloves are used to detect hand movements. To supplement the raw data,window-based statistical features are calculated from previous raw feature vectors and future raw feature vectors. To recognise terms in ISL, the proposed framework was developed using novel techniques based on existing systems (ISL). Describe an approach for a continuous sign language recognition method (B. Bauer et al.). It is a framework that depends on continuous hidden Markov model images (HMM). It employs German sign language (GSL). Feature vectors that represent manual signs are fed into the dev.

**Chapter 3**

**PROPOSED WORK**

The proposed system has four major steps that guide us to achieve the objective. The first step is the generation of the dataset using a capturing device followed by building the model and training our machine learning model. Finally testing the accuracy of the trained machine learning model completes the process. Figure 2 explains the workflow followed to reach the objective. The dataset is generated by using an image capturing device. The dataset is then divided into validation and training sets. Then the model that needs to be trained is fed with a training dataset. We have used a Convolutional Neural Network to make the predictions for the sign language. This model after training is then tested against the dataset which is created by performing a split on the overall dataset. At last, the model is given images from the live camera feed and the predicted alphabets are displayed as output.

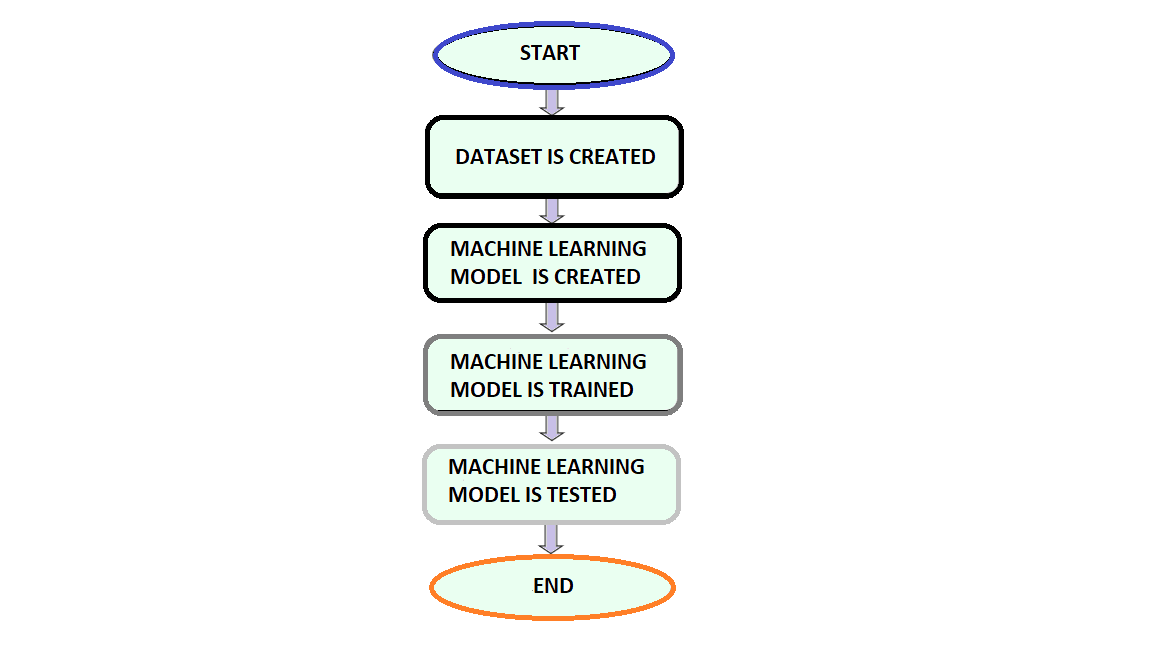


Fig 3: Flowchart for sign language recognition

**ARCHITECTURE OF CNN**

We have used Convolutional Neural Network for the prediction of the Sign Language. The architecture of the Convolutional Neural Network is shown in figure 2.

Diagram

Description automatically generated

Fig 4: Architecture of CNN

* CNN’s effectively reduce the number of parameters without losing the quality of models. Images have high dimensionality (as each pixel is considered a feature) which suits the above-described abilities of CNNs.
* CNN retains the 2D spatial form of images.
* All the layers of a CNN have multiple convolutional filters working and scanning the complete feature matrix and carrying out the dimensionality reduction. This enables CNN to be a very apt and fit network for image classifications and processing.

We can train our model by invoking the fit generator() function, which accepts both our training and validation images as input for training and validation. We should also specify epochs and steps per epoch. The most common method for determining steps per epoch is the number of train images/batch size. In our case, it is approximately 800. The trained model is saved as a file in the format of an h5 file.

The extraction of suitable distinct features that efficiently characterise the variations in signs is an important problem when designing a sign language recognition system.

Because pattern recognition techniques are rarely independent of the problem domain, it is believed that a proper feature selection has a significant impact on classification performance.

Three issues must be addressed during feature extraction. The first issue is analysing the feature extraction region. While some authors adhere to the conventional framework of dividing images into small intervals known as pixels from which a local feature vector is extracted, other researchers prefer to extract global statics from the entire speech utterance. Another critical question is which feature types are best suited to this task.

LayerS of Convolution

Convolution is a specialised operation that extracts various features from the input. The first step is to extract low-level features such as edges and corners. The features are then extracted by higher-level layers. For the 3D convolution process in CNNs. The input is N x N x D in size and is convolved with the H kernels, each of which is k x k x D in size.

Diagram

Description automatically generated

Fig 5: Convolution Method Structure

**GENERATION OF DATASET**

The images are captured using an image capturing device. A window of 64 x 64 pixels is drawn while capturing which contains the gesture image. We have used an HSV colour model for the images. After the images are captured, they are saved locally in the gesture name folder respectively. The images are already in the size of 64 x 64 pixels as it was normalized while capturing. In this way, the quality of the images is also ensured. Figure 3 displays each gesture from our dataset.  Dependencies such as cv2, os, and time have been imported for data generation. The dependency os is used to assist with working with file paths and it provides functions for interacting with operating systems. Time in Python can be represented in code in a variety of ways, including objects, numbers, and strings, thanks to the time module. It can be used to measure code efficiency or wait during code execution in addition to representing time. It is used here to insert breaks between image captures to allow for hand movements.

**LABELLING OF IMAGES**

After all of the images have been captured, they are labelled one by one with the labelling package. When you save a labelled image, an XML file is created. The XML files for all of the images are now available after they have been labelled. This is used to generate TF (TensorFlow) records. The images and their XML files are then separated into training and validation data. Each alphabet has 1750 images captured for the training dataset and 250 images captured for the validation dataset.

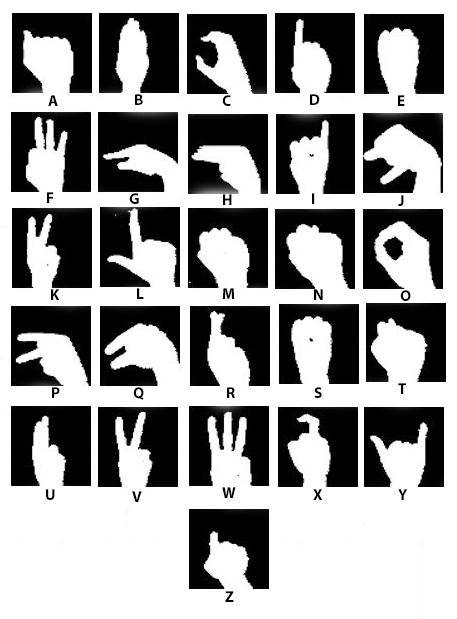


Fig 6: Gestures for each alphabet

1. **Building Our Machine Learning Model**

We have used Convolutional Neural Network for the prediction of the Sign Language. The architecture of the Convolutional Neural Network is shown in figure 4. A sequential layer is initialized using the Keras library and we have added some layers for increasing the performance of the Convolutional Neural Network. A convolution layer that takes an image of 64 x 64 as input is added. The output from this layer is passed on to a MaxPooling Layer. This pair of Convolution and MaxPooling layers is repeated. After three pairs of Convolution and MaxPooling layers, a Flatten layer is added. The Flattening layer converts the 2 D feature matrix into a one-dimensional feature vector. A Dropout layer is added to overcome the overfitting of the model. And finally, a dense layer is added which constitutes the Convolutional Neural Network. Standard gradient descent is used as an optimizer for increasing the performance of the Convolutional Neural Network.

Keras has a useful API which makes us easier to define the layers of our neural network. Here the input shape is 64,64 which is our image size and 3 represents colour channel RGB. If it is a grayscale image, we should specify it as 1.

Conv2D(): Neural networks apply a filter to an input image to create a feature map that summarizes the presence of detected features in the input. In our case, there are 32,64,128 and 128 filters or kernels in respective layers and the size of the filters are 3X3 with activation functions as relu.

MaxPool2D(): Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after the max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

Flatten(): This method converts the multi-dimensional image data array to a 1D array.

**TRAINING OUR MACHINE LEARNING MODEL**

The training dataset is used to perform the training of our model. The proposed model is trained for 40 epochs with a step size of 800 per epoch. The validation step is set to 6500. The trained model is saved as a file in the format of an h5 file. This is done so that the model can be used on different systems. The accuracy of the training set was found to be 96.41% with an error value of 0.14. These values are justified in figure 5.

Learning rate= 0.01, epoch= 25

A picture containing graphical user interface

Description automatically generated

Fig 7: Accuracy graph of the model while training(LR=0.01, epoch=25)

A picture containing shape

Description automatically generated

Fig 8: Loss graph of the model while training(LR=0.01, epoch=25)

Learning rate= 0.1, epoch= 25

Shape

Description automatically generated with low confidence

Fig 9: Loss graph of the model while training(LR=0.1, epoch=25)

A picture containing graphical user interface

Description automatically generated

Fig 10: Accuracy graph of the model while training(LR=0.1, epoch=25)

Learning rate= 0.01, epoch= 50

Shape

Description automatically generated with medium confidence

Fig 7: Loss graph of the model while training(LR=0.01, epoch=50)

A picture containing graphical user interface

Description automatically generated

Fig 8: Accuracy graph of the model while training(LR=0.01, epoch=50)

**Testing our Machine Learning Model**

The test data is fed into the model to see its performance of the model. The model performs well on the test data as well with an accuracy of 94.21%.

After the model has been built and the data has passed the training phase, the testing phase begins. At this point, we attempted to classify test images using two different models with different filters. The number of training epochs to be used is a common issue encountered when training neural networks. Excessive epochs can result in overfitting of the training dataset, whereas insufficient epochs can result in an underfit model. So we experiment with different numbers of epochs to see when the model's performance stops improving. The number of epochs is determined by the best model performance.

Finally, the live camera feed is connected to the model for real-time detection of the Sign Language. The HSV values for the camera feed are set using a track bar for all the values. After the values are fine-tuned, the model predicts the Sign for the indicated gesture in the camera.

**Chapter 4**

**RESULT ANALYSIS**

We propose a model that recognizes the Sign language using a convolutional neural network. The input is given as a live image feed. The proposed model is trained on a labelled dataset that contains images of all 26 alphabets. Our model can identify all 26 alphabets with considerable accuracy.

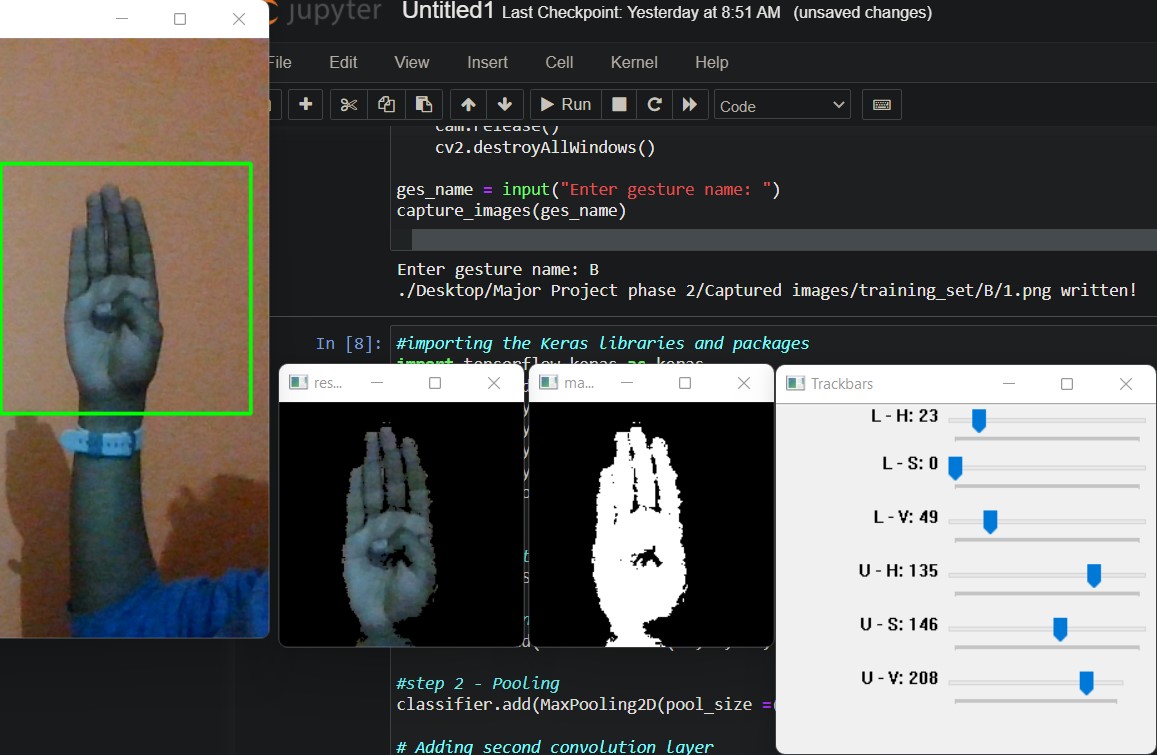


Fig 11: Capturing of Dataset

The images are captured by pressing the trigger key(C). The image is normalized and masked in the process as shown in Figure 6.

**DATA MODEL**

![Table

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDcRXhpZgAATU0AKgAAAAgABAE7AAIAAAAGAAAISodpAAQAAAABAAAIUJydAAEAAAAMAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFJFRVNVAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAMzOAAAkpIAAgAAAAMzOAAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMTowNDowOCAwNToxNDozNAAyMDIxOjA0OjA4IDA1OjE0OjM0AAAAUgBFAEUAUwBVAAAA/+ELGGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjEtMDQtMDhUMDU6MTQ6MzQuMzc1PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPlJFRVNVPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAk8CngMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiuR+Iwv5tDsLPSL+4sLq81KCFJ7eQow5LEZ9Dt5B6jI71ysvjK5/4SLTvFd7cXtrpUdndwS6Ysh8tpIYfMlYr0Z1cNGPTYSOtTzKzb6f5J/qHVLv/nY9YqK6urextJbq9nit7eFC8s0zhERR1JJ4AHqa4tvEniXSYdI1bX10s6bqlzDbvaW0UizWRmIEZMrORLhiqt8idcjpg4vjnV9c8RfDzxjc6bPp9no9nDd2hjmtXmnufKUrIwcSII/mDKAVf7u7vgXZ3t/Wlv8ANBGzt5/rf/JnqQYMoZSCCMgjvVLVdb0rQrVbnW9Ts9NgZ9iy3lwsKlsZwCxAzgHj2rjNa8fR6fryaBBr2gaFJb2MVzNda2+4SF8hY44/NjJOFLFt3GVGDnI09M8Sp4t+F15qyLErPbXUUnkSeZGXj3ozI38SErkH0IqKj5YSkun6Dh7zSfX9Tq4J4rm3jntpUmhlUPHJGwZXUjIII4II70+sTwV/yIOgf9g23/8ARa0/wxc6/daVLJ4qsbaxvRcyqkVtJvUwhyI2JyeSuCefwHQaSXLNx7GcJc0FLubFZ1z4i0WzluY7vWLCCS1aNbhZbpFMJk/1YYE/KW7Z69q0a5bU/h7o2rXuq3Vy1ysmqzWc1x5bKPmtmDR4+XODjnOfbFT1L6M6miuM8Z+L7jQNc03T/wC0NN0O1vI5HOratA8lv5i4xDw8aqxG5ss44XABzweJ/G48P2OhwTajo1reawxVb+7m22cSKm5peWUuD8oVQwyXHzcZJ0uHWx2dFed2PxMB0TXJDcaXrl3pc0EMNxpUu23u2nIWIfek8s7jtb5mxjPfA0rbWPE2n+NrPRtcu9JvIbvTri7DWdjJA0bxNEMZaZwwPmHsOlH9fhf8g/r8bHZUVxNr4z1GfwH4W1t4bUXOsXNpFcIEbYolbDbRuyCO2SfxrI1j4orDqGtCx17wxp6aNM0AsdTuALi/dFBcKRIvlDJKAlJMkE4xxSutfL9Lf5odtbf11/yZ6bRXDL4r13XPFFvpnhxNPtrS50WDVBeXsTytCZHYbDGrrvyAP4l24J+bgVz/AIs1fXPE3wii1FLjT7Mrex295EbR5fMlivli3xt5q7F3Ju2kMcHGQeaqzW/e342ErN/d+Kv+R6zVay1Gy1JJX068t7tYZWhlMEquEkX7yHB4YdweRVOzt/ECadcpf6nps96wP2eaDTpIo4zjjfGZ2L8+jL/WsnwBPLLp+rx3NrpsE9vq1xDK+m2hto52G3MhQsx3HPJLHNJb28r/AJf5ibsl62/B/wCR1dZw8Q6K2tnRl1ewOqKMmxFynngY3f6vO7pz06Vx138QJNO8VWNk3iDwzqS3eoLYy6XZPi8tCxIDFvNbftYAMPLTrnjGDreJf+R/8G/9d7v/ANJ2pX0v/X9ag3ZtHWUUVR1bSk1e1W3mu7y2hDhpBZ3DQNIAD8pkXDqM4Pysp4xnBILGWri4htLaW4upY4YIULySyMFVFAySSeAAO9VrTWtLv7hYLHUrO5maBblY4Z1djE33ZMA52ns3Q1xngoJ4j0rxVpovrrUPDj3clnYXU8zTM8RiAlCyyZMiiQuAxLdCMkAV0Gh+DNM8P6kl9YtOZU0yDTAHZdvlQlipwAPm+Y5PTgcCn6+X5P8A4APTTr/wV/wToKKKKQBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBk67obazPpMi3Rtxp1+l4VCbvOCo67Oox9/OeenSq3iXwpB4mvNNe8lAt7QziaAx7vtEcsLRMmc/L94HOD0x3rfopWVrDu73/AK/rU5C28HapKdNtdf16LUdM0uVJreFLHyppnj/1Rnk8xg+3hvlRMsAeBwaeqfDu9vtK1zRbXxB9j0bWJJp2gWz3TRSS8uBJvAMZbLFdob5iN4Fd3RT3/r+uyEtNv6/q5zN54X1CHXP7Y8OarBY3k1rHaXSXdmbmGdYySjbVkjZWG5hndghuQcAizquma/qHhz7DDq2mw3csbR3Vw2myPG6sCPkj88FDyOrNW7RSkuZNPqC0d0YvhXSdS0PQrfTdU1C1vxaxJDBJbWbW+EVQo3BpHyeOowPaneGNFvdC0qW11LWbnWZXuZZhcXIwyq7lljHJ4UHA/QAYA2KKptt3YkklZBRRRSGZGtWOs3eBo+pWFtE6FJob/TzdI49gssZGckHJYHjAHOcaHwAum6BolnoWpNaX2huz2t1NbrIjb8+YjxKV+QhjgKVIwuDxz2FFAbnN3/hzUte8L3mleIdUtZJ5XWS3utPsmg8hkZXRtryybiHUHqARxjvWBHYajB8YdHbXNSjv7ibRb1T9ntzBDGolg+6hdyCc8ksc4GMAYr0OihWvf1/FNB0t/W5wNt8OdRistF02bxKW0zQruK4sreKxCNIsbZVZ3LneQvAKhBnkhuMav/CNa1puqX9x4Y1qytLbUJvtM1tf6c1yEmIwzIyTRkBsAkHdzkgjOK6mij+vy/yQPV3f9b/5sxodBaPxlL4ge6DNLp0dkYBFgApI7785PXfjGOMday08CR/8IBd+GJdQc/aJ57hLuOIK0TvcNOhCkkHaxH129s11tFHS39dx31v/AForfkZFpbeIo9LuI77VtMnvj/qLiLTJI404/jjM7F/wZay/DPhfW9Fl1FNS1qxvbXUZ5bmVLbTXt5Fkkxna5ncBRjptz711dFKybv8AIXb7zgbD4dalaaToukSeIoW0zRLuGe1hh00RvKsbZCzOXO5v9pAnOSQc4GjrnhrxHqniWx1Sz13S7WPTpJHtYZdJklPzx7GDsLhd3UkYC/jXW0U/+H/L/IVkYN1438LaXctZav4p0S1vYcLNFNfRRMrYzyjNlfXBrN8QRx/EbwvJZeDvF1jFA0wS8uLPF2JExloSY5FKbsjJDA46YzmuwopNJqzGtNjF8N6Vqej2Qsr660uS0hjWO1h07TXtFhUcYw00mR0xjGPetqiiqbuK1gooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoqhq02rwwIdCsbK8lLYdLy8e2VVx1BWKTJz2wPrWUL7xv38PeH/APwfT/8AyHQB0lFc4t942LDf4f0ALnkjXZiQP/ASkN943zx4e8P4/wCw9P8A/IdAHSUVzZvvG2ePD2gHjvrs3/yHR9u8b4H/ABT2gZ7/APE+m/8AkOgDpKK5v7d422n/AIp7QM54H9uzf/IdKl741LfP4f0EDB5GuzHnHH/Lp6/5NAHR0Vzf27xv/wBC94f/APB9P/8AIdBvvG+ePD3h/wD8H0//AMh0AdJRXONfeNs/L4f0AjA667MOe/8Ay6Un27xvj/kXvD+f+w9P/wDIdAHSUVzgvvGuw58P6Bu4wP7dmwfx+yUgvvG2efD2gDjtrs3/AMh0AdJRXN/bvG//AEL3h/8A8H0//wAh0G+8b5OPD3h/Hb/ifTf/ACHQB0lFc2b7xtnjw9oB4767N/8AIdH27xvj/kXvD+f+w9P/APIdAHSUVzgvvGuw58P6Bu4wP7dmwfx+yUgvvG2efD2gDjtrs3/yHQB0lFc39u8b/wDQveH/APwfT/8AyHQb7xvk48PeH8dv+J9N/wDIdAHSUVzZvvG2ePD2gHjvrs3/AMh0pvfGvlgjw/oO/JyP7dmwBxjn7J9f89ADo6K5v7d43/6F7w//AOD6b/5DpVvvG2fm8P6ABg9NdmPPb/l0oA6Oiub+3eN/+he8P/8Ag+n/APkOg33jfPHh7w/j/sPT/wDyHQB0lFc41941GNnh/QDwM512Yc/+AlJ9u8bbR/xT2gZzyP7dm/8AkOgDpKK5sX3jfBz4e0DPb/ifTf8AyHQL7xtnnw9oA47a7N/8h0AdJRXN/bvG/wD0L3h//wAH0/8A8h0r3vjUSMI/D+gsmTtLa7MCR7j7IcUAdHRXN/bvG/bw94f/APB9N/8AIdL9u8bbRjw/oG7PI/t2bGP/AAE+tAHR0Vzf27xv38PeH/b/AIn0/wD8h0q33jUt8/h/QAPUa7Mf/bSgDo6K5v7d43/6F7w//wCD6f8A+Q6DfeNtx2+HtAIzwTrsw/8AbOgDpKK5s33jfjHh7QPf/ifTf/IdH27xttH/ABT2gZzyP7dm/wDkOgDpKK5xb3xrtbd4f0EHHyga7McnPf8A0TjjPrSC+8b558PeH8f9h6b/AOQ6AOkorm/t3jf/AKF7w/8A+D6f/wCQ6Vr7xsGOzw/oBXPBOuzAkf8AgJQB0dFc39u8b9vD3h/3/wCJ9P8A/IdL9u8a7M/8I/oG7PT+3ZsY+v2SgDo6K5sX3jbBz4e0Dpx/xPZv/kOgX3jfPPh7QAPbXpv/AJDoA6Siub+3eN/+he8P/wDg+n/+Q6DfeNtx2+HtAIzwTrsw/wDbOgDpKK5xr3xrtXb4f0EnHzA67MMHPb/ROeMelJ9u8b4/5F7w/n/sPTf/ACHQB0lFc4L7xtg58P6ADjjGuzcn/wABKQX3jfPPh7w/j/sPT/8AyHQB0lFc39u8b/8AQveH/wDwfT//ACHSvfeNQ52eH9AK9iddmBP4fZKAOjormzfeNsDHh7QOnP8AxPZv/kOj7d43x/yL2gZ/7D03/wAh0AdJRXNi+8bYOfD2gdOP+J7N/wDIdAvvG+efD3h/H/Yen/8AkOgDpKK5v7d43/6F7w//AOD6f/5DpXvvGoc7PD+gFexOuzAn8PslAHR0VzZvvG2Bjw9oHTn/AIns3/yHR9u8b4/5F7QM/wDYem/+Q6AOkormxfeNsHPh7QOnH/E9m/8AkOlS98amRRJ4f0FUyNxXXZiQPYfZBmgDo6K5v7d43/6F7w//AOD6f/5DoN9437eHvD//AIPpv/kOgDpKK5w33jbAx4f0AnHOddm4P/gJSfbvG+P+Re8P5/7D0/8A8h0AdJRXOLfeNfm3eH9AHHGNdmOT/wCAlIL7xtuG7w9oAGeSNdmP/tnQB0lFc39u8b/9C94f/wDB9P8A/IdBvvG+ePD2gEe+vTf/ACHQB0lFc2b7xtgY8PaB05/4ns3/AMh0ovfGvlknw/oO/IwP7dmwRznn7J9P89QDo6Kw7C78VyX0a6pouj21qc+ZLb6vLM68cYRrZAecfxD+lblABRRRQAUUUUAYXjbW7nw34G1jWbFIpLixtXmjWYEoWAyMgEHH4itmCQy20cjYyyBjj3FYHxB0q91z4da7pmlw+feXdlJFDFvVd7EcDLEAfiadp+qaxfWUloPDuo6ROtqRDc30lq8XmAYUERTO3Xn7uMA0XsmPt/XYavjKNNas7DUtG1TTUv5Gisru6SLyrhwCwXCyM6EqCQJFXoQcHili8baW/h/WtXmS4t4dEmnhvIpVXzEaLk4AJB3DBXnkMOlcNYeF/EdzqHha4vtI1kXen38c+qXeo6950UjeXIGeGASsm3cf7sZUFQFILbdDX9Jml+KkWjWkSvpuvxxajqYz/qzasBkjHPm5hQ5PSM9aLa29V+Tv91xNpa+j/Fq35HU3fi9IJre0s9H1PUdQmtVu5LG2WJZLeI8AyGSRUBzkbQxJIOAQCRXvPiJotppOj6isd9dQ6vcm1t0trYvIsoVyUZPvAgxspGOD1wMkY/iLwtcp48utdXTtY1S0vrOGBotH1h7KaGSJnwWHnRK6MH/vEqRwPmJp8nhSSNfBv9h6PdWVrYavJe3cF3dLLLCHimDM7mR95LyA8M33vrhqz+/9f8hPTbt+n+Z0uheJY9avL6xl0+80y/sdhmtLzyy4RwSjgxu6lTtYcNnKnIFbVFFIZy2m3l34g8bX88dzLDpWiSGzjhjO0XVwUDSO5/iVAwUDpu3E5IXGSnjTXh8RdR03/hGNZntIbCGSO2jexDKxklBl3GcfKwUADORtOVGeb3guRNM8QeJfD1wdt0moSalCD1lt7g7g4+j70PptGeopdSg1bR/iFJrljolzrFne6YlpKtnNCksEkcjMpIldAVYSHkEkFenNJfZ9P0/z0K099f1uv01Nq18RW1z4outBaC4gvba1iu/3oXbJG5ZcqQxztZSDkDkjrmseX4kaVHYWF0tlqUq6lezWVkkVuHa4kj38qA33W2HDHA7ttGSMv4mLf2M2ia1oMarq9xK2kKjnBMdyvXjOTG6JJjphG5HWm+KLdfDOpfD600rTp71LG8lhitbUxrI4FnKvBkZV9zlh36mmtfy/r5W+/wAiXp91/wAP1dzo7Txpp0mnapdalDdaS+kH/Tre8RTJCCoZT+7Z1cMDxtJyeOoxS6V4tS/1ePTL/SNS0a7ngNxbJfrF/pCKQG2mORwCu5cq21vmHHXHM6n4O1bxTo/ii5uoP7Mu9Ya2NpaSXGGjW3IZBLJESFZm3ZKFtoKkEnNXvDOiQHXre9u/DPiSwurWJzFc6rrn22JCwAZVX7VIckd9g4Hahb6h0Lml3t5o3ju58O31xLdWl/DJqOmzTMWePDgTQFj1Cl1ZfRW29FFdZXH3Df2v8XrJbUq0eg6fMbtwfuyXBTy4/rtjZiOwK+orsKF8Kv8A1q/689wdrsKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKwPEuu3en3mmaTo0UMmqarK6RNcAmOCNF3SSsBgsF+UBQVyWAyOo365jxTpmonWdF1/RrUXs+lvKk1oJFR5oJVAcRliF3gqhG4gEAjIzQMz5vGt54Xn1a18YRw3TWGn/2lFdaZAYxcQg7XXy3dtrq2P4yCGB45FdFe6/b2WtWGltBPLc6hDNNCIwuMRBSwJLDBO8Y7dckVxviDwtrPjO21++m08aVcXOjPpmnWt3KjSZZt7vI0ZZVBYIAAzcDJ64F2CDxFrPj/AELVrvQH0rTtPtLmGQXN1E8xkkEfO2NmXZ8mAdxJOcqowS1qlffW/wCNv0Fon5f8Bfrcj8OeM9ev/E+vWl34Z1h4Le+iiiXfZD7GjQxsQ5E+W5Yt8u/g468DSvPiDaW+qatp9no2salcaOQb0WkCbY0MYcOGd1DcHG1cvkH5cYJhs4dX0DxxrkiaFdajp+sTwXEd3aTQDyGEaxOsiySI2BsDZUNkN6jFY9lqOqweLfHNno+gXWozXF7EqXAmhjt4nNpEB5haQSYHU7EY46ZPFRd203t+Ohdlc7G18Vabfavp9hZtJOdR09tRt50UeW0IZB1Jzk+YpAx69K5jxl49u7fwBrupeGdO1Bp7Frq2N0ot9ttJC20uyyP8yk5xhWPHIFNg8N6p4Q1Dwzcabp0+uQaXokmlSx20sUcu4mJlk/eui7f3RB5yMjg84ZaeENdvfhd4o0XVIba01TVrm+ljWObzIgZXLJ82M45HJAPsK0fKnptr/wClafgTDpfuvy1/E7HRdVvNTjkN7oWoaTsC7TePbt5uc52+VK/THfHUYzzWpWR4f1PUNRtcaroF7o00aJuFzLBIrsRzsMUj8AjqwXqPfGuc4OOT2zSluRHY4jwd4o1nxih1W0vNHjsUuXim0kwu13bAMVxLKJMJJxu2mL/Zz/HVKDxhcaX448ZWpsNX1kWclvKsFoFZbaL7MhYgyOiDLZOxSWJydp5NQ+I/D174k1a3utM8IzaDr8N1E58RGe3UBEID/NFIZZVZAVCOgBBG4L23rDRdQh1zxncSW+2LU3iNo29f3u22VD34+YEc4qW2o3XRP9DWNnJp/wBampP4iU6XZX+jabe63HexiaEWPlrmMgEOWldFHUcZ3HPA4OMyX4i6LB4Xg12aO9W3lvBYvAtuXmhnLlDGyLkkhhj5d2e2Qa5qPQPEtlofhKyu9P1W606x0ZILux0fVFtJku1VAGeQSx70ADDCuRnnB4IoSaVqnhjwPZR6jpsxuj4uS5jtYr0XLyI1wWXEsjLuJHdyp9cVUrKdlte3/kyX4rX+rGevs7ve36N/nod7ZeNLKaTUItVs7zRJ9OthdzxagI8+Qd2JQY3dSuVYEZyCOQMjKab4zgvtStLO60rUtL/tBWewlvo41W7AG4hQrsyNt+bbIEbGeOCBz+r+FdU8br4gub20k0T7do50uzguZEeQEsXMknlMyhd20ABicBs9QKk8O+H421jT59V8L+I7a7ssyJdX2vte2sUuwqSitdMxyGYAmMHB5C0R8/63/SwPy/rRfrc7+iiikMKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMjxZq0+heDtX1a0SN57GyluI1lBKllQkAgEHGR6iufHiHxJo2reHo9dk0vUrPXJvsytY2slrJbSGNpFOGlkEi4RgfukcHnpW5410+61bwJrmnafF511dWE0MMe4LvdkIAySAOT3qHQfB2l6O1teCK6mv4oBGst9fTXbQ5HzCMyu2zPfbjOBnoKI/E7+X63/roOXwq3n+lv66iax4xj0Kdn1DR9UTTUkWOXVAkXkRlmC5K+Z5u3cQNwQjvnHNR33je3tvE1z4fs9I1TU9St7eK5aK0ij2mNywB8x3VBjZ0YgnI27uccL4g8LeK9Y0zW7S80zVr7U5rp3guxrpgsDBvHlokCygFgmAVeMAkMS54z3GmaPfW/wATdc1aaDbY3Wn2cMMu9TueNpS4xnIxvXqO9Japf10E93b+tf8AIt2/iqLUPDlpq+jaZqGpLdD5LaFESVSDhg3mOqKVIIILdRxms5viPpS2EEzWeoC5m1BtMax8tPOiugjP5bfPt5AGGDFTuU5xkhnjey1u71TSGsrfUr3R4zL9utNJvxZ3DuVHlN5hkjJQfNlQ4ySpwcVzWheDNes9Rgln02SGNfFLahiXUftbLbmy8oOZHbex38EHJHbIANNat3/rVL8rsey+/wDJ/rY6rw78RNK8S6na2Vna30LXdtJcQy3EaBH8tlWVAQxJZGcAnG3OcMcGusrzLwd4R1zStZ8LT39l5UdhYalDct5qN5by3CPGOGOcqpPGcd8V6YwypAJUkdR2o6J/1uyerPONV+Ky2d7qENrZX0n2e/s7OINol7/y0ZPNLHZjIV8qOCeMBsjPosEy3FvHNGHCyKGUSIyMARnlWAIPsQCK8tf4feJZdWMj+IdY8ubX1upZdlhnyo7cBJv9T97eiLjGMdU6mvVFBVACxYgYLHGT78UR+DXf/gL9bjfxabf8Fi0UUUAFFFFABRRRQAUUUUAFFZfiPV/7C0Oa/wDM06Lyyo3anffZIBlgPml2Njrx8pycCud0Lx//AGxdTw/bPCEnk27zY0/xL9pcbe7L5C7U9WycehpXA7as3S/D+naPdXV1ZRSm5vGBnnuLiSeRgM4XfIzEKMnCghRk4Aya5TSviR/aWrWtn9u8Ev58oTbZ+KvPmOT/AAR/Z13N6DIz60mqfEn+z9WurP7d4ITyJWj23fivyZRg4+eP7Odjeq5OPWnsB31FcTrvj/8Ase7gh+2eEI/Nt0mxqHiX7M53Dqq+Q25PRsjPoKdd+Pfs3huw1T7Z4RX7W7rvm8SeXbNtOP3c3kHzD6jaMe9GwLU7SiuJtPH/ANp8PX+pfbPCB+yPGu+HxLvt13HH7ybyB5Z9BtO48cU/w747/t3UZLb7X4Tl2QNLjTPEf2uQYxyU8hcLzy2ePQ0bAdPcaTY3WqWmpT26teWQdYJ8kMgcYZeOoOBwcjIB6gVcrz6w+Jn23Ura1+3+Bm86VY9tv4t82Q5OPkT7ONzegyMnjNP1T4k/2fq11Z/bvBCeRK0e278V+TKMHHzx/Zzsb1XJx60B5nWf8I/px18a1JFLLfKpWNpbiR0hBABMcbMUjJAwSoBPOepqa80myv76xvLuHzJ9PlaW2few8tmRkJwDg/KzDnPWuW17x7/Y11bRfbPCMfnWyT41HxJ9lY7s8ovkNuTjh8jPPApLnx/9n8OWWp/bPCA+1SOm+XxLstm2n/lnN5B3n1G0Y96Ng3+f9fkdtRXGWHjz7Z4e1LUvtnhJvsRQb7fxH5tuu44/ezeQPK9vlbJ44pmgeP8A+2tU+yfbPCEn7t326d4l+1S/KM/c8hfl9Tngc4NAdLnVadpVlpMU0enwCITzvcSncWaSRzlmJJJJ/kAAMAAVcrz6w+Jn23Ura1+3+Bm86VY9tv4t82Q5OPkT7ONzegyMnjNS6v8AEb+zNYurL7d4KTyJCm298U/Z5hj+/H9nbafbJoA7yiuJ1zx//ZE9rH9s8IR+fapPi/8AEv2Zjuzyg8ht6ccPxnngUXPj/wCz+HLLU/tnhAfapHTfL4l2WzbT/wAs5vIO8+o2jHvRsC1O2orjLDx59s8PalqX2zwk32IoN9v4j823XccfvZvIHle3ytk8cUzQPH/9tap9k+2eEJP3bvt07xL9ql+UZ+55C/L6nPA5waA6XO2orz6w+Jn23Ura1+3+Bm86VY9tv4t82Q5OPkT7ONzegyMnjNS6v8Rv7M1i6svt3gpPIkKbb3xT9nmGP78f2dtp9smgDvKK4nXPH/8AZE9rH9s8IR+fapPi/wDEv2Zjuzyg8ht6ccPxnngU+48d+R4VtNX+1+Ex9omaLzJfEey1OM8JP5B3txyu0Y554oDc7OiuIsPiB9s0bUr77Z4Pb7EsZ3W/ibzYV3Nj97J5A8oenDZPHFO8PePf7c1YWf2zwjLmNn26Z4k+1zcDP+r8hePU54HrQHS52tFee2XxN+139vb/AG/wK3nSqmIPF3mSHJx8qfZhub0GRk8ZqbV/iN/ZmsXVl9u8FJ5EhTbe+Kfs8wx/fj+zttPtk0Ad5RXGa948/sZrIfbPCUf2q1Sf/iYeI/spbOeY/wBw29OOH4z6CmS+P/L8Mwar9s8IDzbhod7+JdtqcDOFn8j5n9V2jA5zQHb+vM7aiuL0zx79v0jU737Z4Rf7CiNutfEnnQrkkfvpPIHlDjg4bPtTfD/j/wDtrV0svtnhCTcjNt03xL9qm4BPEfkLkccnPA55oA7aivPbL4m/a7+3t/t/gVvOlVMQeLvMkOTj5U+zDc3oMjJ4zVnXPiH/AGRrdzYfbfBsfksBt1DxP9mnHAPzxeQ2089MnjFAHc0VxGr/ABA/syOwb7Z4PT7Xapcf6b4m+zhs55jPkN5kfHD8Z54FOl8e+X4Wh1f7Z4RHmXJg8x/EmLU4GcLP5HzP/sbRxzmgDtaK4nTfH/2/S9Tu/tnhB/sMavutfEvnRLk4/ev5A8oehw2TxTvDnj3+3daisPtnhGXerHbpniT7XOcDPEXkLkepzwOaA6XO0qnZ6VZ2F5fXdpD5c9/Ksty+4nzHCBAcE4HyqBxjpXDx/E/fdpD9v8CHdIEwvi/L9ccL9m5PtmreufEP+yNbubD7b4Nj8lgNuoeJ/s044B+eLyG2nnpk8Yo8w2bR3NFcXrHj3+y7bTpftnhFPtlsJ833iT7OrZ7xHyG8xPR/lz6U1/H+zwvFq32zwh890YN7eJcWpwucCfyOX/2NvTnPagDtqK4zSPHf9p6bql19r8Jv9hhEu6y8R+fGvXmZ/IXyl4+9hu/HFRaD8Qf7Z1u3sPtvg6TziRt07xP9qnOFJ+WLyF3dOeRgZPagPM7iivO4/ifvu0h+3+BDukCYXxfl+uOF+zcn2zVzXviF/Y2uXNh9t8HR+SQNuo+J/ss4yoPzxeQ23r/eORg96A2djuKqahpdnqqQLfw+atvcJcxDcV2yIcq3BGcHseK5TVvH/wDZtrp032zwgn2y3E2bzxL5CNnvE3kN5if7WF+lPPjzHhMax9s8Jc3XkeYfEf8Aon3c4+0eR9//AGNvTnNHn2/z/wAw307/AOX+R2dFcTpPj/8AtK01Gb7Z4Qf7HbmbNn4l89F56yt5C+Wn+1hvpTdB+IP9s65b2H23wdJ5xI26d4m+1TnCk/LF5C7un94YGT2p9bCudxRXnjfE7bfG3+3+BeJNmD4uw/XH3fs3X2z1q3rvxC/sfW7iw+2+DY/JIG3UPE/2WcZUH5ovIbb1/vHIwe9LpcfWx3FFcZq/jv8AszTtMuftfhNPt0Hm5vfEf2eNunML+Q3mrz97C9uOajX4gZ8MPq32zwf8t0LfePE2bX7u7Bn8jiT/AGNvTnPagNzt6K4rR/Hv9qWuoy/bPCL/AGK2M+bLxJ9oVcd5W8hfLT1fDY9Kj0P4h/2xrdtYfbfBsnnMRt0/xP8AaZzwT8kXkLuPHqOMmgVzuaK88b4nbb42/wBv8C8SbMHxdh+uPu/ZuvtnrV3xD4//ALD1uaw+2eEIvLCnbqfiX7JOMqD80XkNt68fMcjB70dLlW1a7HbUVxOpeP8A7BpemXf2zwgn26Nn3XXiXyYmwcfun8g+avqcLg8Yp0fj3f4Vm1f7Z4RPl3Ig8xfEmbUZGcNP5HD/AOxtPHOaP6/QS1O0oridI8f/ANpwahJ9s8IP9jtWnzZeJfPVcd5T5C+Wnq/zY9KZofxD/tjW7aw+2+DZPOYjbp/if7TOeCfki8hdx49Rxk0CudzRXnjfE7bfG3+3+BeJNmD4uw/XH3fs3X2z1q74h8f/ANh63NYfbPCEXlhTt1PxL9knGVB+aLyG29ePmORg96OlyratdjtqK4nUvH/2DS9Mu/tnhBPt0bPuuvEvkxNg4/dP5B81fU4XB4xTo/Hu/wAKzav9s8Iny7kQeYviTNqMjOGn8jh/9jaeOc0f1+glqdpRXE6R4/8A7Tg1CT7Z4Qf7HatPmy8S+eq47ynyF8tPV/mx6UzQviH/AGxrdtYfbfBsnnMRt0/xP9pnPBPyReQu48dMjjJo62Fc7mivPbz4m/Zb+e3+3+BV8qVkxN4u8uQYOPmT7MdreoycHjNW9f8AiB/YusS2X2zwfHsVTt1HxN9lmGVB5j8hsDng55GDxnFHS5Vjt6K4rU/Hv9n6Tpl59s8Ip9ujd9114k8mFtpA/cyeQfNHPJwuDjrSReP/ADPDE+rfbPCB8q5WHeniXdajIzhp/I+V/RdpyOc0PQS1O2orjND8ef2ut+ftnhKT7JatP/xL/Ef2kLjvJ+4Xy09X5x6Gq2j/ABG/tTWLWy+3eCn8+QJtsvFP2iY5/uR/Z13H2yKdtbCbsrneUV57efE37Lfz2/2/wKvlSsmJvF3lyDBx8yfZjtb1GTg8Zq74h8e/2HqzWf2zwjFiNX26n4k+yTcjP+r8huPQ559qV9Lj62O1oridR8f/AGHSNMvftnhBftyO2658S+VC204/dSeQfNHqcLg8c0+38d+f4Vu9X+1+Ez9nnWLzI/Ee+1GccPP5A2Nzwu05455oDc7OiuQ8LeNv+Ei1Z7P7X4Wm2xGTbpHiD7bLwQOY/JTC89c+nHNdfTsAUUUUgCiiigAoorO0zxDoutzTxaNq9hqElsQJ0tLlJTETnG4KTjoevoaANGis208R6Hf6tNpdjrOn3OoW+7zrSG6R5Y9pw25AcjBODkcU268T6DY3EVve63p1vNNMYI45buNGeQYygBOSw3Lx15HrQBqUVDHeW011Naw3MMlxb7TNErgvHuGV3DqMgcZ61n3niCGG183S7W41uTzmg8nTTG5DrneGdmVEK4OdzDnjk8UAa1FZGheJbPXrO6mjjms5bGd7e8trsKsltIoBIbBKkYIYFSQQQQaytM8c6frvjoaRoOp6ZqdkunNcyy2dwszJIJFUKSrEAYJOCM0dUu/+VxN2V3/WtjrKKK5zxX440vwhcabb6hFdXE+pXCwxR2sYcoCyp5j5ICoGdAT6sAAaBnR0UVU1XU7XRdIu9T1GTyrW0haaV8dFUZP1PtRsNJt2RborCstdntPCMmu+Lkh0pER7mSMEt9nh6qrnnc+3Gdv8XAz1L9H8Qz6tKvm+HtW02CRN8VxepCFkHUfKkjOhxzh1X0ODxRZ3sLpc2qKzYfEWiXOsyaRb6xp8upxZ8yyjukaZMdcoDuHUdqbeeJ9B0+dIb/W9OtZZJjAkc13GjNIMEoATy3zLx1+YetG4GpRUEd7ay3k1pFcwvcwKrTQLIC8YbO0svUA4OM9cGqF74hghty+l21xrcona3MGmmN2SRfvKzMyohHcMw54GSQKANaiuUf4g6fH4bu9Vaw1DzbK8SwudOKRi4jnZ1RUOX2HO9SGDlSDnNW9N8Xw3euJo2paZf6LqM0TTW8F+Ij9oRThijxO6ErkZUsGwc4xzRv8A18/y1A6CisDRtfuJvEOo6DrMUcOoWv8ApEDxAiO5tWYhHXJJDLja49QCOGFb9HS4eQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVFd3UFjZzXd3KsNvBG0ksjnARVGST7ACk2krsaTbsiWiuP8A+E2n03wFceLfEGnTRWruklrZWyA3AhkZUjDhmC+YS24jICg45IJOwPE9g1xokUYmf+2gzWzqowoWPzDuycjj2PNU1bcXS5sUVm/8JJof9uf2L/bOn/2r/wA+H2pPP+7u/wBXnd93np05pt94m0HS2K6nrenWbCUQkXF3HGRJtDbOSPm2kHHXBBpAalFU9T1fTdEs/tes6ha6fbbgvnXc6xJk9BuYgZqe1ure+tIrqynjuLeZA8U0Th0dTyCCOCD6igCWiiigAorBt/FaX+qNbaRpOoajbRzm3n1CERLbxSKcOMySKz7TwSisM5GcggR674vGgtcyS6Fq91ZWSeZd31vFH5UC43E4d1eQKvJMav3HUEAvpcdm3ZHRUVHbzxXVtFcW7iSGZA8br0ZSMgj8Kko2EnfVBRTJpora3knuJFjijUu7ucBVAyST6Vj6N4jbXJle10XUotPlj8yDUbhYo4pl7YQyeaM9t0Y/lkA26K5q48aw2WoQRaho2q2djcXQtItSnijWFpWO1QV3+aoZhtDMgBJHOCCelo6XDyCiiqupXk9jYtPa6dc6lIpAFvatErtk9QZHRePdqALVFYWleK7fU11JJrDUNPu9MwbqyuIQ8qqy7lZREzh8gHAQk5BGM8VFp3jCO61yDSdR0fUtGu7qJ5rRb4QkXCpjftMUj4I3AkNtOD0ODg62Dpc6KisC18QTxeMrjw/q8UcTzRm602ePO24hGA6nPSRGIyOhVlI7gb9HS4BRRRQAUUUUAFFFFABRRRQAUVBfX1tpmn3F9fSrDbW0bSzSN0RFGSfyFYFl42iuNV0+zvtF1XS11PcLG5vEi8u4YLv2/JIzISgLAOq/dI68ULV2QHTUVys3xAsIS9ydP1BtHjn+zyayqR/ZkfdsPBfzCof5d4Qr3zgEi/qnieKw1IabY6fe6vqPlCZ7SxEe6KMkgO7yOiKCQQAW3HBwCASADbopkEjS28ckkLwM6hmikILISOh2kjI9iR70+gAooooAKKKKACiiigAooooAKKKKACiiigDl/iPb3d18P9TisYZbhiEM0EClnlgEimVFA5JMe8YHJziqtxrWjeJfB2o2ngDUtPur+PS5EtBYyKTbbkwinb/quQuFOD8vT5TjsqKlxumn1KUrNPseVPqGg6zong/RvB81s+qaffWkgtIQPO0+OMgT+cg5i+TejBgMswXqaiutPs08EfFe7W2j+0XFxeLLLt+Zwluu0E+gySB2yfWvWqKqXvc3nf8AHl/+RJh7rj5W/C/+Z5v4gvm8Happ3iiKPzF1HTP7OmQZzLcKvmWw9OT5ifV1q1qev6d8Ivhxp8V+6z3z4hhjeVY/tV3ISzszthUUuzMzkgAfgD31FNu9/wCut/1f4dgjZW/ry/Rfj3PHJUsG+Gd9cLr2neILi51e1v8AxJJpkyzQrGZY/NQKpJESxpjnkqrE966bTNc0/Xfi/FPo8y3dnHociLdwkNBK3noSI3HD7eASMgHjOQQO9qC8S6ezlXT5oYLkr+7knhMqKfUoGUke24fWi6unba//AKTy29F0Fa8bf1vcnrxfxXLrcem6lqet+E9UW9vNVskjmWezaKK3ju08qJT5+7LcscqBvfnAGa9E+w+N/wDoYfD/AP4IZ/8A5MroxnaNxBOOSBilbVMq5FaTSXNnFNNbS2kkihmgmKF4z/dJRmXP0JHvXL/E/I8DSSt/x7Q3lpNd+nkLcRtJn22g59s111RXVrBfWc1peRLNbzxtHLG4yHVhgg+xBo80JW2Zy3xOsLzUfANyum2z3ssE9vdG1iGWuEimSRkA7kqpwO9WE8XaP4n0m6tvCet2l1qUto7RRQygywMVO0yJ96PBwDuAweOvFaXhzRpPD+iRaY19LfRW5KW8kw+dIs/IjH+IqMLuPJAGea1KUknFx6P/ACBNpp9UeQJe6JqXw/8AC3hzw5JCdfsrqyYWCY+02Ekci+e8qD5o8ASBmbGS2MksM2JLC0Ph/wCLt0baI3E01zFJIVBZkWxjKqT6Asxx7mvV6KqT5ubzv+Nv8hwag4+Vvwv/AJnmesXw8Hz6H4vEbSR3elDTbpF5Mkgj823/ABLh4x7yird9r9l8I/hrp66q6T6nOdixtIsQuryUl3LO2FRN7MSzHAH4A+g0USd7+v63/V/h2JirW8v+G/r5/Lxq6m0q3+F19fN4l0rWNQuNasr/AFi6sbpHghka4hG0EH5Y1VAAW5IUk966W81Ky8Z/ELwxJ4Zuo9RtNHknury/tG8y3TdEY1i8xTtLnfnaCSAMnAIz6BRR1/rsl+g+lv61OP1QCT4w+HhbjMsOl3rXBA6Rs8IQE+7A4+hrsKx9G8PjTNS1LU7q5N7qOoS5knK7QkSk+VCq5OFUE/VizHrgbFLZJf13B6sKKKKACiiigAooooAKKKKACiiigAooooAK5X4nI7/DLXNgLKtsXlUDOYgQZBjv8gbiuqpHRZI2SRQysMMpGQR6Une2gadThviw8cvwnvJLcqYmks2Qr0Km4ixj8Ky75W0z4o6DocZVFNxeXtiMcBJIHLr3ztl3H2DqK7Xwv4f/AOEY0ptLhvJLmxilJso5F+a2hIGId2cuFO7BPIUgc4ydmnJJqy2f/A/HQWtrHi6XOmz/AAfs/CUEsJ8Xq8Ktp+R9rivhKGedkBDABt0hk6FeckHnfs9PtH1b4nXUttE9xKVgkkZAS0YskO36ZY8e9ek0VM486a2un+Nv8i4y5Xc8cNzeaZfeDNWudY0vSLFfDiQ299q9o80Ec7BC67hNGEdkUYLHkKwHeu18Fx2+keE7++ttYs9etprm4vRJpEIEIzy8cSiR/wCIMcburGuuqnqtjLqWmy2kGoXWnNIMfabTZ5ijvgurAZHGcZ9MHmrk2723d/xdzOMUkk+lvwVil4T8Uaf4z8L2ev6OJhaXikos6bXUhipBAJGQQRwSK2GBKMFOCRwfSqekaRY6DpFtpek2629napsijUk4HuTySTySeSTk1dolZt22Gr9TxDRWh0nwdpun6fruof8ACc2dylu2lDUJN27zsPvtM7DFsLMZCn3cPuzg10XxD8a6XeanL4Dj13TtJa4ixq19e3McQtrdxykYcjdK6njqFByewr02ipavuUm07o5rWfFWi+Dvh+NejEt9o9nBEIzpwWYtGSFVlOQpXkHOcYrobeeO6tYriA7o5UDocYyCMis3xD4dt/E1mljqFzcrYFv9JtYWVUul/uSHaW28dFK56HI4rVVVRAiKFVRgADAAqr3u2TaySRzXxH0681b4aeILHS0eS7nsZEjjQZZzj7oHqRx+NczpF/p8/i7QT4J1+81bzjIdZhfUZLqOGExEhpI2Yrbv5oUBVCdXG3A49MooTs7jeqPJdS8caB408bR6XP4h0rTtB0K+SSdrq9jim1C7jYFI40Y58pGwS2PmYALwCa7vxH4x07wtfaLa6nHcn+2bxbK3lhj3Ikjfd3nPAPQYz+QJrfrG1Dw1a6rr1lqWo3FzcR2DebbWLFBBHMMgTYC7mcAkDcxUZyADzQtkvv8A1/r0B6tv+vI2az9d1Cx03R5ptT1iLRYWHli+lljj8pjwCDICmfQEEexrQoqWB554Q1G0stZ8S6iurDWNGFvDcyeIZdp8x1Vw8e9AInVFVT+7UBSzA5YmqHhPxXo3jnx1b+IJ9d0uIQxy22h6SLyI3LK+PMmkQMWDMEGE6quc8nj1KimtGn/X9dF+r1B7af1/X9aHH+K/3nj3wVFbjNyl5czNgcrALZ1cn23PGPqRXYVkWOgC38R32t3dy13d3KiGHK7VtoBz5ajPdsszdScdlAGvR0QdQooooAKKKKACiiigAooooAx/FsrweDdXlj0v+12WzlIsMZ+0/KfkxznPSvL/AAtfWo8TeD00vxJD4wZo5Fks1k806OpjJMq4YsijKxfv974bAYZIPtFFC0d/66/5g9Vb+un+R45rtmZvD2q+BvC3i7T9SW7klhi0u2thNd2m+TLiWVZCqRIWbJeMHGFDbiCbt5b6NpfjrxIPGfiC70lLxbabT2XU5LETKsIjbY0bKZHDA5TLYyp2/Nz6tRStpYPQwvBU2qXHgnSpdfEo1B4AZfPQJJ/s71AGH24yMDnPArdooqm7u4krKwUUUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUNW0LSNegSDXdKstSijbekd5bpMqtjGQGBwcd6yh8OfBA6eDvD/wD4K4P/AImq/wAQpryLS9Kj0+9mspLjWLWB5YWIbYz4YfiPXiuF8LavfWWu6Nc3+tX8ts1/rFnKtzdySRrb27OY8hidzLtzvbLkcZxxUpqzfb9Ev80DbTS/rr/kz0Nfh14JRgyeDtAVlOQRpcIIP/fNIfhz4IJyfB3h8n/sFwf/ABNcXFrFxH4h8Mano9r4nt7PVr8Qy3Wq36y297FJE7ArAZ2MRJUMuI48DIwAdtM8T6pqOjeMrrwzYa7cR6drksL3V/JPI8uiNKSuyNiCFEuzEYJGxsnoVFXvp8vnoN2WvTf5anbn4deCWOT4O0A8Y50uH/4mj/hXPgjAH/CHaBgdB/ZcP/xNb9tbpaWkVvG0jJEgRWlkaRyAMZZmJLH3JJNch46+23HiDwlptnqd5p0N9fzRXLWkmxpIxbSMUz2zt6jkdQQQCF5IS2ND/hXXgnaR/wAIdoGCckf2XD/8TSp8O/BUbZTwfoKnBGRpkI4IwR930OK47UNS1PQrjW9B0rVLsWw1PTraK8ubhriWyW5IEqiSXcScYK7ycGQdsCtHUJLnwV4utrbRrnUL61vNKvLiWzvb2W7KyQBCjq0rM67t5UgHB+XjPNS2uVy6a/grv8ASbdlv/wAGxv8A/CufBH/Qm+H/APwVwf8AxNYUugeB4PF1toV18N9Pia8Ehtr1tMsjBNsQM2MMXHXHzIOfzrPlF7pXgnw/4st9c1G81a8msTc77x3t7wXDorxrAT5SDD/KUUEbRyec9J4h/wCSk+D/APt9/wDRIq2nF2fR2J5rpNdbEGu+EvBOj2cd03w+0/US80cHl2OiwyyKGO0MRgYRe57CtD/hXPgjGP8AhDvD+P8AsFwf/E1n/EqBodGtdUtry/trq3v7OJDbX00SMr3MasGjVgj5BI+YHrS+OTeT+IfCWnWmpXenw32oTR3RtZNjSxi2kcpntnb1HI6gggESrv7/APIu1vSxoD4d+CghUeD9ACtgkf2ZDg4/4DSD4deCVOR4O0AcY40uH/4muRvdW1Hw5oHxAttN1G6f+yZIRp7XU7XM0ZlgjYorybmdizHaGJ+YgdOKseF9La58fQX2lyeKRotvprLPHrd1fKj3LMAuI7hgXOzfu4KD5cc01q7f1tf/AIHzE9Fr/W3+d/Q07Pw78K9Q1WbTNP0fwfdahBu820gtbV5Y9pwdyAZGDwcjipdb8H+AdD0O/wBXvfBehyQWUD3Eqx6VAXZVUscZUAnjuRWffG31/wAf6Fo3hq0jSz8K3JuL67hUJFbnymRbWPHVzuBZRgKoGeSBWL4niHiHwH461nWb+/jaxa9s7a3i1Ca3hiSJSqhkRlWQufmO8NncB0AFTKXuNrs391tfvZpGPvJPZtL77/ojtYfh/wCB7q3inXwboO2RFZd2lwZAI4/hp/8AwrnwRj/kTvD+P+wXB/8AE1taZ/yCLP8A64J/6CK5H4i+JtPsvsXhu71m00htY3faLq6uVgEVquPNKsxA3tkIvcbi38Jq56SsjKDvG7Lem+D/AIdavYm40jw74XvrVmKmW1sbeSMsvBGVUjI/SrI+HXglTkeDtAHGONLh/wDiawPhBqGgvo+r6b4fvdPkjt9XvGitrOZG8uEyny2CqeFI6Hoa9Do6J90hvdrs2jiZvCvw9i8RW+iJ4M0Sa9miad1i0mAiCMcB5Dt+UE/KO5OcDAJEq+E/hpLb3lyugeFHhsXdLuUWVsVt2UZcSHGFIHJBxin+Bc3d74n1OfJup9Zmt2z/AARw4jjUe2AWx6uT3rkfDvh661OzvXt4YpNH1rUr46srMv8Ayxu5tvHV/MG1G/2VFTd8vN5X+/X8vxKtq12dvuvf8Udpb+A/AV9axXVr4U8OXEEyK8U0enQOroRwQQuCMYwRUp+HfgoxhD4P0EopJC/2ZDgE4ycbfYflXEaBqdnqHgvwtosVn4h1O8h0SC4e20W++x+RHjYrvJ50WclWAXc33Sdo61Jb6rql18GbXX7rWLqz1XR7qby2mkP+lGK4eJYJ1jJEhcKEONx3HK5PWpJJt9E/1t/VibPRHV3vgf4e6ZYy3eo+GPDNnaxDMs8+n28aIPUsVwPxqpo3h74W66Zj4f0jwhqRhAEv2G1tZvLDZA3bQcZwevXBrkr6+vNb+FHiTxHqlxO+oXFwsMmkpOxj03yp1X7OFbavmcZZyBuLcHbiumu7XVdV8W2filtAn0ddFsbpSt1LE096zqNsYELuPLBXd8zZ3YwvU1LfLfm6f5X+f9bgvefKjZ/4Vx4I/wChN8P/APgrg/8AiaD8OfBBOT4O8Pk/9guD/wCJrjpGv7D4d6L40i1vULrWrl7KadftjvbXInkRXhWAnylGJMKVUMCoOSck3obe58SaX4i1u/1nUrC7sby6gs1tr6SGKzWAkIWiUhJCcbz5gbIYD7uBTl7l79L3+Vv8wj7yVutrfO/+X9dLHirQvBPhm0S9u/hzp99aLhZbi102zItxuVV3B2ViPm/hDdK1rvwD4JtbGaf/AIQnRZxCjSCGHSYWd8DOFG3knGAPpWL4n1K51j4Dx6lfxeTdXllaTzR4xtdmjJGO3Jrstd06HU9HmhuJbqJVUuGtLuW2cEA4+eNlbHtnFKpeHMnuhU2ptfIwdJ8E+CNV0e2vh4D0qyF1EshtrzR4Y5o887XXacMO4q2Ph14JU5Hg7QBxjjS4f/ia5iXVtST9mdNVS/uRqX/CPpMLwyky+Z5YO/dnJOe9X44Ljw74/wBBt49W1C7TU7G7e++23byJLJGIisgQnbFyzcRhVwcY4GLmuWTT6f8AB/yCN5RXn/X6mx/wrjwR/wBCb4f/APBXB/8AE1m6hoHwtsdWjsdW0nwhbajdEGO3uba1SaYscAhWGWycj3NcXoX27xbb+HbqG68VHVZr/wC0apN9ovbaxe2BfOw5WEoRs2CP5iMFgfnrrfiRPavot54V0ewhute8TRtGsCKBtBUI1zMccIgA+Y5JIVR7R+e35f1fyKVnK34/1/Wps/8ACufBB6+DvD//AIK4f/iaU/DrwTtCnwfoG0EkD+zIcD/x32FbtnAbWxgt2cyGKNULt1bAxmuf+I13PZfDrWZbSRopWt/KEqHDRhyELA9iAxOfanLS9iU72cjGstH+GF3o13rDeGPDtppVvMYvt91p1tFBMAQN6MRgpuO0McZI4yME6v8AwhHw/gnt0/4Rjw1HLc5EC/2fbhpflydvy/N8vJx2rL+KVjbWHwdubC0iWO1tzZwxxqPlVFuIgBj0wKoNJJp/jLw1o0gZ5NJurtIc4zJA1szQ4x6D93nuYyaJNJN9v+B/mGtjqf8AhXHgj/oTfD//AIK4P/iaD8OvBLMS3g7QCScknS4ef/Ha4kzaiPhTaePl1jUW16SOG8MYvHNs5d1H2YW+4RYw2wfLuzg53c1dms7zWNV8eS3et6tFDp0i/YYLW9kgW2f7IjlgUILckfI2UyCduSaU2oJ83Tf5f1/W5UY8z0OpPw58EHGfB2gHHA/4lcP/AMTR/wAK68E7QP8AhDtAwDkD+y4f/ia4+417V9bv/Denz2es39tPoKaldJo13HaSTzMVXJkM0TBV5O1G5LjPArr/AAY2r2vh+5XxJFdWq29xILZtRnjkmFsMFDK6OykjkZLEkKCSTkmmrXv5/g7fmQpcyTXl+Kv+Q5fh34KVWC+D9BAcYYDTIfmGc4Py+oB/CkHw58EA5Hg7w+D/ANguH/4mt6zvbXUbOK70+5huraZd0c0EgdHHqGHBFTUtijhLPRPhLqOqNpmn6Z4Lur9Swa0gt7R5QV+8CgGeO/HFXrzwb8O7O5t49Q8N+GIJ7yTy4FnsbdWnfrtUFcs3sOaz/Hb6gNQ0i41SxtY9DstYtZVu7ecy3W8sETMbIqou9wGYO525wvPEfxA0qyttc8L6hDbqLy68SWgmnYlnZVjlCqCeijkhRgZLHGSSSOtvVL8v8weik+yb+6+n4G7/AMK58EHr4O8P8dP+JXB/8TS/8K78FbNn/CH6BtBzt/syHGfX7tbWoajZaTZPeapeW9lax43z3MqxouTgZZiAOSBVkHIyOlAHLz+AfAdnay3Fz4T8OQQRoXlkk02BVVRySxK4AGM5PpVDR/D3ws15pf7A0jwfqTQYMn2K2tZjHnOM7QcZwfyrt64MtqA+K+kXfiKxtbNpbO5tLBrC4NwJTlZG81mSMr8qAqoVhndlhwCLcNlcvf8ACF/DkaoNMPhrwv8AbzF5wtPsFv5vl5xv2bc7c8ZxirR+HXglmJbwdoBJOSTpcPP/AI7WJ/Zlnpfxq0eOwgWES6PfSyEZLSOZoMszHlifUkngV29zqNlZz20F5eW9vLdP5dvHLKqtM2M7VBPzHHOBR9lPv/m0D3a/ra5iN8O/BTKobwfoJCDCg6ZD8oznA+X1JP41U1Lwb8ONGsWu9X8OeFrC0UgNPdWNvFGCeBlmUCuuqK6a4S0layiimuAhMUc0hjRm7AsFYqPcKfoaWwHK6X4S+G+s2RutE8P+Fb+1YlDNaWVtKhIwSNygjjjiq+k+HfhZrssqaFo/g/UZIMGVbO1tZTHnpnaDjoevpUfhiFpvGniq08RWFtFf38FtNPaQETWr25V41+dgpkYlXDbkXjaACBksspo/FfxQtNT0SIDSvDtvcWcl8vCXU0mwGKPH3lTZyem7AHQ01q0v6/q+gbJ/1/WhYsPCvw8v9W1DTE8GaHDe2DL5sM2kwKWRhlZF+X5kPIB9VYHBFab/AA78FSOWfwfoDMeSTpkJJ/8AHao+KEFp8Q/Bt/b8XFxcXFhLj+OFoHlIP0eJD+ddjR0QdTmz8OvBJAB8HaAcDA/4lcPH/jtH/CufBGMf8IdoGOuP7Lh/+JrpKKAObHw68EgEDwdoAyMH/iVw8/8AjtA+HPggHI8HeHwf+wXB/wDE10lFAHN/8K48Ef8AQm+H/wDwVwf/ABNK/wAO/BUjln8H6AzHkk6ZCSf/AB2ujooA5s/DrwSQAfB2gHAwP+JXDx/47R/wrnwRjH/CHaBjrj+y4f8A4mukooA5sfDrwSAQPB2gDIwf+JXDz/47Wdpnh74X6lqMtto2keEbu9tDmWK0trZ5ISDj5goyuDxz3q98SZ57b4Z6/LbPJG62UmXjzuRcfMRjuFya5XUY7g6r4J1Czs49ItINRS00yxWMCaS3aJ/MZ8fcUouRGOm1Sx3fIotXb0/H+vz7Cekb+r+7+v6udb/wrjwR/wBCb4f/APBXB/8AE0H4c+CD18HeHz/3C4f/AImukooGc4fh14JYAHwdoBCjAB0uHgdf7vvSf8K58EYx/wAId4fx/wBguD/4mukooA5xfh34KXdt8H6ANwwcaZDyPT7tIPh14JVgV8HaACDkEaXDx/47XSUUAc3/AMK48Ef9Cb4f/wDBXB/8TQfhz4IJyfB2gE++lw//ABNdJRQBzZ+HXgkgA+DtAOBgf8SuHj/x2lHw78FCMoPB+ghGIJX+zIcEjODjb7n866OigDDsPBPhXSr6O90vwzo9ldRZMc9vYRRumRg4ZVBHBIrcoooAKKKKACiiigCpqGl2eqpAt/D5q29wlzENxXbIhyrcEZwex4rOg8GaBbyQPFp65t5rieMNI7APcZ84kEkHduPByBngCtyilZf1/XkBzNn8O/DFjcWU8FhK0mnyCSy8+9nlFqQCMRB3IjXB5VcKcLkfKMWj4N0FtM1bT5bAS22sStLfrLK7tOzAAksWLDAAAwRtwMYxW5RTDrc519Yv9KYWFn4Q16+t7ZRHHcpdWjiRQMA7pbkSH6sMnvWNrulah4z1Pw3LdaLqmlW1jqEr3Qe+jgmVDbSKrq9vMWwXZVwGz1yMV3dFO/fUPQx4fCmiwaJc6SLFZLO7Ja5E7tK87HGXeRiXZuB8xJIwMHgU7SPDGl6Jcy3NlHcSXMqCNri8u5rqXYDkIHlZmC552g4zzjNa1FIPI5+x8DeH9Nvorq0s5VMDtJBA93NJb27nOWigZzHGeTgqowCQMZNM1HwJomq6sNSvG1U3aszRvFrV5EItww2xUlAQEdQoFdHRQG5ha74N0fxJ5Y1f7fIkQQLHFqlzCmVbcrFY5FBYHB3EE8DngVYi8NaZH/ZpZLid9LleW0lubuaaRGdWViXdizfK7D5icZ46CtWii9gOP8beFZb7wxri+HbdDqOqPDJdK0mDcLGUBVS2VRzGu1W4w2DkHms3RtF1g+LtPubCy8QaLpkCSi/j1rW2vPtYZcIsaefMFIbDF8oR0GcnHoVFC0B6nLaT8OfDuhCJdIGq2scUvmrDHrd75RYtuJKebtbJ5IIIPfNWNQ8B+HNVur2e/wBPaX7epF1D9plWGY7du9ogwQvgAB8bhgYPAroaKAMC51u/064azs/B+s3kEICRz281mEcAdvMuFb25A6VoaTqFzqNu8l5o97pLK20RXjwszjH3h5UjjHbkg+1X6KPUPQp6dpVlpMc6afD5K3FxJcyjezbpJG3O3JOMk9BxVyiigPM5m10jUdG8d3V5p6JNo+sjzbyPcFa1uUQKJRn7yuqqpA5BVT3ONrS9JstGsfsemw+Tb+ZJLs3s3zSOXc5JJ5Zifx4q5RR0sHW5zQ+HvhpVsxDYzW/2OD7NEbe9niLQ7t3lyFXHmLn+F9w5Pqamt/A/h20tNMtLbTVittKuXu7KBZXEcUrFiW2bsHBdsAghc8YrfooDfcxLrwdoV5PqUs9kS2qCP7YqzyKsxTG1ioYAMNoG4AEgAEkVt0UUB5mBa+B9As9Qju4LOXMMpmhge7me3hkOTvjgZzGjcnBVQRk46mjUPA+ganqE15eWkxe5Km5ijvJo4LkrwPNhVxHLwADvU5AAOQMVv0ULTYDE1/wjpPiaFIdXW8eFV2+Tb6hcW8bDORuSJ1DYIGCQcU6Xwtp82jJpck+qG2Ry4YavdCUk54aUSeYw5+6WI6ccCtmilZAc7aeBNAsvDdzoEUF0+lXMIge1n1C4mVYwMbU3uSgx2XFaGoaNb3dxDfRxqNRtIJorOZ2bbF5gAOVzgg7V6g9K0qKcveWoLTY8g03wp4kttC0vTdO03XtL1m3eET6pJrgOnrsYGRktlnYFGAYLH5KgbhkLjjurvwBoF3r13rTJqMGoXiotxPaavd2/mBRhQRHKowPTHc+tdLRQBFa26WdpFbxNIyRIEUyytK5A9XYlmPuSSah1XTbbWdIu9Mv08y1vIXgmXOMqwIP6GrdFD13BabHLaVoV5qvhB9A8dW6Xn2eVYRcJJtF8kZV45sKQUOQMr/eU9VIzrXPh3S7zxDZ65cWu7UbFGjgnEjDarAggqDtbqeoOMnGM1p0U27u4rK1jn4/A/h+LUheJZy5Wf7StubuY2yS5z5i25fyg275twXO7nrzWgmhadG+pslvhtVbdeHe3707BH68fKoHGPzrQoqbJqxV3e5iXng/RL2y0+2ltpYl01BHZy211LBNCm3btWWNg+CAARuwcDOcUt1pU9h4Xn07QoEv5HVkWPV9QmkVg33t8rCRyME8fhx1G1RTfvXv1EtLW6GB4H8KQeCfBen+H7adrhLNCDKy7d7MxZiBk4GWOBk4Hc1v0UU23J3YkrGFF4N0aPUkvpUvLuaOTzY1vtRuLqOKTs6RyuyKw5wQARkgYqPWvA2ieIL9LzVRqUksciyxCPVrqFInUYDoiSBVbGeQAeT610NFIZyPjXwhN4l8Gf8IraCE2NyFiuLq+uJJpYEUgh03ZMj8cFnGDyd3Q9Ta26WlnDbQ58uGNY1ycnAGBUtFHfzAjubdLq1lt5TIqSoUYxSNGwBGOGUhlPuCCKy9L8K6TpF4by2juZ7rYUW4vr2a7kjU9VR5nYoDgZCkA4GegrYooA5mb4e6DPrA1WU6sb5QwWYa5egorMGZVAlwqkgfKOOBxxVbxX4Qm8WeJdBku47WLTtFu11BJ1kZriSZc4jC7Qqp0YtuJO0DaOtdfRQtLW6B38wqpqemwatZG1upLqOMsG3Wt3LbPx/txMrY9s1booAx7Xwpo9lp17ZwW8u2/QpdTSXMslxOMFfmnZjIcA4HzcDgYqpovgPQ/DxthpH9pwRWoxDbnWLt4UGMY8ppShHPQiujoo63A5uDSL3UfHD6zq0SQ2umxtb6ZDuDMxcAyTtg4BOAijqAGJ+9gdJRRQtFYOtwooooAKKKKACiiigAooooAbJGk0TRyorxupVkYZDA9QR3FYmkeDND0O8S6062mEsUZig8+7mnW3Q4ykSyOwiXgDCADAA6AVu0UdbgFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUVQ1bTrnUYEjs9YvdJZW3GSzSBmcY6HzY3GO/AB96yh4X1f/oevEB/7Yaf/wDItAHSUVzi+GNWVgT4519gDkgwWGD/AOStIfC+rk/8j14gHt5Gn/8AyLQB0lFc2fDGrE8eOtfHHaDT/wD5Fo/4RjV8Af8ACda/9fI0/wD+RaAOkorm/wDhGNW2kf8ACda/16+Rp/8A8i1Guh3qXwtG+IGtNctE0gt2j0/fs+7v2i2zgEjnpnGfSgDqKK5v/hF9X/6HvxB/340//wCRaD4X1fP/ACPXiAf9sNP/APkWgDpKK5xvDGrE8eOdfXgDAgsP/kWk/wCEX1fH/I9eIPr5Gn//ACLQB0lFcvb6Ff3Mcv2fx/rU+yQxuyR6e2xl6qcW3B9QeakHhjVgefHWvnjvBp//AMi0AdJRXN/8Ivq//Q9+IP8Avxp//wAi0Hwvq5J/4rrxAPbyNP8A/kWgDpKK5s+GNWJ48da+OO0Gn/8AyLR/wi+r4/5HrxB9fI0//wCRaAOkornB4Y1YIR/wnOvknGG8iwyP/JWkHhjVgefHWvnjvBp//wAi0AdJRXK22hX95D5tn8QtbuIw7IXij05l3KxVlyLXqCCCOxBFTHwvq5J/4rrxAPbyNP8A/kWgDpKK5s+GNWJ48da+OO0Gn/8AyLSnwzq3lhf+E414EEkt5Fhk9OP+PXHb9aAOjorm/wDhF9X/AOh68Qf9+NP/APkWlXwxqwPPjnX24IwYLD/5FoA6Oiub/wCEX1f/AKHvxB/340//AORaD4X1cn/kevEA9vI0/wD+RaAOkornG8Mas2MeOdfXAA4gsOffm1pP+EY1baB/wnWv9evkaf8A/ItAHSUVzY8Mavg/8V1r5/7Yafx/5K0DwxqwPPjrXzx3g0//AORaAOkorm/+EX1f/oe/EH/fjT//AJFpX8M6s0jMvjjXkBJIVYLDC+wza5oA6Oiub/4RfV/+h68Qf9+NP/8AkWl/4RjVtoH/AAnOv5znPkWGT/5K/wCc0AdHRXLQaFfXEk8dv8Qtbme3fy5lSPTmMT4DbWAteDhgcHsQe9TL4Y1ZWyfHOvsPQwWH9LWgDo6K5v8A4RfV/wDoe/EH/fjT/wD5FoPhjVixI8da+MnoINP4/wDJWgDpKK5s+GNXOP8AiutfH/bDT+f/ACVo/wCEY1baB/wnWv8AXr5Gn/8AyLQB0lFc4vhnVgrA+ONebcMAmCw+XnqP9F/Dn1pB4X1cH/kevEB/7Yaf/wDItAHSUVzf/CL6v/0PfiD/AL8af/8AItK3hjVmYkeOdfUE5AEFhgf+StAHR0Vzf/CL6v8A9D14g/78af8A/ItL/wAIxq2zb/wnOv5znd5Fhn6f8euKAOjormx4Y1YA/wDFda+cj/nhp/H/AJK0Dwxq4OT4618+xg0//wCRaAOkorm/+EX1f/oe/EH/AH40/wD+RaD4Y1YsSPHWvjnoINP4/wDJWgDpKK5xvDOrFVA8ca8u0YJEFh83PU/6L+HHpSf8Ivq+P+R68QfXyNP/APkWgDpKK5weGNWAOfHOvnIwCYLDj/yVpB4X1cH/AJHrxAfbyNP/APkWgDpKK5aDQL+5Mn2b4ha5N5TmOTy49ObY46qcWvB5HFTP4Y1ZnJHjnX0B/hEFhgfna0AdHRXNnwxqxA/4rrXxgf8APDT+f/JWj/hGNXxj/hOtf+vkaf8A/ItAHSUVzY8MasAf+K6185H/ADw0/j/yVoHhfVwf+R68QH28jT//AJFoA6Siub/4RfV/+h78Qf8AfjT/AP5FpX8Maszkjxzr6A/wiCwwPztaAOjormz4Y1Ygf8V1r4wP+eGn8/8AkrR/wjGr4x/wnWv/AF8jT/8A5FoA6SiubHhjVgD/AMV1r5yP+eGn8f8AkrSp4Z1ZZFZvHGvOAQSrQWGG9ji1zQB0dFc3/wAIvq//AEPfiD/vxp//AMi0Hwvq5/5nrxAP+2Gn/wDyLQB0lFc4fDGrEDHjnXxgYJEFhz/5K0n/AAi+r4/5HrxB9fI0/wD+RaAOkornF8MasN2fHOvtkYGYLDj3/wCPWkHhjVgwJ8da+cHoYNP5/wDJWgDpKK5v/hF9X/6HvxB/340//wCRaD4Y1cnjx1r4+kGn/wDyLQB0lFc2fDGrED/iutfGB/zw0/n/AMlaUeGdW8sr/wAJxrxJIIbyLDI68f8AHrjv+lAHR0Vh2GgalZ30c9x4t1i/jTO62uIrMI/Hcx26t78MOlblABRRRQAUUUUAZPifXV8N+HbnUzAbl49iQwBtvmyOwREzg4yzAZwcdapDUNc0DRb/AFbxZdabcW1ravcvHp9rJE0WxdxXLSN5nGRnCdOnPGj4g0O28R6DdaVevJHFcKP3kJAeNgQyupIIyGAIyDyKo2miavdabc6f4s1Wz1W1uLY27JbaebYuGGGLkyvkkf3doGTweMS72dt/6/r+mUrXV9jGbxJ4l0mHSNW19dLOm6pcw272ltFIs1kZiBGTKzkS4YqrfInXI6YNO+8YeKlsvFWp2kOkw2Phy5lUJNHJJJepHGrsuQ6iI4JG7D5J+6MfNrW3g7VJTptrr+vRajpmlypNbwpY+VNM8f8AqjPJ5jB9vDfKiZYA8DgzzeC/O8O+KNL+34/4SCWeTzfJ/wBR5sYTGN3zYxnqM+1VLq4+dv8AyW3/ALcTHePN5X/G/wCg+08TXH/CYTaZqEcMNnNpqahYyjIZgDiZW5wdu6M8AcNUOkfafGfhG1v9bnuNPhuZHuEisJ5LVmtyx8nfIrBwdm1jtZeeDxkF3ifwX/wkOm6fbw6gbGezVoWuEh3NJA8flyx/eBXcMHOeCqnBxR428JX3ifw7DoulaumkWm9Rcx/ZmlFzCB/qSVkRlU9DtYEjjIyabt0/rXf7reejCPS/9af8P5ao5vw/4tudN8IatfWkk2r2Ta2LDQJbydm+0LIyRrmZss8YlZwHO4lV6nitDTYdZh+L8Q16+sbyQ6FJsNlZPbBR56ZBDSyZ+uR9K0J/CGo6j4VbR9R1LTovs7wSadJpmmNbpaPCweP5GmcMoKr8oK8ZHfILfQ5dG8Qy+LvFHiKB2jsGtZiYRbWsKb1YMoZ22Dg53MxJbqAAoNFJPtf7uW3333Jabi1/XxJ/dbY62vLdTWbRda1/VvGFj4qbSlvhNBf2GsyLbW1r5cYyYIrlXwHDk4iPBz0rrv8AhY3gj/ocvD//AINIP/iqZrnh7WfEkdzp95rlnHoN4NksFrYMty8RHKeeZmXDdCRGDgnGDyErp3/roVpazOlR1kjV42DKwBUg9RXP+PNVutI8HXUmnSeVe3LxWdtLjPlyzSLEr/8AAS+fwrfiiSCFIoVCRxqFVR0AHAFY3jLRJtf8J3dlZMqXq7J7Rn+6s8bCSMn23KM+1Dt8v0CN/mZHiZz4C+Gv2Xwwq28weGytZZBv2SzSrH5z5+825y5Jzk9c5rStdEs/CtpNqcmpaxdG3t2e5ku9QmuBKFXLN5bMUU8Z+RV9BgcVAE0/4k+A5bbUbe4tFugYbmAnZNaTo3IBx95HXIPIOAeQas6do+u+S1p4k1qz1WzMJiZYtNMEk2RtJkbzWU5Gc7VTn0HFKXNZ237/ACBW0vsYH/CWeJLLQ9L8TatFpo0nUJbdZLCKNxcWiTsqoxmL7ZCpddy+WvU4PHMV94v8UtH4uvNPj0mGy8MzSACeKSR7xUgSUpw6iI4bG/585Hyjb82ha+B9R+x6fpOq68l7oemyxyQW4silxKImBhWabzCHC4XO1ELFRk9QbbeDN2k+LLL7f/yMcssm/wAn/j33wJDjG75sbM9uuPenK3vcvnb8LfqOFrx5/K/43/QLHxRcy+MY9OvYYYbG+0pNQsZRncWU4mRjnBwHiIwBwT1xUWjPP438Jwajq9xcWNpczSXEEenzyWrSW24+SXkVg4JXa52soOcEYyC7xH4JOu6Fpllb6k1jc6evli7SLczRtEYpExuGNytkHPDBTzil8Z+ELzxJ4Zt9C0fV00eyVlW4i+zNKLiBRjyCVkRlQ9DtYEjjIyacrXdv613+WnnoyY305vn9235+W3y4r+1b26+HGsPZarqEul/8JDBa6Xffa5BNLatcQq+2cEOy7mlUPuJIHU9a6O5jm8HePPDtrp2oX8+m6281rcWl9eyXWyRI2kSVHlZnU/KVKg7SCDjIybd74Q1jUvCLaJc6rpMHlS20lk1jpDwxQCGRXCmMztuHyAYDLgVcsfDF7J4it9c8S6rHqN5ZxvFaRWtqba3gD43PsLuzSEDG4vgDgKOSTr/Wui/P89R/Z+/5f1/wClGh8P8AxWS2tDtsfENpLcSwA/Kl1CUBkUdi6ON3qUB6k57KuP07PiX4hya1EG/szRYJbC1lPS4uHceey99qeWqZ7tv9K7CkvhX9ddPw/AHa7/r+vPzuFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFYnjPWZvD/gzU9TtFVrmCA+QH+75rfKmfbcRn2rbrJ8U6L/wkfhXUdJEvkvdQMkcuM+W/VGx3wwB/Cplsxrc4/xppj+Efgrc2ek3l1b3KNbiW9hmZJ5ZHuIxLIZAQdzFmJPvUy65ejXvC1nd3kiz2t1eWmookhxKY4Cys4HB3LskAPTcO9WWR/iX8PJ9Muy+kalHNFDqETx+YbaeJ0kZcZG4MACGBwVcH2rRv/B/23x3p/iFL3y4rVHE1mYdwnYoyBt275cK2Dwc4XpirnfXz/4FvkT0/r5/MwW8a+IU8Iw+N2j00aDIEnbTjE4uVtmIAk88vs3AEOU8vGMruz81WLnxN4pu9S8VRaQuk29toLKYpbqKSVrnNuspjIV12cn7/PBA2HBJmi8A3q6TD4el1yN/DMEilLMWZFyYlYMsDT+YVKAgL/qwxUYznJOtB4W8mbxK/wBsz/br78eVjyP3CxY6/N93Pbrj3qJ35Xydnb8Lf1+hcbX97+v6/rU53VfiVCkuiWkWp6NoM2p6aNSkutZlBjhQ7QsapvjMjMS3O4YCE4OcVv8AgfxQPFehzXJls55bS6ktJZ7GTfbzsmP3kZyflYEHGTg5GTjJqjwZeWKaNcaFq8VrqOm2C6e8tzZmaG5hAH3ow6sCGUMCH4yQc540b+e903wldPrdzLdXXlshl0XTpd+W4Xy4g0jZGRzkjucCrk0rtef56fhoZxUmkn5flr+JuUVzXw7sNe0z4f6VaeLrqS61eOI/aJJZN78sSqs2TuIUgE5OSOp610tElZtDTujzzxLA2heJPD66PrOqT65qGpp5lvNfSSxzWuSZyYM+WiqnRlRcELzknNLVdXsZtf8AENv4n8SXei6laShdGtYb6S33xGNSkkcSsPtTNJuBUh+V27fXV0nwT4l0rxJqGtHxHpV7eX8mHnu9GkaWOAHKwRkXIVEHsvJ5OTW7q+ka5qtxLbprlvZ6TMu14rewP2rbj5gJmkKDJ7+VkA8EHDVFvdsv+G2/K34spfFd7fn/AF+iHeCdYuPEHgTRNXvlC3N7YxTS7VwN7KCSB2Ga3K4j4h2F5F8PD4f8Hf2jY6hKkdvpp0tHUQ7CuA0o4iTaMEsRkZAyeD2Vqk0dnCl1IJZ1jUSOBgM2OT+daSalJtdyUmkkwuYWuLWWGOeS2eRCqzRbd8ZI+8NwIyPcEe1cJp+NK+K8WieH9R1G8t002SbVoLu/mu1gfcvkHdKzeW7Zf5QRlRnHANdxqEV1PptxFp9ytpdSRMsNw8XmCJyOGKZG7B5xkZrlPBng/XPCaCCXWtMvoZZWmvZv7KlS6vJCOXeU3DDdnH8OMDAAGMSvi/r+vP7vlX2f6/r+vv4mbxVFaeFf7cl8SXY8aW9wj3ukm8kOz98EeE2e7ase04EmzOMPuJOT7TXN3XhzU9Zulj8R6ta3OlxzrPHZWdiYPMKOGQSu0jlwpCnChMkc5GVrM8aR63qXi3wxZ+GrnUrV7S+W61F0jdLV7TBDI7kbXYnACAkjOSABkC2S8/8AL/hxPdv+v66Hb1n65b2tzo8637Xy26L5j/2fLPHMQvPymAiQn2Xr0wa0KqanHqMtkV0e6tbW53DEl1bNOgHcbFkQ5991JjOP8EXDXl/4ksbHUtUjsI2iFnHqMjte2xaM7n23AMqoWGV80clWwNuMxWaf2R8VbDRtB1LUbyH7BNLrEF5fzXaw/d8hsyM3lsx38KRkZ44GNyx8Nahb3Gp6rc6vFLrt9arbLcx2ey3t1TcU2wlyTguWO5zk9CBwM/wd4O13wqSk2uaZfxzyma+nOlSpdXjkfeaU3DDPT+DAAwAKa+Jen/A/4L+7XdLZWX9f10/pN2qJ/YHxM0e+szsg8QGSxvoR915UiaSKXH94LG6E9wVz90Y7KuPmB8TfES0MKv8A2f4aaR5JsYWW8dNgRfXZG7Enpl1HUEDsKFblVv6/r8ge4UUUUAFFFFABRRRQAUUUUAY3i681Ow8Hard6DEZtRhtXe3RY/MJYDsv8R9B3PFcLYeI9HtPFfhhPCPii41xNVne31GKTUJL0AGB5FdgzEW77k+6AmQWG35fl9PuBM1tKtrJHHOUIjeRC6q2OCVBBIz2yM+orDsfD19Lq9vqnibUoNRubPd9kjtbP7PBAWG1n2s8jFyCVyXxg4Cg5JFvdg9v6/rQ5rxxCdBudNutM1nVZPEWo6pClpbNfSNHNGZB5qfZgfK8tIi2W2ZGAS245MfxU1TUr3R9X0rw9qNzpx0vT5NQ1C9tJCkiYVjDCrDoWZdzdwi4/jBq/D4K8SW3i/UPEKeI9Kubq6Jjga90aSRrODORDGVuVAHcnGWPJ7AT678MtH1nT9Zjhn1Cxu9XSQzyw6ldCIysm3e0AlCNwB8pGCAB0qGm4W9f+B/n6+WppFrnT/rz/AMv6sUvGGoy2q+GP7Rub7+x7oGK7j024eO7lmKKYyojIldR85ZY8t0JBUGr3wzvJL/RNQuI7+a701tQkGnrdXPn3EEIC5SViS27fvIVyXVSobBGAT+AZLa70W/8AD2rvbX+kwy26yakst+kscuN4IeUODlVIIfgDGCOml4W8MvoD6nd3l8L7UNWuvtV3LHD5MW4KEUJHuYqAqqOWYk5Oew10vJ97/mv6/Axs+VL0/JnQUUUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFDVptXhgQ6FY2V5KWw6Xl49sqrjqCsUmTntgfWsoX3jfv4e8P/8Ag+n/APkOr/iHxFZ+GbGG71BJnjmuY7ZBCgZt7nC8ZHfjjnmsbRPiNpmuaraWENhqNtJdNPEr3EcYRZ4CRLCSrnLDGcqCh7Nniktbg2luW1vvGxYb/D+gBc8ka7MSB/4CUhvvG+ePD3h/H/Yen/8AkOlXxlGmtWdhqWjappqX8jRWV3dJF5Vw4BYLhZGdCVBIEir0IODxUNz8QNLsYdaN/bXttcaPKkT2jxqZrrfjymhVWO8SH5V5ByCCBg0x21JTfeNs8eHtAPHfXZv/AJDo+3eN8D/intAz3/4n03/yHW/bTPcWkU0kEls8iBmhlKl4yR91tpIyOnBI9zWR4j8V2XhmTT4ry3vLmbUpmt7aK0h8xnkCFwuMjGduMngdSQMkD0Eiv9u8bbT/AMU9oGc8D+3Zv/kOlS98alvn8P6CBg8jXZjzjj/l09f8mmR+O9NXTdRutTtr3TJtNlSG5sriJXmDyY8sKImcPv3ALtJyeOoIE2neL4LvVm0vU9NvtEvvs5uo4dQ8r99EpwzK8Ujr8pIyCQRkHGDmgCP7d43/AOhe8P8A/g+n/wDkOg33jfPHh7w//wCD6f8A+Q6gtviDYTmyuH03U7fSdQlSKz1aaJFt52fhON5kQMeAzooPHPIzX1W3Nj8UvD0ttd36jUBdC5ga+maB9kQ24hLFFx1+VRzzR6ibsaDX3jbPy+H9AIwOuuzDnv8A8ulJ9u8b4/5F7w/n/sPT/wDyHVXx9qC2ljZR3MWvQWr3lvvv9HuIYvLYyqqo+5w5RiwBCqeK0/EXiqz8NS6dDd293cTalO1vaxWsW9nkCFwvUYztxk8DqSBkhJ3GVxfeNdhz4f0Ddxgf27Ng/j9kpBfeNs8+HtAHHbXZv/kOltvG+nPpWrXuowXeltozbb62ukVpIvkDrjy2dX3KwI2k5zjrxVfT/HTXvii30C48M6zp13cWrXaPeG2EYjBwSSkzHOSAQASNwJAHNPfT+u4dCf7d43/6F7w//wCD6f8A+Q6DfeN8nHh7w/jt/wAT6b/5Dqjc654l0PXtEi1x9KubXWbs2gt7OCRJLVzG0inzGciVQEIJ2R9c4HSqfj3xlLF4W8TW+gadql1LYWcyTajZPFHHZy+WW+88iMWUFWJjDYzjrxSk7K/9f1t95UY3lym0b7xtnjw9oB4767N/8h0fbvG+P+Re8P5/7D0//wAh1t6czPpdq7sWZoUJJOSTtFcz4/8AFNz4ZGiLb6hpemR6hfG2mvdVjLwwKIZHzgSR8koBy3eqkrOxEXdXLgvvGuw58P6Bu4wP7dmwfx+yUgvvG2efD2gDjtrs3/yHV3w1eT6jo63c2s6XrSSuTFd6VCY4WUcY/wBbJkghskN7Y4rXoatoM5v7d43/AOhe8P8A/g+n/wDkOg33jfJx4e8P47f8T6b/AOQ6i028u/EHja/njuZYdK0SQ2ccMZ2i6uCgaR3P8SoGCgdN24nJC4w7P4gaxdWfiGL7LZf2nDeSwaNHtcJOgmeFDJyeVeNi+0j5cHipvpf+v66jtv62OhN942zx4e0A8d9dm/8AkOlN7418sEeH9B35OR/bs2AOMc/ZPr/npBpXiy/u/Buiar/YV5qd3f2UVxPFpoiRYmZATzNIgxnIADE/zp7ePtJHh/TNcWO7fTNQnW3a5EYAs3LbP3wJBQBxsJAOD145qnFqVvO36E9B/wBu8b/9C94f/wDB9N/8h0q33jbPzeH9AAwemuzHnt/y6VneIvHjW+i+IJfD9u0s2jukBu7iLdatMzqGjGHDMVDfNjAB4zkEVJHr2vaN4x03RfEcum38GqwzPBdWNs9q0DRDcwdHkk3KQRhgwwRgg5BqbofkXPt3jf8A6F7w/wD+D6f/AOQ6DfeN88eHvD+P+w9P/wDIdV4fiHp8v2S6bTtSh0e9mWC11iWNBbys5wnG/wA1VZuAzIFORzggmTUPHVrZT332fSdU1Cz01tl/fWkUZitmAywIZ1eQqpBPlq+OnXIp7bh5f1/X+RK1941GNnh/QDwM512Yc/8AgJSfbvG20f8AFPaBnPI/t2b/AOQ6y/iP5Vz4FbXdN1C9ilgWKS1nsr+aFGV5E5KowVwQf4gevvXS+Iria20K5MFlf3jOhQrp0kSTRgg5dWkdFBHXrn2NKT5U7hH3nZGcL7xvg58PaBnt/wAT6b/5DoF942zz4e0Acdtdm/8AkOq+l+I9N0f4T2niJ59Tu9Nt9NS6Mt6wku5I9oOXOcM+OvNWtM8ZW2oa1Bp0um6jYNeQPcWMt5Eirdxpt3FQGLIQHU7ZFRsHpwcU1aTj2Fe6uN+3eN/+he8P/wDg+n/+Q6V73xqJGEfh/QWTJ2ltdmBI9x9kOKzJfibag2Mtt4e1q6sNRuxZ2eoRLbrDO5JAI3zK4U7TgsoB4xnIzZ8Yap4m0HTNT1yyl0hdO02A3BtZ4JXluURdz/vQ6iInkAbH6Ak84E30v0KSbfL1LP27xv28PeH/APwfTf8AyHS/bvG20Y8P6BuzyP7dmxj/AMBPrW7aXAu7KC5CMgmjWQK3VcjODVXXtYg8P+Hr/V7pWeKygaZkT7z4GQo9yeB7mnL3b3EvetYy/t3jfv4e8P8At/xPp/8A5DpVvvGpb5/D+gAeo12Y/wDtpWJq2peIPBvwtu9buLiK81+aSGaVbne1vC8skaGNEDAhEBwACMkFjyTWlF4tuJ77wxFHDCg1SS4hvEYEtDJDG25VOR0dSOQeBTegrq1yx9u8b/8AQveH/wDwfT//ACHQb7xtuO3w9oBGeCddmH/tnVY/ETTgkd4dO1IaJJOIF1sxxi1LFtoON/m7C+FD+Xt5Bzt+alvfiDaWuoaxZWuj6tqE2ilTffZYY9sSGMSBwXdQ3BPyrl8g/LjBK6XKs72LBvvG/GPD2ge//E+m/wDkOj7d422j/intAznkf27N/wDIdS3vi+ygGnJptrd6vdalAbm1tbJUDvCApMhMjIqqN6/eYElgACavaHrdtr+m/a7RJYikjQzQTrtkglU4ZHAJGQfQkHggkEGnZk3TszNW98a7W3eH9BBx8oGuzHJz3/0TjjPrSC+8b558PeH8f9h6b/5DrpKKQzm/t3jf/oXvD/8A4Pp//kOla+8bBjs8P6AVzwTrswJH/gJVOy8S65J8SIdD1KytLSyuNMmvI41YyTqUmRBvcHZyHyVAOCPvtVT4meN9S8K6TKvhm2tLvU4bdr2dbvcY4LZOrsFIOWPyqMjJyeimhapNdf8ANr9B2d2uxrfbvG/bw94f9/8AifT/APyHS/bvGuzP/CP6Buz0/t2bGPr9krokO6NSepGaWgSd9TmxfeNsHPh7QOnH/E9m/wDkOgX3jfPPh7QAPbXpv/kOt+7llgs5Zbe3a5lRCUhRlUyHsoLEAZ9zXOeEvEGratrXiHT9bhs4pNLuYokW0LkAPCshBZvv4LY3bVzj7ooW4En27xv/ANC94f8A/B9P/wDIdBvvG247fD2gEZ4J12Yf+2dZGueONStvHulaRo1taTaYb5LHU7qbcWWaSNpFjjwQNyqmWznG9feu7oWquD0dv6/rQ5xr3xrtXb4f0EnHzA67MMHPb/ROeMelJ9u8b4/5F7w/n/sPTf8AyHXSVV1KPUJbFl0i6trW6yNst1bNOgGecoroTx/tD8aAMYX3jbBz4f0AHHGNdm5P/gJSC+8b558PeH8f9h6f/wCQ6i8P65rN3f6/pV8NPvb3SWjEVzah7eGZnj3iN1JkMbDjPLcMpxziobbW/EGl+NNM0PxBLpt8mq288sUthayW5t2i2khg0j71IfAI24I6HPBu7B0uW/t3jf8A6F7w/wD+D6f/AOQ6V77xqHOzw/oBXsTrswJ/D7JVc315oPxCh0+5uJLnS9eWR7UysWNrcxqC0YP9x0BYD+Eq3Y4HV0dLh1sc2b7xtgY8PaB05/4ns3/yHR9u8b4/5F7QM/8AYem/+Q66SigDmxfeNsHPh7QOnH/E9m/+Q6BfeN88+HvD+P8AsPT/APyHXSUUAc39u8b/APQveH//AAfT/wDyHSvfeNQ52eH9AK9iddmBP4fZK6OigDmzfeNsDHh7QOnP/E9m/wDkOj7d43x/yL2gZ/7D03/yHXSUUAc2L7xtg58PaB04/wCJ7N/8h0qXvjUyKJPD+gqmRuK67MSB7D7IM1o+Idah8O+G7/WLmNpY7KBpTGn3nIHCj3JwPxrll8Wa3o3ijRtN8TzaTN/bB2G3sIZFlsHYEx72Z2EiEqybtseWHAPIBu7A9Fc1/t3jf/oXvD//AIPp/wD5DoN9437eHvD/AP4Ppv8A5DrpKKAOcN942wMeH9AJxznXZuD/AOAlJ9u8b4/5F7w/n/sPT/8AyHXSUUAc4t941+bd4f0AccY12Y5P/gJSC+8bbhu8PaABnkjXZj/7Z10lFAHN/bvG/wD0L3h//wAH0/8A8h0G+8b548PaAR769N/8h10lFAHNm+8bYGPD2gdOf+J7N/8AIdKL3xr5ZJ8P6DvyMD+3ZsEc55+yfT/PXo6KAMOwu/Fcl9GuqaLo9tanPmS2+ryzOvHGEa2QHnH8Q/pW5RRQAUUUUAFFFFAHPeMNLvNVt9JWwh802+rWtzKNwXbGj5ZuSM4HYc1xmleD/ElneaVNHaJBLa6nrNx5kkqMsYnMhgYgEkg7l4GSO4FeqUVPLo13/W3+SB6u/wDXX/M8ksPC/iO51DwtcX2kayLvT7+OfVLvUde86KRvLkDPDAJWTbuP92MqCoCkFtt7X/C/iLXfEsvi6LToYNR0GUJoljOYW+1xjJlZ3GdpkBwnI2bQTgsa9Noqv+H/AC/yDf7rfn/n6kVtK89pFLNbyW0kiBmhlKloyRypKkrkdOCR6E1xfj66msvFXgqe2sLjUJE1KfFtbNGJH/0SYcGRlXjryw6fhW5e+DtMv72W6nudaSSVtzLBrt7Cg+iJKFUewAp9r4S0m1ls5Qt5cS2U7XFvJeahcXLRu0ZjJBkdjjaxGOnOcZ5pp9Q6WOTvvCutaz/aevSWKWt/Lf2V3Z6bcSqxKWpyFkZSVDuS+NpIHyc9au3Wjal4y8QQ3up6Zc6HZWmn3VoiXM0TzSyXAVS2IndQiqvdsknoAMnuaKnlVmun+as/wBNp3W/9M85GleIdX8L6R4Uv9CewWxltftepfaIjbyJburZhCuZMvsGA6LtBOTkAG74hm1uTx1o19Z+E9Vu7TSjcCSaKezUTeZGFUoHnU4z13Ba7miqbbd36isrWRxfxEXWNS0WHTtI8OX+oO1xa3TSxTWyIgjnSRkPmTKd2EPQEcjnrifUrTU9c1zwhqa6VcWaWV7PLdw3MkPmQKbeWNSdjspyzL90nrzjnHW0ULQb1VjyzxtptzY6H4/uL2MRW+sT2kdo3DtNmOKIhAMlXLZVSwADbSeBmr2gXKaX4xsx4is/Ex1a/t5LTT7rWWsnXag8x4lFq2FJC7izrzsA3cAV3Oq6TY65pc2narbLc2s4w8bZHfIII5BBAIIwQQCOaoab4P0bS9SXUIYrq4vEQpHPf30948Sn7wQzOxTPfbjOBnOKUdP68rDk7r+v6/r7uO0KLxDfeNB4h8YeENV+2Ixt9OijubN7bTIWOC4/0jc0jD77hc4+UDA5TVdD8TQeGfF/hvStF+1f2tLd3FpftcxLDtnG4owLBxIGLKPl2/dJYc16bRSaTjy+VvvHzNO/nf7v+HILCN4dOtopBtdIlVhnoQBmq+rajc6dAklno97qzM20x2bwKyDHU+bIgx24JPtVG+8IabqN9JdXFzrKSSnLLb65ewoOMcIkoVfwAq9pOjW2i27w2ct7IrtuJvL+e6YHGOGldiB7A4qm23dkJWVkYvg3RL2wvdd1bULdLBtavFuV09GVvs4EaplyvymRtuW2kjOACep6miikM47wXImmeIPEvh64O26TUJNShB6y29wdwcfR96H02jPUUeE/CYg0i3l12wRNUtb3UJbdi4bylnuJGyMEjlGHXkZxxzXS3Gk2N1qlpqU9urXlkHWCfJDIHGGXjqDgcHIyAeoFXKPs8vlb9PyHd3b87/wBfeeS6Z4c8VWdjoFlq+navdWFro6WwtdK1gWZt7oNhnmdJUZ1K7cbS+3DfLk83NM8N+JLX4W2vhQ6Wsc+o3l1FfzS3KXKWdtJPI5fLnMzFGAXIJy2WAwRXp1FNu6s/61/r5A273/ra3/B9Ty5/B2tWPw71rwPZ6b5tjC6nSbqGWOLzYWlVzG2GDCRPmy+Bu4OdxNdlB4L0i0juzbpcvc3Vs1s13eXk13MkbDlVeZmZVzztBAJAJFb9FTJcyd+olo7o83/sXxBqPgzTPBV1oj2cdo1rFc6oJ4vs7wwOjbolVzLuYRgAMigZOTwM2obLxB4etNb0XTdAk1FNQu7i4s79biFIU88liJwziQbGZs7EfKhcc5A76im/evzdb3+dv8gXupJdP+D/AJnnvibRNT0/4YWnhHQ9GvtYkis4IBcxS26IvlFPveZKpyQpxgEV013q+oP4eNxH4Z1R7mYtGbES2vmxjB+ZiZvLx9HJ5HHWtyiiXv3v1CKUWrdDzK30XX7z4CXHhWXQLqz1WDR1skjnntytxIEx8jJKwAyOrbetdHrmnXX/AAmOhauI1Fjp1jei6mdhiMuse3Izkg7G6A9K6qkdFkRkkUMrDDKwyCPSnUk53fXUIWjbyPEvCcsejeFPD+q+JLbxNPotq8dxBIxs/sEMkrYSVEQi5MeZPkVwdoYfLkDHT+MLXX/EHipNP1HwvqV74Usik3lWVxa/8TOUcgSiSZCIlP8ABg7mGTwAD0Nt8PPDdrJbmO0uXhtZBJBaTahcS2sTD7pWB5DGu3PygL8vGMYFdNR1uu9w1vcitZXns4pZbaS1d0DNBKVLRnH3TtJXI9iR71h+PdLudY8B6tZ2CeZdNB5kMeceY6EOq/iVA/GuhoqXqCt1OH8Vl/H/AMJzL4XUXh1D7NNCocJkLOjMCWIAICtkHnIx1pup+GtTb4l6bdWls0mjs89zczCVR5ErQGIgKTnDYQ/KMZ3E9a63TtIsNJa7OnW624vLhrmdUJ2tKwAZgM4GcAnGMnJ6kmrtOST06f1p+AtbHmUfh/xBJ8Prf4fvoskUcSR2UmsC4iFu1ujDMiqH83zCg4UoAHP3sDJ3bTQtRin8bl7fC6pKGsz5inzR9kSP14+ZSOcflXYUVM1zpp9f1Ki+V6Hl8/gzULZ/DmpTabql8bbRY9NvLPStXaznidQGDArNGjrncpBf+6Rnt1mhWEWieFb2XTNMutKnlMty8WsXvnuZduN8kvmycHaP4zgeldJVXUdMsNXsns9Wsre+tXILQXMSyIxByMqwIOCAaqV5X6Xv+Lv+ZMUo28rfgrfkYnw88Q6j4r8AaVretWcdleXkRd4os7MbiFZcknDABhyeD1NdLQBgYHSinJpu6BaI89nk19/ipa64vg/VjYwabNYM/wBosslmmRg4H2jO3CH36cVT8W+BfFd1ofiw6Xqum30+txsBDLprLOYwMRwLMbgIoUE4JXGWYkZJr06ilHRJdr/i7/mVfW/p+BxvjPXPEHg/4YXmqwvpl9rFmqkb4Hhhly4G1U8xm3bTgDfy31xXW2ssk9nDLNEYZJI1Z4yeUJGSPwqO502xvbm2uLyyt7ie0cvbyyxKzQsRglCRlTjuKs0X3uTa1kiO5leC1lligkuXRCywxFQ0hA+6NxC5PuQPeuE8Lrr8Pi7xJcXXhnUtOh1q4jlt7qaW0kWDZAqZkVJyx+Zeig9R07d/RSKueW3PgTxlY2vh61sNY0m9FnqwvJ7htJdJC5STfPLm6/eEs/KqFOWGMAYro/GPibWfDup+FbbTrW0vhqupLZ3ituWQIVJaSNc9FwWOc4GPXNdfVUaZYDVTqgsbYagYvIN35K+aY852b8Z255xnGapfrf8ALT8BPX7rfn/mWqz9dYf2PNG+iy63HKPLksYhCfMU8HImdUIx1BP4GtCipeoHE+GtJv8ARrnWNS07QG0nTpLVBZeHo5olJmTeWfajGGIvuUfKxzgMxB4FDwXba7/wkUuteLvDGqLrV8PKa6aa0a10+EciGILOz7cgZYLlm5IAHHotFPrf+v6/TRaA9V/X9f1c4/X/APibfEfw1p1qQzaU0uqXZB/1aGJ4YwfdmkYj2jauwqpZaVZafcXlxaQCOa+l864kyS0j7QoJJ7AAADoO1W6NbJMOoUUUUAFFFFABRRRQAUUUUAZHizRG8SeEdT0iOYQSXdu0ccpGQj9VJHcAgVyOl+H9Uv59HtrvQrjSUsbxdS1K9u7qKd7+4VCo2MjFmG4g7nCYVFUKBwvotFC0d/60B6q39ahRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFUNW1m20WBJryK9kV22gWdhPdMDjPKxIxA9yMVlDx3pB/5c/EH/hOah/8AGKAOkornF8daSzBRaa/knAz4dvwPz8mkPjvSAcfY/EH/AITmof8AxigDpKK5s+O9JB5tNf6Z48Oagf8A2hR/wnekYB+ya/8A+E5qH/xigDpKK5v/AITvSdpP2TX+uP8AkXNQ/wDjFKnjnSXbAtNeHBPPh2/HQZ7w/wD66AOjorm/+E80j/nz8Qf+E5qH/wAYoPjvSAf+PPxB/wCE5qH/AMYoA6Siucbx1pKnBtNf6A8eHb8/+0aT/hO9Ixn7H4g/8JzUP/jFAHSUVzg8daSULfZNfwMZ/wCKdv8AP5eTSDx3pJPFpr/TPPhzUB/7QoA6Siub/wCE80j/AJ8/EH/hOah/8YoPjvSASPsfiD/wnNQ/+MUAdJRXNnx3pIPNpr/TPHhzUD/7Qo/4TvSMZ+x+IP8AwnNQ/wDjFAHSUVzg8daSULfZNfwMZ/4p2/z+Xk0g8d6STxaa/wBM8+HNQH/tCgDpKK5v/hPNI/58/EH/AITmof8Axig+O9IBI+x+IP8AwnNQ/wDjFAHSUVzZ8d6SDzaa/wBM8eHNQP8A7QpT450kRh/smvYJIH/FO3+eMdvJz3/zigDo6K5v/hO9I/58/EH/AITmof8AxilXx1pLHAtNf6E8+Hb8f+0aAOjorm/+E80j/nz8Qf8AhOah/wDGKD470gHH2PxB/wCE5qH/AMYoA6Siucbx1pK4zaa/yAePDt+f5Q0n/Cd6TtB+ya/ycf8AIuah/wDGKAOkormx470gg/6Jr/H/AFLmof8AxigeO9JJ4tNf6Z58OagP/aFAHSUVzf8Awnmkf8+fiD/wnNQ/+MUr+OdJjkZGtNeJUkHb4dvyPwIhwaAOjorm/wDhO9IH/Ln4g/8ACc1D/wCMUv8AwnWk7Q32TX8E4/5F2/z+Xk+9AHR0Vzf/AAnekf8APn4g/wDCc1D/AOMUq+OtJdsC01/Pv4dvx/OGgDo6K5v/AITzSP8Anz8Qf+E5qH/xig+O9JViDaa/wcceHNQP/tCgDpKK5s+O9IGP9E1/n/qXNQ/+MUf8J3pO0H7Jr/XH/Iuah/8AGKAOkornF8c6SysRaa98oyc+Hb8d8cfueevakHjvSCcfY/EH/hOah/8AGKAOkorm/wDhPNI/58/EH/hOah/8YpW8daSrFTaa/kHBx4dvyPz8mgDo6K5v/hO9I/58/EH/AITmof8Axil/4TrSdm77Jr+M4/5F2/z+Xk5oA6OiubHjvSSD/omv8DP/ACLmof8AxigeO9IJwLTX/wAfDmof/GKAOkorm/8AhPNI/wCfPxB/4Tmof/GKD470lWINpr/Bxx4c1A/+0KAOkornG8c6SqqTaa98wyMeHb898c/ueOnek/4TvSMZ+x+IP/Cc1D/4xQB0lFc4PHWkkEi01/5Rk58O34/9o80g8d6QTj7H4g/8JzUP/jFAHSUVzf8Awnmkf8+fiD/wnNQ/+MUr+OtJRyptNfJHp4dvyPzENAHR0VzZ8d6SAP8ARNf5Gf8AkXNQ/wDjFH/Cd6RjP2TX/wDwnNQ/+MUAdJRXNjx3pJB/0TX+Bn/kXNQ/+MUDx3pBOPsfiD/wnNQ/+MUAdJRXN/8ACeaR/wA+fiD/AMJzUP8A4xSv460lHKm018kenh2/I/MQ0AdHRXNnx3pIA/0TX+Rn/kXNQ/8AjFH/AAnekYz9k1//AMJzUP8A4xQB0lFc2PHekkH/AETX+Bn/AJFzUP8A4xSp450mSRUW014FiAN3h2/A/EmHAoA6Oiub/wCE80j/AJ8/EH/hOah/8YoPjvSB/wAufiD/AMJzUP8A4xQB0lFc4fHWkgAm01/5hkY8O35/9o8Un/Cd6RjP2PxB/wCE5qH/AMYoA6SiucXx1pLbsWmv/KMnPh2/H/tHmkHjvSWYAWmv8nHPhzUB/wC0KAOkorm/+E80j/nz8Qf+E5qH/wAYoPjvSAebTX/w8Oah/wDGKAOkormz470kAf6Jr/Iz/wAi5qH/AMYpR450kxl/smvYBAP/ABTt/nnPbyc9v85oA6OisOw8XabqV9HaW9trCSSZ2tcaJeQIMDPLyRKo6dyK3KACiiigAooooAKztM8Q6Lrc08WjavYahJbECdLS5SUxE5xuCk46Hr6Gsj4j293dfD/U4rGGW4YhDNBApZ5YBIplRQOSTHvGByc4qrca1o3iXwdqNp4A1LT7q/j0uRLQWMik225MIp2/6rkLhTg/L0+U4luyb7FJXaXc6C08R6Hf6tNpdjrOn3OoW+7zrSG6R5Y9pw25AcjBODkcU268T6DY3EVve63p1vNNMYI45buNGeQYygBOSw3Lx15HrXnz6hoOs6J4P0bwfNbPqmn31pILSEDztPjjIE/nIOYvk3owYDLMF6morrT7NPBHxXu1to/tFxcXiyy7fmcJbrtBPoMkgdsn1qpe7fyv87cv+fnsTH3nFd7fK9/8j1SO8tprqa1huYZLi32maJXBePcMruHUZA4z1rPvPEEMNr5ul2txrcnnNB5OmmNyHXO8M7MqIVwc7mHPHJ4rjfEF83g7VNO8URR+Yuo6Z/Z0yDOZbhV8y2HpyfMT6utWtT1/TvhF8ONPiv3We+fEMMbyrH9qu5CWdmdsKil2ZmckAD8AW1b8vx/r70Ebytbr/l/X3M6fQvEtnr1ndTRxzWctjO9veW12FWS2kUAkNglSMEMCpIIIINZWmeOdP13x0NI0HU9M1OyXTmuZZbO4WZkkEiqFJViAMEnBGa4iVLBvhnfXC69p3iC4udXtb/xJJpkyzQrGZY/NQKpJESxpjnkqrE966bTNc0/Xfi/FPo8y3dnHociLdwkNBK3noSI3HD7eASMgHjOQQCz5l87/APgN/uvt6Et+42vL/wBKS++253tYN14qRNWm07SdJ1DWZ7VlW7NkIlS2LAEBmlkQFsEHapYgYyBkZ3q8Yl/s/R7bxTb6pr+pWXiZdRu7rTbO2v5IZJ/MO+AxW6kLcBjgfMrjIZTwMCb2fyv+Whdrr52/PU9nqpqup2ui6Rd6nqMnlWtpC00r46KoyfqfanaabptKtDqIUXhhQzhegk2jdj2zmua+J+R4Gklb/j2hvLSa79PIW4jaTPttBz7ZqmteUmLTVzRstdntPCMmu+Lkh0pER7mSMEt9nh6qrnnc+3Gdv8XAz1L9H8Qz6tKvm+HtW02CRN8VxepCFkHUfKkjOhxzh1X0ODxWX8TrC81HwDcrpts97LBPb3RtYhlrhIpkkZAO5KqcDvVhPF2j+J9JurbwnrdpdalLaO0UUMoMsDFTtMifejwcA7gMHjrxUt2TaW3T5DSvZPr1NSHxFolzrMmkW+safLqcWfMso7pGmTHXKA7h1Ham3nifQdPnSG/1vTrWWSYwJHNdxozSDBKAE8t8y8dfmHrXmiXuial8P/C3hzw5JCdfsrqyYWCY+02Ekci+e8qD5o8ASBmbGS2MksM2JLC0Ph/4u3RtojcTTXMUkhUFmRbGMqpPoCzHHuacvd5utr/O1v8AMcE5OK2vb5Xv/keoR3trLeTWkVzC9zAqtNAsgLxhs7Sy9QDg4z1waoXviGCG3L6XbXGtyidrcwaaY3ZJF+8rMzKiEdwzDngZJAridYvh4Pn0PxeI2kju9KGm3SLyZJBH5tv+JcPGPeUVbvtfsvhH8NdPXVXSfU5zsWNpFiF1eSku5Z2wqJvZiWY4A/AFyVm120+d/wCvvRMfetbrr8rf5/kzWf4g6fH4bu9Vaw1DzbK8SwudOKRi4jnZ1RUOX2HO9SGDlSDnNW9N8Xw3euJo2paZf6LqM0TTW8F+Ij9oRThijxO6ErkZUsGwc4xzXnV1NpVv8Lr6+bxLpWsahca1ZX+sXVjdI8EMjXEI2gg/LGqoAC3JCknvXS3mpWXjP4heGJPDN1HqNpo8k91eX9o3mW6bojGsXmKdpc787QSQBk4BGS2tv62X6/gPTlv6/wDAOh0bX7ibxDqOg6zFHDqFr/pEDxAiO5tWYhHXJJDLja49QCOGFb9cfqgEnxh8PC3GZYdLvWuCB0jZ4QgJ92Bx9DXYUlrFP+t2v68we/8AX9f8AKKKKACiiigAooooAKKKKACiiigAooooAKiu7qCxs5ru7lWG3gjaSWRzgIqjJJ9gBUtcr8Tkd/hlrmwFlW2LyqBnMQIMgx3+QNxSew1uQ/8ACbT6b4CuPFviDTporV3SS1srZAbgQyMqRhwzBfMJbcRkBQcckEnYHiewa40SKMTP/bQZrZ1UYULH5h3ZORx7Hmud+LDxy/Ce8ktypiaSzZCvQqbiLGPwrLvlbTPijoOhxlUU3F5e2IxwEkgcuvfO2XcfYOopydk7f1a34/5E30v/AFqd5/wkmh/25/Yv9s6f/av/AD4fak8/7u7/AFed33eenTmm33ibQdLYrqet6dZsJRCRcXccZEm0Ns5I+baQcdcEGvK0udNn+D9n4SglhPi9XhVtPyPtcV8JQzzsgIYANukMnQrzkg879np9o+rfE66ltonuJSsEkjICWjFkh2/TLHj3qZy5It9k362t/X4+RcY8zsd7qer6boln9r1nULXT7bcF867nWJMnoNzEDNT2t1b31pFdWU8dxbzIHimicOjqeQQRwQfUV5Cbm80y+8Gatc6xpekWK+HEht77V7R5oI52CF13CaMI7IowWPIVgO9dr4Ljt9I8J399baxZ69bTXNxeiTSIQIRnl44lEj/xBjjd1Y1crRvfpf8AB2/4JnFuSVlvb8Vf/gHXUVjeE/FGn+M/C9nr+jiYWl4pKLOm11IYqQQCRkEEcEitmhpp2Y07nO2HjbTdT8VDQrOG7d3tZLqO7aLbBKqSBGCEkM3LDDBdpHRjRrvi8aC1zJLoWr3VlZJ5l3fW8UflQLjcTh3V5Aq8kxq/cdQQOSufGXhj/heVjL/wkek+XHolxbO/26LCzG4ixGTu4bg/L14NO+IfjXS7zU5fAceu6dpLXEWNWvr25jiFtbuOUjDkbpXU8dQoOT2FRduEWt3f82vusrl6czT2Vv0/4Y9Jt54rq2iuLdxJDMgeN16MpGQR+FSVzWs+KtF8HfD8a9GJb7R7OCIRnTgsxaMkKrKchSvIOc4xXQ288d1axXEB3RyoHQ4xkEZFW7XdiFeyuJd3UFjZy3V5KsMEKF5JHOAqjkk1k+HvFVp4kutSt7S1vLd9OlSOQXcPlFt6B1IUncPlYcMFYdCBWvc3MFnay3N3NHBBCheSWVwqooGSSTwAB3rzrwL4q8PXvxE8Xw2eu6ZcS319A1qkV5GzXAW2QMUAPzYwc4zjBpLV28v8h9LnSXHjWGy1CCLUNG1Wzsbi6FpFqU8UawtKx2qCu/zVDMNoZkAJI5wQT0teS6l440Dxp42j0ufxDpWnaDoV8kk7XV7HFNqF3GwKRxoxz5SNglsfMwAXgE13fiPxjp3ha+0W11OO5P8AbN4tlbywx7kSRvu7zngHoMZ/IE01rFee34W/rtYHpJrt/X9edzfqlq+r2Wh6bJf6nKYoEKrlUZ2ZmIVVVVBLMSQAACSTV2uJ+J1s0mnaJeSzzW1hYavDcX80JwYodrqXJx8qgspLfwjLZGMiX0+X5j7/AD/I6XSNUutTSRrvRL/SdhG0XrwEyA9x5Uj4x/tY61mWfjnS7/x5deE7WK7e8tbdp5LnygLclSgaMPnJceYmQBgZ654rlLPUr+G710/Da5uvElgunxmAy6ibyJbxnKkRzzSfNtTDMm/A+X7pY5p6C1xovxK8PadF4X1mERaLcxSSXUtm0khe4gMlw5Scg/Nktj5iW4U84uC5pJPz/J/5fj6kvRP5fp/n+Z3lr4gni8ZXHh/V4o4nmjN1ps8edtxCMB1OekiMRkdCrKR3A364/wAV/vPHvgqK3GblLy5mbA5WAWzq5PtueMfUiuwqeiH1CiiigAooooAKKKKACiiigCK6uoLK0luryZIIIUMkksjYVFAyST2GKx/D/i2y8R6hqdnZ2t7byaa0Qk+1weUXEib1IUncOD0YKR6Vs3FxDaW0txdSxwwQoXklkYKqKBkkk8AAd6838FeKvD158U/GKWmvaZO9/PZ/ZFivI2NxttgG2YPzYwc4zjFC1dv63QPa5vXnxEsbSO6vBpWqT6PZTtBdatCkRghZX2OdpkErKrZBZUI4OCQDV7xl4z03wRoB1bVY7mdC22OCzjEkspwWO1SQMBVZiSQAFJryLUdSt103XZl1OKz1BNYdx4GLE/bWEnAMefNLSkrJmPERyCyONxPQeO31uax8Walq3hbUmhi0me006WKe0aGCJo8yysDMH3M2AQFJCoMZJIqG3yc3z/C/9eWrLik5287fjb+u/Q9C1LxRb2LWUFtZXeo399EZoLG0CeaYxjc5LsqKo3KMsw5IAyeKjt/GekS6DdarcSS2kdlMbe7gmjJmgmyB5RRN25yWXATdu3DbnIrmNQu7e3h0HxFda5Z+DdTS0e0WHXTDIlzCdpI2pOuSCqMCr5GcEc4rnrrRdUXQrnX9TvLpLS58T22p3Ey2xgdbVI0iaTyyC0Shl3DOWVFBJzlq0as2v6+JL8tX/kjNfCn/AE9G/wA9D1TR9VutUEputC1DSQmChvWgPm5z0EUrkYxzux1+uNOuD8IXVrL4zvI/CusXOr+HlslM00moSX0Ud1v4VJnZjnZyyhiB8pwM895R0uF9WgooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAqC8S6ezlXT5oYLkr+7knhMqKfUoGUke24fWp6KAOb+w+N/+hh8P/wDghn/+TK6MZ2jcQTjkgYpaKACorq1gvrOa0vIlmt542jljcZDqwwQfYg1LRQ9dGNNp3Rl+HNGk8P6JFpjX0t9FbkpbyTD50iz8iMf4iowu48kAZ5rUoooFtsFFFFABRRRQAUUUUAY+jeHxpmpalqd1cm91HUJcyTldoSJSfKhVcnCqCfqxZj1wNiiigAooooAKKKKACiiigAooooAKKKKACiiigApHRZI2SRQysMMpGQR6UtFG4GN4X8P/APCMaU2lw3klzYxSk2Uci/NbQkDEO7OXCndgnkKQOcZOzRRTbb3AKKKKQBVPVbGXUtNltINQutOaQY+02mzzFHfBdWAyOM4z6YPNXKKNwKWkaRY6DpFtpek2629napsijUk4HuTySTySeSTk1dooobvqwCiiigDJ8Q+HbfxNZpY6hc3K2Bb/AEm1hZVS6X+5Idpbbx0UrnocjitVVVECIoVVGAAMAClooAKKKKACsbUPDVrquvWWpajcXNxHYN5ttYsUEEcwyBNgLuZwCQNzFRnIAPNbNFABRRRQAUUUUAZFjoAt/Ed9rd3ctd3dyohhyu1baAc+Woz3bLM3UnHZQBr0UUbKweYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFUNW0LSNegSDXdKstSijbekd5bpMqtjGQGBwcd6yh8OfBA6eDvD/wD4K4P/AImgDpKK5xfh14JRgyeDtAVlOQRpcIIP/fNIfhz4IJyfB3h8n/sFwf8AxNAHSUVzZ+HXgljk+DtAPGOdLh/+Jo/4Vz4IwB/wh2gYHQf2XD/8TQB0lFc3/wAK68E7SP8AhDtAwTkj+y4f/iaVPh34KjbKeD9BU4IyNMhHBGCPu+hxQB0dFc3/AMK58Ef9Cb4f/wDBXB/8TWFLoHgeDxdbaFdfDfT4mvBIba9bTLIwTbEDNjDFx1x8yDn86A2PQaK4vXfCXgnR7OO6b4fafqJeaODy7HRYZZFDHaGIwMIvc9hWh/wrnwRjH/CHeH8f9guD/wCJoA6SiucHw78FBCo8H6AFbBI/syHBx/wGkHw68EqcjwdoA4xxpcP/AMTQB0lFcRZ+HfhXqGqzaZp+j+D7rUIN3m2kFravLHtODuQDIweDkcVLrfg/wDoeh3+r3vgvQ5ILKB7iVY9KgLsqqWOMqATx3IpNpK40m3bqdlRXMQ/D/wAD3VvFOvg3QdsiKy7tLgyARx/DSXHgHwFa2sk914T8OQwQqZJJJNNgVUUDJYkrgAAdab03EtdjqKK47SfC3w11yyefQtC8KajbB9jyWdnbSoGAzglQRkA9Pero+HXglTkeDtAHGONLh/8AiaAOkoriZvCvw9i8RW+iJ4M0Sa9miad1i0mAiCMcB5Dt+UE/KO5OcDAJEq+E/hpLb3lyugeFHhsXdLuUWVsVt2UZcSHGFIHJBxijpcZ2NFcvb+A/AV9axXVr4U8OXEEyK8U0enQOroRwQQuCMYwRUp+HfgoxhD4P0EopJC/2ZDgE4ycbfYflQ7oR0dFcpe+B/h7pljLd6j4Y8M2drEMyzz6fbxog9SxXA/GqmjeHvhbrpmPh/SPCGpGEAS/YbW1m8sNkDdtBxnB69cGgDtqK5v8A4Vx4I/6E3w//AOCuD/4mg/DnwQTk+DvD5P8A2C4P/iaAOkorgPFWheCfDNol7d/DnT760XCy3FrptmRbjcqruDsrEfN/CG6VrXfgHwTa2M0//CE6LOIUaQQw6TCzvgZwo28k4wB9KTdlcN3Y6miuQ0nwT4I1XR7a+HgPSrIXUSyG2vNHhjmjzztddpww7irY+HXglTkeDtAHGONLh/8AiaYdNDpKK5v/AIVx4I/6E3w//wCCuD/4ms3UNA+Ftjq0djq2k+ELbUbogx29zbWqTTFjgEKwy2Tke5o6gdtRXN/8K58EHr4O8P8A/grh/wDiaU/DrwTtCnwfoG0EkD+zIcD/AMd9hQB0dFcBZaP8MLvRrvWG8MeHbTSreYxfb7rTraKCYAgb0YjBTcdoY4yRxkYJ1f8AhCPh/BPbp/wjHhqOW5yIF/s+3DS/Lk7fl+b5eTjtQB1VFc3/AMK48Ef9Cb4f/wDBXB/8TQfh14JZiW8HaASTkk6XDz/47QB0lFc2fhz4IOM+DtAOOB/xK4f/AImj/hXXgnaB/wAIdoGAcgf2XD/8TQB0lFc4vw78FKrBfB+ggOMMBpkPzDOcH5fUA/hSD4c+CAcjwd4fB/7BcP8A8TQB0lFcVdeGvhbY6rBpl9ovhC21C4x5NpNaWqSy5OBtQjJyQRwK0m+HXgl2LP4O0BmY5JOlwkk/980B1Ojorm/+Fc+CD18HeH+On/Erg/8AiaU/DzwSI9reENACA7sHTIcD3+7QB0dFcXpPhr4X66sx0LRfCOpLEQs32O0tpQmegbaDjp39K0B8OfBAOR4O0AEdxpcP/wATQB0lFc3/AMK48Ef9Cb4f/wDBXB/8TVDV/C/wy0KFLjXtD8J6dHK+1Zb20tog7dcAsBk96AOzormG8B+BGtFnfwr4dNuke5ZDp0GxU5bIO3AHJP45rPn8M/DlfDcmt2PhHQNUsUQyh9O0y3n8xQcEptHzY54GTwQMniltuC1O3orlbLwN4Av7GK70/wAK+HZ7W5jV45YtNgKyIcEEELyOhqYfDnwQDkeDvD4P/YLg/wDiabutAWp0lFc3/wAK48Ef9Cb4f/8ABXB/8TSv8O/BUjln8H6AzHkk6ZCSf/HaAOjormz8OvBJAB8HaAcDA/4lcPH/AI7R/wAK58EYx/wh2gY64/suH/4mgDpKK5sfDrwSAQPB2gDIwf8AiVw8/wDjtA+HPggHI8HeHwf+wXB/8TQB0lFc3/wrjwR/0Jvh/wD8FcH/AMTSv8O/BUjln8H6AzHkk6ZCSf8Ax2gDo6K5s/DrwSQAfB2gHAwP+JXDx/47R/wrnwRjH/CHaBjrj+y4f/iaAOkormx8OvBIBA8HaAMjB/4lcPP/AI7Wdpnh74X6lqMtto2keEbu9tDmWK0trZ5ISDj5goyuDxz3oA7Wiub/AOFceCP+hN8P/wDgrg/+JoPw58EHr4O8Pn/uFw//ABNAHSUVzh+HXglgAfB2gEKMAHS4eB1/u+9J/wAK58EYx/wh3h/H/YLg/wDiaAOkornF+Hfgpd23wfoA3DBxpkPI9Pu0g+HXglWBXwdoAIOQRpcPH/jtAHSUVzf/AArjwR/0Jvh//wAFcH/xNB+HPggnJ8HaAT76XD/8TQB0lFc2fh14JIAPg7QDgYH/ABK4eP8Ax2lHw78FCMoPB+ghGIJX+zIcEjODjb7n86AOjorDsPBPhXSr6O90vwzo9ldRZMc9vYRRumRg4ZVBHBIrcoAKKKKACiiigDk/iFNeRaXpUen3s1lJcaxawPLCxDbGfDD8R68VwvhbV76y13Rrm/1q/ltmv9Ys5VubuSSNbe3ZzHkMTuZdud7ZcjjOOK9b1DS7PVUgW/h81be4S5iG4rtkQ5VuCM4PY8VnQeDNAt5IHi09c281xPGGkdgHuM+cSCSDu3Hg5AzwBUpNJ/P8l/k/vE1dr5fr/wAD7jg4tYuI/EPhjU9HtfE9vZ6tfiGW61W/WW3vYpInYFYDOxiJKhlxHHgZGADtpnifVNR0bxldeGbDXbiPTtclhe6v5J5Hl0RpSV2RsQQol2YjBI2Nk9CortbP4d+GLG4sp4LCVpNPkEll597PKLUgEYiDuRGuDyq4U4XI+UYtHwboLaZq2ny2AlttYlaW/WWV3admABJYsWGAABgjbgYxir2/rppp+v6FN3/rrrr/AF95r21ulpaRW8bSMkSBFaWRpHIAxlmYksfckk1yHjr7bceIPCWm2ep3mnQ31/NFctaSbGkjFtIxTPbO3qOR1BBAI0n1i/0phYWfhDXr63tlEcdyl1aOJFAwDuluRIfqwye9Y2u6VqHjPU/Dct1ouqaVbWOoSvdB76OCZUNtIqur28xbBdlXAbPXIxR8TuLZf12MbUNS1PQrjW9B0rVLsWw1PTraK8ubhriWyW5IEqiSXcScYK7ycGQdsCtHUJLnwV4utrbRrnUL61vNKvLiWzvb2W7KyQBCjq0rM67t5UgHB+XjPNdXD4U0WDRLnSRYrJZ3ZLXIndpXnY4y7yMS7NwPmJJGBg8CnaR4Y0vRLmW5so7iS5lQRtcXl3NdS7AchA8rMwXPO0HGecZqbOzV/wCrW/B6grXu/wCtThJRe6V4J8P+LLfXNRvNWvJrE3O+8d7e8Fw6K8awE+Ugw/ylFBG0cnnPSeIf+Sk+D/8At9/9Eirtj4G8P6bfRXVpZyqYHaSCB7uaS3t3OctFAzmOM8nBVRgEgYyaZqPgTRNV1YaleNqpu1ZmjeLWryIRbhhtipKAgI6hQKqTu9F1+7yJtoZ3xKgaHRrXVLa8v7a6t7+ziQ219NEjK9zGrBo1YI+QSPmB60vjk3k/iHwlp1pqV3p8N9qE0d0bWTY0sYtpHKZ7Z29RyOoIIBGtrvg3R/EnljV/t8iRBAscWqXMKZVtysVjkUFgcHcQTwOeBViLw1pkf9mlkuJ30uV5bSW5u5ppEZ1ZWJd2LN8rsPmJxnjoKUbLfuW327HBXuraj4c0D4gW2m6jdP8A2TJCNPa6na5mjMsEbFFeTczsWY7QxPzEDpxVjwvpbXPj6C+0uTxSNFt9NZZ49bur5Ue5ZgFxHcMC52b93BQfLjmtrxt4VlvvDGuL4dt0Oo6o8Ml0rSYNwsZQFVLZVHMa7VbjDYOQeazdG0XWD4u0+5sLLxBoumQJKL+PWtba8+1hlwixp58wUhsMXyhHQZycEd7+X6f18xStbT+tv69H9xfG31/x/oWjeGrSNLPwrcm4vruFQkVufKZFtY8dXO4FlGAqgZ5IFYvieIeIfAfjrWdZv7+NrFr2ztreLUJreGJIlKqGRGVZC5+Y7w2dwHQAV2ek/Dnw7oQiXSBqtrHFL5qwx63e+UWLbiSnm7WyeSCCD3zVjUPAfhzVbq9nv9PaX7epF1D9plWGY7du9ogwQvgAB8bhgYPAqZJuNurT+921/CxalaSfZr7lfT8bmvpn/IIs/wDrgn/oIrj/AB6l/f8AizwnpNotqYLie4nc3iGSISxRboi0YIMmCSwXcvKg5yBW3c63f6dcNZ2fg/WbyCEBI57eazCOAO3mXCt7cgdKVrSHxfpvl6/oN/pvkzB4RcXEaTIwHEkclvKxQ8kZDA4yOh5uXvO6M4rlVmReGtZ1O61jWNG1z7JNdaY0RF1ZRtFHMkilh+7ZmKMMEEbmzweM4HR1n6Poen6FavBpkLIJZDJLJLK80szn+J5HJdzgAZYnAAHQAVoUgRyPgXN3e+J9TnybqfWZrds/wRw4jjUe2AWx6uT3rkfDvh661OzvXt4YpNH1rUr46srMv/LG7m28dX8wbUb/AGVFdta6RqOjeO7q809Em0fWR5t5HuCta3KIFEoz95XVVUgcgqp7nG1pek2WjWP2PTYfJt/Mkl2b2b5pHLuckk8sxP48UmrwS62t/Xz1Kvq153/Pb0vY8x0DU7PUPBfhbRYrPxDqd5DokFw9tot99j8iPGxXeTzos5KsAu5vuk7R1qS31XVLr4M2uv3WsXVnquj3U3ltNIf9KMVw8SwTrGSJC4UIcbjuOVyevYj4e+GlWzENjNb/AGOD7NEbe9niLQ7t3lyFXHmLn+F9w5Pqamt/A/h20tNMtLbTVittKuXu7KBZXEcUrFiW2bsHBdsAghc8Yq273fVu/wCP9f8ADCdr6bf8D/P+rnnd9fXmt/CjxJ4j1S4nfULi4WGTSUnYx6b5U6r9nCttXzOMs5A3FuDtxXTXdrquq+LbPxS2gT6Oui2N0pW6liae9Z1G2MCF3Hlgru+Zs7sYXqa6K68HaFeT6lLPZEtqgj+2Ks8irMUxtYqGADDaBuABIABJFbdZtXWmn6aWEtJXZ5VI1/YfDvRfGkWt6hda1cvZTTr9sd7a5E8iK8KwE+UoxJhSqhgVByTkm9Db3PiTS/EWt3+s6lYXdjeXUFmttfSQxWawEhC0SkJITjefMDZDAfdwK6i18D6BZ6hHdwWcuYZTNDA93M9vDIcnfHAzmNG5OCqgjJx1NGoeB9A1PUJry8tJi9yVNzFHeTRwXJXgebCriOXgAHepyAAcgYqpe9e2m/y229LPtuOOlr67fPf+v6RzfifUrnWPgPHqV/F5N1eWVpPNHjG12aMkY7cmuy13TodT0eaG4luolVS4a0u5bZwQDj542Vse2cVV1/wjpPiaFIdXW8eFV2+Tb6hcW8bDORuSJ1DYIGCQcU6Xwtp82jJpck+qG2Ry4YavdCUk54aUSeYw5+6WI6ccClU97mtpcUFytXONl1bUk/ZnTVUv7kal/wAI+kwvDKTL5nlg792ck571fjguPDvj/QbePVtQu01Oxu3vvtt28iSyRiIrIEJ2xcs3EYVcHGOBjatPAmgWXhu50CKC6fSrmEQPaz6hcTKsYGNqb3JQY7LitDUNGt7u4hvo41Go2kE0VnM7Nti8wAHK5wQdq9QelXUkm3KPn+o4LRRf9bHk+hfbvFtv4duobrxUdVmv/tGqTfaL22sXtgXzsOVhKEbNgj+YjBYH56634kT2r6LeeFdHsIbrXvE0bRrAigbQVCNczHHCIAPmOSSFUe3O6b4U8SW2haXpunabr2l6zbvCJ9Uk1wHT12MDIyWyzsCjAMFj8lQNwyFxx3V34A0C71671pk1GDULxUW4ntNXu7fzAowoIjlUYHpjufWpaVrdL/5fh/wRqTUubr/wX/X3G/ZwG1sYLdnMhijVC7dWwMZrn/iNdz2Xw61mW0kaKVrfyhKhw0YchCwPYgMTn2robW3SztIreJpGSJAimWVpXIHq7Esx9ySTUOq6bbazpF3pl+nmWt5C8Ey5xlWBB/Q0T1uKOlktDjPilY21h8HbmwtIljtbc2cMcaj5VRbiIAY9MCqDSSaf4y8NaNIGeTSbq7SHOMyQNbM0OMeg/d57mMmuk0rQrzVfCD6B46t0vPs8qwi4STaL5IyrxzYUgocgZX+8p6qRnWufDul3niGz1y4td2o2KNHBOJGG1WBBBUHa3U9QcZOMZpzu7q++t/u/H/PuJbWPODNqI+FNp4+XWNRbXpI4bwxi8c2zl3UfZhb7hFjDbB8u7ODndzV2azvNY1Xx5Ld63q0UOnSL9hgtb2SBbZ/siOWBQgtyR8jZTIJ25JrrY/A/h+LUheJZy5Wf7StubuY2yS5z5i25fyg275twXO7nrzWgmhadG+pslvhtVbdeHe3707BH68fKoHGPzqJpuL5dNHby2t/Xz3Li0nrqed3Gvavrd/4b0+ez1m/tp9BTUrpNGu47SSeZiq5MhmiYKvJ2o3JcZ4Fdf4MbV7Xw/cr4kiurVbe4kFs2ozxyTC2GChldHZSRyMliSFBJJyTavPB+iXtlp9tLbSxLpqCOzltrqWCaFNu3assbB8EAAjdg4Gc4pbrSp7DwvPp2hQJfyOrIser6hNIrBvvb5WEjkYJ4/DjqLk9+Xz/O6+7YzjF2Sk+35Wf+ZqWd7a6jZxXen3MN1bTLujmgkDo49Qw4IqViQjFRkgcD1rB8D+FIPBPgvT/D9tO1wlmhBlZdu9mYsxAycDLHAycDua36JJXaTGr9TyLwtqviDQPhza+K5DpUkN5cC41G1NvJ9qneWbYzefvxvGVAQx4AUJngGvVbnUbKzntoLy8t7eW6fy7eOWVVaZsZ2qCfmOOcCshPBHh+PVPty2cm/wC0fahAbqU2yzZz5gty3lB93zbgud3zdeaoeK/CE3izxNoMt3Haxadot2uoJOsjG4kmXOIwu0KqZwxbcSdoG0daF0X9Jf1sN9WddXE/E8XVxp2h6ZbpDJBqWsQW10s5PltHh32uByylkXK8bvukjdmu2qnquk2Wt6bJYapAJ7eTBK7ipUg5VlYEFWBAIYEEEAgg1Pb1X5j7+j/Iw9E1bWIfFlz4c117C6eOyS9hubC3e3UIXKbGjZ3wcjIIbkZ4GOegtdRsr6S4jsry3uXtZPKnWGVXML4ztYA/KcEcHmqWleHtP0Fbh9Lhka4uAPMnu7qWeWXaDtVpZCzlRk4GSBk4HJrI8F+EJNA1LXtYvVtor/XroXE9vZszQw7VwAGYAuTyzNtXJboMZLXn/Wv+X5eZO239af5/n5HWV5pZJ4ivvG/jLUtKn0m3ubGdLOF9RtJLhjCII5BGCsieWhZ2Yn5slunyjPpdYeq+DdE1m+e8vrabzpYxFOYLuaBbhBnCyrG6iVeSMOCMEjoTS1vdf15/13KVupw2r+PtE8UaboOk6vqVjolrrFjHqOqJeXiRj7OekCsxG7zGBBIH3FboWWtv4QalpV38P7e10i8s5/sk06vDayo3kgzSFAVU/KCvI9uldvFFHbwpDBGsUUahURFwqgDAAA6CstrI+HvDUtv4X05ZpYg7W1q85VWkdix3OxJA3MSevGcA8Cm7K7X9f0v61Js20Yvw8T7IviPTId32Sw1ueO2BPCo6pKVHsGkYAdsV2NY3hXQB4c0JLSSUXF1LI9zeXAXb588jFpHx2GTwOwAHatmnskuyS+5B3fdv8wooopDCiiigAooooAKKKKACiiigDmPiTPPbfDPX5bZ5I3Wyky8edyLj5iMdwuTXK6jHcHVfBOoWdnHpFpBqKWmmWKxgTSW7RP5jPj7ilFyIx02qWO75F9PkjSaJo5UV43UqyMMhgeoI7isTSPBmh6HeJdadbTCWKMxQefdzTrbocZSJZHYRLwBhABgAdAKFpK/p+H9W+bCWsbev4/1+Ru0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFDVtOudRgSOz1i90llbcZLNIGZxjofNjcY78AH3rl4khm1mTSIfipfyalHnfZLJphmTAycoLbcMDnpXV6rLqUNiW0a0tbu53D93d3TW6Y7ncsbnPtt/GuJ+DrawPAOkLdabp8FjJbtKLmG/eSaWRnLEvGYVAJJYk7z+vCWrY+ly9o8Ca6ZH0T4oahqiwMBKLRtNmCE9A2y24zg1oHwvq5P8AyPXiAe3kaf8A/ItV9C/5Kf4s/wCuFj/6DJWX478dT+E2vLiPxB4ZiazhE40W8fF3dqOSFfzV2MwyFHlvyAcnOAXQld7G4fDGrE8eOtfHHaDT/wD5Fo/4RjV8Af8ACda/9fI0/wD+Ra53xSviC78eeErjTtT023hnkna1SfTZJGizbEtvInUPnnGAuPfuvxPt/EI+Hq79U0wstxai5I02QCVjdR7Cn7/5AOMg7s9ivZ9bedgjZuyOh/4RjVtpH/Cda/16+Rp//wAi0qeGdWVsnxxrz8EYMFh6deLXt1qtrniHUvBXhG71XxJe6bfyrLHFb+TbtZRBnYIu9nlkwoLZLdgDwazfDnj433igaJLrvh/X2mspLqK50Q7RC0ZUNHInmydd4KtuGcMMcZpaWf8AXS4G1/wi+r/9D34g/wC/Gn//ACLQfC+r5/5HrxAP+2Gn/wDyLXOaN408UXPh/wANeINUg0qOy1i4gtpbKGOQyr5uVWVZS+OuD5ZQ4BxvJr0eq/4b+vvQdbf1/WhzjeGNWJ48c6+vAGBBYf8AyLSf8Ivq+P8AkevEH18jT/8A5FpfFunaY1jPquv6zqVhY2duS32bUZbRI/VyYirMx4ADFhwMAEnLvAVzq154B0a58Rb/AO0pbVWnMibHJPQsuBhsYyMDnNJa38v+D/kGw0eGNWCEf8Jzr5JxhvIsMj/yVqFdCvzeNaj4g621ysYkaDy9P3qhJAYr9mzjIOD0yDXSzzLb28k0mdkaF2x6AZrlvh1AZ/C0fiG6G7UNfUX9xITkhXGY4weyohVQOnU9SaO/l/S/X7g6Fn/hF9X/AOh78Qf9+NP/APkWs4Qo+uNoy/FLUP7UHJsA+meeBjd/q/s27pz06Vg/Dq40zUbTTZ7648V3GrvLKWlll1Q2jMJHxlifs5XAA/u9utdP4l/5H/wb/wBd7v8A9J2pX0TFfQsHwxqxPHjrXxx2g0//AORaP+EX1fH/ACPXiD6+Rp//AMi0eLtdbRVtx/wknh/QBLu2yayN3nEdVVfNi6ZGTk9eneuH8Ta/q3jH4N22tWVxp1pFJdQxXMX2drgSSJepGHikEigJuXdyGJBxkdaa1+8q2qR3I8MasEI/4TnXyTjDeRYZH/krSDwxqwPPjrXzx3g0/wD+Rame18UDRblJtY0hr0nMcy6TKsapjkGP7SST77hj0Nc58Lf7Zs/hvot5qep6fJpUemK6wxae8cqALnLSGZg2ADnCDPtSutb9AtdG7/wi+r/9D34g/wC/Gn//ACLQfC+rkn/iuvEA9vI0/wD+Ra4/Tvi4l1/ZOovrXhue21S5ihGi204a/tllYKjM3mEOwJUunlrtBbk7fm0b7xf4paPxdeafHpMNl4ZmkAE8Ukj3ipAkpTh1ERw2N/z5yPlG35nsrvpf8LX/ADQJNuy/rf8AyNp9Cv1vI7VviDraXEiF0hMenb3VcBmA+zZIBYZPbI9amPhnVvLC/wDCca8CCSW8iwyenH/Hrjt+tUvGsD6r4D/t3TQINV023/tPT5SeY5FTcUJ/usuUYdwxrp9NvF1HSrS9jGEuYUmUegZQf60W1ae6/r/Mm6smuv8AX6oxP+EX1f8A6HrxB/340/8A+RaVfDGrA8+OdfbgjBgsP/kWujooGc3/AMIvq/8A0PfiD/vxp/8A8i0Hwvq5P/I9eIB7eRp//wAi10lFAHON4Y1ZsY8c6+uABxBYc+/NrSf8Ixq20D/hOtf69fI0/wD+Ra6SigDmx4Y1fB/4rrXz/wBsNP4/8laB4Y1YHnx1r547waf/APItdJRQBzf/AAi+r/8AQ9+IP+/Gn/8AyLSv4Z1ZpGZfHGvICSQqwWGF9hm1zXR0UAc3/wAIvq//AEPXiD/vxp//AMi0v/CMattA/wCE51/Oc58iwyf/ACV/zmujrE8Z6zN4f8GanqdoqtcwQHyA/wB3zW+VM+24jPtSbsrhu9SjBoV9cSTx2/xC1uZ7d/LmVI9OYxPgNtYC14OGBwexB71MvhjVlbJ8c6+w9DBYf0ta5zxppj+Efgrc2ek3l1b3KNbiW9hmZJ5ZHuIxLIZAQdzFmJPvUy65ejXvC1nd3kiz2t1eWmookhxKY4Cys4HB3LskAPTcO9N2V32/4H4ah0ubn/CL6v8A9D34g/78af8A/ItB8MasWJHjrXxk9BBp/H/krXPt418Qp4Rh8btHpo0GQJO2nGJxcrbMQBJ55fZuAIcp5eMZXdn5qsXPibxTd6l4qi0hdJt7bQWUxS3UUkrXObdZTGQrrs5P3+eCBsOCSm1HV9P6/r/MaV3obB8Maucf8V1r4/7Yafz/AOStH/CMattA/wCE61/r18jT/wD5FrB1X4lQpLolpFqejaDNqemjUpLrWZQY4UO0LGqb4zIzEtzuGAhODnFb/gfxQPFehzXJls55bS6ktJZ7GTfbzsmP3kZyflYEHGTg5GTjJqzu1/W9vzJ5k0n3/wAr/kC+GdWCsD4415twwCYLD5eeo/0X8OfWkHhfVwf+R68QH/thp/8A8i10lFIZzf8Awi+r/wDQ9+IP+/Gn/wDyLSt4Y1ZmJHjnX1BOQBBYYH/krXP+JYG0LxJ4fXR9Z1SfXNQ1NPMt5r6SWOa1yTOTBny0VU6MqLgheck5k+JX2TT9OvL6CfxFDrBtmezksbi8FrFIo+VpMH7Mig4LGXAxknIpX92/9fL+u40ryt/XzNv/AIRfV/8AoevEH/fjT/8A5Fpf+EY1bZt/4TnX85zu8iwz9P8Aj1xXOaxrEd/4mtdN1nVrqKwt9Mguimk3UkT6jczF1RUaEiRgBGzBVIBzk8LTfDni7UdJ8K61b6xBPcajpepfYLG3nuBLPOZEjeCJ5BwzgShWYbuFJy2Cxff+utvzF2+X4q/5HSHwzqqKxfx3rwGOSYdP+X3/AOPWo7XQNQu7aK6tPiBrdxbzIJI5Y49PZJFIyCCLXkEdxWX8NYtVs77xRYa5rFzq1xbahHmWeQlVZ7eORljU/cQMzYUdBirXhlDoXjrXPDcBxpzQxapZxZ4t/NZ1ljX0Xem4DtvOOKFv6/8ADhtqXv8AhF9X/wCh78Qf9+NP/wDkWg+GNWLEjx1r456CDT+P/JWukooA5xvDOrFVA8ca8u0YJEFh83PU/wCi/hx6Un/CL6vj/kevEH18jT//AJFrpKKAOcHhjVgDnxzr5yMAmCw4/wDJWkHhfVwf+R68QH28jT//AJFrpKKAOb/4RfV/+h78Qf8AfjT/AP5FpX8Maszkjxzr6A/wiCwwPzta6OigDmz4Y1Ygf8V1r4wP+eGn8/8AkrR/wjGr4x/wnWv/AF8jT/8A5FrpKKAObHhjVgD/AMV1r5yP+eGn8f8AkrQPC+rg/wDI9eID7eRp/wD8i10lFAHN/wDCL6v/AND34g/78af/APItK/hjVmckeOdfQH+EQWGB+drXR0UAc2fDGrED/iutfGB/zw0/n/yVo/4RjV8Y/wCE61/6+Rp//wAi10lFAHNjwxqwB/4rrXzkf88NP4/8laVPDOrLIrN4415wCCVaCww3scWuam8a6xc6B4I1bU7AL9qt7ZjBvGVDnhSR6AkGuKjvZNE8TaHPoWo6tqllNfDTtVu72+ea3uZXBH7pGJUOjrkmIKg+ZOSCqi1dv61/r8RPRX/rT+vwOu/4RfV/+h78Qf8AfjT/AP5FoPhfVz/zPXiAf9sNP/8AkWukooGc4fDGrEDHjnXxgYJEFhz/AOStJ/wi+r4/5HrxB9fI0/8A+Ra6SigDnF8MasN2fHOvtkYGYLDj3/49aQeGNWDAnx1r5wehg0/n/wAla6SigDm/+EX1f/oe/EH/AH40/wD+RaD4Y1cnjx1r4+kGn/8AyLXSUUAc2fDGrED/AIrrXxgf88NP5/8AJWlHhnVvLK/8JxrxJIIbyLDI68f8euO/6V0dFAGHYaBqVnfRz3Hi3WL+NM7ra4iswj8dzHbq3vww6VuUUUAFFFFABRRRQAEZBFY/hfQW8M+EbHRY7oXDWcPlLO0W0Meeduffpn8a2KKB3OR0fw14k0/xVeaxe69pVyt+IluYItIkiJWMMF2Mbltp+bkkN9Kz7/4dalcWfiXTrDxHHZ6d4gllnmB08SXMbyIFZRKX2lOOhTcFOAwwCO+opWQJtbHO634YuNQh0abTNSWy1HR5fMt55bfzo3BjMbq8YZSQVY9GBBAOeKl1fw/ceIfB0+j6xfR/ap0GbuztzGqSKwZHWNmfGCFOCxzjtW7RTet79RJWtboc3d+G9S1zw7PpniXVbaeVpI5be606yNsYXjYOj7XkkDEOoPYdsVc0yz8QxtIuu6tpt7E0e1RaabJbsG9SWnkBHsAPrWxRQ9QOTg8EC38HeH9DOo5GiT20xnMOPO8k5xt3fLn1yce9WD8RvBAJB8Y+HwR1H9qQ/wDxVdJRR3+/+vuDrf8Ar+tTh/EPhy68cX2k6to/iTTJtGth50NnLZG7triYE4lLRzpv29gcgHJ5OMddpseoRWKrq9zbXV0Cd0trbNAhGeMIzuRx/tH8KtUUKy2DfVjZY0mheKQbkdSrA9wa5HwBPJptjL4Pv0eO80FFiiZhxcWhJEEqnoflXaR1DKfauwoo/r+v66h0Mnwvof8Awjfhu10n7R9p+z7/AN7s2btzs3TJx97HWsjXPDXiPVPEtjqlnrul2senSSPawy6TJKfnj2MHYXC7upIwF/GutooDpY5W/wDCmpP4uXxDpWsW9pdy2K2N0J7Azgorlw0X7xfLbLN13g/LkHHMNt4AWL4cz+FbjVZp2llmmW/8lVdZHnaZX2j5cqxHTAOOg6DsKKOlg63/AK7GdpNvq8NtImv39lfyE/I9pZNbALjnIaWTJ9wR9KyPDHhjU/Dtiujzata32hwo8VvA1iyXCRljtVpRKVYKp28RgnA/HqKKHqC0Vjl9B8O694dhs9NtNcsp9FswI4YbjTWNyIR91POWYKSBgBvL6AZyckjeDN2k+LLL7f8A8jHLLJv8n/j33wJDjG75sbM9uuPeuoooet79dP6+4cW4tNdDjvFpmh8LW3hHTHeTU9Wt/sMTouPKiChZZz2UKpyM9WKr1NdXZ2sdjYwWkAxFBGsSD0VRgfoKmop3u231Jskkl0/r/IKKKKQwooooAKKKKACiiigAooooAKyfFOi/8JH4V1HSRL5L3UDJHLjPlv1Rsd8MAfwrWopNXVhrRnEMj/Ev4eT6Zdl9I1KOaKHUInj8w208TpIy4yNwYAEMDgq4PtWjf+D/ALb470/xCl75cVqjiazMO4TsUZA27d8uFbB4OcL0xXTUU3Zk20scTF4BvV0mHw9Lrkb+GYJFKWYsyLkxKwZYGn8wqUBAX/VhioxnOSdaDwt5M3iV/tmf7dffjyseR+4WLHX5vu57dce9dBRSklJNPqUm07o5IeDLyxTRrjQtXitdR02wXT3lubMzQ3MIA+9GHVgQyhgQ/GSDnPGjfz3um+Erp9buZbq68tkMui6dLvy3C+XEGkbIyOckdzgVuUU5e8mn1/XcUbRtboc18O7DXtM+H+lWni66kutXjiP2iSWTe/LEqrNk7iFIBOTkjqetdLRRTk+ZtiSsrHCaT4J8S6V4k1DWj4j0q9vL+TDz3ejSNLHADlYIyLkKiD2Xk8nJrd1jS/EWozTwWmuWFlps6+WyDTGkuUUjDbZTNs3dcExkDjIbHO9RS6WGtHdHIy+BFsbvSL7wtdW+n3ulWH9mxyXlobpXthjCkB0bcCoIYMOrZBzwsHw60qSzh/tia7vNRW6kvpL+C6ls3e4ddjOPJdSo2YQLk4UYyeSetop3e/8AW9/z1D+v0/LQ5Xw34FsvC+uatqsWoX8/26QOqXN/cSrEojRTu8yRhIfkzvYZAOBwKi8Ihtc8Qap4vKulpexRWemhxjzLaMs3nY6gO7sRn+EKe9dfRSQPUKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigClrGlWmu6LeaVqKGS1vYWglUHBKsMHB9a5/TfB1+lzpP9u6xb39nouDYW9tYfZvnCFFklO9gzBSQAoRcsTt6Y62ihaO4dLBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFUNWm1eGBDoVjZXkpbDpeXj2yquOoKxSZOe2B9ayhfeN+/h7w//AOD6f/5DoA6SiucW+8bFhv8AD+gBc8ka7MSB/wCAlIb7xvnjw94fx/2Hp/8A5DoA6SiubN942zx4e0A8d9dm/wDkOj7d43wP+Ke0DPf/AIn03/yHQB0lFc39u8bbT/xT2gZzwP7dm/8AkOlS98alvn8P6CBg8jXZjzjj/l09f8mgDo6K5v7d43/6F7w//wCD6f8A+Q6DfeN88eHvD/8A4Pp//kOgDpKK5xr7xtn5fD+gEYHXXZhz3/5dKT7d43x/yL3h/P8A2Hp//kOgDpKK5wX3jXYc+H9A3cYH9uzYP4/ZKQX3jbPPh7QBx212b/5DoA6Siub+3eN/+he8P/8Ag+n/APkOg33jfJx4e8P47f8AE+m/+Q6AOkormzfeNs8eHtAPHfXZv/kOj7d43x/yL3h/P/Yen/8AkOgDpKK5wX3jXYc+H9A3cYH9uzYP4/ZKQX3jbPPh7QBx212b/wCQ6AOkorm/t3jf/oXvD/8A4Pp//kOg33jfJx4e8P47f8T6b/5DoA6SiubN942zx4e0A8d9dm/+Q6U3vjXywR4f0Hfk5H9uzYA4xz9k+v8AnoAdHRXN/bvG/wD0L3h//wAH03/yHSrfeNs/N4f0ADB6a7Mee3/LpQB0dFc39u8b/wDQveH/APwfT/8AyHQb7xvnjw94fx/2Hp//AJDoA6Siuca+8ajGzw/oB4Gc67MOf/ASk+3eNto/4p7QM55H9uzf/IdAHSUVzYvvG+Dnw9oGe3/E+m/+Q6BfeNs8+HtAHHbXZv8A5DoA6Siub+3eN/8AoXvD/wD4Pp//AJDpXvfGokYR+H9BZMnaW12YEj3H2Q4oA6Oiub+3eN+3h7w//wCD6b/5Dpft3jbaMeH9A3Z5H9uzYx/4CfWgDo6K5v7d437+HvD/ALf8T6f/AOQ6Vb7xqW+fw/oAHqNdmP8A7aUAdHRXN/bvG/8A0L3h/wD8H0//AMh0G+8bbjt8PaARngnXZh/7Z0AdJRXNm+8b8Y8PaB7/APE+m/8AkOj7d422j/intAznkf27N/8AIdAHSUVzi3vjXa27w/oIOPlA12Y5Oe/+iccZ9aQX3jfPPh7w/j/sPTf/ACHQB0lFc39u8b/9C94f/wDB9P8A/IdK1942DHZ4f0ArngnXZgSP/ASgDo6K5v7d437eHvD/AL/8T6f/AOQ6X7d412Z/4R/QN2en9uzYx9fslAHR0VzYvvG2Dnw9oHTj/iezf/IdAvvG+efD2gAe2vTf/IdAHSUVzf27xv8A9C94f/8AB9P/APIdBvvG247fD2gEZ4J12Yf+2dAHSUVzjXvjXau3w/oJOPmB12YYOe3+ic8Y9KT7d43x/wAi94fz/wBh6b/5DoA6SiucF942wc+H9ABxxjXZuT/4CUgvvG+efD3h/H/Yen/+Q6AOkorm/t3jf/oXvD//AIPp/wD5DpXvvGoc7PD+gFexOuzAn8PslAHR0VzZvvG2Bjw9oHTn/iezf/IdH27xvj/kXtAz/wBh6b/5DoA6SiubF942wc+HtA6cf8T2b/5DoF943zz4e8P4/wCw9P8A/IdAHSUVzf27xv8A9C94f/8AB9P/APIdK9941DnZ4f0Ar2J12YE/h9koA6OiubN942wMeHtA6c/8T2b/AOQ6Pt3jfH/IvaBn/sPTf/IdAHSUVzYvvG2Dnw9oHTj/AIns3/yHSpe+NTIok8P6CqZG4rrsxIHsPsgzQB0dFc39u8b/APQveH//AAfT/wDyHQb7xv28PeH/APwfTf8AyHQB0lFc4b7xtgY8P6ATjnOuzcH/AMBKT7d43x/yL3h/P/Yen/8AkOgDpKK5xb7xr827w/oA44xrsxyf/ASkF9423Dd4e0ADPJGuzH/2zoA6Siub+3eN/wDoXvD/AP4Pp/8A5DoN943zx4e0Aj316b/5DoA6SiubN942wMeHtA6c/wDE9m/+Q6UXvjXyyT4f0HfkYH9uzYI5zz9k+n+eoB0dFYdhd+K5L6NdU0XR7a1OfMlt9XlmdeOMI1sgPOP4h/StygAooooAKKKKAKOsavZ6DpU2o6lIyQRYB2IXZmJCqqqOWYkgADqTXKaf4g1C9+J6xX+nalo1omiySm3vp4SrsJk/eYildcgZGTg1reOtIvdX8OIulRLPeWd7bX0UDOE84wzLIU3HgEhSATxkjOBzWJLo2ueJfFd5ealpj6Tpt1oU+nRiSeNp43d1yXEbsozyRtY8LkkE7ROt+ZdL/wDpLt970G0mren/AKUv0NbT/Hllf3Fgf7N1K20/U22WGpzxxiC6YjKgAOZF3AEqXRQfqQDWu/iTp9tJq3laVq11Bok7RalcQwoI7UABi+WcFxtOcRhmAHKjK5xPDXhT7O2k2OueGfETT6eyf6Y3iBrixEkX3ZVja63bTgEAxcZxjir9z4b1aTwb48sUtM3OrT3b2SeYn70PCqqc5wuSCOcU5aN21sn89Vb9RR1aT0u1fy3v+hv6n4risdSt9OsNNvtYvbi3N0ILHylKQggb2aV0UAkgAZyeeMA1LqfiW20uxsppLW7mudQcRWlhHGBPLJtLFcMQq4VWJLMFAHXpWB4o0JLy305pvDWqX15bWvlwX+i6hHa3NqxXDKXaWI7ehABcE5yowM0NV8Ja9d6L4VvdW+1atqej71vYbDUWtJp1kXaSkqtGC64UnJQNhuRnBb0uvP8AUSvo32/r8TsdC8QQ659qi+y3NheWcgjubK7CiWEkZUkozKwI5BViO3UEDWrmvB+kWtj9tu4tF1fS7m5ZFmOr6h9slmCA7SG8+XAG4jGR9K6UnAPGfYUMFc4+bVvEmieJtFstWu9J1O31aeS3EdnYyW00O2Nn8z5ppA6jbg8LjcDnsewrgG0WfVPE9nqOleEJPDN5HeLPfapM1tHJdRD70R+zyO0u7jIkwoxuzkAV39C21G/iOc8Zape2trY6Xosoh1PWLkWsExXd9nTaXllx3KorYzxuK54qnrF7rOl654c8NeHry1jF1a3Bku9UgkvHIhWMDOJUJY7zliTSeNpV0jW/DXiK5IWysbx7e7kPSFJ0MayH2D7AfQNntS+KLfVIfGvh7WdO0a61a3soLuKdLSWBXQyCLacSyICPlPQ0L9f0/r7xrfXsGneM3sbnXdP8XtbW9zocUdzLdWqt5VxbyZ2yLHlmVsqylMscgYJyKvab4vjvNUXT9Q0jUtHuprdrm2S/WL/SEXG/aY5HAK7lyrbW+YcdccrqfgXWPFNn4m1LUkjsL/Vo7WOxs/tLZgit381FlkT7rs5bOzcF4wW61oeH9ChfV4ru58MeJLG8t4JPJuNV1z7bEjMNrKoN1Ickdyg4HalrZ+n9f8H5+QadP62/4Ja0r4labq1vpV5BpeqxabqsiwW+oTQosQmbOI2XfvHIK7gpTPRq0Ljxeia5dabYaPqep/YSi3txaJF5dszLuAIeRXchSGIjVzggdeK5+z8MavF8N/B2lyWmL3Tbqyku4vNT92sbZc5zg49ic9qj8aeHby91S7ufDmhapZa9Okaw65p2oxwQEqRs+0oZAZFXoQYpPl4X0DW/z/Cy/wA2v+GYutvL9X+nqdVq/iePTdSi02z02+1fUZIvPNpYiMNHFnG9mldEAJ4A3ZPOAcEivNdHxl4OF54Z1CexuSd9vIylWimjYgxSp3G5SjqffHY1z/iDwreDxoddew1bVobqwhtZotH1d7GWKSNnO4jzoldDvPVsqRwDuJHR6LFpnhXwfJcGzuNGs4RLd3EeoXPnSRkku7PJ5kmSTk53nrSduVtv+rh10/rT/MseE9e/4SXwvZaq0P2eWZCs8Gc+VKrFJEz3w6sPwrYrl/h1Y3Nl4Jt3vojDPezz37Qt1i8+Z5Qp9wHAPvmuoq5biWwUUUVIwooooAKKKKACiiigAooooAKKKKACiiigArP17WIPD/h6/wBXulZ4rKBpmRPvPgZCj3J4Hua0K57x7pdzrHgPVrOwTzLpoPMhjzjzHQh1X8SoH41MtmNbmFq2peIPBvwtu9buLiK81+aSGaVbne1vC8skaGNEDAhEBwACMkFjyTWlF4tuJ77wxFHDCg1SS4hvEYEtDJDG25VOR0dSOQeBVPxWX8f/AAnMvhdReHUPs00KhwmQs6MwJYgAgK2QecjHWm6n4a1NviXpt1aWzSaOzz3NzMJVHkStAYiApOcNhD8oxncT1qp3V9Nb/lb/AIJP2S6fiJpwSO8OnakNEknEC62Y4xali20HG/zdhfCh/L28g52/NS3vxBtLXUNYsrXR9W1CbRSpvvssMe2JDGJA4Luobgn5Vy+Qflxgnno/D/iCT4fW/wAP30WSKOJI7KTWBcRC3a3RhmRVD+b5hQcKUADn72Bk7tpoWoxT+Ny9vhdUlDWZ8xT5o+yJH68fMpHOPyqZtxi3HWydvPb+v+CrlxSbs9DRvfF9lANOTTbW71e61KA3Nra2SoHeEBSZCZGRVUb1+8wJLAAE1e0PW7bX9N+12iSxFJGhmgnXbJBKpwyOASMg+hIPBBIINcBP4M1C2fw5qU2m6pfG20WPTbyz0rV2s54nUBgwKzRo653KQX/ukZ7dZoVhFonhW9l0zTLrSp5TLcvFrF757mXbjfJL5snB2j+M4HpVyajd9r/g9Pw1M480kvl+Wv46HSUVzXw88Q6j4r8AaVretWcdleXkRd4os7MbiFZcknDABhyeD1NdLRJOLsxp3Rxusa54l8PX2m3WpHSrjT7/AFKKxNnbQSCaESsVVxMXw+OCR5a8Z545oan40vY/HepaGPE/hnQltfs4t4tUtzJNcmRcnb/pEecHjAB61Utk8Sal46bWvFPhDVpYrGVo9ItLe5smgt1PBuH3XAZpWBP8PyDgZJJrpfFNzqd9Z3+h2fhW5v0u4DAt1cTW62Z3rglx5hl2jPOIyeOAetRqop7/APDLT+u/kNaya/r1/wCG7eZB4p8XnQ7y10tb/S7C5lgM9xqOptstrZAwUHZvUuzMcKu9e5LcYbTgin1/wfsOuwzS3URMeqaOvlJ1yrxgvJ046swODng4rmYfC194U1vQ9WtNOm8RPZ6IujzmJ4kuBsIZZF811Ug4YEbgeV6843PBulXGi6Tfz6lDHYtf301+bVXBW1V8HYSPl3cbmI43FsE9TVlZpvv+en4aiu9NO35a/joTeDNcutc0N/7UjWLUrG5ksr1UUhTLGcF1B/hYYYezVv1yHw7Vrqw1jXNu2DW9UlvLbnO6EKkUb/8AAliD/RhXX030vvZX9ba/iC627v8APQKKKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAo63eyab4f1C+gVWltbWSZA4JUlVJGcduK5jwL4mu/EyQ3D+KvDWqA2ySz2Wl2xWaBmAIDN9pkxg5HKjOO1djcSNDbSyxwyXDohZYoyoaQgfdG4gZPTkgepFcebDU/EvjXRNYudEuNDh0XzmL3ksLT3BkTZ5aiF3Aj/iJZgcquB3Avif9dwlsv67FKfxprqeGbnxjHHp/wDYVtI5bT2hc3TQJIUaTzt+1W4LbPLPTbuyci3qnijV7q+1ldAvNH06x0RFW4vNUhkmWWZkDhAEkTYoVky2WJL4C8c5OueB4fEU2oaVaeGtQ0u1vp2a7u7nUh9jZSwLyRWsczqZWxkFo0wSXJ3DBsXHh2+0Lx1qOr22g3XiC2vGS6s4YbqGOOzuVjETM6yMvVVTDjeVG8ADJ3JbL+tf6vuHX+v6/wCAWLDx3f8Aii10ODw3b29lf6lay3V01/G8qWKxOI3QoCjO3mEqASnQk9NppJ8Tp18L6lPKdPkvLDUYbF9QQOtkY5mUJd8kkRgNyNxBKkB8HdVaX4e3ulxaHcSWsutpFFcRavp9nMkX2ozS+eceYyhoxJkFGZQynnIypl1Xwx4lv9H1XUYLeS2uNV1C2kudJguljmaxiUIbcTKwVZGAJJDAcld2Pmp6fj+v+X6vcFv/AF2/O/fy6GrD4yvtI1DXrfxFcafqMGj6auoS3mmwPCI87j5Lo0kmGKruB3cg9BwTXPjHXdG1bQV8TPpJh1uRYjY2kMgnsDJ/q2aQuVkXdiMtsj+Zhj0qhY+CLvUJ/EFlZ6RN4V8OappTWh02SWJs3LZHnxxRO8cY28HBBY8kcZMmmeG9WvrfStL1HQ57B7e9hv8AVtVubqGb7dLAQV8soxcguqkb1QKg2hRwA42ur+X3Xd/w+e3VsT2dvP8AJW/G/l30SPS6KKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAoato1trUCQ3kt7GqNuBs7+e1YnGOWidSR7E4qlY+ENN06+ju7e51l5IjlVuNcvJ0PGOUeUq34g1uUUAFFFFAEV1awX1nNaXkKT286GOWKRdyupGCCO4IptjZQabp9vY2aslvbRLFErOzkKowBuYkngdSSanooAKKKKACiiigAqpqWlWWr26W+pQC4hSVJhGxO0ujblJA6gEA4ORwKt0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUtO0iw0lrs6dbrbi8uGuZ1Qna0rABmAzgZwCcYycnqSau0UUAFFFFABVXUdMsNXsns9Wsre+tXILQXMSyIxByMqwIOCAatUUAAGBgdKKKKACiiigAqrqem2msabPp+pQ+faXC7JYixAde6nB5B7joRweKtUUAIqqiBEUKqjAAGABS0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBQ1bWbbRYEmvIr2RXbaBZ2E90wOM8rEjED3IxWUPHekH/lz8Qf+E5qH/xiujd1jRnkYKqjLMxwAPWuT03x3p2t+OP7J0PVNL1KxTTXupprS4WUxyCRVALKxAGCTgjNLrb+tr/oD0V/63sWl8daSzBRaa/knAz4dvwPz8mkPjvSAcfY/EH/AITmof8Axir8Hinw/dahb2FtrumzXl1GJYLeO8jaSZCNwZVBywwCcjjAzT7zxJoenapBpuoazp9rf3G3ybWe6RJZdxwu1CcnJ4GByaYGafHekg82mv8ATPHhzUD/AO0KP+E70jAP2TX/APwnNQ/+MVqaj4g0bSFkOratY2IiVWkNzcpHsDEhSdxGASrAeuD6VJc6vptnpP8Aal3qFrBp+xX+1yzqsW1sbTvJxg5GDnnIoDyMf/hO9J2k/ZNf64/5FzUP/jFKnjnSXbAtNeHBPPh2/HQZ7w//AK61INe0e50ldUttVsZtPZgq3cdyjREltoAcHGd3y9evFS3Wq6fYyMl7fW1s6wtcMs0yoREuN0hyfujIyegyKGBi/wDCeaR/z5+IP/Cc1D/4xQfHekA/8efiD/wnNQ/+MVU8P/E3wzr+oXdnHrWkxzx3rWttGuoxO12MKQ6LkEgliABnp1rr6OiYX6HON460lTg2mv8AQHjw7fn/ANo0n/Cd6RjP2PxB/wCE5qH/AMYq5rOvT6XOIbTQdV1d/L82T7EkQWNeg+aWRFYnB+VSxGOQMjNrRtYstf0W11XS5fOtLuMSROVKkg+oPII6EUbhtuZQ8daSULfZNfwMZ/4p2/z+Xk0g8d6STxaa/wBM8+HNQH/tCukrm9L8TTakNY1QQKuh2O+O2kAzLctFu82Qc42ZG1e5Ksc4IpNpbjs3sH/CeaR/z5+IP/Cc1D/4xQfHekAkfY/EH/hOah/8YqtY/EG2uE02fUtE1XR7LVCi2d5eiBopHcAohMUrlC2eN4UZ4zkgHYv/ABV4e0rf/amu6ZZbJfJf7ReRx7ZNobYdxGG2kHHXBBqrMSaexQPjvSQebTX+mePDmoH/ANoUf8J3pGM/Y/EH/hOah/8AGK1dV17SNBt459c1Wx02GRtiSXlwkKu2M4BYgE4qSfV9NtraK4udQtYYJkMkcskyqrqFLlgScEBQWz6DPSkBjjx1pJQt9k1/Axn/AIp2/wA/l5NIPHekk8Wmv9M8+HNQH/tCtnTdW07WLEXukaha39qSQJ7WZZYyR1G5SRxUGleJND13zv7D1nT9S+z4877HdJN5ec43bScZwevoaAM3/hPNI/58/EH/AITmof8Axig+O9IBI+x+IP8AwnNQ/wDjFbK6rpz2dvdpf2rW10yrbzCZSkxb7oVs4YntjrXLv8U/CsHi+50W513R4Ire1WVruTU4lXzfMZGhIJ4ZdoJ5zz0o62H0uXz470kHm01/pnjw5qB/9oUp8c6SIw/2TXsEkD/inb/PGO3k57/5xTvEniKXw/FY6sVhm0IsFv5lJLwK+AkwIOCgJ+b2bcD8pB6EHIyOlAjm/wDhO9I/58/EH/hOah/8YpV8daSxwLTX+hPPh2/H/tGujooA5v8A4TzSP+fPxB/4Tmof/GKD470gHH2PxB/4Tmof/GK6SigDnG8daSuM2mv8gHjw7fn+UNJ/wnek7Qfsmv8AJx/yLmof/GK6SigDmx470gg/6Jr/AB/1Lmof/GKB470kni01/pnnw5qA/wDaFdJRQBzf/CeaR/z5+IP/AAnNQ/8AjFK/jnSY5GRrTXiVJB2+Hb8j8CIcGujooA5v/hO9IH/Ln4g/8JzUP/jFL/wnWk7Q32TX8E4/5F2/z+Xk+9dHUV3dQWNnNd3cqw28EbSSyOcBFUZJPsAKG0tQs27IwP8AhO9I/wCfPxB/4Tmof/GKVfHWku2Baa/n38O34/nDVL/hNp9N8BXHi3xBp00Vq7pJa2VsgNwIZGVIw4ZgvmEtuIyAoOOSCTsDxPYNcaJFGJn/ALaDNbOqjChY/MO7JyOPY802rbhpa5T/AOE80j/nz8Qf+E5qH/xig+O9JViDaa/wcceHNQP/ALQrS/4STQ/7c/sX+2dP/tX/AJ8PtSef93d/q87vu89OnNNvvE2g6WxXU9b06zYSiEi4u44yJNobZyR820g464INIDPPjvSBj/RNf5/6lzUP/jFH/Cd6TtB+ya/1x/yLmof/ABitjU9X03RLP7XrOoWun224L513OsSZPQbmIGantbq3vrSK6sp47i3mQPFNE4dHU8ggjgg+ooAwV8c6SysRaa98oyc+Hb8d8cfueevakHjvSCcfY/EH/hOah/8AGK6SigDm/wDhPNI/58/EH/hOah/8YpW8daSrFTaa/kHBx4dvyPz8mmy+NobXUraDUNG1Wxs7u6Fpb6jcRRrDJKSQq7d5lTcRgFkUE455GS68YSJr99pOm+HNW1WWwEZnktXtURd67lH72ZCePalfS4dRf+E70j/nz8Qf+E5qH/xil/4TrSdm77Jr+M4/5F2/z+Xk5q5q3iODSfssJtLq71C8BNvp1sqtM+3BY5LBFC5GWZgvQZJIBdLeatdeG5LrTtN+x6nsLRWepOh+YHhWaJ2UbgOCCcZBI4Ip9Gx9igPHekkH/RNf4Gf+Rc1D/wCMUDx3pBOBaa/+PhzUP/jFaPh3XbbxJocGpWivGJCySQyffhkUlXjbH8SsCD9K06NhI5v/AITzSP8Anz8Qf+E5qH/xig+O9JViDaa/wcceHNQP/tCukooA5xvHOkqqk2mvfMMjHh2/PfHP7njp3pP+E70jGfsfiD/wnNQ/+MV0lFAHODx1pJBItNf+UZOfDt+P/aPNIPHekE4+x+IP/Cc1D/4xXSUUAc3/AMJ5pH/Pn4g/8JzUP/jFK/jrSUcqbTXyR6eHb8j8xDXR0UAc2fHekgD/AETX+Rn/AJFzUP8A4xR/wnekYz9k1/8A8JzUP/jFdJRQBzY8d6SQf9E1/gZ/5FzUP/jFA8d6QTj7H4g/8JzUP/jFdJRQBzf/AAnmkf8APn4g/wDCc1D/AOMUr+OtJRyptNfJHp4dvyPzENdHRQBzZ8d6SAP9E1/kZ/5FzUP/AIxR/wAJ3pGM/ZNf/wDCc1D/AOMV0lFAHNjx3pJB/wBE1/gZ/wCRc1D/AOMUqeOdJkkVFtNeBYgDd4dvwPxJhwK3L28t9OsJ72+mWC2t42lllfoiqMkn8BWDp3ja3vdSsrS70jVNLXUQxsJ76KNUusDdtG12ZGKgsFkCNgHjIIo30QbK7F/4TzSP+fPxB/4Tmof/ABig+O9IH/Ln4g/8JzUP/jFdJRQBzh8daSACbTX/AJhkY8O35/8AaPFJ/wAJ3pGM/Y/EH/hOah/8YrpKKAOcXx1pLbsWmv8AyjJz4dvx/wC0eaQeO9JZgBaa/wAnHPhzUB/7QrpKKAOb/wCE80j/AJ8/EH/hOah/8YoPjvSAebTX/wAPDmof/GK6SigDmz470kAf6Jr/ACM/8i5qH/xilHjnSTGX+ya9gEA/8U7f55z28nPb/Oa6OigDDsPF2m6lfR2lvbawkkmdrXGiXkCDAzy8kSqOncityiigAooooAKKKKAOQ+J6k+Ci0iF7KO+tH1BAu4G1E6GbcO67Mlv9kHtWVFrmn678UHn0eZbuzj8PTIt3CQ0EreahIjccPt4BIyAeM5BA9EopNXX3/irDv+n4O5598MPDmkXXwd8KRy2EQVYbe/Hl5jJuBhhISuCTnrnqODkcVjLqGiad4U8X6F4mkhbXb+8vWbT3wbm/EjN9nMSfelBTy1UrnBXHG049aopz9+/nf8bf5Ch7iXlb8L/5nk2nXmj+G/iRbz+OdS0201S28L2UK3N7OiEyb5RLsLHknAzjJ/OoNIhXQYfCOpeI0+waDb3Woyxfa08tLJppWa1MgYDysRsyjONpYLwTXpMGg+T4zvNf+05+02MNp5Hl/d8t5G3bs858zGMduvNa9Pmb1fd/m3+uguVLRbafgkjxzX/L1Ox8aa3oRW40OaXTZRcW43RzyQTBriZCOGVUCAsMjMZH8JrobPxBpmv/ABp0+TRbpb62i0G6H2uD54HYzwZVJB8rkY5wTjIB5r0KiiL5ZJ+v4qw3qmv66f5HB+Ftc0vSPFXiTRdV1C2sdSudZaa2trmZY3uUkjj2NGGILgkEcZ5Uiu8oOcHHB7ZrmzY+N8nHiHw/jt/xIZv/AJMpLRJdkl9ysFtW/wCtdTI+IfxAtvDs9v4fstS0+x1vUULR3GozrFBZxZwZnLEZPXag5Yj0BNb3gq20ex8G6dZeG9Qh1LTrWLyY7qGZZVkIPzHcvGd2c471rWSXcdlEupTw3F0B+8lghMSMfZCzEf8AfRqehaX/AK/r+vJI3IbxZXsZ1tziZo2EZzjDY4/WuL8JmP8A4UPYrENpj0QxyqRgrIsRWQH33hs+9d1WLo/hxdD1rVLmyunFjqLic2JXKwzknzJEbPAf5SV6ZBI+8amUeaMod1/n/n+RSdmn2/r+vmec3Osaf4r+Ffh7wv4avoNT1SdNPWVLKQTGyWNo2kklKn92FCkfNjJwBzXQeENKsLrxz8Qprizhkln1CO2kkZBuaI2kJKZ67ck8V6BRVyfM231v+Nv8jOMeWKXa34X/AMzzuB9D8E/EK6bXbq30ux/sm1tNIuL+cLGsUZfzIlkc/ezsJBOSNp5xxyiw6Zp1t4Pm8RLbWOhnxNf3VmmogQxwQFJ2gyHwF52soOMZXivb6x9X0H+1dc0LUftPlf2RcyT+X5e7zd8LxYzkbcb8556Y96E3e7/rW/3ldGv62sedazE2uaf4x1TQ42u9Eu7zT2ke2Tet9FEyfamjAB8xTGNhIzu2MvOK25tQ0rxP8QPD914Pu7TUFsLW6W9ubFxJHHC6AJCzqcBi+1gp5wjHAr0KipsrW/ra39fMDxXTvEGnP4C8CaDDOZtTtNSsYr22iUs1kyPhhOB/qjuG0BsEnpnBx1+qavp3hv4tm+1++t9Ns73REggubuQRRNJHM7Mm9sLuxIpAzkjPpXd0U9nfzb+9WB6380l9zuc747nsz8Ndemuts1o+mzfd+YSBoyABjrnIx9RWnoUc8Ph3TorzP2hLWJZc/wB8IM/rVbW9AGu3mn/a7php9pN9oks1Ti5lUgxlmz91SN23HJC5OBg7FC0v52/C/wDmD6eV/wAbf5BRRRQAUUUUAFFFFABRRRQAUUUUAFcr8Tkd/hlrmwFlW2LyqBnMQIMgx3+QNxXVUjoskbJIoZWGGUjII9KTvbQNOpw3xYeOX4T3kluVMTSWbIV6FTcRYx+FZd8raZ8UdB0OMqim4vL2xGOAkkDl1752y7j7B1Fdr4X8P/8ACMaU2lw3klzYxSk2Uci/NbQkDEO7OXCndgnkKQOcZOzTkk1ZbP8A4H46C1tY8XS502f4P2fhKCWE+L1eFW0/I+1xXwlDPOyAhgA26QydCvOSDzv2en2j6t8TrqW2ie4lKwSSMgJaMWSHb9MsePevSaKmcedNbXT/ABt/kXGXK7njhubzTL7wZq1zrGl6RYr4cSG3vtXtHmgjnYIXXcJowjsijBY8hWA712vguO30jwnf31trFnr1tNc3F6JNIhAhGeXjiUSP/EGON3VjXXVT1Wxl1LTZbSDULrTmkGPtNps8xR3wXVgMjjOM+mDzVybd7bu/4u5nGKSSfS34KxS8J+KNP8Z+F7PX9HEwtLxSUWdNrqQxUggEjIII4JFbNUtI0ix0HSLbS9Jt1t7O1TZFGpJwPcnkknkk8knJq7RK13bYavbU8muvHHh/xt42gs5vEOk2GhaHfLJtubyKOfUbxD8oRGOREjc7sfOwGOBk6vjL/hDrbUNUmGvzWHieWFdttpuryrcySBP3QFqkmJG5GFKEEHkEV6JRU29239X7jXxNnlF7G8HijQ9S+I2qSaRDc+Hktp5or1rKIXgcO6GdCpUkchQwB2nriut+H091caBcvLc3N3Yi9lGmXN0xeSa1yNjFzy4+9tY5LLtOTnJ6qqmq2t1faTcWthetYXEyFEukjDtFnqyg8ZxnGeAcHB6U72vZd/xd/wDgegrbfL8Fb/g+pzXw+2vL4pntwPsk2v3BgIXAO1USQj1/erJz65rsKp6TpVpomkW2m6dGY7a2QJGpYscepJ5JJ5JPJJJq5T6JLokvuVg7vvd/fqFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAc58QdOutW+Hut2Wnw+fcy2j+XCP+WpHOz8cY/GuIiubPxfrnhq4tNVh1nWo79Ly5S1Y+XpFuqMWjKdY2JKod+JHJPAVdi+tUULSV/T8Aeqt6/iFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBQ1bQtI16BINd0qy1KKNt6R3lukyq2MZAYHBx3rmT4T8AJ4mTQ/+EK0T7U9o12HGlW+zYHCEZxnOW9Oner/AI+1O80zw1H/AGdO1rNeX1rZfaVALQLNMkbOMgjIDHGeM4rmJre18J/Ea8m0t729nt/DE9yYbvUZrpyVlUqAZXYqDtPTANS3bV7a/hFv/IbvbTfT8Wl/mdUvw68EowZPB2gKynII0uEEH/vmkPw58EE5Pg7w+T/2C4P/AImuN8Nt4muF8O6ra6d4le7uWik1O7vtSt2s7iGRcyMsAuGEeMgrsjUjGD1OWajHqE2g+Ptal17VhPpF3ctpsUN48UdsY4UcfKpAkBb+GTcoHAAy2al7rafS/wCFv8xR95q3W343/wAjtT8OvBLHJ8HaAeMc6XD/APE0f8K58EYA/wCEO0DA6D+y4f8A4msXxTDrdzdaXqJs9W1TRVsWNzaaLqJs7hZiAwk+V4zIuAV2B+pztPGKmu+JZdRsPCNroA1i/wBP1dpGna0lS3u5UijJ8svI8ZRi3LEMG+QgHmjbT+v60/rYSa/r+v6Z0v8AwrrwTtI/4Q7QME5I/suH/wCJpU+HfgqNsp4P0FTgjI0yEcEYI+76HFVvBEetW8+qQajY6naaYJEfT11a7jubhQVPmKXWWQlQwBG9ifmI6AAdFqmpWuj6Td6lqMvlWlnC080hGdqKMk478CiVktRxu3ZHNy+DPhxBqUGnTeG/C0d9cKXhtnsbcSyKOpVduSB3Iq0fhz4IJyfBvh//AMFcH/xNeZ2HiPw1c+PPDfifUvEOitqt/PcGfbfwsbGAwsIbckMcYzzzy7N7Y9wo1tqJO7OQ1jwh8PNF0qfU9W8K+H4ra3QF3bS4SccAKBtySTgADkkgCqV1ofwv0vT7K58Q+HvC2h/bI98cOqWlpA4OASvPBYZAOCcVe8Wr9s8X+EdNn5tZL2W6dezvDEWjB9cMd31QHtSa5/yVfwl/156h/KGhf19xSteyJLDwX8PtT09LrS/DXhm7s5/mSa3sLeSOTBIyCq4ODmpx8OvBKnI8HaAOMcaXD/8AE1wuu6xd+GPEfj+48MoQot9PeQQhSIryZ2jeQBjt3+X5bHOAcAt61t6Hba7Y+Jbc2mleJLPSpLWVL863qkV2DIADHJHi4lZWzuBCgKQw44FK6tf+v6/zXyVun9f1/wAE3/8AhXPgj/oTfD//AIK4P/iaD8OfBBJJ8HeHyT1P9lw//E1w2hf2la+B/BPiGfXdVu9Svbu0huGmvHMUkMpKlGizsOAR85XeSMlq0/FMmoaL4k1PWvEcGu3GhRrC9neaLqDIunoAA5mtldfMw2XLFZQV4IAGKfW3nb8n+v8AXV63t8/xsdMfh14JY5Pg7QDxjnS4f/iaztc8KfD/AMP6W2oah4L0U2kbqJpI9JgYQKTgyN8vCL1YjOBz0BrN8Qape6t4/k0hNP12/wBMtNNhu1j0W/SzaSSV5BveQzxOVATAUEjJJYfdrpfB0Ory+EUtfFlvMLkPLDtu3jeWSDewjMhjZkLFNu7B5OaNbX/rcWzt/W1xqfD3wQ0P7vwj4fMb4b5dMgw3ofu89aB8OvBKnI8HaAOMcaXD/wDE1U+F7MPAVvalmeOxurqygZjnMUNxJHH+SqB+FdfTktRepzf/AArnwR/0Jvh//wAFcH/xNB+HPggkk+DvD5J6n+y4f/ia6SikM5s/DrwSxyfB2gHjHOlw/wDxNKfh34KMYQ+D9BKKSQv9mQ4BOMnG32H5V0dFAHN/8K58Ef8AQneH+f8AqFw//E0q/DrwSpyvg7QAcEZGlw9Dwf4a6OigDm/+FceCP+hN8P8A/grg/wDiaD8OfBBOT4O8Pk/9guD/AOJrpKKAOcb4d+Cnxv8AB+gNgADOmQnAHb7tJ/wrrwTtA/4Q7QMA5A/suH/4mukooA5sfDnwQAQPB2gYPX/iVw8/+O0D4deCVOR4O0AcY40uH/4mukooA5v/AIVx4I/6E3w//wCCuD/4mlf4d+CpJGeTwfoLuxJZm0yEkn1J210dFAHN/wDCufBB6+DvD/8A4K4f/iaU/DrwTtCnwfoG0EkD+zIcD/x32FdHXM/Ea7nsvh1rMtpI0UrW/lCVDhow5CFgexAYnPtSbdg06mNZaP8ADC70a71hvDHh200q3mMX2+6062igmAIG9GIwU3HaGOMkcZGCdX/hCPh/BPbp/wAIx4ajluciBf7Ptw0vy5O35fm+Xk47Vl/FKxtrD4O3NhaRLHa25s4Y41Hyqi3EQAx6YFUGkk0/xl4a0aQM8mk3V2kOcZkga2ZocY9B+7z3MZNOTSTfb/gf5i1sdT/wrjwR/wBCb4f/APBXB/8AE0H4deCWYlvB2gEk5JOlw8/+O1xJm1EfCm08fLrGotr0kcN4YxeObZy7qPswt9wixhtg+XdnBzu5q7NZ3msar48lu9b1aKHTpF+wwWt7JAts/wBkRywKEFuSPkbKZBO3JNKbUE+bpv8AL+v63KjHmeh1J+HPgg4z4O0A44H/ABK4f/iaP+FdeCdoH/CHaBgHIH9lw/8AxNcfca9q+t3/AIb0+ez1m/tp9BTUrpNGu47SSeZiq5MhmiYKvJ2o3JcZ4Fdf4MbV7Xw/cr4kiurVbe4kFs2ozxyTC2GChldHZSRyMliSFBJJyTTVr38/wdvzIUuZJry/FX/Icvw78FKrBfB+ggOMMBpkPzDOcH5fUA/hSD4c+CAcjwd4fB/7BcP/AMTW9Z3trqNnFd6fcw3VtMu6OaCQOjj1DDgipqWxRxEXhz4V3GtPo8Gj+D5dTjzvsktbUzLgZOYwNwwPapdY8MfDLQ4ludf0Pwnp0cz7VlvbS2iDt1wCwGT3rP8AEP2TXfFOj+FPDlrGG0bUIdTv7iBQsVgqksI8gf6yTJ+Uc7SxPB5rWSeIr7xv4y1LSp9Jt7mxnSzhfUbSS4YwiCOQRgrInloWdmJ+bJbp8oyltd/0lbX73+Fw62/q/b+vQ09T8NfDrSf7Pe88H6F9lvplgjul0qBoo2b7m5tvAY4UHpkgdxWr/wAK78FbNn/CH6BtBzt/syHGfX7tRJcWvjz4WC5v7byLfWNM8ySEtu8sOmeGwM46g4HQHirXgXULjVfh9oF/fEtc3OnQSysf4mMYJP49aqzXMu1vxv8A5Cvs+/8AwP8AMhHw68EgEDwdoAyMH/iVw8/+O0D4c+CAcjwdoAI7jS4f/ia6SikM5v8A4Vx4I/6E3w//AOCuD/4mg/DrwSzEt4O0AknJJ0uHn/x2ukooA5xvh34KZVDeD9BIQYUHTIflGc4Hy+pJ/Gk/4Vz4Ixj/AIQ7w/j/ALBcP/xNdJRQBzg+HXglQQPB2gAMMEDS4eR1/u+1IPhz4IByPB3h8H/sFwf/ABNdJRQBzf8AwrjwR/0Jvh//AMFcH/xNK/w78FSOWfwfoDMeSTpkJJ/8dro6KAObPw68EkAHwdoBwMD/AIlcPH/jtH/CufBGMf8ACHaBjrj+y4f/AImukooA5sfDrwSAQPB2gDIwf+JXDz/47QPhz4IByPB3h8H/ALBcH/xNdJRQBzf/AArjwR/0Jvh//wAFcH/xNK/w78FSOWfwfoDMeSTpkJJ/8dro6KAObPw68EkAHwdoBwMD/iVw8f8AjtH/AArnwRjH/CHaBjrj+y4f/ia6SigDmx8OvBIBA8HaAMjB/wCJXDz/AOO1n2/hz4YPrraXaaP4SbVoPna0itbYzx4wclANwxkHp6Vp/EC/vNK+HPiC+0xmS7t9PmkideqMEPzD6dfwrC0m41bwfL4Z0qVdIfR9Sc2kNtY20kUlq3lNKGLtIwmHyMGO1CSd3qKFq7P+rg9Ff1/As6h4Z+F2k3lvaaronhCyubogQQ3NpaxvMc4+VWALcnHFN1fw98K9AaIa/pHg/TGmBMQvba1hMgGM43AZxkdPWuY0TVfEWjeC9c8aIdJcLfXc9/bT27m4uVhndNvnhwEKxqFVfLYfKOfmOL97e6t/wn3iWXwZZQarMdOgTUIr4CP7PJtJiSNs/vCVYsYyFXJ/1qkkUndRv/W39a/kPrb+t7G7qfhb4Z6PYR3ms6F4TsbRiEjnu7O2jjJOSAGYAc8nH1qxaeBPh/f2cV1Y+FvDVzbTKHjmh063dJFPQghcEe9ZfhrWPC2gfCfQ9dV2ls7KxENrNNAPtLE4UxIoz87OoXYpIJA5IANafw60m90rwzM+p2/2OfUb641D7GGz9lWZy4j44yAeccZJqmvea7f1/wAEXRP+v66fMsL8O/BS7tvg/QBuGDjTIeR6fdpB8OvBKsCvg7QAQcgjS4eP/Ha6SikBzf8AwrjwR/0Jvh//AMFcH/xNB+HPggnJ8HaAT76XD/8AE10lFAHNn4deCSAD4O0A4GB/xK4eP/HaUfDvwUIyg8H6CEYglf7MhwSM4ONvufzro6KAMOw8E+FdKvo73S/DOj2V1Fkxz29hFG6ZGDhlUEcEityiigAooooAKKKKAKmq6VZa3pc+napbrcWs67ZI2JGecggjkEEAgjBBAI5qjpnhLRdIvje2Vq/2xomie5nuJJ5ZVJUkO8jMz/dUAsSQAAMDitmigOljB07wXo2kXiXGmLfWojdnjto9TuRbJnOQLfzPKC8n5QuB2FWn8N6TJp2qWL2mbbVmke9TzH/el1Csc5yuQAOMVqUUB1uY+peFtM1TyvtH22ExRiINZahcWpZB0VjE6lwOcbs4ycdTS3fhXRb3RbfSpbBI7O0ZWtkt2aFrdl6NG6EMh68qQeSO5rXooAwfs/8Awilht0fS9a1rzpcvH/aP2iROOu67nGF46KevbqadY6/qV3fRwXHhHWbGNzhri4msykfHUhLhm9uFNblFAFS50uzvNQsr25h33FiztbvuI2Fl2twDg5BxzmrdFFAGB4v0K61rS4JdJljg1bTrhbywllGU8xQQUb/ZdGZD6Bs9RReeHbLxRBpl/rtneWWoW8JKrbajLBJbmQL5ieZA67uVA6kHHFb9FHSwXf8AX9f1cyrPwxotho8+l2+mwfY7rcbmORfM+0lhhmlZsmRiOrMST3NRaT4U03RZS9jJqRHl+UI7nVbq4jVeOAkkjKOnUDIraooAyY/DGkRaPp+lx2mLLTZI5LSLzX/dtGcoc5yce5Oe9Q6l4O0bV7tp9QiupBJjzbdb6dLebH/PSBXEb56HcpyODkVuUUeYGRq3hfSdangnvIZo7i3UpFcWd1Layqh6p5kTK23gHbnGQDjio9TtNR03wwbHwsjzXbYhhmvbp5vI3HmV3kYu4UEnbkk4A46jbopNXVg8zP0HRrbw9oNnpVlkw2sQQM33nPdj7k5J9zWhRRVN3dwWgUUUUgCiiigAooooAKKKKACiiigAooooAKKKKACqmq6bbazpF3pl+nmWt5C8Ey5xlWBB/Q1bopNXVhptO6OW0rQrzVfCD6B46t0vPs8qwi4STaL5IyrxzYUgocgZX+8p6qRnWufDul3niGz1y4td2o2KNHBOJGG1WBBBUHa3U9QcZOMZrToqm7u5Nlaxz8fgfw/FqQvEs5crP9pW3N3MbZJc58xbcv5Qbd824Lndz15rQTQtOjfU2S3w2qtuvDvb96dgj9ePlUDjH51oUVNk1Yq7vcxLzwfol7ZafbS20sS6agjs5ba6lgmhTbt2rLGwfBAAI3YOBnOKW60qew8Lz6doUCX8jqyLHq+oTSKwb72+VhI5GCePw46jaopv3r36iWlrdDA8D+FIPBPgvT/D9tO1wlmhBlZdu9mYsxAycDLHAycDua36KKcm5O7ElY5Wy+HHh3TZZZNOGq2nnTtcyrBrd6iPKxyzFRLgknrkc1e1Xwboms3z3l9bTedLGIpzBdzQLcIM4WVY3USryRhwRgkdCa3KKQ/M53xTpl/qGhxaDokSW1ref6NdXCkKLS2x8wRRyWI+RccLnJ6YO7bW0NnaRW1rGsUEKCOONRgKoGAB+FS0UdwCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAGyRpNE0cqK8bqVZGGQwPUEdxWJpPgzQ9EvIrmwtZvMgQx2/n3c062yHgrEsjMIhgAYQKMADoBW7RR1uBgz+CdAuNUe+ms5C8kwnkgF1KLeWUYw7QBvKZsgHcVJyAeoBo1TwVoer6lLf3UN1DdTRCKaWyv57QzIM4D+U678ZON2cZreopWVrDuzndR8BeHNUt9LgnsZII9IybBbK7mtfs5IxlfKdeccZ9z6mtPSdGttFt3hs5b2RXbcTeX890wOMcNK7ED2BxV+imIKKKKACiiigAooooAKKKKACiiigAooooAy/Eekf27oc1h5enS+YVO3U7H7XAcMD80W9c9OPmGDg1zuheAf7Hup5vsfhCPzrd4c6f4a+zOd3Zm89tyeq4GfUV21FKwHBaV8N/7N1a1vPsPglPIlD7rPwr5Ewwf4JPtDbW9Dg49KTVPht/aGrXV59h8EP58rSbrvwp50pyc/PJ9oG9vVsDPpXfUUwOJ13wB/bF3BN9j8ISeVbpDnUPDX2lxtHRW89dqei4OPU0678BfafDdhpf2Pwi32R3bZN4b8y2Xcc/u4fPHln1O459q7SqOq63pWhWq3Ot6nZ6bAz7FlvLhYVLYzgFiBnAPHtQ/MF5HL2ngD7N4ev8ATfsfhAfa3jbZD4a2W7bTn95D558w+h3Daeeaf4d8Cf2FqMlz9k8Jxb4Gizpnhz7JIc44L+e2V45XHPqK7CCeK5t457aVJoZVDxyRsGV1IyCCOCCO9PofmJeR59YfDP7FqVtdfYPAy+TKsm638JeVIMHPyP8AaDtb0ODg84p+qfDb+0NWurz7D4Ifz5Wk3XfhTzpTk5+eT7QN7erYGfSu+ooGcVr3gL+2bq2l+x+EZPJtkgzqPhv7Uw254RvPXanPCYOOeTSXPgH7R4cstM+x+ED9lkd9kvhrfbLuP/LOHzxsPqdxz7V21FG4LQ4yw8B/Y/D2pab9j8JL9tKHZb+HPKt22nP72Hzz5vt8y4PPNM0DwB/Yuqfa/sfhCP8Adum7TvDX2WX5hj7/AJ7fL6jHI4yK7aigOljz6w+Gf2LUra6+weBl8mVZN1v4S8qQYOfkf7QdrehwcHnFS6v8Of7T1i6vfsPgp/PkL7r3wt9omOf78n2hdx98Cu8ooA4nXPAH9rz2sn2PwhJ5FqkGb/w19pYbc8IfPXYnPCc455NFz4B+0eHLLTPsfhA/ZZHfZL4a32y7j/yzh88bD6ncc+1dtRRuC0OMsPAf2Pw9qWm/Y/CS/bSh2W/hzyrdtpz+9h88+b7fMuDzzTNA8Af2Lqn2v7H4Qj/dum7TvDX2WX5hj7/nt8vqMcjjIrtqKA6WPPrD4Z/YtStrr7B4GXyZVk3W/hLypBg5+R/tB2t6HBwecVLq/wAOf7T1i6vfsPgp/PkL7r3wt9omOf78n2hdx98Cu8ooA4nXPAH9rz2sn2PwhJ5FqkGb/wANfaWG3PCHz12JzwnOOeTT7jwJ5/hW00j7J4TP2eZpfLl8Ob7UZzykHnjY3PLbjnnjmuzooDY4iw+H/wBj0bUrH7H4PX7asY22/hnyoW2tn97H5580enK4PPNO8PeAv7D1YXn2PwjFiNk3aZ4b+yTcjH+s89uPUY5HpXa0UB0see2Xwy+yX9vcfYPAq+TKr5g8I+XIMHPyv9pO1vQ4ODziptX+HP8AaesXV79h8FP58hfde+FvtExz/fk+0LuPvgV3lFAHGa94D/tlrI/Y/CUn2W1SD/iYeHPtRXGeI/367E54TnHqaZL4A8zwzBpX2PwgfKuGm2P4a3WoyMZWDz/lf1bccjjFdtRQHb+vI4vTPAX2DSNTsvsfhFPtyIu218N+TC2CT++j88+aOeBlce9N8P8AgD+xdXS9+x+EI9qMu7TfDX2WbkEcSee2BzyMcjjiu2ooA89svhl9kv7e4+weBV8mVXzB4R8uQYOflf7SdrehwcHnFWdc+Hn9r63c3/2LwbJ5zA7tQ8L/AGmc8AfPL567jx1wOMV3NFAHEav8P/7TjsF+x+D3+yWqW/8Apvhn7QFxniMeevlx88Jzjnk06XwF5nhaHSPsfhE+Xcmfy38N5tRkYysHn/K/+3uPHGK7WigDidN8A/YNL1O0+x+EE+3Rqm218NeTE2Dn96nnnzR6DK4PNO8OeAv7C1qK/wDsfhGLYrDdpnhv7JOMjHEvntgeoxyOK7SigOljzuP4YbLtJvsHgQbZA+V8IYfrnhvtPB98Vb1z4ef2vrdzf/YvBsnnMDu1Dwv9pnPAHzy+eu48dcDjFdzRQG7ucXrHgL+1LbTovsfhF/sdsIMX3hv7Qq47RDz18tPRPmx601/AG/wvFpP2Pwh8l0Z9jeGs2oyuMiDz+H/293TjHeu2ooA4zSPAn9mabqlr9k8Jp9uhEW2y8OeRG3XiZPPbzV5+7le/PNRaD8Pv7G1u3v8A7F4Oj8kk7tO8MfZZxlSPll89tvXng5GR3ruKKA8jzuP4YbLtJvsHgQbZA+V8IYfrnhvtPB98Vc174e/2zrlzf/YvB0nnEHdqPhj7VOcKB88vnru6f3RgYHau4ooDd3OJ1bwD/aVrp0P2Pwg/2O3EOLzw156LjtEvnr5af7OW+tPPgPPhMaP9j8JcXXn+WfDn+ifdxn7P5/3/APb3dOMV2dFAHE6T4A/s201GH7H4QT7ZbmHFn4a8hG56Sr57eYn+zlfrTdB+H39ja5b3/wBi8HR+SSd2neGfss4ypHyy+e23r/dORkd67iinfW4rHnjfDHdfG4+weBeZN+T4Ry/XP3vtPX3x1q3rvw9/tjW7i/8AsXg2TziDu1Dwx9qnOFA+aXz13dP7owMDtXcUUulh9bnGav4E/tPTtMtvsnhN/sMHlYvfDn2iNenEKeevlLx93LdueKjX4f48MPpP2Pwf810LjYPDOLX7u3Jg8/mT/b3dOMd67eigNjitH8Bf2Xa6jF9j8Ip9ttjBiy8N/Z1bPaVfPbzE9UyufWo9D+Hn9j63bX/2LwbH5LE7tP8ADH2accEfJL57bTz6HjIruaKBWPPG+GO6+Nx9g8C8yb8nwjl+ufvfaevvjrV3xD4A/tzW5r/7H4Ql8wKN2p+Gvtc5woHzS+eu7px8owMDtXbUUdLFX1b7nE6l4A+36Xplp9j8IP8AYY2TbdeGvOiXJz+6Tzx5S+oy2TzmnR+AtnhWbSPsfhEeZcify18N4tTgYy0Hn8v/ALe4ccYrtKKP6/US0OJ0jwB/ZkGoR/Y/CCfbLVoMWXhryFbPaUee3mJ6p8ufWmaH8PP7H1u2v/sXg2PyWJ3af4Y+zTjgj5JfPbaefQ8ZFdzRQKx543wx3XxuPsHgXmTfk+Ecv1z977T198dau+IfAH9ua3Nf/Y/CEvmBRu1Pw19rnOFA+aXz13dOPlGBgdq7aijpYq+rfc4nUvAH2/S9MtPsfhB/sMbJtuvDXnRLk5/dJ548pfUZbJ5zTo/AWzwrNpH2PwiPMuRP5a+G8WpwMZaDz+X/ANvcOOMV2lFH9fqJaHE6R4A/syDUI/sfhBPtlq0GLLw15CtntKPPbzE9U+XPrTNC+Hv9j63bX/2LwbH5LE7tP8MfZpxwR8kvnttPPXB4yK7mijrcVjz28+GX2q/nuPsHgVvNlZ8zeEfMkOTn5n+0jc3qcDJ5xVvX/h//AG1rEt79j8Hyb1UbtR8M/apjhQOZPPXI44GOBgc4zXb0UdLFXOK1PwF/aGk6ZZ/Y/CL/AGGN023XhvzoV3EH9zH548occjLZOOlJF4B8vwxPpP2PwgPNuVm2J4a22pwMZaDz/mf0bcMDjFdtRQ9RLQ4zQ/Af9kLfj7H4Sj+12rQf8S/w59mDZ7Sfv28xPVOM+oqto/w5/svWLW9+w+Ck8iQPusvC32eYY/uSfaG2n3wa7yinfW4mrqx57efDL7Vfz3H2DwK3mys+ZvCPmSHJz8z/AGkbm9TgZPOKu+IfAX9uas159j8Iy5jVN2p+G/tc3Ax/rPPXj0GOPeu1opdLD63OJ1HwB9u0jTLL7H4Qb7CjrtufDXmwruOf3Ufnjyh6jLZPPFPt/AnkeFbvSPsnhMfaJ1l8uPw5stTjHLweed7ccNuGOOOK7OigNjkPC3gn/hHdWe8+yeFod0Rj3aR4f+xS8kHmTznyvHTHpzxXX0UU7gFFFFIAooooAp6npcGr2f2a7kuo49wbNpeS2z5H+3Eytj2ziuB8Fi10fwhqnijU77WLt9NudRDfaNWup18qGaRVXy3kKEhUABIz3znmvSq5/SPCVvYeF77Q76UXttfT3ckuY9mVnldymMnoHxnvjPFJ35Xy7jVm1zbX/Axm8SeJdJh0jVtfXSzpuqXMNu9pbRSLNZGYgRkys5EuGKq3yJ1yOmDmwWnitvi9q4t9a0aOX+ybY75NIlZfL86fauBcj5hzls4OR8oxzuW3g7VJTptrr+vRajpmlypNbwpY+VNM8f8AqjPJ5jB9vDfKiZYA8Dg2tV8N6nJ4pXX/AA9q1tYXT2f2O5ivLE3MUqK5dGAWSMqylnGckEN04qnbmTXd/dbT8RRvytPsvvvd/gN8Wa9Loq2yt4m8OaC0ikiTWFJE7DqEXzo8AZHO5uvTuYdN8QnxV8KZdZaKOJ7mwn3rFJ5iblDKSrd1JUkH0Ip974U1JvFz6/pGsQWc9zZR2V0s9j5/yIxYNCfMXy2+Y/eDqcKSDjmO28IarpngSPw7pWt22/8AfLNd3mnmUyLIzscIkqBW+frkjjoO2VSLlTlFbu/6lRdpp9Eafgr/AJEHQP8AsG2//ota2ZWKQuw6qpIzWP4V0nUtD0K303VNQtL8WsSQwSW1m1vhFUKNwaR8njqMD2rWn/49pf8AcP8AKta0ruUl5mdKLUYxZ51pXjbxTJ4X0DxRq9vpMenalPb281nbpIZVEziNZhIXwPmZT5ew4XPzk1Z8d+Op/CbXlxH4g8MxNZwicaLePi7u1HJCv5q7GYZCjy35AOTnAo/D7w5qWtfD/wAKnWtUtZdItoYLqGytrFo3d0wYxLI0jBgpw2FRMlV7ZB0L/wCHWpXFn4l06w8Rx2eneIJZZ5gdPElzG8iBWUSl9pTjoU3BTgMMAgldXtvr+ll+f9bWrc2u3/D/APA/re3qHiTX7nxvaaD4fj06GG50r7e93eo8hgPmBceWrLvyDjG5ccnJxtNqbWdasPCb3PiC60XQbyK5aF769bNqUDEJKE8wEbxtwhkBXd1OMG3beGfs/i231v7Xu8nSxp3k+Vjd84bfuz7Yxj8aZ4n8NXGt3mk3+nX8VnfaTcNNAbi2+0QtuQodyB0OcHghgR+NJ6LTz/8ASnb8CY36+X5f5nJR/E3UJPDupzWL6PrF3p+r2lgt3Yufst2k7xjK4dtjASFT8zAFc89Kfonj7xBc+IdPstTi01reXWbzR5nt4XRpHhSRxKuXIRcKoKncSSTkcCtI/Du6nTUn1DXvtNzqOpWWoSS/YwgQ25jPlqob7p8vAySQCMlyCTJZ/Dv7Jqtre/2pv+z67daxs+z43efGyeXndxt353d8dBVRt1/r4f8A7b+rDltp/Xxfry/1ch8J+NL3xDrktvdanottPHNLHLoDwyR39uqnhizSfPkbW4iC4bhuMnotSufEEXijR4NLsLafRphN/aVzJJiSAhR5ewZ5ycg8H8KzLjwlqWq65pt3r2rWd1b6Vc/abQQab5VxuAIAeUyMCCD8wRE3EDoOK09S0S9vvFGj6pb61c2lrp4mFxYRj93eb1AXfz/CRkcHr2qY7K/9af5/8OD3f9f1/Whs1xnjzXLu01LQ9CsBe+Zq0kxlGnhfPeKJNzIjsQsZYso3llwM4IOCOzrE8Q6BNq01hfaderY6npsrSW08kHnRkMu10dNyllIPZlIIBz2Keo0Z3hqSyv8ATdT0ay/t3SbyCQC5i1G9a4uoN6gqyyPJKu0gcFWIBz0bNZ2gE2XxQv8ASND1C+vNKttNDX8d3ey3Yt7ouPLUSSszKxTcSu7GNpwO+rD4W1SDTtZmj11I/EGrKA2pR2WI7faNqCOEueFBJ+Z2yxJJxhRD4M8J6v4Ss47BtU0u4shueUxaZLHcXErcmV5WuH3MTySV56cU93d9vv0t+H52I1tbz/X+vlc5rwiJNAbRI/Gtl4ptNVmfyPt15rUlxaT3BzhdiXLqMjON8YXI7HFep1y0nhfVdX1Cxl8VaxZ3lrp9wt3Ba2GntbK8y52NIWlkLBScgDb8wBOcYql4hh1rUfiX4bGh3OpW9rpryyaqux47SWJ0woJICyvu6Bd23knacZfZPv8A1+vyKlu3/X9bD/Hd/e34PhnQryeyuprZ7q9vbZ9slnbr/db+F5G+VfYOR92sFNdnl8DeArTUtZuLKDU7BLjUtQ+0FJnjjtw7L5udyl2K5YHdjOCCc11mqeAtG1S9v75n1K2vL9As0lrqt1Cr4XauUjlVTgdsevrWZbfDO2sfCuhadZ6hI2oaHJHPa3t4HuUEqx+Wf3TyHbGRn5EZQpwVIIzSXwtPy/X9LadfnoS6W8/0/ryKXhW4v5vEmt+HDNq1ppk1lDfadJeXLS3kKOzo+Wl3OoYpuVXJZcnO0/Ktnw+32L4pX2kaJqF9eaVbacrX8d3ey3Yt7ouPLUSSszKxTcSm7GNpwO+lH4U1VLbWLw69GviLU4VgXUo7HEVqiZ2LHCXPQszfM5JZsngBRH4L8J6v4StIrB9T0u5sQWebydMljuLiVuTK8rXD7mJ5JK8+1Nb38vv3/L8xS8v+G2v97/A66iiikMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKep6pBpFn9pu47qSPcFxaWcty+T/sRKzY98YrM0Pxto3iO48rSP7Rl+Z1Mkuk3UMQZCQymR4woIIIwTnIx1rfriPh9/yIupf9hTU//SqWk3ZNvoO17Jdzt6K8v+GngbRLz4c+EdWMLxarFb290uoq2644HMXmMCREVJUxjC7SQMdax/iFfWWr6J4r1iz8P6OkmjO9oNc1DUPIvYLmNQV8j90xXDFdg3puJOB82S5Wi2n0FC8kn3PV9avdTsbaB9H0n+1JXuY45Y/tKw+VET80mW67RztHJrRrzz4h6bY3nhnRNXurG1m1KLUdMEd48KmWMNdRbgr4yAcngetWvH9hBqnijwXZXm9rabUpxLGrlRKotZTsbHVTjBU8EZByDVcrt82vy/zEmnr5X/M7mmyNsjZ8Z2gnFeXJf6b4Lj8dwRaeP7Ltr60ittNtmEMQeeGIbRjCxoztlj0wWOD0LPDsJ0b4tS6VBpWi6Ek2hSzXOnaLeebGzrLGEkkj8mMI2HcA4OR34qNWtO36XKtbf+tbHoPhfW/+Ek8K6brP2f7N9ut1n8nfv2bhnG7Az9cCqVz4h1Ky8bWGjXWmWv2LURL9nu471mkHloGO+IxADrjhz6+1eaaVoun6b8MfBGr6bEF8SXN1Yrb3eN08is4EkW7r5Qh8zKfdABOM816B4h/5KT4P/wC33/0SKp7/ADa/r7yZXX4fmdbRXmfia3vNb+Kj6VdaLo+t2UGlRz2llrF40URcyOJJVTyZA7gBFyQNobj75rrfBOlaponheHT9bkheaGWXylhuHnWKEuTHH5jqrNtQhckdqS1V/wCt7Dejt/W1zforkvimpf4Xa6odkLW4AZDhl+Ycg9jWV/ZGl+GPiNoq6BbrbvLYXUmq+VlpLiNQhSWY9ZH35AZiWO5uvNLmSev9b/5B6HoVFeKSX0FzeeE/E2l6Bo+i/wBr6rCbe+j1HOoXkLsVZZUEQ35UgsDK2047itzRvCOieJ/GXj0a9YpfqNSjijS4/eJDm0hy6I2VV+fvgbuAM01fXyv+Fv8AMV1e3p+N/wDI9PorzvXksrjVrbwo+g6f4mm0vS0up7nxJdhI0jyUD58qTdIdrFmCqAMc84rgLONNe8L6CupSLewQ6ProRfO8+N0imVYhvI/eKoVCpI/hU9qUmoxcuiv+F/8AIuMW2kz6Dorxnwjp5lu9YsbbUpNMl1Twnp802obvnFzL5wM7HIy+SOcjoAMADHQ+CIk8N+KU8OX3hXSdH1GbTRcC90SXMF4qOFfzEKoysCykFg/3mAbrm+W0uX1/C9/yM1K8eb0/G3+Z6LRXKaTB4eT4neIJ7DUZpdektbZb+0ZyUhjAbyyoxgEgnPJxxwM89XU9LlPexm6v4g03QxENRncSzkiG3gheeabGN2yKMM7YyCcA4HJxTLrxJptlpMGo3TXMcVyQsMRs5vtEjEE7RBs8wtgEkbcgAk8A1yV1Y32ofGTUIV1q40kDRbdrV7aOJpXXzZPMCmVHULkpuwuf9XyOht6F44S0+HX9ueJrtZxFczWsNxBFhtQ2StHE0aL955AoIC8EnIwOiv7t3/Wth21t/W1zaj8Z6DJ4evtb+2sllp+8Xhkt5EltyvJDxMokU4wcFckEEcEVPpPiSx1qd4rKHU42Rd5N5pVzaqR7NLGoJ9gc1zGieDrvXPDPiNvFyNYXPiuTzLi1tXBazi8tY0j3kEM4Vck4xknjAqxp15q2k/EaDw9JrU+u2c+nPdTG6hhWayZXVUJaFEXY4LABlJyhIOMiqtrb+trsT2uv610O1rI8Q+K9E8Kx2j6/fraC9uFtrcFGcySN0ACgn8eg7mnw+JdHuPFFz4dhvkbVrWBbiW12sGWNjgNkjB6joTjIz1ryXxtrUWoWusaprFhrMNxHeW1np0EujXWyCBLuIvJ5nl7N0pXPB+6sajnIqVq1/XW39f8AAHb+v6+//hz1nWfEul6DJBFqE0xnud3k21rbS3M0gX7zCOJWbaMjLYwMjJ5FRf8ACX6GdBttYivTPZ3bbbf7PDJLJK3OUWJVLlhtbK7crtbIGDjkZL7Ubr4st/ZSfZxqmiQm1ur+JozBHHNJ522FwrmT54vlIAHDN0CsnhLV9P8ACPhnxHql7ezz6XJrcxsZGJllvGIRW8sD77PMJMBeCTxgdHpZt/1ra39fiHXT+tLncaLr2neILWSfS52kEMphmjlieGSFx1V43AZDgg4YDgg9DWjXJ+BNJ1S3j1XW/EEKWmo67dC6kso23C0RY1SOMt/EwVcsRgZJxwK6ym1YQUUUUgCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqC3srW0gaC0toYInZnaOOMKrMxJYkDuSSSe5Jrn/AB/4g1Lw34aju9Eitpb2a+trWNbpWMf72VU52kH+L1qEeNPM1rQkCxQafqOmXN7cNOpEkBi8r5TzgY3sGBHUdqWlr/L8Ljs72/rsdPaWltYWcVpY28VtbQqEihhQIiKOgCjgD2FULjwt4fu9Sl1C70LTZ72eMxS3MtnG0kiFdpVmIyQV4we3FZdn48tLm5sftGlapY2OpSCOx1C6ijWG5Y8oMBzIm4AlfMRM9OpANe7+JOn20mreVpWrXUGiTtFqVxDCgjtQAGL5ZwXG05xGGYAcqMrl9df67/mvvEvL+v6sbOp+EPDWtTpNrHh7StQljQRpJdWUcrKg6KCynA5PFWrbRNKs4bSG00yzgisWLWqRW6KtuSCCUAHykhmBxjqfWrkciyxrJGdyOAykdwa5D4k+JNV8N+G/tGjafeTStNApuoPIKRAzxqVYSOCSwYqMKcE8kdaNU7AtTppdJ06dLxJ7C1kW/GLtXhUi4G3bh8j5vlGOc8cVDp3h7RdI8v8AsnSLCx8pGSP7NbJHsViCwG0DAJAJHcgVmX3jFdH8L3+u6/omp6XbWOC8cxgkkdSQNyiKVxgZ7kH2p+m+MYL/AFuDTLjStT06S7heeykvYkRbpEK7ioDFkIDKdsio2D04OBb6B0LmneFfD2j3732k6DpljdyAq9xa2cccjAnJBZQCckA1XvfA3hLUr2W81Hwvot3czNuknn0+J3c+pYrkmoZfGMdrq9raajo+p2NveTi2tb+dIvJlkIJVcLIZE3YON6LzxwSM1NQvw3xM0e1uYtfsyEuFtmjuIRY3vyBmMiBy5K/wllXBz1oVnYG7XN7U/D+ja3axW2s6TY6hBCd0cV3bJKqHGMgMCBxxVixsLPS7GOz0y0gs7WIYjgt4xGiDOeFGAOTWLqnjK307xKfD9vpmo6jqhtFvEgtI0w8ZdkJ3u6opBX+IjORjJ4qW08VQ6l4Zg1jStOv77z2KLZxIizK6kq6NvZUUqVYElgMjAJyMnS4W1t/XcXxpoc3iXwZqWj2phEt5F5amcnZ1B5wDxgelW9I8P6N4fikj0LSbHTElIMi2dskIcjoTtAzXI+KPGC3/AMN/ElxY/bdI1HS/3VzDMwjmtXyrA7kZlwVIIZWIwevUVu6Z4xtr/WYNMudO1DTZbuFp7J72NEW8RcbigDllIDKdrhGwenBwlvp/XUXqXLfwr4es7mW4tNB0yCaaVZpJYrONWeRSWDkgZLAkkHrk1oQWVrazXE1tbQwy3LiSd44wrSsAFDMR944AGT2ArkPD0Dad8Udf06G8v5bNdOs7hIbu+muAkjyThivmM23IVeBgcCr+peObSwu76ODS9S1GDTCBqN3ZxxtHaHaHIYM6s5CEMRGrkA+vFPZD3bNbUvD+jaxcW8+r6TY381qd1vJdWyStCcg5UsCV5A6egpV0PSUUKul2ShVlQAW6cLKcyDp0c8t6nrmrkE8Vzbxz27rJFKgdHU5DKRkEfhWN4j8V2XhmTT4ry3vLmbUpmt7aK0h8xnkCFwuMjGduMngdSQMkD00YJ31RoRaPpkDMYdOtIy8C2zFIFG6Fc7Yzx90ZOF6DJqLSPD2i6Asq6DpFhpizEGQWVskIcjpnaBnGTVXSPFEGqxagHsb6yu9NYLdWNxGrTJldykeWzqwYHgqx7jqCKi0zxcl5rv8AY2o6RqOjX725uYIr7yWE8akK5VopHXKllyCQfmBAIzg/r9f+CLS39ehsx2NpFfTXsVrCl1OqpLOsYDyKudoZupAycA9Mmp64nTvijpmp6dZanBpGsJpN1Mlv/aM1uiRRSu+wKQX3sNxUb0VkyfvcNjV1/wAWjw9589zomq3Gn2qh7rULdIjFAvViVaQSMFBySiN3HJBANirO9jR1bQNH1+GOLXdJsdTjibdGl5bJMEPTIDA4NGpeH9G1myis9X0mxv7WEhooLq2SVIyBgEKwIHBxxXP6t8TNE0fUZbWeG9mWA2xmngiV441nOI3+9uILYGACeeAQCRPH4/0+TQpNQaw1COdL/wDs0aeyRm4a5yAIxhynOQclwAOpHNL+v0/VIm6/r+vI0LXwd4YsbS6tbHw5pNtb3ihLmGGxiRJ1HQOoXDDk8GrelaHpOg2z2+h6XZ6bA772is7dIVZsYyQoAJwBz7UaVqMuo2zSXOmXmmSq21re8EZYehBjd0IPsx98GqnhrxRZeKbe/l0+K5iWxvpbGUXEWwtJGcMV55Xng/oKrqPoVdM8KfZ/GV94o1O6iu9SngFnAYbfyUgtg24JjcxZiTksTg4GFXvu3VpbX1uYL23iuYSVYxzIHUlSGBweOCAR7gVNXn+q+M9cFnq+s6VJpMGlaXctaRQXkMsk2oyo211RkYeWS3yIAkhLDOOQKm/Tt/X5sdm9f6/qyOy1bQ9J163S31zS7LUoY33pHeW6TKrYxkBgQDgkZ96jv/Dmiarp8Fhqmjafe2dvjybe4tUkjiwMDarAgYHHHapdF1W313QrHVbLP2e+t0uI9w5CuoYZ9+au1TTTaZKaaujM0jw1oXh9pToOi6fphmwJTZWqQ+ZjOM7QM4yevrWnRRSGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBzHj7R77WtDsbfTIPPli1WyuHXeq4jjnRnbkjooJx19K5vW/A2pap40urWKAx6HdaZfoLwSJ+5muvLDJsJ3feRnzjHzkV6XRSaTVvX8Ul+hSk07r+rf8OcBJY6/wCI7HRNF1LQJdKj0+6t7m8vWuYWhk+zsGCwBHLncyrjeqYXOecCnXPhvVpPBvjyxS0zc6tPdvZJ5ifvQ8KqpznC5II5xXe0U5e9e/W/42/yRMfdcWulvwv/AJmLceG7bVbCxTUZdRhktoQmLLU7i15wM58mRQ3TvnHbrVXxnoNzqvgS60vSR5lzGIpLdZ5ifMaKRZFVnbJ52Y3H15PeukopybYorlt5HFeK4tY8XfDrVbGDw/e6ffSCNYre7mt8yYdWJDRysuBg9SD7V2Ugk+zsISqybSFLDIBxxmn0VLVxrR3PIbbwt4nvF0FtT0vWJNUs9TguNSvbzXi9tLtkJd4LdZSm30BSMqMYBOa6XxDNrcnjrRr6z8J6rd2mlG4Ek0U9mom8yMKpQPOpxnruC13NFP8Azv8Akv0Fa7v/AF/Wpzdvpd7/AMLNudYe3KWUujQ2yyF1z5omkYrgHPAYc9PeuUj8PeJ7DwppVktpfPbpq17PqNlpt+lvczQyTSvEUl8xABlkYgOpxx6g+n0UdLf1vcpu7v8A1tY8oi8G66fC/jiBNKuIJNVaGSwt7rU/tcsgWJBtaWRzhsqRgsVB4DFQGrp4rXVfEHi7StVvtGuNHtNHjmKR3kkLzXEsihPl8qR1CBc5ywJJHHGa7Cijrclq5welS65/wsvUNWuPCOq21lf2VraLLJPZnymjeUszBZyduJB0BPB4rLm8HzadrWtrd6F4i1i01K7e7il0jX2t0CyD545ImuYgCCDyAwKkcjGK9Qoo3/If9foV9PtILDTbaztITBBbxLFFEW3FFUABc5OcAY6muO8fXU1l4q8FT21hcahImpT4trZoxI/+iTDgyMq8deWHT8K7mqd5pNlf31jeXcPmT6fK0ts+9h5bMjITgHB+VmHOetO93diSSVl2OLutI8VXGleLNZ0uH+zNZ1WOJLG0eVDJFHEuPmcFkWRt0mCCyrlMnrVbQvDepR/ELTdWXRtXsdPg0+5glbV9ba9m8xmiI+QyyKqkKeVbnB3BcLn0mipsh9LHL/DfR77QPh1pGl6tB9nvLeJlli3q20l2PVSQeCOhrkfFnhzxPrH/AAlFrPpurajNeLImlTW+uG0sY4TFhUeJZVJfduzujYMSAWC/d9Wool7zbZUZcrTR5VqvgzXrmK+EFhuMsmiFP30YyLeQNN1b+ED8e2a1IvC8o0PxJba74cbV7e/1uW6jtYp41leJguySNi6hWBHd0YDJBzgH0Gin1b73/Fp/oZqKSS7f8H/M5TwNp2sWEepf2mdQisZrgPYWmqXi3Vzbpj5g8gZ8gtyoLuQOp7DS8NXPiC6t78+KLC2spUvpUtFt5N4ktgf3btycMRnI4+g6Vs0UFBXmkXhzWPD2ra0NP0OfWJby4ml0u9e7hW3sBOSZA6MwdSHZiWRHZl2jPG0el0UrJ7+g02l+Jm+HdHTw94Y03RopDKlhax24kYYL7FC5x74zWlRRVSbk7slJJWQUUUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFYPizxt4f8D2MF54o1D7Db3EnlRv5Mkm5sE4wikjgGuVH7QPwxY8eJu2f+PC5/8AjdK6HZnpFFecR/H/AOGcsqxx+Jcs5CqPsFzyT/2zpG/aB+GKsQfE3IOD/oFz/wDG6Yj0iivOG/aA+GSHDeJcHAP/AB4XPQjI/wCWdIf2gfhkFDHxNwTgf6Bc/wDxugN9j0iivOB+0B8Mihf/AISX5QQCfsFz1P8A2z9qdF8ffhpO5SLxJuYKzkfYLkcKCx/5Z+gND01YHotFeb/8NBfDE/8AMzf+SFz/APG6D+0F8MQSD4m5H/Thc/8AxugD0iivOH+P/wAM42Cv4lwSA3/Hhc9CMj/ln6Gk/wCGgfhjtB/4SbgnH/Hhc/8AxugFqro9Iorzlfj98NGheVfEuUQgMfsFzxnp/wAs/akX9oD4ZOcL4lycE/8AHhc9AMn/AJZ0Aej0V5v/AMNBfDE/8zN/5IXP/wAbob9oH4ZKxVvE2CDg/wCgXP8A8boA9Iorzhv2gPhkhw3iXBwD/wAeFz0Iz/zzpP8AhoH4Y7Qf+Em4Jx/x4XP/AMboDc9Iorzlfj98NGheVfEuUQgMfsFzxnp/yz9qRf2gPhk5wviXJwT/AMeFz0Ayf+WdAHo9Feb/APDQXwxP/Mzf+SFz/wDG6G/aB+GSsVbxNgg4P+gXP/xugD0iivOG/aA+GSHDeJcHAP8Ax4XPQjP/ADzpx+Pvw0WBJj4k/duzIp+wXPJABP8Ayz/2h+dAHotFebj9oH4YkEjxNwOv+gXP/wAbpU+P/wAM5GKp4lyQC3/Hhc9AMn/ln6CgHpqz0eivNv8AhoP4Y/8AQzf+SFz/APG6U/tA/DFWIPibkHB/0C5/+N0AekUV5zJ8fvhpEVEniXBZQw/0C56EZH/LOkP7QHwyCB/+El+UkgH7Bc9Rj/pn7igD0eivNx+0D8MipI8TcLyf9Auf/jdKv7QHwyc4XxLk4J/48LnoBn/nnQGx6PRXm3/DQfwx/wChm/8AJC5/+N1JN8ffhpbzyQzeJNskbFGH2C5OCDg/8s6L9APRaK83P7QPwxGM+JuoyP8AQLn/AON0p+P/AMMxEsh8S/KxKg/YLnqMZ/5Z+4oDY9HorzcftA/DEg48TdBk/wCgXP8A8bp0Xx++Gk0myPxLubBOPsFz2GT/AMs/agD0aivNv+Gg/hj/ANDN/wCSFz/8bpz/ALQHwyR2R/EuGU4I+wXPX/v3QB6PRXm5/aB+GS4z4m+8Mj/QLn/43Sn9oD4ZBA//AAkvykkA/YLnqP8Atn70Bsej0V51H8ffhpKkjp4kysS73P2C54GQv/PP1Ipg/aB+GLMAPE3J/wCnC5/+N0bgekUV5t/w0H8Mf+hm/wDJC5/+N06T4/8AwzilaOTxLhkJVh9gueCP+2dAHo9Febn9oH4YgDPibqMj/QLn/wCN07/hf3w0EIl/4SX5CxUH7Bc9QM/88/egD0aivOF/aA+GTBiPEvCjJ/0C54Gcf88/ekX9oH4ZOwVfE2STgf6Bc/8AxugFroj0iivNv+Gg/hj/ANDN/wCSFz/8bpz/ALQHwyR2RvEuGU4I+wXPX/v3QB6PRXnUnx9+GkSRu/iTCyrvQ/YLnkZK/wDPP1Bpn/DQPwx25/4SbjOP+PC5/wDjdG+wHpFFecL8f/hm6uy+JchBub/QLngZA/55+pFIP2gfhizADxNyTgf6Bc//ABugFroj0iivNv8AhoP4Y/8AQzf+SFz/APG6fL8fvhpBM0UviXa6nBH2C5OP/IdAHo1FecN+0B8MlCk+JeGGR/oFzyM4/wCeftSf8NA/DLaW/wCEm4Bxn7Bc/wDxugFrsekUV5wP2gPhkwYjxLwoyf8AQLngZx/zz96QftA/DFmAHibknA/0C5/+N0BvsekUV5t/w0H8Mf8AoZv/ACQuf/jdPl+P3w0gmaKXxLtdTgj7BcnH/kOgD0aivOG/aA+GShSfEvDDI/0C55Gcf88/ak/4aB+GW0t/wk3AOM/YLn/43QC12PSKK84H7QHwyYMR4l4UZP8AoFzwM4/55+9Oh+Pvw0uJ44YfEm6SRgij7BcjJJwP+WdD01YHotFebf8ADQfwx/6Gb/yQuf8A43Sn9oH4YqcHxNz/ANeFz/8AG6APSKK84b4//DNFRm8S4Djcv+gXPIyR/wA8/UGk/wCGgfhjtJ/4SbgHH/Hhc/8AxugFrsekUV5zH8fvhpKHKeJchF3N/oFzwM4/55+9In7QHwyd1RPEuWY4A+wXPX/v3QHmej0V5t/w0H8Mf+hm/wDJC5/+N0rftA/DJGw3ibB6/wDHhc//ABugD0iivOD+0B8MlCk+JeGGR/oFzyM4/wCeftTh8ffho0DzDxJ+7RlRj9gueCQSP+Wf+yfyo23A9Forh9A+MngPxPrlvo+h679qv7kkRQ/Y503YUseWQAcA9TXcU7MAooopAFFFFAH/2Q==)

**ACCURACY OF THE MODEL**

The test data is fed into the model to see its performance of the model. The model performs well on the test data as well with an accuracy of 94.21%.

After the model has been built and the data has passed the training phase, the testing phase begins. At this point, we attempted to classify test images using two different models with different filters. The number of training epochs to be used is a common issue encountered when training neural networks. Excessive epochs can result in overfitting of the training dataset, whereas insufficient epochs can result in an underfit model. So we experiment with different numbers of epochs to see when the model's performance stops improving. The number of epochs is determined by the best model performance.

Chart

Description automatically generated

Fig 12:Model Accuracy

When we train the model, the accuracy and loss in the model for validation data may vary depending on the case. Normally, the loss should decrease, and the exactness should increase with each increasing epoch. However, with validation loss (Keras validation loss) and validation accuracy, numerous cases are possible, as shown below:

1)Validation loss increases, while validation accuracy decreases. This implies that the model is cramming values rather than learning.

2)Validation loss begins to rise, and validation accuracy follows suit. In situations where softmax is used in the output layer, this could be an example of overfitting or diverse probability values.

3)Validation loss begins to decrease, and validation accuracy begins to improve. This is also acceptable because it implies that the model created is learning and is doing well. We obtained the following results after testing our model: we plotted the graph of accuracy and loss concerning epochs.

**STEPS OF TRAINING**

Graphical user interface, text

Description automatically generated

Fig 13: Steps of training

Predicting the action

Just as we did when we created the dataset, we generate a bounding box for detecting the ROI and calculating the cumulative average. This is done to identify a foreground entity. Now we search for the maximum contour, and if one is found, it indicates that a hand has been identified, so the ROI threshold is used as a test image. We load the previously saved model with Keras. models.load\_ Model and then feed the threshold image of the ROI containing the hand to the model for prediction as an input.

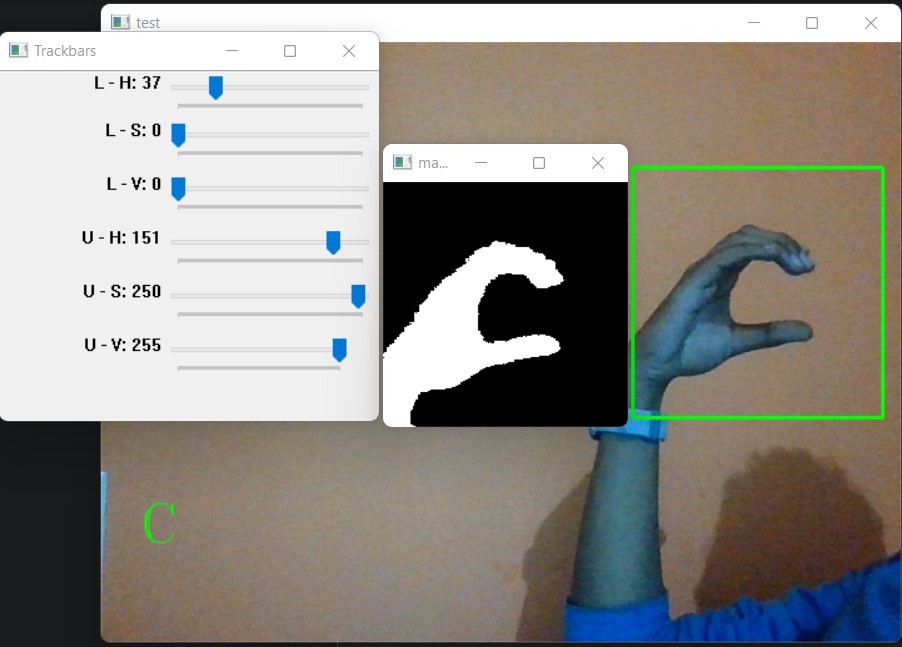


Fig 14: Testing the Model(C)

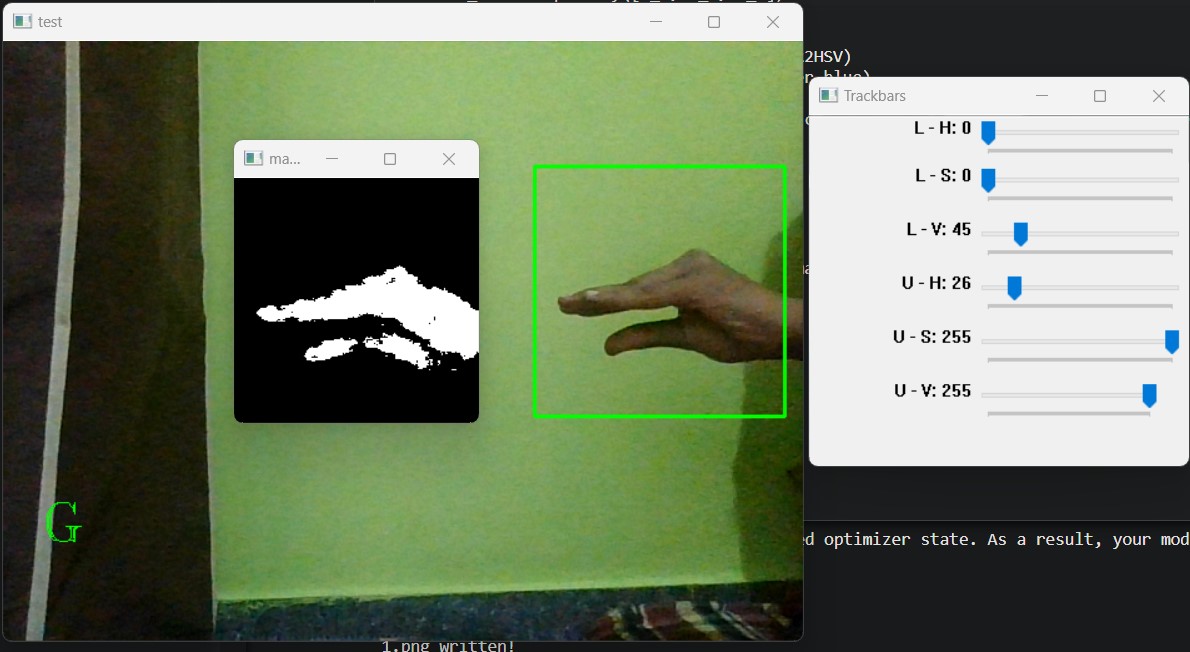


Fig 15: Testing the Model(G)

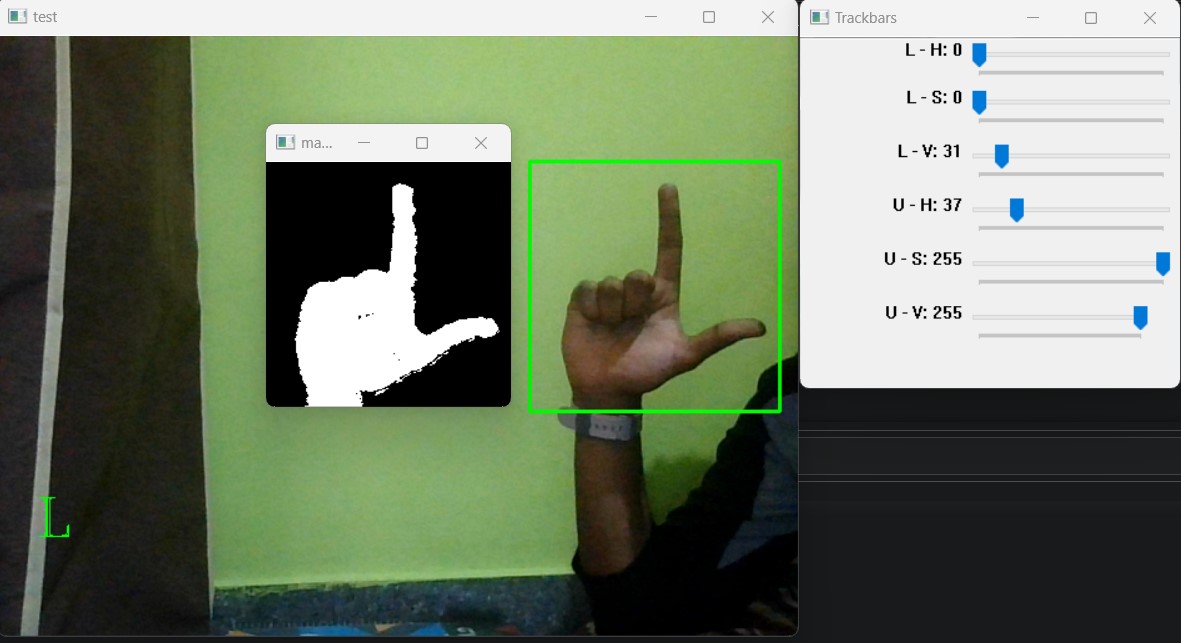


Fig 16: Testing the Model(L)

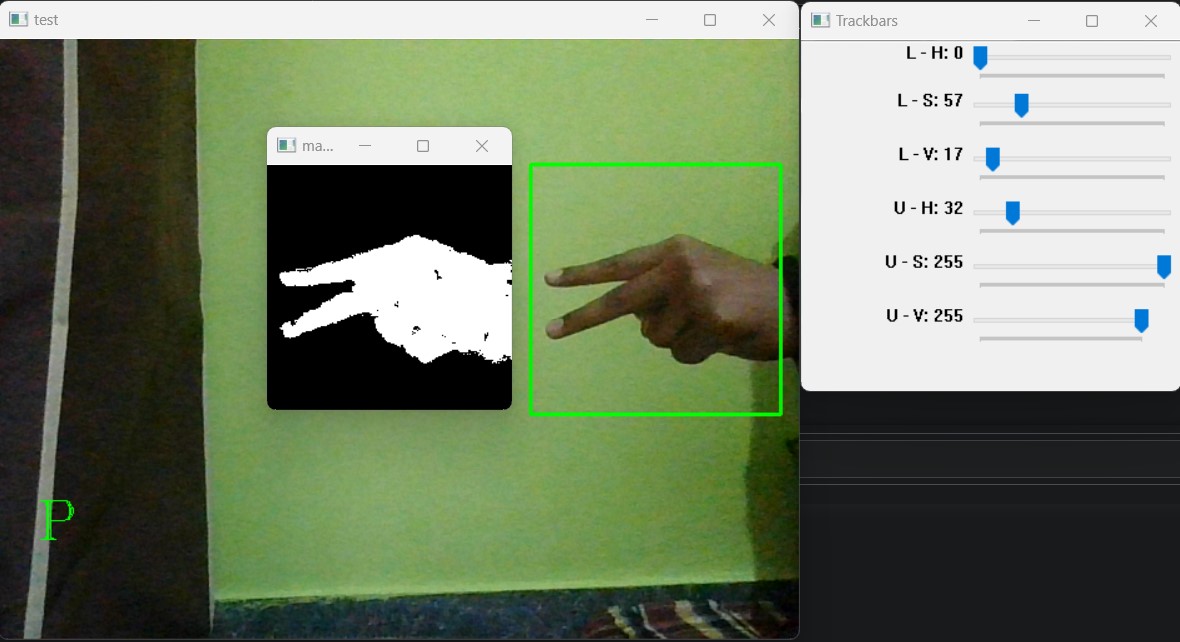


Fig 16: Testing the Model(P)

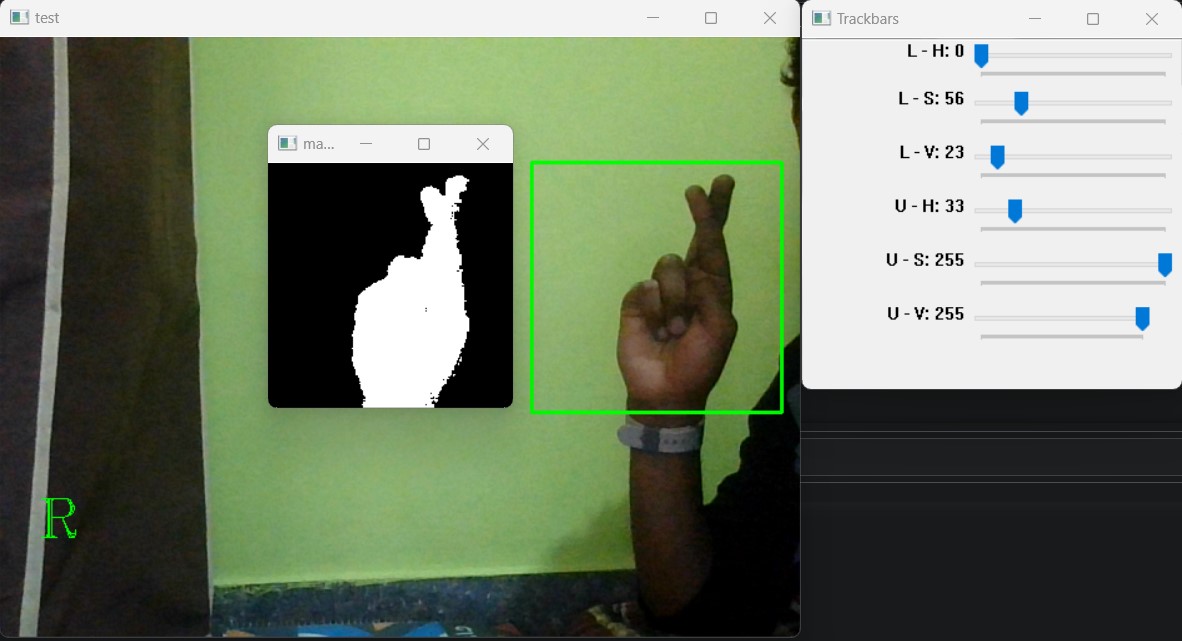


Fig 17: Testing the Model(R)

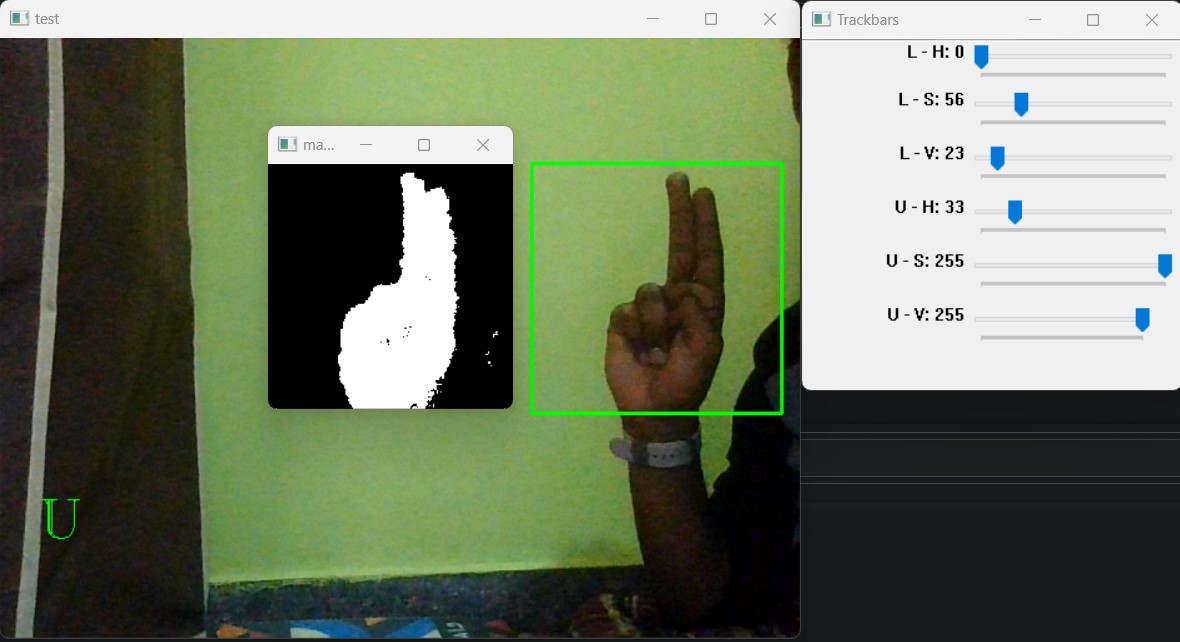


Fig 18: Testing the Model(U)

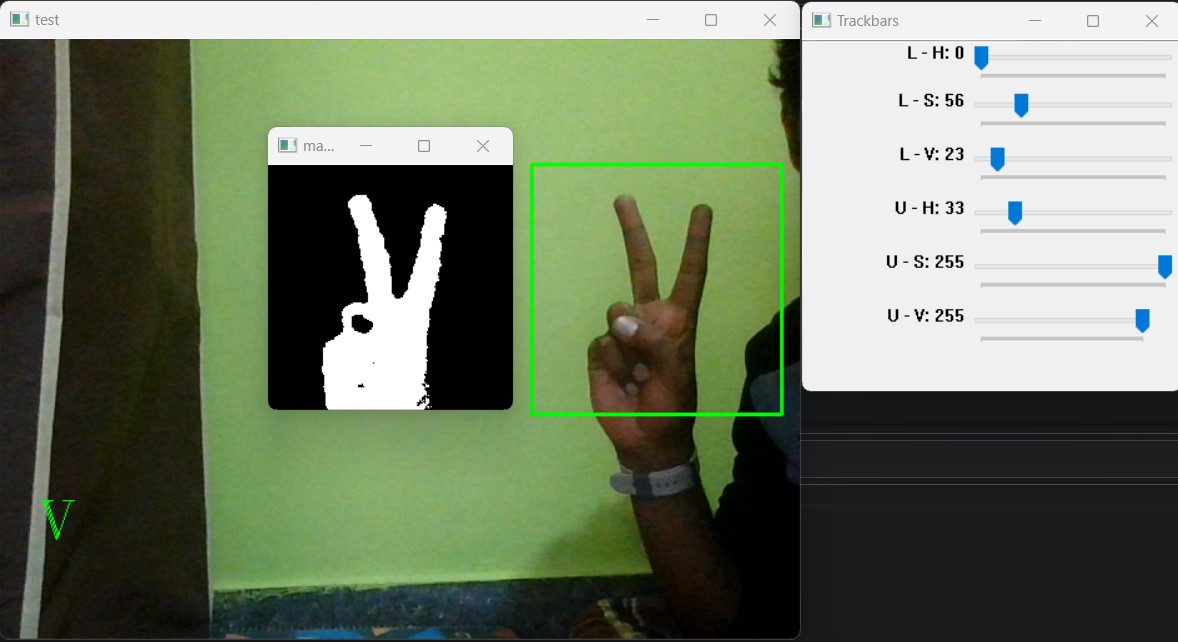


Fig 19: Testing the Model(V)

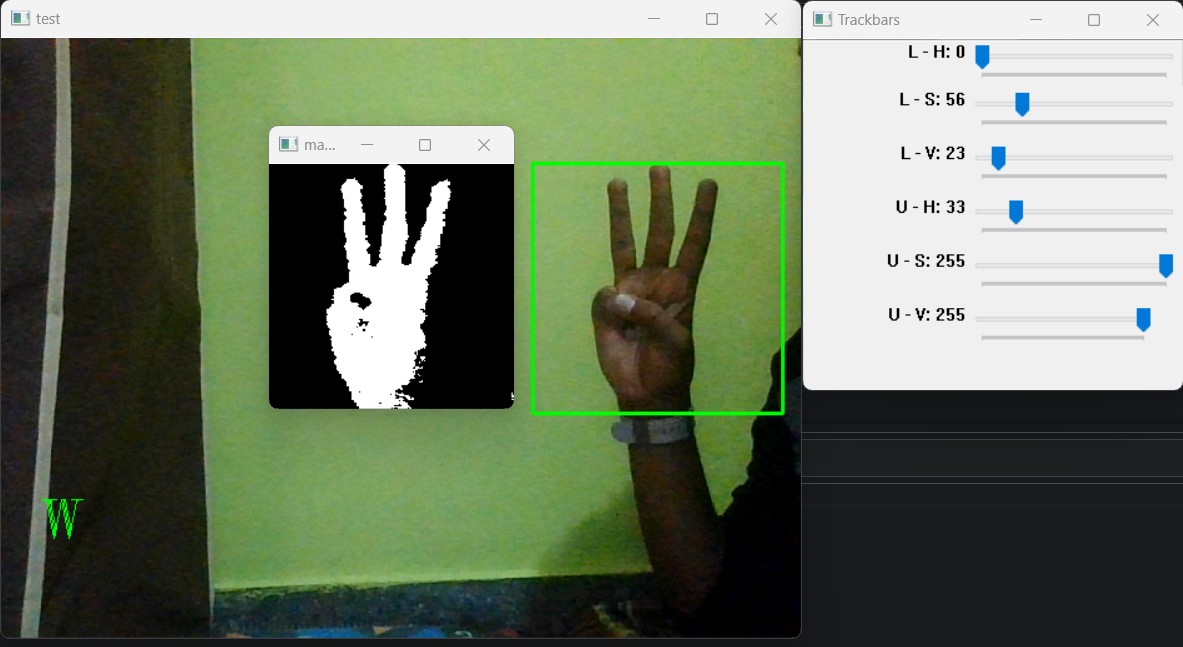


Fig 20: Testing the Model(W)

The model is given images from the live camera feed and the predicted alphabets are displayed as output in the bottom-left section of the test window. The model is tested with the alphabet C, G, L, P, R, U, V and W as shown in the figures above. It detects sign language with high accuracy as proposed.

From our results we can conclude that our trained model can recognize the alphabets with the following accuracy values:

Accuracy = 96.41 % (during the training of the model)

Accuracy = 94.21 % (while using the validation Set)

**COMPARISION OF RESULTS**

|  |  |  |
| --- | --- | --- |
| Paper | Method | Accuracy  (in %) |
| Nagarajan [3] | Support Vector Machine | 93.75 |
| Agarwal Rajat [2] | Multiclass Support Vector Machine | 94 |
| Rokade [5] | Support Vector Machine | 92.12 |
| Izzah [1] | Support Vector Machine | 81.39 |
| Proposed method | Convolutional neural network | 96.41 (training)  94.21 (validation) |

Table I

Our model can identify all 26 alphabets with considerable accuracy. The accuracy of the training set was found to be 96.41% with an error value of 0.14. The model performs well on the test data as well with an accuracy of 94.21%. By using the convolution neural network of deep learning, we achieved higher accuracy and minimal loss.

**Chapter 5**

**CONCLUSION**

Sign language recognition using a convolutional neural network was introduced in our work. The machine learning model is created using the Keras API with TensorFlow as the backend. The model was trained using a labelled dataset with 1750 images. Our model can identify the 26 alphabets and we have achieved an accuracy of 96.41% during the training of the model and an accuracy of 94.21% while using the validation set. So, our model can recognize the gestures with considerable accuracy and is feasible for hearing impaired people.

The Sign Language Recognition (SLR) system is a method for recognising a collection of formed signs and translating them into text or speech with the appropriate context. The significance of gesture recognition can be seen in the development of effective human-machine interactions. We

attempted to build a model using a Convolutional Neural Network in this project.

**FUTURE SCOPE**

The proposed method for the identification of sign language using a Convolutional neural network identifies the 26 alphabets of the English language. In future, we can improvise our model to recognize numeric characters. Further can be improved by recognizing some of the words which are commonly used. For more convenient use of our model, we can develop a mobile application.

**REFERENCES**

[1] Izzah, Abidatul, and Nanik Suciati. "Translation of sign language using generic Fourier descriptor and nearest neighbour." *International Journal on Cybernetics and Informatics* 3, no. 1 (2014): 31-41.

[2] Agarwal, Rajat, Balasubramanian Raman, and Ankush Mittal. "Hand gesture recognition using discrete wavelet transform and support vector machine." In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 489-493. IEEE, 2015.

[3] Nagarajan, Sathish, and T. S. Subashini. "Static hand gesture recognition for sign language alphabets using edge-oriented histogram and multi-class SVM." *International Journal of Computer Applications* 82, no. 4 (2013).

[4] Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image-based hand gesture recognition techniques." In *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, pp. 1-5. IEEE, 2016.

[5] Rokade, Yogeshwar I., and Prashant M. Jadav. "Indian sign language recognition system." *International Journal of Engineering and Technology* 9, no. 3 (2017): 189-196.

[6] Ma, Daniel Nareshkumar, and Vijayalakshmi Sb. "Recognition and Analysis of Indian Sign Language Using Improved K-means Algorithm." *European Journal of Molecular & Clinical Medicine* 7, no. 11: 2020.

[7] Rosero-Montalvo, Paul D., Pamela Godoy-Trujillo, Edison Flores-Bosmediano, Jorge Carrascal-García, Santiago Otero-Potosi, Henry Benitez-Pereira, and Diego H. Peluffo-Ordóñez. "Sign language recognition based on intelligent glove using machine learning techniques." In *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pp. 1-5. IEEE, 2018.

[8] Wadhawan, Ankita, and Parteek Kumar. "Deep learning-based sign language recognition system for static signs." *Neural Computing and Applications* 32, no. 12 (2020): 7957-7968.

[9] Sawant, Shreyashi Narayan, and M. S. Kumbhar. "Real-time sign language recognition using PCA." In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1412-1415. IEEE, 2014.

[10] Pratama, Yohanssen, Ester Marbun, Yonatan Parapat, and Anastasya Manullang. "Deep convolutional neural network for hand sign language recognition using model E." *Bulletin of Electrical Engineering and Informatics* 9, no. 5 (2020): 1873-1881.

[11] Kuznetsova, Alina, Laura Leal-Taixé, and Bodo Rosenhahn. "Real-time sign language recognition using a consumer depth camera." In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 83-90. 2013.

[12] Ahuja, Mandeep Kaur, and Amardeep Singh. "Static vision-based Hand Gesture recognition using principal component analysis." In *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, pp. 402-406. IEEE, 2015.

[13] Campos, Necati Cihan, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. "Neural sign language translation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7784-7793. 2018.

[14] Aryanie, Dewinta, and Yaya Heryadi. "American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier." In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pp. 533-536. IEEE, 2015.

[15] Campos, Necati Cihan, Simon Hadfield, Oscar Koller, and Richard Bowden. "Subunits: End-to-end hand shape and continuous sign language recognition." In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3075-3084. IEEE, 2017.

[16] Bhagat, Neel Kamal, Y. Vishnusai, and G. N. Rathna. "Indian Sign Language Gesture Recognition using Image Processing and Deep Learning." In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-8. IEEE, 2019.

# APPENDIX

**Machine Learning** is the study of computer algorithms that can improve themselves automatically based on experience and data. It is regarded as a component of artificial intelligence.

**Deep Learning** is part of a broader family of machine learning methods based on artificial neural networks with representation learning.

**A *Convolutional Neural Network* (CNN)** is a type of artificial neural network (ANN) that is commonly used in deep learning to analyse visual imagery.

**CONFERENCE/ ARTICLE PAPERS PUBLISHED**

**1. ICRTST-2022**

**Graphical user interface, text

Description automatically generated**