# **INFO 5505 Applied Machine Learning for Data Scientists**

## Assignment 6

Name: Ravi Varma Kumar Bevara

### **PART I: Model and Dataset Description**

Neural Networks

A neural network is a set of algorithms that attempts to recognize underlying relationships in a piece of data using a method that is like how the human brain works. A neural network (NN), just like a regression or an SVM model, is a mathematical function...

**𝑦=𝑓𝑁(𝑥)**

The function has a particular form as it’s a nested function. You have probably already heard of neural network layers. So, for a 3-layer neural network that returns a scalar, function looks like this...

**𝑦=𝑓3(𝑓2(𝑓1(𝑥)))**

In the above equation, f1 and f2 are vector functions with the following

**𝑓𝑙(𝑧)=𝑔𝑙(𝑊𝑙(𝑧)+𝑏𝑙)**

where 'l' is called the layer index and can span from 1 to any number of layers. The function 'g' is called an activation function. The parameters 'W' (matrix) and b (vector) for each layer are learned using the familiar gradient descent by optimizing, depending on the task, a particular cost function (such as MSE).

* Information is saved on the entire network, not in a database, as it is in traditional programming. The network continues to function despite the loss of a few pieces of information in one location.
* The capacity to work with incomplete information: After ANN training, the data may provide output even if the information is incomplete. The importance of the missing data determines the absence of performance.
* Because of their structure, artificial neural networks demand computers with parallel processing power. As a result, the equipment's manifestation is contingent.
* The most serious issue with ANN is the network's unexplained behavior. When ANN provides a probing answer, it does not explain why or how it was chosen. This decreases network trust.

Convolutional Neural Network

A convolutional neural network (CNN) is a special kind of FFNN that significantly reduces the number of parameters in a deep neural network with many units without losing too much in the quality of the model. CNNs have found applications in image and text processing where they beat many previously established benchmarks.

![Diagram

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDsRXhpZgAATU0AKgAAAAgABAE7AAIAAAALAAAISodpAAQAAAABAAAIVpydAAEAAAAWAAAQzuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFNhaWx5IFNoYWgAAAAFkAMAAgAAABQAABCkkAQAAgAAABQAABC4kpEAAgAAAAMxOAAAkpIAAgAAAAMxOAAA6hwABwAACAwAAAiYAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMjowMToxMyAxNTo1Njo0NQAyMDIyOjAxOjEzIDE1OjU2OjQ1AAAAUwBhAGkAbAB5ACAAUwBoAGEAaAAAAP/hCx1odHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTAxLTEzVDE1OjU2OjQ1LjE4MjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5TYWlseSBTaGFoPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAWECxAMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APe7vVNQXWZbDTrG2n8m3jmd57povvs4AAEbZ/1Z9OtN+2eIP+gXpv8A4MpP/jFEX/I46h/14Wv/AKMuK0qU58rtYIxur3M37Z4g/wCgXpv/AIMpP/jFH2zxB/0C9N/8GUn/AMYrSoqPaeQ+TzM37Z4g/wCgXpv/AIMpP/jFH2zxB/0C9N/8GUn/AMYrSoo9p5ByeZm/bPEH/QL03/wZSf8Axij7Z4g/6Bem/wDgyk/+MVpUUe08g5PMzftniD/oF6b/AODKT/4xR9s8Qf8AQL03/wAGUn/xitKij2nkHJ5mb9s8Qf8AQL03/wAGUn/xij7Z4g/6Bem/+DKT/wCMVpUUe08g5PMzftniD/oF6b/4MpP/AIxR9s8Qf9AvTf8AwZSf/GK0qKPaeQcnmZv2zxB/0C9N/wDBlJ/8Yo+2eIP+gXpv/gyk/wDjFaVFHtPIOTzM37Z4g/6Bem/+DKT/AOMUfbPEH/QL03/wZSf/ABitKij2nkHJ5mb9s8Qf9AvTf/BlJ/8AGKPtniD/AKBem/8Agyk/+MVpUUe08g5PMzftniD/AKBem/8Agyk/+MUfbPEH/QL03/wZSf8AxitKij2nkHJ5mb9s8Qf9AvTf/BlJ/wDGKPtniD/oF6b/AODKT/4xWlQSFUliAAMkntR7R9g5PMzftniD/oF6b/4MpP8A4xR9s8Qf9AvTf/BlJ/8AGKxfCPjmLxf4i1+0sYCtjpRgjimdSGnZg5Zh/s4C7fUc9wAz4ga/r3hvSLnVdK/s6GxsbSSaeW9iaUyy5AiiRVdMZOcsc9VwDzV3lzctlcmyte5u/bPEH/QL03/wZSf/ABij7Z4g/wCgXpv/AIMpP/jFYXhbWfEc8NhdeLpdItrfUrWJ7dIY2hk+0PlvJ+eRtxCDPABPPpWzJ4s8Ow7PN1/S03q7JuvIxuCZDkc8hdrZ9MHPSk5NO1vzGku5J9s8Qf8AQL03/wAGUn/xij7Z4g/6Bem/+DKT/wCMVnav4lRk0lfDus6A0t/cptW9u+LmDJD+TsPzPnAHUeta39t6V/bH9k/2nZ/2ljd9j+0L52MZzszu6c9OlHPLt+Ycq7kX2zxB/wBAvTf/AAZSf/GKPtniD/oF6b/4MpP/AIxTofEOi3OpHTrfV7CW9DvGbZLlGkDJ95doOcjuO1YXw+8S6n4o0jUNS1M2f2VL6WCyltomjEsUZx5hDO3U56HtRzSte35hyq9rm39s8Qf9AvTf/BlJ/wDGKPtniD/oF6b/AODKT/4xXL6H4u1rxzdXk/hNLGw0S0me3TUL+B7hr1xt5jjV02oOfmLHOV4BDAb0OvrpOi28/ja80zR7qZ3Xa90qR8MdoVnI3HbtJ/HgdKblJaWVwST6lr7Z4g/6Bem/+DKT/wCMUfbPEH/QL03/AMGUn/xinT+INGtb2WzudWsYbqGIzywSXKK8cYGS7KTkKBzk8Ypy65pLxwyJqdmyTwtPCwuEIkjX7zqc8qMjJHAzU877fmPlXcj+2eIP+gXpv/gyk/8AjFH2zxB/0C9N/wDBlJ/8You/Emh2Fjb3t9rOn21pdDME810iRzDGcqxOG454rB8QeKr/AE34gaJollJYi0ubea61B7iNt1vDGM794cAAnjlSBjOaalJ9PzE0l1N77Z4g/wCgXpv/AIMpP/jFH2zxB/0C9N/8GUn/AMYpI/EuhS29vcRa1pzw3LMsEi3aFZSvLBTnDEY5x0rnbr4h6bqlrNH4S1zQvtsd6tqjandBIrg/KXEe07mOGwCBgnpkUKUn0/MLJdTo/tniD/oF6b/4MpP/AIxR9s8Qf9AvTf8AwZSf/GKkl1vSoNWj0ufU7OPUJRujtHuEEzjnkJnJ6Ht2NQDxV4eMcrjXtM2QzCCRvtkeEkOcIeeGODx14NLnl2/MfKu4/wC2eIP+gXpv/gyk/wDjFH2zxB/0C9N/8GUn/wAYp2o+IdF0eVYtX1ewsJGTeqXVykRK5xuwxHGeM1hr4k1Sf4sv4ctfsj6bb6cLu6cwt5sbs21U378ZP3vu9KalJ9PzFypdTa+2eIP+gXpv/gyk/wDjFH2zxB/0C9N/8GUn/wAYrlfiF411XwXf6feW8Ntc6KHj/tQGBzNbRs+3zA4cDnOAu08jrzx0Md7qk/ix44LrTzo0Nqk0n7hjKzPuwBIH24+Xcfl6EeuQ+Z2vZfiKyva5Z+2eIP8AoF6b/wCDKT/4xR9s8Qf9AvTf/BlJ/wDGKhHi/wANHT2vx4h0o2ayCE3AvY/LDkZC7t2N2Ocdalu/Euhae0y3+tadbNBs80TXaIY94yu7J4yOmetTzy/l/MfKu4v2zxB/0C9N/wDBlJ/8Yo+2eIP+gXpv/gyk/wDjFLJ4i0SLT5b6XWLBLOGXyZLhrpBGj/3C2cBuenWuc8Z+MrjT9K0qXwhdabfXOpanHYRB1M8bEkhjlHXG3ByecdDjrTUpN2t+YNJdTovtniD/AKBem/8Agyk/+MUfbPEH/QL03/wZSf8AxiqvjC/1bTfDktz4eFtJqAdFhhuYmkWZmOAmFdSDkg5ycAHg1gaP4y1XxX4J0PUtAewttQvbkW13Hd27yJG6hjKAA6kYCMwBJyMDjOaak2r2X4g0k7XOp+2eIP8AoF6b/wCDKT/4xR9s8Qf9AvTf/BlJ/wDGKSfxLodndzWd7rWnQXdvH5k8Ml0iPGuAdzKTlRgg5PY1X0DxhofiW3SbSdQhlEskqQqZFDyiMgMwTO7AyOoHBB6EZnnla9vzDlXcs/bPEH/QL03/AMGUn/xij7Z4g/6Bem/+DKT/AOMVPa6vpt9LHHZaha3EksPnxpFOrl487d4APK54z0zVOPxb4cl0+a/i1/S3s4GCS3C3sZjjY9AzZwCfQ0c8u35j5V3JftniD/oF6b/4MpP/AIxR9s8Qf9AvTf8AwZSf/GK57xt43OleE/t/hO703UL972CzhiZvOR5JCMIdjgg7Tu+nbnNdLpWs6drFuz6XqdlqIiISWSzmWRVbHQ7ScfQmm5SSvb8xWV7XI/tniD/oF6b/AODKT/4xR9s8Qf8AQL03/wAGUn/xiue1L4i2Vkmragu3+xdDkMN7dY3PPPjHkQrkDcGZcsTgfdwcllSw1Lx3qugLrMVppFm1yiyW+kzJI8iowGC8+9RkZLbfL6ALkE5DvK12kKy7nRfbPEH/AEC9N/8ABlJ/8Yo+2eIP+gXpv/gyk/8AjFcm3jjU9O+KsXh/V47RtHvi8NldxQsjCdVV/LYl2DfK2MgDLdhinaxrnjKy0DVL/Tn0i7uI714dOtBYy7rhYywcMfO4b5XPHGIzx8wCv3tNFr6hp3Z1X2zxB/0C9N/8GUn/AMYo+2eIP+gXpv8A4MpP/jFc5pni3UPFPhnw5qfhuewhk1J9t0l1bvL5e1WMm0LIpGGUgZzncvTNb3hbWx4h8OwX52CXc8Mypwokjco+AeQNykjPYipcpLdDST6kv2zxB/0C9N/8GUn/AMYo+2eIP+gXpv8A4MpP/jFaVFT7TyHyeZm/bPEH/QL03/wZSf8Axij7Z4g/6Bem/wDgyk/+MVpUUe08g5PMzftniD/oF6b/AODKT/4xR9s8Qf8AQL03/wAGUn/xitKij2nkHJ5mb9s8Qf8AQL03/wAGUn/xij7Z4g/6Bem/+DKT/wCMVpUUe08g5PMzftniD/oF6b/4MpP/AIxR9s8Qf9AvTf8AwZSf/GK0qKPaeQcnmZv2zxB/0C9N/wDBlJ/8Yo+2eIP+gXpv/gyk/wDjFaVFHtPIOTzM37Z4g/6Bem/+DKT/AOMUfbPEH/QL03/wZSf/ABitKij2nkHJ5mb9s8Qf9AvTf/BlJ/8AGKPtniD/AKBem/8Agyk/+MVpUUe08g5PMzftniD/AKBem/8Agyk/+MUfbPEH/QL03/wZSf8AxitKij2nkHJ5mb9s8Qf9AvTf/BlJ/wDGKPtniD/oF6b/AODKT/4xWlRR7TyDk8zN+2eIP+gXpv8A4MpP/jFH2zxB/wBAvTf/AAZSf/GK0qKPaeQcnmZv2zxB/wBAvTf/AAZSf/GKPtniD/oF6b/4MpP/AIxWlRR7TyDk8yLSr7+09Gsr/wAvyvtVuk2zdnbuUHGe+M0VV8K/8ido3/XhB/6LWitpK0miYu6TGRf8jjqH/Xha/wDoy4rSrNi/5HHUP+vC1/8ARlxWlWFX4vu/IuGwUUUVmWFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFczrltqniK/k0I219pukGMtPqUT27faDxiJUYsdpyd25OQpXGDk9NXMeJfhx4U8YalHf8AiLSvtl1HCIUk+0Sx4QEsBhGA6sefeqg0nqTK7Whg/DTQ9e0jxN4pu9dsL2GPVLiOeC4u5bZnkCgghxC2A3PYYqT4q6brniOwsdD0rQrq9097yGbUZo54EDQK2WjUPIpLcA84HA564d/wo34d/wDQvf8Ak7cf/HKP+FG/Dv8A6F7/AMnbj/45W/PDn57/AIf8Ejlly8v6/wDAI/iHarq154bbQrW2vdZ0rW7Z3gEqiSCPYXZXIzsUqFPpwCAeKxb+wupfiClloekRaq/hu0aYhRFFEmo3bFxMys3CqF3fLuYZGB3re/4Ub8O/+he/8nbj/wCOUf8ACjfh3/0L3/k7cf8AxynGpTStd/d/wROMm/6/yMTT/AfiDQ9U8LaXY2qXOn6Fp07x3zzIIhfTFtzMhO8qoJK/KeSBxyRPovh/xBZfDiSyt/DAtPElvaXEialc3EDSyXcvDPE6sx3MM5ZymMIOR93U/wCFG/Dv/oXv/J24/wDjlH/Cjfh3/wBC9/5O3H/xyh1Yvd/h/wAEOR/0/wDgHL6Z4V8VabJdS6Z4V+xtZeHRpmlb7yDJkYlpZDtY/vDI2eoGNx3ZOD3fhDw/PZ/DK28P3do2mstmbYqZEdwWXDOdhKglixADHjGTkkDM/wCFG/Dv/oXv/J24/wDjlH/Cjfh3/wBC9/5O3H/xylKpCXX8P+CCjJf1/wAAi8A6d4l8H+EYfC7+H99xbSyrFqS3EX2WRWcsJGG/zf4iNuznAGQDkZXiDwd4p1nxJ4lQ2iOdYihsbXVpZE8myswu6ZVj3eZuZu2MEnO4Vtf8KN+Hf/Qvf+Ttx/8AHKP+FG/Dv/oXv/J24/8AjlHtIczlf8P+CHLK1v1/4BzXiX4f69f3XiG9s9Okna00y30nQreS6j/fRgjfKxzgMpy67sfMAcZCkaq/D7ULbW/CF/odjZ2D6fps1nfvO4c7CipGCAP3jjc7Y+7kEbsYJ0P+FG/Dv/oXv/J24/8AjlH/AAo34d/9C9/5O3H/AMcp+1ja1/w/4Icj7f19xzl/8PdefUtd0zT9PRbW+tbfTNP1KaVGhsLBUzIgTd5hdiP7uCTksKuP4e8Vaf4i8S3mlaK8zrpMel6LcS3kQAhROf4t28yFWwQq4DEtnAbX/wCFG/Dv/oXv/J24/wDjlH/Cjfh3/wBC9/5O3H/xyj2se/4f8EOR/wBP/gHK6T4U8Z6Xf6U9voAFvoehSw6dG13CNt5Ip3u2GPzMfT5cEfN1zb0fwh4jtYvAunS6FtsNJd7y9L3UR/0o52u+GJO0s7DbuzlRkZO3f/4Ub8O/+he/8nbj/wCOUf8ACjfh3/0L3/k7cf8Axym6sX/w3/B8xcj/AKf/AADnvDfgTxHcf2RH4gsmtWsNVfWdTu3njkk1G6BYRCPaxIQLjO/b1wB1NZ5+HPiO5h069v8ARzc3+pa6+parE95GFtow37uLIJ+RsKWKBmAUYBIArsf+FG/Dv/oXv/J24/8AjlH/AAo34d/9C9/5O3H/AMco9tG97/h/wQ5H/T/4By174R8WX2tXTXvhz7YLvxCLu5u3u4Ar2kX/AB7xL824Jx82V3AHgMSQOu8DaVrdp4t8Uanr2lG2k1K8LR3DXEbhoEAWFFCknpvJ3bcfLgHJ2xf8KN+Hf/Qvf+Ttx/8AHKP+FG/Dv/oXv/J24/8AjlKVSDVr/h/wRqMk7/1+Rpx6TL4nt/Elt4i0i4sYdQAtovPkifdAqEKRsdsEOXbnGNw965CfwR4rtvhjpGikyXs4v4zqqW0kQkktY/kRE8392w2JHlX4PQ5ya3P+FG/Dv/oXv/J24/8AjlH/AAo34d/9C9/5O3H/AMcpKcFs/wAP+CDi3/w//AMrXPCGqS6BpVjpegXEkV1q63+sC4vYWuJwmCDKxITLMBkJuCgfL6VmL4O8Zatpus2ms6WsVx4h1yF7+7FxERHZRkEBVDEkAKAAeSGII4yeo/4Ub8O/+he/8nbj/wCOUf8ACjfh3/0L3/k7cf8AxymqsUt/w/4Icj/r/hjI8S+C9T03xMs/hHw4t3YQ6fcS2sS3MaompS4UyyCRgT+7UYIz83pkmo9F8J+INP1rwTFJ4fI03QtPfObuLCXUhKu0gBzwoLDaG+ZgM9WG3/wo34d/9C9/5O3H/wAco/4Ub8O/+he/8nbj/wCOUe1ja1/w/wCCHI73/r8job839z4w0uD+yrhtNthJO16JItgm2FUBUvvIwz9F6le2SOZsPAuq6B4k8W32hSxJb31u1xpULEbIryRSHJHb5kXnptfA6Gpf+FG/Dv8A6F7/AMnbj/45R/wo34d/9C9/5O3H/wAcqFKCVk393/BG1J9Px/4BzFr4K1+08AX32XQr6TxK2mGzSW/vrfIeZh55hEZ24OXcySESE4GWzxrt8NJrvVNDmigGlWtlokllcCFl86YkBUjZlP8AdyWIJAJIBOd1aH/Cjfh3/wBC9/5O3H/xyj/hRvw7/wChe/8AJ24/+OVftY9/w/4IuR9v6+45d/hx4iX4Z6XY3FlBJqEIhs7qCynUTPYmXzJo/MYqu4nbkZ2/IcE7jVzUvA3iK/0vXZks47e68TXVrZtaxSIF02wj4y2CAzbRghd3X5c81uf8KN+Hf/Qvf+Ttx/8AHKP+FG/Dv/oXv/J24/8AjlP2se/4f8EXI/6f/AMnxZ8NLm61PT7HwpavpdhAJ9QnvoZkDPe+SI4ThjkNlclgO5IIYk12ngmz1Wz8F2VtrVna6depEA0FrgqjY5Y4+UszbmOOPmxk9Tg/8KN+Hf8A0L3/AJO3H/xyj/hRvw7/AOhe/wDJ24/+OVEpwlHlbf3f8EpRad0v6+45pPh9e6/+z7ZaNYtGuqRztdypO3yzTrK+9GI78kfVRnHUehQazr08KOPC9zYiGPfNDPPbu8xxxHDsl29erOVAAGAc/Lhf8KN+Hf8A0L3/AJO3H/xyj/hRvw7/AOhe/wDJ24/+OU5ThLd/h/wQUZL+v+AQ+IPB2peJvhdBGtq2n+Jbaf8AtGASvGTHd+YXYBlJXa2WC88ArnGMDetLO+g8Q6VYvplw2nabaNi/MkWyS4Kgbtu/fnHmDO3kue3Jx/8AhRvw7/6F7/yduP8A45R/wo34d/8AQvf+Ttx/8cpc0GrXf3f8EOWXb+vuKem6FJ8NrzxPqvM2nyyl9GsIhuZpZgpeJVHJJdI1A9AT610vw88O3PhbwPY6bqMnm33zzXTZz+9kYuwz3wTjPtWN/wAKN+Hf/Qvf+Ttx/wDHK6jw14U0Xwfpslh4ds/sdrJMZnj815MuQFJy5J6KOPalOcWtH+H/AARxi0zXooorA0CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAoeFf+RO0b/rwg/9FrRR4V/5E7Rv+vCD/wBFrRXXP4mYR+FDIv8AkcdQ/wCvC1/9GXFaVZsX/I46h/14Wv8A6MuK0qwq/F935GkNgooorMsKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACio7i5gtITLdzRwRjq8jhVH4mmWt9aX8ZksbqG5QHBaGQOB+Ip2driur2J6KKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCh4V/5E7Rv+vCD/0WtFHhX/kTtG/68IP/AEWtFdc/iZhH4UMi/wCRx1D/AK8LX/0ZcVpVmxf8jjqH/Xha/wDoy4rSrCr8X3fkaQ2CiiisywooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqO4uYbSEy3MixxjqzH9KqS6kZJmt9NjFzMpw7k4jiP+03r7DJ+nWnW+mhZhc3khurkdHYYWP8A3F/h+vX1NXypayM+a+kTyPxBrMuv6vLcyuzQK5W2jJ+VEHAIHqcZJ9/aqlhqFzpF8l9YOUljOSAcCRe6t6g//Xp+p6bJo2qz6dMu0wsfL/2oz91h+H65FV44ZbmZLe2RpJpW2RooyST/AJzX1kI0vYpL4bHzc5VPatv4rnuNhew6jYQ3dswaOVQwIPT2/CrFc/Ywx6Lptpc2B82w8iNZ1UdAFAEoHrgfMPTnqOd9WV0DIwZWGQQcgivlKkVF6bH0kJNrXcWiiiszQKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCh4V/5E7Rv+vCD/0WtFHhX/kTtG/68IP/AEWtFdc/iZhH4UMi/wCRx1D/AK8LX/0ZcVpVmxf8jjqH/Xha/wDoy4rSrCr8X3fkaQ2CiiisywooooAKKKKACiiigAooooAKKja4hS4jt3mjWaVWaOMsAzhcbiB1IGRn0yKkoAKKKKACiiigAooooAKKKKACio5bmCCSGOaaON7hzHCrsAZG2ltqjudqscDspPapKACiiigAooooAKKKKACiiigAoqMXEJumthNGZ1QSNFuG4KSQGx1wSCM+xqSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiisXW/GPh7w5fWlnrmr2tlc3jBYIpXwz5O0HHYZ4yePegRtUVm/8JJof/QZ0/wD8Ck/xqW31vSry4WC01OznmbO2OK4RmOBk4APpRZgXaKCQqkscAckntWcb+a+OzSVUx9DdyD5B/uj+P+Xv2qoxchOSRau72CyRTO3zOcIijczn0AHJqp9nu9R5vS1rbHpbxt87j/bYdPov5npVi00+K1dpctNcOMPPKcu3t7D2GBVqq5lH4fvJ5XL4hsUUcESxQIscajCqowAPpTqKKzLKOqaLp2swrHqdqk4Q5UnIZfowwR+BrMGg6b4bEd9pdt5PlyDz2Ls5aM8HJYk4GQ34VqanrGn6PCsmpXSQKxwoOSzfQDk/hUGn65pGvxyw2N1HcZUh4iCrbTwflYA4963g6ijfXl/Aykqbl05vxHwf8S/U2tjxb3RMkPor9XT8fvD/AIFUcf8AxJrpYW/48J2xEf8Ang5/g/3T29Dx6UlvA17pDWcshW5tH8sS91deUf8AEbT+JFWraSPVNNZLqJcnMU8J52sOCP8AA+mDTem/z/R/1+olrt8v1X9foXKKzrKaS0uRp145dsE28zdZUHY/7Q7+o59caNYyjys1jK6CiiipKCiiigAooooAK4C++JV5Dp+ta1p2gLeaDol09vd3LX3lzyeWwEzwxeWVcLzjdIu7aenFd/XAXPw1vH0/WdEsvEC23h/Wrp7i5tjY77iMSMGlSObeFVWOfvRsQGOD0NNW6iN6bx34bt47x5dSwtlLbwzkQSHY84UxDhedwdeRkDPOKxtO+K+izHXn1iO40q20fUfsJuJreYpJkoFYnywFJZ8bckgfMeDmquufCqTUtQ1FtP1pLGw1GexuJrZrIysHtQqqFfzBhSqAEbScjOeoNHxf8OtZfRfEEPh+6S6/tnWbbUlg8lUkt3V4vMbzGfaygR7gu0Htk09A1OxtPHfh288gQ30iyT339nJFNayxSLcbC/lsjKGQ7QTlgB71kXfxT0iPxFoWnafBc38GrXNzbPcw20x8h4AcgKIyX+ZSDjG0fMfl5qq/wyu3lbUm1yH+3W1lNWNz9gP2fckRiEfk+bu27D18zOec9qNL+F9xpJ0Ke311Jb3R9Su73zprLKzLchg6FFkGCA3DZxkfd7UaBqehUUUVIwooooAoeFf+RO0b/rwg/wDRa0UeFf8AkTtG/wCvCD/0WtFdc/iZhH4UMi/5HHUP+vC1/wDRlxWlWbF/yOOof9eFr/6MuK0qwq/F935GkNgooorMsKKKKACiiigDzhj4XbxtrC+Nza/2yL6I6St4374QeXH5ZtB97mQSZ8v5t+c9qXTfE9/c+LLeAeIPOv5NUuLa78PeTF/olsnmBZeF81eFjbzGYo3mcD5lx6NRTuI8w0fVPFt3pOmGLXXu7/WvDk+oQrLbQqkFxEbfYF2oDtbzyG3FuRkbelPfxrd6rpUGtrrDaF4f1LUEt4754Yw1pEsDFmJkUqjPODHlwVAUAAMwx6ZRRcDyqXWdXurK21C1u2vr2HTda/s69WBQboIYxDKFA2sWwpyBtbqBggUzxD4oTxTomuJpevh9OtNJsbt57IROqOZpfNJYqw4VFyP4dvbmvWKKLgea32uvY3d9NZa1EbOSLTEm14QwMywu0wacuqBD0ABYFF3E4xnK2et61qY0+xsPEkk1nc65NaQa1FbwM95bLZvLuHyeUSsoaPeq4PldDzn0mii4GL4QvrrUfC9vNqE32i5SSaCSYqFMpjlePeQAACQmTgAZPAFbVFFIAooooGeLeDdI1TQfAkXirS9O0vTDHoJQCxV5pL52KMLidAiZaNUYhQXJ8xhuHfa/4Su7ig1Iab4t/tbRo5rJH8QeXbyfYvMdlmXfGgiO1RG2WUhPNy+VwK9Pop3EeaWGpz6rqfh/zNT/ALbsYfE00VlqhSMfa4v7MuCxzGAj7ZGkTcoAOzHUEnLttd16PwNHfafqv2CPSvBdlrAtbazgWOaYxzMykbPljIjA2ptxxtK859foouB5jN4p1O6+LUmhL4h+wJb6lGn2Tz7FEmt/s6SFfLkBuGkaRsZUBNnQgjNenVhv4N0WTWjqjwXBna4W5MX22f7OZVAAkMG/ytw2g525yAevNblDAKKKKQzzbw2/hr+1pW1kWr+OE1G5PlzOPt3338tU/jEPklMY+TYcn+Ks238aaq2h3d3Y+J/7UvP7Bu72+tfs0Q/si5SMMiYVQ0eGLLsmLsdn+y2fW6KdxHnl3q/iDRH1gSatNqS6ZaWGrSGS2iDGF5ZhcRKEQfKEgJXq2eCxqvc+KLx4tNudY8Uf8I3pWqfa7u3v/LgX5A6C3h3yoyDdETIQQWY5CnCnPpdFFwPKZvEGr29u2uXL/Zb4aLpkt1IYNgSM3j+axR87B5ZckHlRnoRRr3iCPXo7q+j8Tmz0fTvEVqkV/beR5UcbW8RLeZIjKV8x2wx4+bHIwK9WoouB5rda69jfXkUutRaDps2syxz6ukMCEYtYmQF3QxguSfncEkKFGMrhLXV/Eesw6Xbpr81iJNN1C4+1wWkO+7WG4jjt5sSIyqHjYOQAAd/GOMel0UXAz9Av5NU8NaZqE4VZbu0incL0DMgY4/OtCiikAUUUUDPN7vX9ff4iahY6Ze6nMbXU7OGOwTT1az+zPFE87ST+VlGCtIwHmA5CgKc4OvF46uRNa3V7pCQaLfSzw212l3vm3RJI+Xh2AKrLC5BDsfu5AydvUW2m2tneXl1bRbJr6RZbhtxO9lRUBwTgfKqjjHSs208HaHZaqdQt7R/O3SMsb3ErwxNISZGjhZjHGzZbJVQTub+8cvQRyniHxlq7+DdZW703+yJbvw5eajptxa35lkURxDIfCL5cg82MjaXGd2G4GdW68WX1nqt9Z2Vk2oXDa2mmQR3F0sUaZsUud25YyQvJBB3t1IJ4QX7fwF4dtbK8tI7KV7e8s2sZI5ryaUR27DBii3OfKQj+GPaOF/ujGifD+mG/N4bb/SGvRfl/Mb/XiAQB8Zx/qgFx074zzRoBxur/ABTbS9JtJ202yjvJkvt8F5qXkxmS0m8l4Yn8tjLIz/cXau4AnjpXe2Us81hby3cKwXDxK0sSOXEbEcqGIBIB4yQM+g6Vx/iL4ePqN1azaBqp0jyRdF1Bufma4lEzsGhuIm+/uO1iy8jCjHPX2Nu9pp9vbS3Ely8MSxtPLjfKQMFmx3PU0OwE9FFFIYUUUjusaFpGCqOpY4AoAWisjxN4lsvCeitqepR3UsIdYwlrA0sjM3AAUVjeFvGms+J9XZX8F6ppGk+UzLfakyRSM+RhfJ+9gjJznt70WEdhRXJeKvB+r+JtUjMPjHVNG0sQhJLPTVSOR3ycv52CwyCBjHbPetXwv4XsfCWkHT9MkupY2lMzyXdw00juQASWb6DgYFAHJ3nxn0R7yax8MaXrPia8hcxvHpti5SNgcEM7YA+vNdhrsOsX3h2VPDl5FpupSBDFPcw+YsXzDdle525H1rTRFjXbGqqvXCjAqO5u7eygae8uIreJeskrhVH4mmBx/hvwJq2ma5FrPiLxtrGuXkasBAdsFp8wwT5K5GeeDmtrxTp9lPoOo3k1nBJdQ2M6xzvEpdAUPAbGRRpXjTw1rury6Zomt2WoXkURleK1mEm1AQCcrkdWA696xvGXiDWku5PDujeEdR1QXltte/V0it4Q+VPzMeSvUrx29aNbgclV7QpWh8TadIkTzMJHxGmMsTE478VJ/wAI1r3/AECJv+/0P/xdaGgaBrFt4ksbm606SCCF3Z3aWM4zG6jhWJ6kVtzIizOuGny3rCTVmVlzlbWM/u1/3v75+vHt3rRAAAAGAOgoorKUnIpRSCikd1jQvIwRVGSzHAFUP7XSbjTYJL0/34xiP/vs8H8M0KLlsDkluaFQXV9bWSg3U6RlvuqT8zfQdT+FVvsuoXX/AB93a2yH/llajn8XPP5AVPa6da2bFreFVdvvSN8zt9WPJ/E07RW7+4m8nsjxfVNSk1nVrjUJm3GVj5f+zGPugfh+pJqvDPNaXEd1auY7iFt8bjsR/Q9CO4q/r+jy6DrEtrKhWFnLW0mPldDyAD6jOCPb3qpY2c2p6hDY2ih5pmCgZ+6O5PsBk/hX1kHT9imvhsfNTVT2rT+K567Z3YkubHUEG2HVLdMj0fbvX81LD8BU9z/xL9SW8HEFwRFcf7LdEf8A9lP1X0rIiiv/AOwVtrRVuPsoVFibCS28iYKj+6w4H93IPfNa9pqNhrdm0QYHzI8SW8nyuARyCDz+NfNSjy67rZ/1/Wx9BGV9Ou6/r+tyze2aX1sYnJRgQySL96Nh0Ye4qLT7x5t9teAJeQY8wDo47Ovsf0ORSaZNIFks7li1xbEKWPWRD91/xAwfcGnahZPOEntWEd3BkxOeh9Ub/ZP+B7Vlt7kv6/4Bpv76LlFV7G8S9t/MVTG6nbJE33o2HVT/AJ561YrNpp2ZommroKKKKQwooooAKw73xloFhcNBcaipkQ4YRRvLtPodoPNZfxE1iaw0uCxtmMb3pYO4OCI1xuA+pYD6ZrzMAKAFGAOgFephMCq8OebsjzcVjHRlyRWp6x/wsDw1/wA/0v8A4Bzf/EUf8LA8Nf8AP9L/AOAc3/xFeUUV2/2XR7v8P8jk/tGr2R6v/wALA8Nf8/0v/gHN/wDEUf8ACwPDX/P9L/4Bzf8AxFeUUUf2XR7v8P8AIP7Rq9kev2fjPQL+4WG31FRI5woljeLcfQFgOa3K8DIDAhhkHqDXpfw61ea+024sLljI1iV2OxyTG2cA/Qqw+mK4sXgVRhzwd0deFxjrS5JLU7GiiivLPSKHhX/kTtG/68IP/Ra0UeFf+RO0b/rwg/8ARa0V1z+JmEfhQyL/AJHHUP8Arwtf/RlxWlWbF/yOOof9eFr/AOjLirUc11MrNFbwlA7IC05BO1iOmw+lctepGM7Pdrs327GtOLcbliiov9N/597f/wACG/8AiKiuLqa0j8y5W0iT1e6I/wDZKx9rFdH/AOAy/wAjTlf9NFqisZtffP7myM3ursB+bIKrzeJL2ISN/Y0rqqAqEkVixz2/T0+tcjzPBp8rnr8/8ivZT7fkdDRWVZa0b+9ktLWKKSWJA0hMjqFz0HKZzx+FaH+m/wDPvb/+BDf/ABFdSrQkrq//AIDL/Inlf9NEtFRf6b/z72//AIEN/wDEUtvN9otYptu3zED4znGRmqjUi5cut/NNfmhOLSuSUUUVoI86j+Kk8kSuuhx4YAjN8f8A43Tv+FoXH/QDi/8AA4//ABuvPo3ZNOgKY3FUUZGRzgf1qTbcf89Yv+/R/wDiq+n+oYf+X8X/AJnzv12v/N+B3v8AwtC4/wCgHF/4HH/43R/wtC4/6AcX/gcf/jdcFtuP+esX/fo//FUbbj/nrF/36P8A8VT+oYf+X8X/AJh9dr/zfgd7/wALQuP+gHF/4HH/AON0f8LQuP8AoBxf+Bx/+N1wW24/56xf9+j/APFUbbj/AJ6xf9+j/wDFUfUMP/L+L/zD67X/AJvwO9/4Whcf9AOL/wADj/8AG61/DHjWTxDrDWMmmra7YGmDrceZnDKMY2D+9+leV7bj/nrF/wB+j/8AFVf0XVtQ0LUje2jWzyGFocSwsVwSpPRxz8orKrl9Lkfs46+ppTxtTnXPLT0PcaK57wZr154g0q5uNQWBZIbkwjyEKqRsRuhJ5+Y966Gvn5wdOTjLdHtwkpxUlswoorl/HurX+kaPaS6Zcm2lluxGzhFYlfLdsfMCOqispyUIuTHKSirs6iivGT4611WIbxAQQcEGKDj/AMco/wCE81z/AKGH/wAhwf8AxFcn1yn2ZzfWodmezUV4z/wnmuf9DD/5Dg/+Io/4TzXP+hh/8hwf/EUfXKfZh9ah2Z7NRXjP/Cea5/0MP/kOD/4ij/hPNc/6GH/yHB/8RR9cp9mH1qHZns1Fcz4M8Swarolol9qttcanI0oaPzEEjAO2PkXH8IB4HTmumrrjJSV0dMZKSugrP1//AJFvU/8Ar0l/9ANaFZ+v/wDIt6n/ANekv/oBqK38KXozSHxI8+lhsIWVXtYyzAkBYN3Tr0HuKZt0/wD58/8AyUb/AOJqd/8AkIw/9cpP5pSJPcS7jHDFtDsoLSkE4JH932r8ZTdv+CfWEO3T/wDnz/8AJRv/AImjbp//AD5/+Sjf/E1Pvu/+eEP/AH+P/wATRvu/+eEP/f4//E0Xf9MCDbp//Pn/AOSjf/E1s+F9T0/Srq9aVZII5UiC7bWTkgvngL7is3fd/wDPCH/v8f8A4mjfd/8APCH/AL/H/wCJrrweLqYOsq0NWu7010MqtKNWDgz0m3njuraK4gbfFKgdGwRlSMg81JXK6Brs6nTdJms41XyxB5yzk/cjJzt2jrt9e9dVX6rg8XSxdL2lJ36P1+Z83VpSpS5ZIKKKK6zIKKK4T4paVN4gttH0aO/u7OK6uDv+yzmLzDwoVyOSvzkkfT0qKk404uUv66GlKnKrNQjv/TOzj1Gylvnsory3e7jXe9usqmRVzjJXOQPeuV8TeN9W0nWTpHh/wXq+u3YRX85NsFrz0BmbjPqMVzPhrwPD4GmmHh/UdA0+5YGKWVpVaYjOSpZ1LdQOM9RXRfbfEH/Q26L/AN/Y/wD4iuf63FfZf4f5nV9Rk/tr8f8AI6Hw5ca1eaFFL4nsINO1Fy3mW9tN5qoMnb83c4xn3rj4Pgtot1dJd+LNW1rxNcI29f7Rvm8tG7bUTAH05q/9t8Qf9Dbov/f2P/4ij7b4g/6G3Rf+/sf/AMRR9cj/ACv8P8x/UJfzr8f8jtqr22oWd5LNHZ3cE8kBAlSKUMYyegYA8dD1riNV8M6J4ysbeLx3rtvem2kcrFaaj5MciMF4cJt3YKnHpk+tdL4b8G+H/B0EsHhrSrfT0m2+aYgS0m3ONzHJOMnqe5rphKM48yOOpTlTlys5/Xfilb6Vrlzo2meGfEGt6hbMFkWxsSYlJAIzISABgjnmun0+81HVPDKXUlidJ1GeBiLa5YSeRJzgNt6jODxWnRVGZ59Y+DfHl1qFtd+JPiC4jhlWQ2Ol2CQxtgg7S5yzKcYII6V1viDwzo3iqxis/EOnxX9tFMJ0ilztDgEA8deGP51qUU7gZ+leHtG0KPy9E0my09cYItbdY8/XaBmn63cS2fh/ULm3bZNDayyRtgHDBCQcH3q7Wb4k/wCRV1b/AK8pv/QDS6gcF/wkuvf9Beb/AL8w/wDxFaGga/rFz4ksba61GSeCZ3V0aKMZxG7DlVB6gVztXtCgS58TabDMCUaR8gMVz+6c9RzW7SsRdnpE+qWsEph8wyzj/ljCpd/xA6fU4FRb9Tuv9XHHYxn+KX95J/3yDtH5n6VcgtobWIR20KQoOiooA/SpKz5orZff/X+YcsnuyimkW5cSXZe9lByGuDuAPsv3R+Aq9RRScnLcpRS2CiiipKIrm0t72Ew3kEVxEeqSoGU/gaqNotlFYvb6fbQWWSHQwRKm1xyrYHoafqesafo8KyaldJArHCg5LN9AOT+FR6X4g0rWWZdNvEmdRkoQVYD12sAce9aJVFHmV7fgZtwb5XuQCSR0XVLaIieMeXeW68lwvUD1ZeSPUHHcUy2t7K7uJraREmhf/S7WQHBCv97aw5BDZPH94Vbm/wBB1NbgcQXREc3+y/RG/H7p/wCA1namh0W/t7+Mf6H537zH/LLfw3/AScN7MvvW8fe0XXb/AC/roYy93V9N/wDP+upJe2l/psqX9rMbxLdTvikH7xoz1AYfexgEZGeOvNaFnq9resqKximZdwilG1iPUdmHuCRV2sdLe2hmk0y+jje1bM1t5g4UZ+ZQexUnI9iPSs1JVI2lui+Vwfu7MnvoJba4/tGyQvIABPCv/LZB6f7Q7evT6VJ/Gnh+3kVJdSTcwBwiM+M+pAOPoaxPFOr/ANl6NJFpOr+cJ28jYW8xosg5KyA5BwD1yfTpXnQAVQBwBwK9DD4FV4802cVfGexlaCPb59UgGhz6naSJcwxwPKrI2Q20E4z+Fcn4Z1LW4/EGi2+qavJqkWtaNJqDpJBFGLWRGh4j2KDsPn4w5Y/KPm65wvAmqfYNdazndVs71GEiyH5Q4GQeeBkAg+vHpXY6Z4T8M6dFPHpZljE8AtwU1KZmhiByI4W8zMK89Iyo4HoMcWIw7oTcdzroV1WhzEOsT6ra+M9LjsNYlm+1T5k0kQReUlqqYklZtvmZDlcEMBllXaeSerrCbwfo7+IZddT7fHfTvG8rw6ncxxymMYQNGsgRgAPulcHJyOTW7XKdB558UP8Aj80f/rncfzirh67j4of8fmj/APXO4/nFXll39uu/EU1pa6nNZRxWccqiOONgXZ3GW3KSR8o4BFfTYGVsNH5/mz5/GRviJfI26K5y28S3l1p0U9tYRTulkl1dfv8AywA27Aj+U7idjddo6c88TP4il+0MYrFWs45YI3mafDfvdm0hNpzguM8jjpnpXZ7WFrnL7KV7G7RXOnxG82pXNgIFEeyYRXMLyEFk7EmNVz1ztZsEYqrc6zfPonl2kJcQQW/2i5NwVkDOFPyjHzcEEksOpxnpSdaPQapS6nWV3Hwv/wCPzWP+udv/ADlrh67j4X/8fmsf9c7f+ctc2Yf7tL5fmb4H+Ovn+R6HRRRXy59GUPCv/InaN/14Qf8AotaKPCv/ACJ2jf8AXhB/6LWiuufxMwj8KGRf8jjqH/Xha/8Aoy4q9p3/AB6t/wBdpf8A0Y1UYv8AkcdQ/wCvC1/9GXFXtO/49W/67S/+jGrln/vUf8L/APbTWP8ACfr/AJkWqaibNUigCvcy52BuigdWPsP1P51iiLMvnTM0056yvyfoPQewpxl+03lxdHne5RPZFJA/qfxpa+EznMamIrSpRdoR09fNndRpqMbvcKrzwXMtzG8V2YIk5aNY1PmHPcnoMemD71Yo/su51a1Jt7k2aBgY5gm4sQc9P7vGPcZ+teZg8LWxVTkpK7/BGs5xgrsZLCkuC4+ZTlWBwyn1BHIrR03U5BMtpftuZuIZ8Y3/AOy3o36GsfTYLm2shBfSrLNExRmUH+Hjv16Zz71NcJvt3G1mOMjZ97I5GPfPSuzBY6tl1dwveKdmv66kTpxqRudVVHTv+QXa/wDXFP8A0EU3QtS/tbRbe8KsruuHVl2kMODkdue1O07/AJBdr/1xT/0EV97UkpVYNdn+cTij8L9V+pYooorQR88r/wAg23/7ZfzWpbgbmhXLAM+DtYjPyk9qiX/kG2//AGy/mtTTf623/wCuh/8AQWr7U+SD7Mn96X/v63+NH2ZP70v/AH9b/GpaKdkTdkX2ZP70v/f1v8aPsyf3pf8Av63+NQB7ua4uFhlhjSJwgDRFiflU9dw9af5d9/z82/8A4Dt/8XWCqX2g393+Z3PCqNuerFNpO3vdVdbRa2ZJ9mT+9L/39b/Gj7Mn96X/AL+t/jXNi08WzqLq01yz8uYCRIZbPAjB5C5BycZxmt//AE6PR+kMuoLB05WN5dv5hd36VVOoprmsYVqUqU3BvVOxetbi6sY2Syvr23Rm3MsV1IgJwBkgN1wB+Ver+Bbia68G2ct1NLPKXmBklcuxAlcDJPJ4AFeAWupeK/tkMd7oFt5LOqyTQ3o+RSeW2kZOBzivX9J8Q3Hhr4a6dd2mhalrbPdTRtBp0YeRF82U78EjjgD8a8vMeV0k0rO/a3RnfgeZVGm76d7noFcZ8Tv+QDYf9f4/9FSVP4U8fJ4p1OWwPhrxFo00UJlZtWsPIQgEDAbccn5s/QGuY+MPjrw7o7WOi6lqHkXwmW68swuR5ZSRc7gpHXtnNfO1ot05JHqVdabscjC4S3d3OFV5CT7bjTvtKf3Zf+/Tf4Vn2Gp2eraJc3GnXC3EQMqll9eTj8iKtrqmnvI0a31szqxVlEy5BHUEZ614Li09UePZkv2lP7sv/fpv8KPtKf3Zf+/Tf4VKCCAQcg9CKKnQRF9pT+7L/wB+m/wo+0p/dl/79N/hUtFGgF3w1qlrp/inT7y7MscELuZH8hzgGJ1HAGepFet6R4h0zXWmGl3DTGAKZA0Lpt3Zx94DP3T09K8XruPhf/x+6x/1zt/5y134Oq1L2a2OzDVGpcnQ9CrP1/8A5FvU/wDr0l/9ANaFZ+v/APIt6n/16S/+gGvRrfwpejPUh8SOCf8A5CMP/XKT+aVCGZbFvLYoxuCu4AcZlwev1qZ/+QjD/wBcpP5pUH/Lj/29f+1q/GY7L5fqfWMn+zy/8/k35J/8TR9nl/5/JvyT/wCJpjQRTajL50SSYiTG9QccvUS/2cyhltAVIyCLRuf/AB2j+tgLH2eX/n8m/JP/AImj7PL/AM/k35J/8TUG3T/+fP8A8lG/+Jo26f8A8+f/AJKN/wDE0Wf9IC1Ck8FzHPFezLLESUbahwSCP7voTXYeF726vbK6N7O07xXGxWZVU7diHHygDqxrhdun/wDPn/5KN/8AE1Hcx2bWsqw2Z8woQuLVhzjjnbXr5ZmVTA1L6uPbZX7nLiMPGtHs+56xRWfY67p+pXJt7SV2lCF9rwunyggE/MB6j860K/UKdSFWPNTaa8tT5yUZRdpKwVy/iz/kYPDX/X2f/Q466iuX8Wf8jB4a/wCvs/8AocdZYr+C/l/6UjqwX8dej/8ASWY0bw20niu9ks7W6lt7tfLFzEHUbpWB/wA+1TwQ65c28c8HhTRWilQOjeSgyCMg8vVSf/jx8Zf9fcf/AKPai+0z+2NY8P2Hm+T5ulRfPt3YwrnpkeleVrol9ystXKS6pntaat29Xd2ShF7Jov8A2LxB/wBClov/AH6j/wDi6PsXiD/oUtF/79R//F1g/wBk+H/+hm/8kJP8aP7J8P8A/Qzf+SEn+NTzv+pw/wAiuRf1Cf8Amb32LxB/0KWi/wDfqP8A+Lrf0a58RXF4665p9tbweWSrxMCS+RgffPbNcF/ZPh//AKGb/wAkJP8AGj+yfD//AEM3/khJ/jWlOvKm7r/0uH+RnUw8KkbP/wBIn/meqkEdaSua8P61odlp9npMGqfaJQ5RG+zum8s5IHI464610x4NexTqRqR5k18mnb7jwatKVKTi0/K6av56iUUUVoZBVTV50tdFvriaBbiOK3kd4X6SAKSVPB4PTpVus3xJ/wAirq3/AF5Tf+gGhCOI/tzTf+hO0n/vpf8A41WhoOs2E3iCzgh8NafZSSs6rcQMpZMIzf8APMdQCOveuXq/oIlbxPpot3SOTzH2s6FgP3T9gRnj3rflViLs9Soqj9m1I/e1GMf7ltj+bGj7Ddn72rXA/wByKIfzU1nyr+Zfj/kPmfb8v8y9RVH+zZD9/Ur1v+BIv8lFH9kxn79zet/29Ov8iKOWPcLy7F6iqP8AY1ofvfaH/wB+6lb+bUf2Jpx+9aRv/v8AzfzotDu/u/4IXn2/H/gHjmqalJrOrT6hM24ysfL/ANmMfdA/D9STUEM81pcR3Vo5juIW3xuOxH9D0I7ir3iDR5tB1ia2mVhC7lreQj5ZEJyAD6jOCPb3qrY2Fxq18ljYoZJpDg4HEY7sx7AV9ZCVL2Ka+Gx81NVPatP4rnraavaatpKYguJxcwgskUTfLkdNxwAR9aZbz6jqFnLY3FnCGQeVObiTlgRwdqgjkH+91yO1O0fVLWK3i02ZGs57YLbiOU8EqAAFboeMH19qtah/odxHqK/dQeXcD1jz97/gJ5+havmvhfLb0Pf+Jc1/UyNHN60r6VqOozI9sTHGYlVfNUAdSQTnBU8diPer2paLZCxeZsedD+9Sa6kMgBHPJYnAPQ/WotbSG2vEuJGaNbgBEeNdzLMuTGQO5ILD34FRxXLXUT3euJtltWVV09OcSH7p/wBok9Ow+oJrT3pNVI6ehn7qThLUxfEtw/iHQ4kt4RDGGEkMRHzllO05x90ZJQDqSewFcAcozLIrI6HDI4wVPcEHpXpqhbLXnutRYeSMS3LKBsS42sVXPoFP5lT1Na1poVhdWsdxqum2txdyEyu00CuyljnbkjtnH4V208UsNGyWhyzwzxErt6nF/D3RJLvVf7VmiItIEZYmZeJXPBx6gDPPqfY16I+nWUn+ss7dv96JT/Sp1VUUKgCqowABgAUteZXxEq1Tn2O+jQjShyblE6JpfUadag+qwqD+go/sWw/hgKf7kjL/ACNXqKy9pPuzXkh2PNfiPaRWl5pQhMpDxz58yZ3xgx9NxOOvavPr3Q7HULo3FykxkMYiby7iSMOgJO1grAMOT1B616R8UP8Aj80f/rncfzirh6+lwPvYaPNrv+Z8/jPdrvl0/wCGM+60LTrxkM0BASPydsUrRq0f9xlUgMv+ycjk+pqV9MtJBKGh4mkSVwGIyybdp68Y2r+VW6K7eWPY5OaXczY9A06O7a5SF/MbfwZnKrv5fapO1cnk4AyaSXw7pk7q0lux2oke0TOFYJ93coOGI7EgkceladFLkj2Hzy7hXcfC/wD4/NY/652/85a4eu4+F/8Ax+ax/wBc7f8AnLXHmH+7S+X5nVgf46+f5HodFFFfLn0ZQ8K/8ido3/XhB/6LWijwr/yJ2jf9eEH/AKLWiuufxMwj8KGRf8jjqH/Xha/+jLirunnFm3/XaX/0Y1Uov+Rx1D/rwtf/AEZcVe07/j0b/rtL/wCjGrln/vUf8L/9tNY/wn6/5nOWP/IPt/eNT+lT1HAhgD2zfet3MZ+g+6fxUg1JX5XiIShWnGW6bPUi7xTRz/jrxCnhbwVqOrOwVokCRls4DuwUE47AnP4VyXwX8c3viXxnq1mNSk1Kzt2xFcmExCVCpwdhJ24K4HfB5rd+KXhbUfGPgK60fSGgFxI6SKJmKhtrZwD6n34rH+Cvg4+DvCU5mP8AxNr5ylzgY+zAcFD/ALXH9ehFfYZBiMPhcHUqTkr9e/l+pxYmEpzSR6NJMm6ecnCNIzA+ozgfn/Wq+p6bqe2yuob2O2UybXg2b9ynn7wPX5e3qee9eRfGLx1d2PiSw8IaNZvczTQbysUrRkSvlYvu8kLgtjjORngc+r6THdjStN09cS3EMG1c/d3H70h/2RnA9e3WuWng4Sw8sXUV6lRvlj69f18kW6jU1BbLc6DQP9VeEfdNycf98KD+uasad/yC7X/rin/oIqeytEsbOO3jJYKOWPVieST9Tk1Bp3/ILtf+uKf+givo6dJ0fZUn0i1/6SY35lJ+f+ZYooorqJPnlf8AkG2//bL+a1NN/rbf/rof/QWqFf8AkG2//bL+a1NN/rbf/rof/QWr7U+SJaKKKokrWv8Ax83v/Xcf+i0qzVd7GF5XkJlVnOW2TOoJwB0Bx0ApPsEP9+4/8CZP/iq5oqrBNWW7693fsejVlharUnKSdor4V0il/N5Faz1GyttKtRcXlvERCmd8qrj5R6mrj3dvHYteNMn2ZY/NMoOV2Yzuz6Y5rIHgrw55zStpFu7u25i4LZP4mtlLaGO1W2SJFgVPLEQUbQuMbcemOMVVGNSMFGVtEYYqpTq1pVIXs23r5v5mHF458OT3sNpBqaSzzyLHGqRudzE4AzjFet6T4pg8J/DXTry507U9REt1NCsOm2pnk3GWU52gjj5Tz9K4CKCGAYhiSMeiKBXrfw//AORIs/8ArpP/AOjnrzsy5vYrm7/ozqwHL7V8vYr+FfHUninU5bb/AIRTxFo8McJlW51ay8hHIIG0cnnnP0BrlfjLqviOFbKx0zwul9Yecsn2+S+RFMuyUeV5ZGenzbunavVq4z4nf8gGw/6/x/6Kkr5us0qcnY9arpTZ47pH206BdHUbKCylLy4jgbcuPXPrnI/Cub1LwFf6nfXEjJocCSSMytHasXIJ4J5HPrXaN/yDbj/tr/Nqt14sa0oNuJ5Sm4u6M2XSTN4Z/skzmIm2EHnRDG3C4yB/SsrQ/Bh0a+iuTruqXXl5/cyTfumyCOV79c/UV09FQqs0mk9yeZpWCiiisyQruPhf/wAfusf9c7f+ctcPXcfC/wD4/dY/652/85a6sJ/GR0Yb+Kj0Ks/X/wDkW9T/AOvSX/0A1oVn6/8A8i3qf/XpL/6Aa9et/Cl6M9iHxI4J/wDkIw/9cpP5pUH/AC4/9vX/ALWqd/8AkIw/9cpP5pUH/Lj/ANvX/tavxmOy+X6n1j6k6f8AIRm/65R/zeooJXi0+yEaK7SIqgM20D5M+h9KlT/kIzf9co/5vUEP/Hrpn/Af/RbUf8D8gJ993/zwh/7/AB/+Jo33f/PCH/v8f/iasUVnddirFffd/wDPCH/v8f8A4mjfd/8APCH/AL/H/wCJrK0fTIbrQ7G4nnvnllto3dvt84yxUEnh6peKfCml6lpMUN59slQXcGA19McbpFQnl+u1mH411Rp0XV9nJ9bbf/bHROjFScIyble21l993+R1mmX93p2pC7+ywyYhaLZ55HUqc52f7P6122kagdU0uO7aIQs7OpQPuxtcr1wPT0ryXQPBWneG9Qe60651Bt8RiMM900kYBIOQD346+5roV8Ia1rtja3uj+M9S0IIJY/s9vGkkTETyHeVbvzj8BX1/DmJbrSw0J3ppXWltbr1f4nz2PprkVRr3r9z0auX8Wf8AIweGv+vs/wDocdXvCuk6zo2lSW3iHxC+v3BmLx3MlqkBRNqgJhODggnPX5vavO/GMXxNg8V6a9vceGrq3a+drCOVJkZV8xdquRwf4Rke9fX4iN6TV+35o4cJLlrJ26P8mbM//Hj4y/6+4/8A0e1X7L/kcPDP/YKT/wBFyVgtNqcfhHxdLcW0M2qq0LSQQvtjabzGyqs3Rd2QCe1Zug+KvE83jrwxDq3ga6sVNrHAJI76KdfLKsPNO3GBgk468V5tKDlKLXdf+lyPXqzUYyT7Po/5Im1ptxDp3gtr3+z7K6mbUDDm6hD4XyweO/Ufqan+2av/ANCbZf8Agreue1rW4tD+Gcc09reXKyaxtxawmQr+5zk46Co7j4mJ4j8UXkWiXerWioiv5E6SQbRtUHg8dawSlGip62SW1vM6G4yrOnpdt7t9LW/M6X7Zq/8A0Jtl/wCCt6Ptmr/9CbZf+Ct642++IS6Ze/ZNR8TS20+0Nslu2Xg9D1q9a+KL6+tluLLXLi4hfO2SK7ZlODg8g+tZOtZXfNb0RsqF3yrlv6s6aPUNaikWSLwhZo6EMrLpjgqR0IOa6nw3qeramt1/bFh9kMezy/3LpuznP3jz0H515t/bur/9BS9/8CH/AMaP7d1f/oKXv/gQ/wDjV0sdGnK92120M62XyqxtZJ99T2Tp1pKqaTI8uh6fJKzO72sbMzHJYlRkk1br6FO6ufLSXK2gqtqUMFzpV3BeSeVbywOksm4LsUqQTk8DA7mrNZviT/kVdW/68pv/AEA0yTlP7D8Kf9DX/wCTlv8A/E1e0XR/DsGtW09hr/2y5jLGOH7TC24lGB4VQTwSa42r/h9zH4o011jaUiSQhExk/uZOmSBWzi7bkXPUqKo/2si/661vIv8At2Z//Qc0f23pw/1l3HD/ANdsx/8AoWKj2c+w+ePcvUVFDd29x/qJ4pf9xwf5VLUNNblJp7BRRRQMiubS3vYTDeQRXER6pKgZT+Bplnp9np8ZjsLSC1RjkrBEEB/ACvOPFPjO+u9SmtNLuGtrOBynmRHa8rDgnd1AzkDHXrVLRfGep6Rdobu6mvbMkeak7F2Ve7Kx5yPTof1r0Y4Cu6XN+BwPG0VU5fxPSdUtAD9sSETgLsuICu7zo/p3YdR+I71Tmk+wxwjSJvtiXIzHYyHeHU91bqq898jtV6fUjNILbSgs87KGLn/VwqehY/yXqfYc1VsrNNBvtrnzIrxhmdlAKy/3Tjop5wOxyO4rCDaj733d/wCvvNpWcvd+/t/X3GXZQ+ZHNaaizQ6rCoNmJ24UDBUJ2IBGCckkDn0E8tw19DH4jiQKIQFt4WHM2TtIJ9SSQvp/wI1Y8QW41WVYIYBP9j/eXH+0pwfKH+0wGfbA9RVCzjkXWDb6XKl3Y2xF1FbyORksOgbHON27n+8v1roTUo83X9P+Dp+Jg04vl6f1+Rc+zx32jWeniTzGv3M1zIvB4O5z7fNtXHb8K19NuZJoWiucfabdvLmwMZPZh7EYP6dqwLTU7eHW7u/KiCJm8uWJ+HjxjdIBnBG4ENjP3QfWsm8+IVsmtLc6bYySxBTHI7vs85eoIGCRg5wT6njniXQqVPdivP5/1oNVqdP3pPy+R6DRWDZ+I4df8P3s2jbxeRQt+4YfOj7Tt+uSOCK4TwTdeG7LW9Nv9BubK2t4NAkbxFOsioFn3w+WbpjyJc/aOZPm+/muGUJRbUtztjJSV0es0VwGral4fsfjVogl1Szi1GayuYJYpb0bwzG38qMIzfKW5IUAbjk8nNd/UjPPPih/x+aP/wBc7j+cVcPXcfFD/j80f/rncfzirh6+oy//AHaPz/M+dx38d/L8gooorvOMKKjuLmC0t2nupo4IUGWkkYKq/UnpT0dZI1eNgyMMqynII9aQC13Hwv8A+PzWP+udv/OWuHruPhf/AMfmsf8AXO3/AJy1w5h/u0vl+Z2YH+Ovn+R6HRRRXy59GUPCv/InaN/14Qf+i1oo8K/8ido3/XhB/wCi1orrn8TMI/ChkX/I46h/14Wv/oy4q9p3/Hq3/XaX/wBGNVGL/kcdQ/68LX/0ZcVe07/j1b/rtL/6MauWf+9R/wAL/wDbTWP8J+v+ZR1iycSfb7ZC7Bds0ajl1HRh6kfqPwqgjrIivGwZWGQQeCK6ese/0go73NgUQsd0kLnCOfUH+E/oe/rXg5xk7xL9vQ+Lqu//AATejW5fdlsUa84vLvxIniqayiuJbxZJGEe8+SIsfdGU4546gnvXfR6javc/ZjPGlz/zwdxv/AZ5HuMip/Kj83zPLXzMY37RnH1r5ClUlhZSjUhr5noRkt0crP4X0K11GHxXrdvIdXSOKKS4Xe5JJCKgRcg5JA4HNdDovifToJrmOW11PzZG3q40q5Yuo4xgR5G38uc9zVHxFcRtp8UcbebJ9us/kj+Y/wDHzH6dPxrqNK0+dLn7ZdqImCFI4s5KgkEliOM8DgdPxr3sl+tzrqpZ8tmtdkvL59iascPGgpTTbbezS2t5PuZWq/E7wnoM1vFruozaW1zu8n7dYXEAk243YLoBxuH51b8K+JNE8Q6VGdC1az1D7PFGJhbTq5iJHAYA5XOD19DT9StoLvxrpkV1BHPGdNvMpIgYH95bdjTvD+i6XpVgkul6bZ2UlzFGZ3toFjMpC8FioGcZPX1NfVTd6yT3s/8A2045wjGClDZ66+rW+nbsa2aK4jxB8I/CviLXLjWrmO+tdTudplurO+liZtqhRxnaMAAcDtXS6Joq6D4eh0q1vby5ECsqXN7L50pyScsx64zx7ACtDnPC1/5Btv8A9sv5rU03+tt/+uh/9BauVj0vxXb2tuU8Q211FmPCT2QUgZGBlTzW3r95c2FjHcWMEU8yyjCSzCJSMHPzHgV9ip6Xaa/ryPlnDWyd/wCvM1KK5bRPGUmp6xHpt5p0VpM6sQY7+KYHAz0U5rQ1vxXpPh25hh1ad4TMpZGEbMMA+wpqrBx5r6CdKaly21Nmiqumanaaxp8d7p0vnW8mdj7SucEg8EA9QatVommrozaadmFFFFMAr1f4f/8AIkWf/XSf/wBHPXlFer/D/wD5Eiz/AOuk/wD6OevJzT+CvX9Gell38V+h0lcZ8Tv+QDYf9f4/9FSV2dcZ8Tv+QDYf9f4/9FSV8xX/AIUvQ9et/DZ5a3/INuP+2v8ANqt1Ub/kG3H/AG1/m1W68FnkBRRUF5LJDb7odocuiAsMgbmC5xketSXSpurUjTju2l95PRVby7//AJ+bf/wHb/4usvVYNfnureLTdWt7Q7Hd/wDQ9wfBUAHLHHU9KcVd2bsbrDRlfkqJve3vdNXvFdDdruPhf/x+6x/1zt/5y15nosGtweeNdu7W6Hy+S1vEUI67t36frW/4V1TxxYa1q/8AwjHh2w1a12QZ82+8hwPnx1GOu78hXZhY2rpXJw6tVR7fVfULX7dptzab/L+0QvFvxnbuBGcd+tVdBvdUvtAgute0oaVqDBvOskuFn8vDEDDrwcgA+2cdq4ofF6WBgup/D7xlaknG9dMEiD/gQavYceZNM9VOzuWNU0eXStRtfOukuPNilxshKbcFP9o561lf8uP/AG9f+1q7Txld+HtI0U6z4qkeGzs/lMyGXKb2UdI+TzjseleXWHjrwprV19g0LVo55JLsmCIq4Zk83cPvgH7vPPNfnmcZPPDT9rQj+7utFd2stW/+HPdwuLVRcs37x0if8hGb/rlH/N6gh/49dM/4D/6Lam3Gp2FhqjJfXtvbNLEnlrNKqF8Fs4yeeoptncQ3Fpppt5o5QNuSjBsfu29K+a5ZKPNbT/gHoXV7GnRRRXOaGBomt6VF4f06OXU7NHS1iVla4QFSEGQRmqfivxfoenaTDJJfxT7ryABLVhMxxIrn5Vyein9B3rq6K6o1aSq+0lFvW+//AADonVi5upBNO997/ojnfD/jSz8Sag9rZafqkISIyefdWhijOCBgMe/OcegNdCPFuvaJY21lofgu/wBeDiWU3EM6RRKTPINhLdxgH6EUtdb4U/5FyH/rrN/6Nevp+GpU5Y2bpxsuXa9+q9DwcwUlRXM76jfCuqa5q+lSXHiXw/8A2BdCYrHam8S5LR4BDlk4GSSMdRj3ri/EGk/FXWteimsF8J2dpYXTtaPcSXDSOm8FSwUYzhRx9a9OpRwa/QGoyVpLQ8WMpQd4vU8qeHU5PCPi6G5uYYdUZoVkuII90azeY2WVW6ruyQD2rN0Hwp4mg8d+GJtU8c3d8otY5xGllHCvlhWPlHGcggEZ6816Trfh+OTQ9UTSLX/S750eQeZ98iTcT8xwOrelZUUElt458OwTrtli01Udc5wQkgI4rzOSpRnFPa61t3m+tuzPZ9pTrwlJb2el+0Etr90cZrOixa38Mo4Zrq8tlTWN2bScxlv3PQkdRWff+CtL8J+Lr7+zpL2ZyiIZLu6aZiCqtzn3rof+ad/9xX/2lR42/wCRwvv+2f8A6LWvNqTksNyp6e7+Uj06VOLxXM1r735xONv/AAzouqXou9R02C5nChN8i54HQfrV60s7awtUtrKCO3gTO2OJAqjJyeB71NRXnOcmuVvQ9JQgnzJahRRRUlns2i/8i9pv/XpF/wCgCrlU9F/5F7Tf+vSL/wBAFXK+1h8C9EfAVPjl6sKpa3by3nh/ULa3XfNNayxxrkDLFCAMn3q7Wf4gleHw1qcsLtHIlpKyOhwVIQ4IPY1aMzgP+Ea17/oETf8Af6H/AOLrQ0DQNYtvEljc3WnSQQQu7O7SxnGY3UcKxPUisP7bff8AQT1D/wADZf8A4qtDw/qNzH4msPtOpXRgZ3Egnu3ZCPKc8hmx1Ard81iND0uimxyJKgeJ1dT0KnIp1YFlabTrG4/19nby/wC/Ep/mKi/sWwH+riaH/rjK8f8A6CRV6iqU5LZk8kXuij/ZZX/U397F/wBtQ/8A6GDR9k1BP9VqW/8A6726t/6CVq9RT55ByI8Gkt5LSeW2nyJYZGjfPXIOCfx60xvunjPHQDrXq/iLwTZ67cG7imazvCAHlVdyyAcDcuRkj1BBqlpfw7hsJRcXGoNcXMZDQsIQqRsDkHaSc9PX9ea9+GZUvZ67niSwFTn02NbRYr7S9FtLUaYmY4lDlZgCzYGWIx1JpdVvrl7Q2baVI8t0GSNPNQgnGc9cgDrn6VZGrC0BTWAttIoJEg5jlx/dPr/snn0z1qTT4JHke/vFKzzDCxn/AJYx9l+vc+/0FeNez55L89fxPVtdckX+X+RkWmqyaPZyWt3p101zGS5bMeZtzYVj855JwDjOOPasyW8n0C3WWW1lWZGMys6BVYuAJE4PTdtYfQDtW7qUMmo6oPsxRW00eYHccNKcEIT2G0ZP+8p7VnX123ir7Pb6Si/uP9Ileb7qtggJ/tfxA44461002m7taPf+vMwmmlZPVbf15Gb4yuUh8I2ljHDMG81FaSaErnAJYgnuSO3rXBV6LbabF4i0ibTnRzcxqUmu5yWaJlP7tR+QLAccHuc1xM+g6tb3MkDadcTtG5QvbRNKhI/2lHH44Nelg6lOClTbs73PPxUJzamloXvBd5NZ+LLfyI5JvOR43jjIyw2lu5A4IHevUP7TYff0+9X/ALZhv5E1zHgfwpc6bM2p6qgjnZNkMOcmMHqW7ZOBx2H147WvLxtWnUrNxVz0sJTnCklLQo/2vAPvw3i/9ucp/kpo/tmy/ikkT/fhdf5ir1FcV4dvx/4B12n3/D/gnm3xHvLe9utKNrMsoRJw20/dyY8Z/I/lXn2uG6Gg3p0/zDc+Q3l+X97OP4ff0969H+KH/H5o/wD1zuP5xVw9fS4FJ4ZJef5nz+MbWIbfkcZbahpttfTr4Z0/T55hYSzRPazlpnZQuFmTaDkscfMxOQehJqVdanW2un07WBqsC28bvdeWh+zsXAY/IAOEJbackbeeDXXUV0ezl3MfaR7HG3M02o25it9Zlu7WO/tfKvY1iJYlxlchNjbTtOQOpwc4rSsr+/k146VJMxa1lklmlKKPMhIHlLwMfxkZGD+5Prz0FQx2sMV1NcomJpwokYknIXoPYDJ6eppqm073E6iatYmruPhf/wAfmsf9c7f+ctcPXcfC/wD4/NY/652/85a5sw/3aXy/M3wP8dfP8j0Oiiivlz6MoeFf+RO0b/rwg/8ARa0UeFf+RO0b/rwg/wDRa0V1z+JmEfhQyL/kcdQ/68LX/wBGXFXtO/49W/67S/8Aoxqoxf8AI46h/wBeFr/6MuKtLayR7hFeTopZm2gIQCSSeq56muOs5RrqaV9LaW627tdjanZwav1/zMXWW1K31XfPfyx2k2Y4I7bAKYGdx+UktnjGSOnHWqojkYRf2laXckzLkLcssr/XapOPwArpfIn/AOf+4/75j/8Aiar3GkJdPvuLmd227c4Qcdf7teRjMPLEcztPW2nMkvzf5G0LR6r8TnL9NKfyY9RtVBZwsXm2rAhz0wdvB9+KfqkNvN9ljuUmk3zbYxHIygNtJBbB5HHfI56V0U2mm4SNZr25cRncudnXGM/d5696Z/Y6f8/Vx/45/wDE14lbJ68Zp4dOy7tXv5WN1Ujb3mvxMGK8l1zw5b21klus7T29zbeafJSXypkkZGKqdr/IRwD6jvjVudT8R2kSyTaPpZVpEjG3VJCcuwUf8u/TLDPtSw+GbO3tRbQzXCxBi4G5chixYkHGRySfamahp7WmnIsd9dMpu7fiRw+D5ydMj8a+nwc8TdRrxd9rprbpfz9DOdSEaTikna715uqXZrsY3iLTPiBqerWF7oM2haNJbQTwyPLLJd7xI0R4Xy0Ax5XXJ61b8Dad4osdPd/FWu22qrLHF9mjt7MQCAAHdkgktnK9em33rovIn/5/7j/vmP8A+JqSGJYLeOJCSsahQT1wBitlKUpp8rWj3t5dr9iKkk1+SV7Lfv6nG+IPAmta5rtxeQeO9a0uzl27LKxCIsWFAOGIJOSCfxro9A0h9E0G302bUr3VGhDA3d/Lvmkyxb5mAGcZwPYCtKitrmJ8uS/C60t7GOXUtL1ghTGHe689VXLAHJ4A64rb1SxttSgitb+FZ4JJPmjccHAJH6ivbfGVhdap4UurTT4TNcO0RWMMFztlVjySB0B715TqugavpS20+pae9tE02wO0sbZbYxxhWJ6A19Hg8RTqRaaSbdrdzwsVRqQkmm2l17GJZeHtH06ZZrHS7SCVc7ZEhUMOMcHrV2SCKVlaWJHZfullBxUlFekopKyR57k27tiKqooVFCgdABilooqhBRRRQAV6v8P/APkSLP8A66T/APo568or1f4f/wDIkWf/AF0n/wDRz15OafwV6/oz0su/iv0OkrE8U+Hm8SadBbJdi1aGcTBzF5gPysuMbh/e9e1bdFfOSipKzPbklJWZ4Df2psV1C0ZxIYJZ4i4Xbu2uwzjJx06ZqSvVvFOg6P8A8I3rN3/ZVj9p+yTy+d9mTfv2Md27Gc55zXkn2u3/AOfiL/vsV4uIouk/U8qtS9myWq2of8eyf9d4v/Ri1J9rt/8An4i/77FRzS2dxEY5Z4ypIPEuDkHI5Bz1Fc1mPDTjSrwqS2TT+5lmqVzPFBqlu08qRr5EvLsAPvR+tN8qx/5+5P8AwMf/AOKqnf6HoGqNE2pJHdmHPl+dcs+3OM9W9h+VOKV/e2NoSw9JuUZSbs18KW6a/mfc1La+tLxnW0uoZzHjeIpA23PTOOnQ/lWz4T+IvhPwjqerw+ItZhspSkACFHduPMJ4UHsyn8a5rTbHRdIVxpkVpaiTG/yiBuxnGfXqfzr0L4X21jcalrE4gt5ZNlv+82Kxz+87/QCuvCqKrq2xlh7e1VjudB8Qab4n0GDWdDuDdWFwGMUvlshfaxU/KwB6qeoril+MSXDAad4E8aXQP/LQaTtT8y39K9Hor2tD1DM8RX+qaboc11oOj/21fIVEdl9pW38zLAH52GBgEnn0rgw3jzW7lRqngLS9MhLKzTJqEc1wuGDfKQAOcY616fRWVWnGrTlTls0195UZOMlJdDwjx54O1fW9VgiWw0dDHAC51WHznUFmwUKEgA4OcnsKr+AvCE3hWHfcS2jm+kjcLawGMLiN+uSc/er2LWPDsup6l9rhvUg/crEUeAv0LHOdw/vfpXEQ/wDHrpn/AAH/ANFtX53mlPE4GDw3/Lp7bXdtfz9D3sNKnWftPtLc5PVfhm2p+I7vVV8T6xZC5cN5NpN5YTgDAP4eldnY2v2HT7e18+a58iJY/OuH3SSYGNzHuT1JqxRXz9bFVq0IwqO6jtsd0aUINuK3CiiiuY0Cut8Kf8i5D/11m/8ARr1yVdb4U/5FyH/rrN/6NevrOFf97n/h/VHl5l/CXqbFFFFfop4QVSk0e2n1+21Z3lFxbxmNVBGwjDdRjP8AEe9XaUHBzSsnbmHGUo35X5HlX/NO/wDuK/8AtKjxt/yOF9/2z/8ARa12PirQZLzQI7TRLGJSLoStHEFjH3WBPYdxXHeNv+Rwvv8Atn/6LWvnsXRlRpOL/u6/KR9Rg68a1VSX97T5xMGiiivJPZCiiigD2bRf+Re03/r0i/8AQBVyqei/8i9pv/XpF/6AKuV9rD4F6I+AqfHL1YVm+JP+RV1b/rym/wDQDWlVbUrz+z9Ku73Z5n2aB5dmcbtqk4z26VZmeQ/bbX/n5h/7+CtTwxPBN4u0xY5Y5D5kmQrA8eTJW5/wsC6/6A8P/gaf/jdXNJ8aTalrFrYy6bHCLhmXzFuS+3CM3TYP7uOtbNu2xOhvyaPp8jl/skaOerxDY35rg0z+zZI/+PXULuIf3XYSj/x8E/rV+io9pLuHJHsUNurRdHtLkejK0R/Mbv5Uf2hcx/8AHzps4HdoWWRf5hv0q/RRzLqg5X0ZRGtWGQJbj7Ox6C4UxH/x4CrqSJIoaNldT0KnINKQGBDDIPUGqT6Pp7sXFrHG56vEPLY/iuDR7j7r8f8AIPfX9f8ADhqesafo8KyaldJArHCg5LN9AOT+FR6X4g0rWWZdNvEmdRkoQVYD12sAce9eP6nqMmsapPfysW81sR5OdsY+6B+H5kk1WSWW3lWe2do5ozujdSQQfqK9iOVp07uXvHlyzFqdktD2V4k1u6ZZV32EDFcdppBwT9F/n9Kq32rN4dzHIzX8ZX92gOZo+wDeqk8bjz9adpd1Lq+mW/8AZu2xs/LUblIaTp0UchR7nn271JBZwNqwgt0xBZkSSsTkyTEfLuJ5JCnPP95fSvPVou09l0/rqdru1eO76kelaUbiwSXUbmO6SQmUxxH907E5JY/x/Q8AAccZqi11JYFtUtSudSY/uyvCRKMJLx2AGT/vfSrXiKP+zbGWewlMEl03lNCPuylhyQOzYycjr6GrOi+XI0zXC+XeYCPbsOYIx91R6j3HBOfoL5vddR6p9P6/PyI5feUFo11/r8ihNE/h2+V7HdOupKIsu2c3H8Ln2ILE49K6CytVsrOOBCW2j5mPV2PJY+5JJ/GuYNp9tmntPMdYEDppUh4CyKQxwfZlwv8Asq1b+i6j/ammRzsNsw+SZP7rjr/j+NTXT5E/v/T/ACKotczX3fr/AJl+iiiuM6wooooA88+KH/H5o/8A1zuP5xVw9dx8UP8Aj80f/rncfzirh6+oy/8A3aPz/M+cx38d/L8gooorvOMKKKKACu4+F/8Ax+ax/wBc7f8AnLXD13Hwv/4/NY/652/85a4Mw/3aXy/M7MD/AB18/wAj0Oiiivlz6MoeFf8AkTtG/wCvCD/0WtFHhX/kTtG/68IP/Ra0V1z+JmEfhQyL/kcdQ/68LX/0ZcVpVmxf8jjqH/Xha/8Aoy4rSrCr8X3fkaQ2CiiisywooooAK53xRr1npwitbrzFkaWGYEJkFVlVjz64U10VYvibw8niCzhj3COWKUEP6KThh+XP1ArfDuCqL2mxjW53TfJuaOn38Wp2Md3bBxFJnbvXBIzjOKs0yGGO3gjhhUJHGoVVHYDgU+sZWu7bGkb21CiiikUFYnirw83iTToLZLsWrQziYOYvMB+VlxjcP73r2rboqoTlCSlHdEyipx5ZbHnn/Cr7j/oORf8AgCf/AI5R/wAKvuP+g5F/4An/AOOV6HRXX9fxP834I5vqdD+X8zzz/hV9x/0HIv8AwBP/AMco/wCFX3H/AEHIv/AE/wDxyvQ6KPr+J/m/BB9Tofy/meef8KvuP+g5F/4An/45R/wq+4/6DkX/AIAn/wCOV6HRR9fxP834IPqdD+X8zzz/AIVfcf8AQci/8AT/APHK7Hw9pB0LQoNOacXBiLkyBNm7c7N0ycfex1rSorKriatVcs3dGlPD0qTvBWCiiiuc3CiiigAooooAKKKKACiiigAooooAKKKKACuM8SaNpmjWllPao0GLjYS87soXy34wzEDoK7OiuLG4OnjKLpytdrR2va/Y1o1ZUpKSPMvt1p/z9Q/9/BR9utP+fqH/AL+CvTaK+V/1Tj/z+/8AJf8Agnpf2m/5fx/4B5l9utP+fqH/AL+Cj7daf8/UP/fwV6bRR/qnH/n9/wCS/wDBD+03/L+P/APMvt1p/wA/UP8A38Fdp4SYN4agZSCplmIIPX969bNFetleSLLqrqqfNdW2t1XmzmxOMdeKjy2CiiivoThCiiigArzP4k+H/Ev9qLq/hjTotYil/wCPq184RTIAqgFN3DdCccHoBXplFZ1KUKseWaujWjWqUZc1N2Z5N4Q0WHxba3LZ1TSbm0cRz22paaYnViM8fPhh7j2rov8AhWo/6C//AJLf/ZV29efar8Mr2LV7vWPBXi3VNBvrqVp5oJH+1WkrsckmJ+mT6HjsK5/qGF/k/F/5nV/aWL/n/Bf5Fr/hWo/6C/8A5Lf/AGVH/CtR/wBBf/yW/wDsq6HUdVk8OeFG1HU4bnUpbSFDOun25eSVuAxSMdsknHYZ9KzvDHxG8KeMHEOh6xDJd87rOXMU6kdR5bYJx3IyKX1DDfyfi/8AMP7Sxf8AP+C/yHQ+M/Culaivhi58Q2MWp2MUcUkM8oiYnYCMbuCSCDgE9a6RWV1DIQysMgg5BFZGueEvD3iWMrr2i2OocYDzwKzr9Gxkfgal0zQbPQPDiaN4djXT7aGN0tgMyCIsSc/Mcn5iTgmu3RKyOBtt3Zp1m+JP+RV1b/rym/8AQDXH2UPxW0TULeG7udC8Tac0qpLcFGs7lEJwX2jKHA5wOa6bxZrOk6TpKQ69PJb2+pyiwEkaFtrSKQCSAdo/2jwOM0+ojzqr2hRNP4m06NJXhZpHxJHjKnynPfIrW/sPwp/0Nf8A5OW//wATV7RdH8Owa1bT2Gv/AGy5jLGOH7TC24lGB4VQTwSa15+xHLc3vtl5Y8ajF50Q/wCXm3UnH+8nUfUZH0q9DPFcwrLbyLLG3IZDkGn1Rm0tDM1xZSNZ3DHLPGPlf/eXo316+9TeMt9P6/r/ACC0o7al6is7+0pbP5dWhES/8/MWTEfr3T8ePetBWV1DIwZWGQQcg1MouJSkmLRRRUlHiuv6PLoOsS2sqFYWctbSY+V0PIAPqM4I9veqtjYXOrXyWNiheaQ4JAyI17s3oB/9avb7m0t72Ew3kEVxEeqSoGU/gaZZ6fZ6fGY7C0gtUY5KwRBAfwAr145nJU+W2vc8t5fF1Oa+hn3Wmabp2kqzWiu1tEscbINkjEAKqhhg5JwKLHR7uxtFEOpSCZvnlEqCRGc9Tzhv/Hqmk/0/Wlj6wWOHf/alI+Ufgpz/AMCX0pmq3Il821DlIYo/MvJQfuJjOwe7D8h9RXCnNpRv59zrajdyt5GXb3V5qusfaXtUvLbTyVjMDbd8hwd4DHBwOnPfI61PrWo2k0cUUglsr1nCxTTKY/KH8Tb+hAHbJBOAa1NHtja6XH5iCOSTMsijgKW52/QDA+gqPTR9tuJdTflZB5duD2iB+9/wI8/TbVucea9tI7f1+JKhLltfVmXeXtvpvh9odQXNvBHm2urQZDsv3cf3XyO/B9eorg18WaxFdS3FnOloZm3PHHGrKTkn+IHuSePWur+ItlbW+hxTW0MMMhuV8wooUuuCPx+YrXndergaVOpTcmr3PNxlScJqKdrHpOheL7nXNKv7Vo1TVobZ3h8vpNwcEA9CDjI9x+HMeCbrw3Za3pt/oNzZW1vBoEjeIp1kVAs++HyzdMeRLn7RzJ83381B4NaZPF1pJBFJNsSRpEjxkrtI7kDqV716l/abD7+n3q/9sw38ia87GUFTqtQ2O/C1nUpXnuclrnhzQdZ+JGmQRaNp3262ZdV1C+Fonn4Q7YEMmNwLON3usLDvXd1R/teAffhvF/7c5T/JTR/bNl/FJIn+/C6/zFcXJPsdfPDucV8UP+PzR/8Arncfzirh67L4j3lve3WlG1mWUIk4bafu5MeM/kfyrzjWNSv9OeE2tlbXEUsiRbpLpo2Ds2BwI246c5/CvpcE+TDR5vP8zwMX7+IfL/Whq0VQXV7dFCXkiRXAmS3eNdzASsoYKCVGRgjnGPyNJPrlhAuWkkk+d02wwSStlDhjtVScA8E9M967eePc5OWXY0KKoSa1YRTRxmYt5gUh442eMBvu7nAKrntkjNQWfiK1uppInSaBlujbIZIJArtgkclQOx/T1FLnje1w5JWvY1q7j4X/APH5rH/XO3/nLXA211DdxtJbvvRXaMtggblJVh74II/Cu++F/wDx+ax/1zt/5y1x4/XDS+X5nXgtMQvn+R6HRRRXy59EUPCv/InaN/14Qf8AotaKPCv/ACJ2jf8AXhB/6LWiuufxMwj8KGRf8jjqH/Xha/8Aoy4rSrNi/wCRx1D/AK8LX/0ZcVpVhV+L7vyNIbBRRRWZYUUUUAef+FviNJrXhiTxHqX2W00nT7Mfb3CMHa52q7Kik/Kiqy4LZLlwRhQC9uw+KWj6npkl1YWt1dTR3cVp9ktZba4dnlz5eHilaPBII5cbcZbA5q1F8OtHh0tdPjmvFtzpw0+dVlCm4Rf9W7kL/rE52su37xznAxoReG3aGBNT1vUdUa3ukuonuVgQqy5wv7qJAV57jPvT0FqUpfHVvBqRs59H1SMRT29vdXBSIx2sk4Ty1ciTJ5kVSUDAHk4GCasPxO0C98VS+HrSfN2JZbeOUTwMHmjVi6+WJDKuNjjc0YUleCdy7ti78LWV4b8yy3A+3XdteS7WXh4DGUA46HylznPU4I7R2HhYaZqE8tlrGoR2M8sszaZiEwB5SWcgmPzBl2Z8b8ZPTHFGgE/hK+uNT8FaJf30nm3N1p8E00m0Dc7RqWOBwMknpWvVTSdNh0bRbHS7VneCyt47eNpCCxVFCgnAAzgegq3SAKKKKBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVQfQdJk1qLV3020OpQgrHeeSvmqCMEb8Zxg4q/RQByni6x8cNe2994J1bTY0hjKy6ZqNuTHcHOd3mL8ynHGBx6mtDwpqeu6ppLyeKNDXRb6OUxmFLlZ1kAAPmKw6A5PB54rbooEU9O1nTNYWVtK1C1vRC5jk+zzK+xgcFTg8HI6GofEn/Iq6t/15Tf8AoBrmNa+D/hXVNQOpafBcaBqmci+0aY20mepyB8pz3OM+9b3ixNUbwvc22hWS31zcL5BSScR7Ub5WfcepAJOO5p6XA85q/oE0dv4o0yWd1jjErgsxwBmJwP1NP/4RrXv+gRN/3+h/+Lq/oXh7VofENlNe6a0VvGzmRnkjYYMbrjAYk8sO1bXj1Is+h6BRWb9muNL+bTwZ7Xvak/Mn/XMn/wBBPHoR0q5a3cN5D5lu+4ZwQRgqfQg8g+xrNxsrrYalfR7k1Z7aX5DmXS5fsjk5aPG6Jz7r2+q4/GtCilGTjsNxT3KCap5LiLVIvsjk4WTduic+z9vocH61fpHRZEKOoZWGCpGQRWf/AGdNZ/NpUoRP+faXJjP+73T8OPaq92Xl+RPvR8yTU9Y0/R4Vk1K6SBWOFByWb6Acn8Kpw+K9JvLOabT7pbiSJciHBV2JOANrAHkkDPvXk+p6nLrOqT6jOctM3yDOQifwqPw/M5NV4p5bWeO5tm2TwsHjbHQg5H8q9iGWLkTk9Ty5Zi+eyWh7Ipk0rTooE2zX9yxPs8h5Zj/sj+QA9KZNaLFHa6YrGRriXzbiRurquGYn6nauPQ47VFp+oQGGLVL2QNdX0SvBBEC7LERkBVHPPUn19gKjga71G6utReX+zrVAYckBpAqE7jnlV5znr932rz1Fptv+n/wDuurWX9L/AIJZ1+/ijWHT2kKvdnDhAS/lj72AOcn7o+pPapl/tC7UJCg022AwCQGlI9h91fxz9BUWg6ekEUl60bCa7O/MjFnVP4QSec45PucdhVm8u5Wm+xWGDcsMu5GVgX+8fU+g7/Sodk+SPTq/6/zKV370uvQzrzTbS5WfS7aITSzx7bq5mJdo0Pqx53eg7dfryl58NdSS6YafeWstuT8puGZHUeh2qQfrxXolpax2duIockZJZmOWdj1YnuTU1VDFVKT9xkyw1OovfRgeF/CsHhyGR2l+0XkwAklxgAD+FR2H8/yA36KK55zlUlzSd2bwhGEeWOwUUUVBZ558UP8Aj80f/rncfzirzzULH7dHAvmeX5VxHNnbnO1s4/GvVviLpE19pcF9bKXaxLF0A58tsbiPoVB+ma8zBDAEHIPQivpcvlGWHUex8/jlKNdyMfUPD4vtQmuRcmLzLfYFCZ2yjcElBz1AYjHfjniq114TjuIrIlrOea2iaNze2YnjkLEMz7dwKtuBOc9yOa6Kiu50oPdHIqk1sznpPCsJ1RbyOLTW+WMOk+niTbsGP3Z3DYMAcc4xT9Q0e8Gm6otrKszzSfabSMR7WjmBDDLFsEblB6DjPWt6il7KGtg9pIrafZrp+nQWqnd5SBSx6se5PuTk/jXoPwv/AOPzWP8Arnb/AM5a4YkAEk4A6k16X8OdImstMuNQuFKNfFfLRhg+WucE/UsT9MVx5hKMcO497HVgVKVdSOxooor5k+hKHhX/AJE7Rv8Arwg/9FrRR4V/5E7Rv+vCD/0WtFdc/iZhH4UMi/5HHUP+vC1/9GXFaVZsX/I46h/14Wv/AKMuK5z4uX95pvw7nudOup7ScXloolgkKMA1xGCMjnBBIPqDWNT4/u/IuHwna0V5hD8RPEDeImEkWmnS08Vt4eMKwSCcgxl1l8zzNoIyoK7OeeRVTR/ih4gu7jS5LuHS5bfUrvUbJYLaKQTRPbK7K5JcgghQGXAxkNnnaM7Mu561RXjOp/EjXtY8K/Z7SSzs7q+8E3GtSS26SeZDKMDEeJAVGN4U8kMAecEHb17UNZtP2cLjUZ9Tc6mNGSZby23wuCVUg5Ls27B5bPJycDOKLBc9LoryTVPiN4nsNU1iGwj0p7TR7vTLXbcwytLcfakjyS4cBSrMedrZB6DGWsD4j679nutNb7G+uQa7NpkX2awLx3SRwiUsEkuYxGdpyS0p+7wDngswuep0V44Pir4p1bQ9Fk8P6daNqd9pKXr24sJ7tZJWm8rbmN18hPldt8hI7Z+Uk7kHjTxhqHjm703SdDhu9N0vUIrK+k2ojbWRWeYM04Zcb8hBE+QPv5PBZhc9HooopDCiiuX+JPia88H/AA/1DXNNFsbm2eBV+1KWjAeZIyWAZTwHJ6jpQI6iiuAvfilo/hvSbGXVta07Xp755TDLo7QwRFIwN3M1wUyCQMeZuJPC8HEuofF3w7p0Wk3EqzvaatFDNDOssCkLK4QEwtKJmwSCxRGAGeuDh2YHdUV51bfFCe1u/Fj69otzFp+h6hFaRTW4iLN5jRIoYeaSWJl3ZAA29fm4rT1P4naTpl5eWjWOpXFxa6nDpXlwRITLPLF5iBcuOD0ycYJ9MmizA7KiuGu/i54dsPFMegXYniuzPDbS5kg/cTSgFUZPM3tjcoLIrICfvcHGYPiRca18Q/CVlocF9b6NqFzfxS3E8MXlX4gifBjILOoDpnkIT7jOCzC56ZRRRSGFFFFABRRXH/EjxZeeEdJ0m4sZ9Ptft2rQWM1zqKForeNwxaQgOnTaDywGM0COworz3QPiZNd6TZi+0ubUtTvLy6hs49JRVS9ggPN0nmyBVjIK4y5JJGM9rT/FbR5NNt7vStO1TVfM019UlgtIoxJbQIxVi4kdRu3K67VLElGwOhLswO4orzOT4oWum+KNdvL/AFDzvD0WnafcWCIiLva4L4IZtuN3y/fYKBycDNaNl8W9E1SxtZNJsr/ULy6vpbGOwtTBJJ5kcfmOd4l8oqEKncHI+YDrkAswud3RXH/CrXtQ8T/DXTdX1idp7y4kuA7tGsZIW4kRQVUADCqB07V2FIAooooGFFFFABRXH/FbXNQ8N/DTUtU0a6a0vIZLdUmSJZCoe4jRsKwIJ2sR0PWsnUPiDJ4ct9NsbCDWfEl/qHnyqdSsZrWVY49ucxw2u/qyhf3WOpLDjLsI9GorzfWfi2dIWwlfQ5NksNrLe20rTJdWXnybMOggaNSM8CSWMsQQB0zTi+IfiHSJ/Gt5q9nbXtjpGqwWltFHdbHQStCgX/UjIxKXJJJz8vT5qLMLnqlFee6t8UZtP1S90+10MXVxb63baNDuvPLWV5ofMVydh2gHAI54yfYuf4m3Y8XPott4Zu7xLW6hs764tFnlEMjqrMykQbDGm8ZLujYydvTJZhc9AoryWDxxq/iX4ieB7i3tX0/Q7661OOLy75mN4sUUigzRBQF+ZdyjL49iOfWqGrAFFFFIYUUVz3iTxhD4c1XStM/svUNTvdV877PDYiLP7pQz5MkiDoeOecUCOhornLP4geF7rw/pmsza3ZWFpqkZktft86QNJjAYAMRkqSAcZ/UVpXniLRdO1O303UNYsLW+utogtZ7pEll3NtXahOWyeBgcnigDRorl9O8d6fPN4gOrvbaRa6LqH2Frm6ulVJDsVgxLABc78YyenWrOveNdC0Lw9/as2q6cyTW7z2SveIgvSq5Cxtn5s5A+XPUUWA36KzvDmr/2/wCFtK1nyPs/9o2UN15O/d5fmIG27sDOM4zgVo0DCiiigAorO17X9N8M6PJqmtTtb2cbIjOsTyHLMFUBUBY5JA4Hes6Lxtpk+vaVpUdvqCz6rHPJA1xZvb7RDt3bllCuM7hj5SDzQI6KiqGt65p/hzSJdU1m4+zWcLIry7GfBdwi8KCeWYDp3oh1vT59eudFiuN2oWsKTzQ7GG1HJCndjByVPAOeKAL9U7rTxLN9ptZPs12BjzAMhx6OP4h+o7EVcopqTi7oTSa1KVtqBaYWt9H9nuuy5ysvuh7/AE6j0q7UVzaw3kJiuYw6E5wex7EHsfcVS8650vi7L3Np2nAy8Q/2wOo/2h+I71dlL4d+3+RN3HfY0qKSORJY1kiZXRhlWU5BFLWZZ4rr+jy6DrEtrKhWFnLW0mPldDyAD6jOCPb3qrY2Fzq18ljYoXmkOCQMiNe7N6Af/Wr2+5tLe9hMN5BFcRHqkqBlP4GmWen2enxmOwtILVGOSsEQQH8AK9eOZyVPltr3PMeXxdTmvoUha23h/RW+wQIJFjWNDj5pH4VQT9SBUbWoYWmixnfFEiyXTf3lB4B92YEn2B9afqdzGt9EJj+5s0N1LjueVQe+TuP1UUyB57WPy1VZNVvT50ik5WIdAW/2VAAHqR9TXEublu9/6/Lf5o63a9lt/X5/5ly9vJBKLOxAa6cZJIysS/3m/oO5/Eiazs47KDy4yWZjueRuWkY9WJ9aSys0s4iAxkkc7pZW+9I3qf8ADsOKsVhJq3LE2Sd7sKKKKgsKKKKACiiigArDvPBfh++uGnn09RIxyxileIE+pCEDNblFVGcoO8XYmUYyVpK5zf8Awr/w1/z4y/8AgZN/8XR/wr/w1/z4y/8AgZN/8XXSUVp9YrfzP72Z+xpfyr7jm/8AhX/hr/nxl/8AAyb/AOLo/wCFf+Gv+fGX/wADJv8A4uukoo+sVv5n97D2NL+VfcYVn4L8P2Nws8GnAyKcqZpXlAPrh2IzW7RRWcpym7ydzSMYxVoqwUUUVJRQ8K/8ido3/XhB/wCi1oo8K/8AInaN/wBeEH/otaK65/EzCPwoZF/yOOof9eFr/wCjLird7YWepWpttRtYLuAsrGKeMOpKkEHB4yCAR6EVUi/5HHUP+vC1/wDRlxWlWFX4vu/I0h8JRGhaSGLDS7IE3X20n7OnNxjHndP9Zjjd196y/DHgTQfCe6TTbON7x3lZr6aKM3BEkhcp5gUHbk8D2FdFRWZRlWXhXw9psskmnaDplo8qukjQWcaF1fG4Egcg7Rn1wPSpYtA0eHRTo8Ok2MemMCpsUtkEJBOSPLxt5Jz0rQooAoPoGjytO0mk2LtcPHJMWtkJlaPHls3HJXA2k9MDFQ3fhbw/qEEsN/oWm3UU1x9qljms43WSbbt8wgjl8cbjzjitWigDhPEnwg8NeJbpJZlks4ltBZi2toLcxpHuZvkEkTGJssfmjKn3yM108nhfQZtTg1K40axn1C3CiK9nt1knTb90+YwLZHrnNalFF2AUUUUDCsjxT4bs/F3hy40XUpJ4ra4eJ2e3YBwY5FkGCQR1QZ46ZrXooEYWveFYdb1TT9Ti1C90vUdPEiQ3dkY92yQAOjLIjqwO1TyuQRwRzWPrXwv0/XryW5vNZ1ZWuobWK9WJoFF4bdy8bv8AuuG3HJCbVOBxXakhQSTgDkk9qwf+E78I+XI//CU6LsiVXkb+0YsIrY2k/NwDkY9cinqBn6h8ONL1I6+st7fxwa88c1zBG0YWOaPZtlQlCwYGNTgkr14qCP4XaWLs3VzqeqXdy2r2+sSTTSRZeeFNighYwApHUAD2wOK6K58TaFZajbafea1p1ve3YQ29tLdoskwY7VKKTlskYGOpoXxLoT6qulprWnNqDOyLaC7Qyll+8Amc5GDkY4xRdgUYvCEdp4lu9Y0zV9RsPt0yT3lnD5LQXDqoXJDxsy5UAHYy5x681m6X8MdK0nXNO1G31DU3TSri4nsLKSWPyLbzw4kVQEDFTvyNzEjaMHGQeksNd0jVLu5tdM1WyvLi0bbcw29wkjwtkjDqCSpyCOfQ1fouwCiiikMKKqatqUOjaLfapdK7wWVvJcSLGAWKopYgZIGcD1FZOleMbS+8Of29qdpPoGmNFFNHc6rPbxo6SY2tlJWC5yow205YDFAjoayPEHhuz8R/2Z9uknj/ALM1CLUIfJYDdJHnaGyDlfmOQMH3pt14w8M2Vla3l74i0m3trwM1tPLfRIk4XGSjFsNjIzjpkVTPxB8OJ4wv/DdxqMFte2FsLmZp540TbtZmAJbOVVdzZAwpB6U9QLOv+FYNdv7DUEv73TNQ0/zFgu7Ix7wkgAdCJEdSDhT0yCowRWJJ8KtHSyt7bStQ1PShFpr6XNJaSRl7qB23EOZEb5tzO25drZduemOnk8QaNDu87V7CPZEkzbrlBtjc7Uc8/dY8A9CeBUD+LvDccdlI/iDSlS/4s2N7GBc87f3Z3fPyccZ5o1Aw7v4W6BcyXLRtd2vnQWcEIt5FAtPspzC0WVOGH+1uBx0q3f8Agj+0o9Pku/EOrtqOmzvNa6motlnjDoUZMCERlSD3TPTmo/FfxB0vwzeWNgk9nealdX9vaPYC8VJollYL5pTlsDIPQZz1FbeqeIdF0MqNa1ew07epdftdykW5QQCRuI4ywH4j1o1DQh8LeGrLwh4ct9E0t53tbd5GQ3D73+eRpCCcDOC5A74xnJ5rXrLh8T6BcLeNb65psq2KeZdlLuNhbpjO58H5RgE5OKbF4s8Oz6b/AGjBr+lyWPm+T9pS8jMXmYzs3A43Y5x1xSA1qKx5fF/huHSYdUm8Q6VHp9w5jhu3vYxFIwzlVfdgkYPAPY1sAhgCDkHkEd6ACiiigZQ1vQ9P8R6RLpes2/2mzmZGeLeyZKOHXlSDwyg9e1Qa34Y0rxBNaTalDN59kzG3uLa6lt5YtwwwEkTKwBGMjODgegrWpGYIhZjgKMk0COZ1L4c+GNXuUuNRsrieVYoYWY39wPOWEkxmUB/3rKSSGfc3vVi88D+Hr+XVnu7F3/thVF8guZVSUqFCtsDbVcbF+dQG461lx/FvwXLbCeLU7h4mt2ukZdNuT5kS/edP3fzBedxGduDnGDWrfeNtB05bSS7uZ1trxYWivEspntsSsFj3TqhjTJK/eYfeBPBFPUNCvb/DvwxbOHj0+RpBfxaiZJbuaR2uY12pIzM5LEDrnIPU5NXH8I6Q3iB9ajjure+lZHma2vp4UnZBhTJGjhJMDj5gcjg8VVT4geGZNbGkrqLfazeNYc20ojFwAT5RkK7AxCkgE844zV/RfEumeIWn/siS4njgdkaZrSWOJyGKnZIyhZMFSMoTRqGhQsvh94Z07XIdXs9OaO8t5pZ4GNzKyQPKGEhSMsUQNvbIUAE4OMgY6SiikAUVXv8AULLSrGS91S7gsrWLBknuJRGiZOBlmIA5IH41Hpesabrdn9q0XUbTULbcU860nWVNw6jcpIzyOKALlcd4v8EyeKvF3hq/kmMdhpf2v7UsV1LbzP5sYVdjR4PUc/MOPXpXY1Bb39pdzXENrdQTy2r+XOkcgZomxnawH3Tgg4PrQB5/r/w8v0vbr/hErbSRY3mgHRDbXkrxraLuYh0Co+8HecqduSoO7nihqPws1hrXV9HsLqyuNN1q306Ce+u5XW5tfsyqjFIwhVyQgZSXXDsevf1aindhY8q1P4X6zc6xeanBc2zs2vtqkNst/Pa+ZE1uIirTRLvjcHJBXI7HqaLv4Y6jDFANBstJtoptGu9KuLO6v57hLbz5fMM0cjRlpCSWJVgmeBnvXo19rmn6bqenafe3Hl3Wpu8dpHsY+YyKXYZAwMKCecVfouwsZPhTSp9C8GaLpF20b3Fhp8FrK0RJRmSNVJUkA4yOMgVrUUUgCiiigZyHxS0HUvEngG40zRYmlvJLm2dQropASdGZgX+XgKTz6dDWJ4q8J+JZPEOkXmlSNrb2Wl6nEbnUjAoMsqIIkdUVAVJBHC9Opr0qoGv7RNQSwe6gW8kjMqW5kAkZAQCwXqQCQM+9O4jwiP4X+ILrwj4usJNAaIXf9m3Gn21yLFN00TE3BRLfEUZK/KCcFlIyeoHR3ngG8PinWb/w/wCHItIbU/Dq22n3Ua28Z0y5CSKVJRiyEqUXdHuGO+BXrdFPmYWPDo/hnqA8EXdnp+i6vaXlxNpxntLuXT44G8qUNI8YtioyADlnwzDb1PT0PwJ4duPDd54lt/sUdjpk+qGfToISgjERijB2op+Qbw3GB645rrqKVwsFFFFIZnyWMtpI0+lFV3HdJascRyH1H91vccHuO9T2d9FeblUNHNH/AKyGQYdPqPT3HBqzVa8sI7va+Winj/1c8fDJ/iPY8GtOZS0l95nyuPwmJ4h8a2WhXBtI4mvLwAFokbaIweRubsT6AE1W0Px/Z6peJaXtu1jNKdsRL70c9l3YGCfcV5nJcS3c8tzcEmWZ2kfPXJOcfh0qN+UPOOOCO3vXuxy2l7Oz37njSzCp7S627HqAupLzVmjii8yZ5fPMbcAY4iDeiqMOfdlHeuksrNbSNiXMs0h3Syt1c/0HoO1UfDNqkeiW92wZrm9iW4nkflizDOPoM4ArXrxq1RN8sdketSg0uaW4UUUVzm4UUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFDwr/yJ2jf9eEH/AKLWijwr/wAido3/AF4Qf+i1orrn8TMI/ChkX/I46h/14Wv/AKMuK0qzYv8AkcdQ/wCvC1/9GXFaVYVfi+78jSGwUUUVmWFFFFABRRRQAUUUUAFFFFABRRRQAyZDJA6DqykDP0ryjSfhNqljoukWc8mmNJY+Hb7S5CrMQZp2yrDKfdxuyeDyeDmvWqKd7CPHdW+F/i6/0bTdJGpWctrbabptuVbUbiGOGa3YGVhEibZw+1cNJgrjgDApNG8Fa9q3iXUzNaQafp1v43bV/tU4kS4mWOMbRGuzayMTjfuGBv4Oa9joo5mFji/A3hXWfDOp6mtzcQRaPNte10+C5knSOUkmWRDIoaJX4byt0gDFiGrtKKKQBRRRQMzPEumzaz4T1fS7VkSe9sZreNpCQoZ0KgnAJxk+hrhr74XXX/CC+HtO0+8le/0m4tbmeOfV7tYpmjjMbrHKCXgHzMVKKAMD5fT0yincR5Lrnwz1+78PppmhLaadaXFpeRXdh/b14UE8zDE5l2b5+C+Y5AFyx65NaMnw71ZrnU4hNZG11bwxHo08plcPBKkci7lTZh1O8dWUjng16TRRdhY8b1P4WeKtb0rVob2TRoJrzRrHTYkjuJXQG3mDszExA4YA4wOCcdtx3fFnw5uNa8X3GpJa2+p6de6fHZT2M2r3WnhAjsR/qFYSKQ/3WHBGR1Nej0UXYWPK9T+GeuT6vOlm+ltp0niS111bi4lkNwojVFaHbsI4Cttbd3wR82R0Xijwdea74x07VoHtRBa6Ze2jLMTu3zKoUgBSMcHPOeehrsqKLsDyaD4YeJbCyh/snU7GwvIfCCaKk8TvlbpZd+8fJwv3vn+8CcgZrI1jwD4j021E4sYrye/8V6deJbJeXF+sMcY2EzTSJvKcAliCAD6AV7hRRzMLHk03w08SDTJrqxuLG116fWLvVYp4L2WNdOeUpgI3lkTqVj+dHjUPvblcc+rxK6wosr+Y4UBnC43HucdqdRRcAooopDCmyp5kLpnG5SM+lOooA86034T/ANn6Vptl/bXmfYdDu9I3/Zcb/PYHzMb+NuPu859RWdqXwTbUINPhOuW7JZ6dYWQkudM86WM2pzugfzB5Ik43DDZ9ea9Wop3YrHluj/DnWLvX9Sm1u6Wz0tPFr65b2ixK8lztRfKfzQ/yKW5Kld3ydRmul8I+B38L65quof2hG8eoBM2lpam3h3rkGZk3splYbQzIEU7c7cmutoouwsFFFFIZxPxisL3VPhNrNnpdrNd3Unk+XDBE0jtieMnCrycAE/QVjePrfxj5ugwT6pL9geS4+3Xei6ZefI+weTuit5/O2/6zkPt3bdy9K9Pop3EeG+J5fGVvJpsOnT69d3Npaae8GoRWN8v28mYmXzI1cQwlUX5lmR2YNjjjE1/oesaTN8Q30aDXo9Sm1C2u7R4HumWa33wNL5bZ2M+A67Rl8ZUfLxXtlFPmCx4vfy+K9Z1a8NoPE1tp134qsUQhLi3ZbFoCspUEBo488k8bTgnDDi7NY+Jm+J00VxrWsadaW95bDThHp97dwT2wRQyPIknkglt4ZplLjO7djGPW6KVwseIaRpuu3XxE8K3mr2viGbUbPVtSOpy3CTmyhRllEBi3fuwpXABj45w3JWtP4h2nim48QeJp9Hk19Y7XQ4JtNWwlnWNroSvnaqHDttxlecgjI4GPXKKdwsePXy+KNJfxVYQQ69eaebzTPs08sl5M0KSKPtMsZidZZFVhkxxuMEkcDIrFtdU163Hg6x8S3fiRLa51TVI5o4Vu4rqe2VC8XyAmcgcEZLOFyMnFe91BNYWdzeW13cWsEtzaFjbzPGGeEsNrbGPK5HBx1FFwseN6d/wmsFr4IuNXj1y+uFjWO509muoSyPM2yaWaFtolSMRF0uBtYFgDu3Y9roopN3AK8p+IOrtoPxLTVUW4Js/Ct9KPswUyLiSPldwKgjrkggYyQRwfVqKFoB4jpfj/AMTHRNd1T+2odTs/D2p2ks8lu0NyLiyeNfPQSpDEG2Es+5UUgqy5bimXvjfxnJp+iSTa3Z6DFrdpdahDe6hLDaxxZkHkW+9oJVO2JlbaQGbn5+OfZdX0qz13R7rS9Uiaazu4zFNGsjJvU9RuUgjPsatRRJBCkUKBI41CoqjAUDgCndBY4DwrrPiDV/iNqNjqesRG103TbGdraxjjaCaWaJt5EjLvKbhuXBB6dsg+g0UVIBRRRQMKKKKAPL/FPg2+stSmu9LtnurOdy+yEFniY8kbepGckY6dKpaJ4O1PWLtFubWayswR50k6GNivdVU85Pr0H6V67RXoxzCsqfJ+JwPA0nPn/ASNFijWONQqKAqgdgKWiivOO4KKKKBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUPCv8AyJ2jf9eEH/otaKPCv/InaN/14Qf+i1orrn8TMI/ChkX/ACOOof8AXha/+jLitKs2L/kcdQ/68LX/ANGXFaVYVfi+78jSGwUUUVmWFFFFABRRRQAUUUUAFFFFABRRQDnpzQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFGcdaKACiiigAooooAKKKKACiiigAooooAKKKKACiignHWgAooooAKKKKACiignHXigAooooAKKKKACiiigAooooAKKKKACiiigAooooAoeFf8AkTtG/wCvCD/0WtFHhX/kTtG/68IP/Ra0V1z+JmEfhQyL/kcdQ/68LX/0ZcVpVmxf8jjqH/Xha/8Aoy4rSrCr8X3fkaQ2CiiisywooooAKKKKACiiigAooooA5fxbZ+I7mI/2RcL9mx80MXySH8e/4Y+hqfw9bavH4fs0FzbQ7Y8eXLaOWXnoT5g5/AV0NFdH1h+zVOy08jD2K9o53f3lDydX/wCf6x/8An/+O0eTq/8Az/WP/gE//wAdq/RWXtH5fcjTkX9NlDydX/5/rH/wCf8A+O0eTq//AD/WP/gE/wD8dq/RR7R+X3IORf02UPJ1f/n+sf8AwCf/AOO0eTq//P8AWP8A4BP/APHav0Ue0fl9yDkX9NlDydX/AOf6x/8AAJ//AI7R5Or/APP9Y/8AgE//AMdq/RR7R+X3IORf02UPJ1f/AJ/rH/wCf/47R5Or/wDP9Y/+AT//AB2r9FHtH5fcg5F/TZQ8nV/+f6x/8An/APjtHk6v/wA/1j/4BP8A/Hav0Ue0fl9yDkX9NlDydX/5/rH/AMAn/wDjtHk6v/z/AFj/AOAT/wDx2r9FHtH5fcg5F/TZQ8nV/wDn+sf/AACf/wCO0eTq/wDz/WP/AIBP/wDHa8m8N+Jb7R/DnizxNeWOu6lNpt1qHlXFxqjNaMqSkLEITMduB38rgA81tr428Y2WgTT67pcNpPdXNpb6RNHaxyfbHmOCnkreEDGMh2mUEHpkYNc0vL7kLlXn97O+8nV/+f6x/wDAJ/8A47R5Or/8/wBY/wDgE/8A8dryey8YeKfE/iTwJOl5Dppnv9TtLuBIXMUxhR+XRZsHheBuYK3zBmHFbOn/ABL1S6vvDttNDYK+qapqdnMgRgyx22/aVG7g/Ku4nI56DIo5peX3IOVef3s7/wAnV/8An+sf/AJ//jtHk6v/AM/1j/4BP/8AHa8ysviV4su/B3ha/WysJ9R8TTOsMdtbEpbqiMxBElxGHZtmQN6YyR85A3a3jXWPEcXwC1TU9VgfQ9eS0JlW0nwYnEgXcrIxxuGGwGON2MnFHNLy+5Byrz+9nb+Tq/8Az/WP/gE//wAdo8nV/wDn+sf/AACf/wCO14H4y8WeI4/B+oeHLXV9TttS8JGeW/v453SWdBIqWm5wctvSXceeTHz1r0HXPiPqWneP7bTLGKG50s6vbaVcs9rsMUsybsCUz5ZgDuwISuMjdmjml5fcg5V5/ezu/J1f/n+sf/AJ/wD47R5Or/8AP9Y/+AT/APx2vNdF+KOu3V3pUt/FpMlpqN5qFn5EAeOSFrYO6yM7OVClUAYFeM7s4O0dX8PfE2peIdOu4/EXlQazYyrHeWUdk0H2ZiucbjLIJVPVXUgFSDgUuaXl9yDlX9Nm/wCTq/8Az/WP/gE//wAdo8nV/wDn+sf/AACf/wCO1fope0fl9yHyL+myh5Or/wDP9Y/+AT//AB2jydX/AOf6x/8AAJ//AI7V+ij2j8vuQci/psoeTq//AD/WP/gE/wD8do8nV/8An+sf/AJ//jtX6KPaPy+5ByL+myh5Or/8/wBY/wDgE/8A8do8nV/+f6x/8An/APjtX6KPaPy+5ByL+myh5Or/APP9Y/8AgE//AMdo8nV/+f6x/wDAJ/8A47V+ij2j8vuQci/psoeTq/8Az/WP/gE//wAdo8nV/wDn+sf/AACf/wCO1foo9o/L7kHIv6bKHk6v/wA/1j/4BP8A/HaPJ1f/AJ/rH/wCf/47V+ij2j8vuQci/ps5bxZaazNoTIs0NwxlTalvaur5z1B3n+VW/DFtr1va41y5jkXHyIfmkX6t/wDr+tb1Fauu3S9nZfcZqilU9pdhRRRXObhRRRQAUUUUAFFFFABRRXlWhXt9N8RPGE00GuXw06+Itpl1NhZ2wFsrbHg85d2Se0bcsOR2Yj1WivD/ABJ8QtR8QeBZdPuo7FRqfgifVrgQqwZJchAFyxwnLdcnI68U3RfEN54V1/xTq9vHBJYQ3GhpqZlVi0ds9miNImCOVLKTnIwGp8oXPcJA5jYRMqvj5Sy5APuMjNcHPp3ij/hKrOW/m86ITfu5UUtCn1QEY/H86F8f6hF8HtQ8b3FlC5CTXNjborLmDeVhL5JJJXaxIxweAKfeeJPEltrGh+HlvNEk1HU5bgtqCW0jwRpFCkmzyfNDB28wY/eY2ruxzgbUarpXsk7mNWkqlrt6HWeTq/8Az/WP/gE//wAdo8nV/wDn+sf/AACf/wCO15bb/FPxNqun2MmnQaTazP4fu9VnaaCWZHkt5TGRHiRTsfG5SckAj73d2p/FbxBHaanqVhbaallpuj6fqrwTQyPJKLjl4w4cBSB0baenIOeI5peX3Ivlj5/ez1DydX/5/rH/AMAn/wDjtHk6v/z/AFj/AOAT/wDx2uJTxr4t1Hx9e6doegx3OlaZqUVleM3lq4RkVml3tOpGA+QoiYMF+8CeOb8T+M/E+tfBDVvEdrqFjpq3McktpFZrKl3bpHc+WR5gk+ZsbdzAKFORtO7IOaXl9yDlXn97PWvJ1f8A5/rH/wAAn/8AjtZviG21eTw/eIbm2m3R48uK0cM3PQHzDz+BrldW8ZeN4PFdzoOgaPa6vc6ZbW090ViSFbnzGbOzzLlTEAqkA4m+brjjPP8AivxJqniDxJpAmfT4NN0/x3b6fDaiNvtRaMZ8xn37Sp3MQoQcFTuPeoVJRknp9yJlTUota/ez0PwlZ+I7aIf2vcL9mx8sMvzyD8e345+grqK8t8c61H4d+Kttq80qwpZ+GL2XzGiMoUiWPB2AqW5xxuXPqOogtvid4kt7TVJ9YsLOOLQ9VtIdRYw+W62k6LlwqTyqGRmBJ3sCuchSDSqzdWXO0l6DpwVOPKnc9ZoryJfiZ4w1VNNh8PaNazXWq211qNoDCCDbJLshUiSeL5ipVmYMcAjCdcdN4f8AFHiHWfH17o97a2OnW2nWNpc3VuVaWcSTRsTGJA+z5XX7205Ax3yMrGlzt6KKKQwooooAKKKKACiiigAooooAoeFf+RO0b/rwg/8ARa0UeFf+RO0b/rwg/wDRa0V1z+JmEfhQyL/kcdQ/68LX/wBGXFaVZsX/ACOOof8AXha/+jLitKsKvxfd+RpDYKKKKzLCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAK1vplhaWstta2VvDBMzvLFHEqrIz8sWAGCTnknrWdF4L8LwafPYQeG9Ijs7llae2SwiEcpXlSyhcEjtnpW1RQIyh4W8Piyt7MaFpgtrabz4IPsceyKXrvVcYVueo5pY/C+gRao+pRaHpqX8khke6W0jErOQQWL4yThmGc9CfWtSigDNm8N6HcaNHpFxouny6bEQY7J7VGhTHTCEbR1PbvUn9iaT/Yv9kf2ZZ/2Zs2fYvs6eTtznGzG3GecYq9RQBn3GgaPd/bftWk2M/9oBFvPMtkb7SE+4JMj59vbOcdqZP4Y0G61hdWudE06bUlKst5JaRtMCuNpDkbuMDHPGK06KAOc8M+A9A8Kq76fZRyXcjys17PFGbgrI5dkLhQduTwPYVraVoul6FatbaJptnptu7mRorOBYkZiACxCgDOABn2FXaKACiiigYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABUFvYWlpNcTWtrBBLdP5k7xxhWlbGNzEfeOABk+lT0UAY0Hg7wxbLMtt4c0mEXCPHMI7GJfMV8b1bC8htq5B64GelN1fwjpOraHqumLbpYJq0At7uezijSV0C7ByVIOF+UZBwOmK26KBFaHTbODSY9MS3jNlHALcQMoK+WF27SDwRjjFUX8I+G5NIj0qTw9pT6dFJ5sdm1lGYUfn5gm3aDyecZ5Na9FAFH+wtJ+T/iV2X7u1Nmn+jp8sBxmIccJwPl6cdKjbw3obwTQto2ntFPClvLGbVNskSfcjYY5VewPA7VpUUAZtx4b0O71iPVrrRdPn1KIqY7yS1Rpk29MORuGO3NInhnQYvt/l6Jpyf2nn7dttIx9ryST5nHz5LN97PU+tadFAGPP4P8ADV0toLnw7pMwskCWoksY28hQcgJlflGecCpZPDWhTas2qS6Lpz6gzIzXjWiGYlMbSXxnI2rjnjA9K06KAKd3pGm38zS32n2tzI0D2zPNArkxP96Mkj7p7r0NZV54G0CbwvqWg6fp9vo9pqUJhuDplvFCxUjB/hIzgkZIOM10NFAGTc+FNAvtMs9P1HRbC+tLFBHbRXdskyxAKFG0MDjgAVcttL0+yuJJ7OxtreaSNI3kihVWdEGEUkDJCgkAdu1WqKACiiigYUUUUAFFFFABRRRQAUUUUAUPCv8AyJ2jf9eEH/otaKPCv/InaN/14Qf+i1orrn8TMI/ChkX/ACOOof8AXha/+jLitKs2L/kcdQ/68LX/ANGXFaVYVfi+78jSGwUUUVmWFNSVJHkVGDNE21wP4TgHH5EH8adVDT+NQ1Uf9PKn/wAgx/4VSV02S3Zov0UUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRVbUdTsdIsXvdWvbextY8B57mVY41ycDLMQBkkCm6bqunazZi70e/tb+2LFRPazLKhI6jcpIzQBbooqC2v7O9edLO6guGtpTDOsUgYxOACUbH3WwQcHnkUAT0UUUAFFZmreJ9B0GWOPXdb07TZJQWjW8u44S4HcBiM1pI6yRq8bB0YZVlOQR6igQtFQWN/Z6nZpd6bdQXds5YLNbyCRGIJU4YcHBBB9wanoGFFQXt/aadb+fqF1BaQ7lTzJ5Ai7mOFGTxkkgAdyagtNZsL7VtQ0y1n8y800xi6j2MPL8xdyckYORzwT70AXqKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAprypG8auwVpW2oD/ABHBOPyBP4U6qGo/8f8ApQ/6emP/AJBk/wAaqKuyZOyLN3ci0hWRlLBpY48D1dwuf/HqmqhrX/IOB9LiA/lKhq/Q0uVP+ugX95oKKKwJfHXhaOxmvE16xuYLeaOCZrSYXHlySNsRWEe4jLcc+/pUjN+iiigYUViX3jHQdNOri9vxEdGSJ77MTnyhL/q8YHzFuwXJzx1qxo/iHTdee6TTZpHks3VLiOW3kheJmQOFKuoIO1gcduhweKBGnRRWVf8AijRNMs7i6u9Sg8q1uEtZzEfNMUrsqqjKmSCS68Ed89KANWiiigYVDBciea5jCkG3lEZJ7korZ/8AHv0qaqGm83mqH1ux/wCiYx/Sqik0yW7NEunXD3Nq7y43LPNGMDHCyso/QCrVUNH4trgel3P+sjH+tX6c1aTsKGsUFFcZffE7S9P8QX2m3OnakIdPu4LO61AJEYIpJlUx5HmeYQdwGQhwa3/+Eo8P/wCl/wDE803/AEJS11/pcf7gBipL8/KAwI5xyCKixRqUVyOufEbR9Jh0S7truxvtN1S+No9/HeoIbcBHZnLjKnGzBGR9a1rnxf4asrG1vbzxDpVvaXgJtp5b2NY5wOuxi2GxkdKLMDYorlD8QNLt/HV94c1Sez0/7PDbyQXFxeKn2ppt2EVTjkbexOc9KrX3xAksNdg06TTIHE+uJpKPHqCOwDQmTzWRQShyCNjYPfODTswO0orm9I8b6ZdaPp11rV7pelXWoySRwWx1SGYSskhTCOpw5zjIXoTg81mWnxS0fWInbw+9tdtBrKaXcJPqEMJVWk2eenzHep5KAYL4wKLMDt6KpWus6Xfahc2NjqVnc3locXFvDOryQn/bUHK/jV2kMq3lw8N1YRpjbPOY3yO3lO381FWqoX/Oo6WPS4dv/ITj+tX6uS0X9dWRF6v+uhQ8K/8AInaN/wBeEH/otaKPCv8AyJ2jf9eEH/otaK6J/EzOPwoZF/yOOof9eFr/AOjLitKs2L/kcdQ/68LX/wBGXFaVYVfi+78jSGwUUUVmWFULLjVtSX1eNv8AxwD+lX6ypbu307XZnvbiK3juII9jSuFBZS2Rk98MtaQTaaXb9UZzdrNmrRVD+3tI/wCgrY/+BKf40f29pH/QVsf/AAJT/Gl7OfZj9pDuX6Kof29pH/QVsf8AwJT/ABo/t7SP+grY/wDgSn+NHs59mHtIdy/RVD+3tI/6Ctj/AOBKf40f29pH/QVsf/AlP8aPZz7MPaQ7l+iqH9vaR/0FbH/wJT/Gj+3tI/6Ctj/4Ep/jR7OfZh7SHcv0VQ/t7SP+grY/+BKf40f29pH/AEFbH/wJT/Gj2c+zD2kO5foqh/b2kf8AQVsf/AlP8aP7e0j/AKCtj/4Ep/jR7OfZh7SHcv0VQ/t7SP8AoK2P/gSn+NH9vaR/0FbH/wACU/xo9nPsw9pDuX6Kof29pH/QVsf/AAJT/Gj+3tI/6Ctj/wCBKf40ezn2Ye0h3L9FUP7e0j/oK2P/AIEp/jR/b2kf9BWx/wDAlP8AGj2c+zD2kO5y/wAX9PvtT+G9za6TbS3N013aMkcULSni4jJO1eSAASfYHpWJ49tvGD3ui29xq1xHYNFcfarzRNMvCFm+Xyt0VvP5oG3dg7ymfvL0x6H/AG9pH/QVsf8AwJT/ABo/t7SP+grY/wDgSn+NPkn2YueHc8g8TTeMotU06Kwk1yee0g00xX0NlfAahmQmdnjVxBDtUfMkqMx3Y44wHRdY0NfGraXb6/HfHxDBeRmFrp1nsy8LOUOSsjYDghcvgFTxgV6//b2kf9BWx/8AAlP8aP7e0j/oK2P/AIEp/jT5J/ysOeHc8mnbxXrN1J5X/CTW1jd+M4QMC4gddOa2IYjo0cRb6bWI+6w4vQWPidvidML7WtXsILXU4/sUSafe3Nvc2YjUbGlWTyF3ZcM0iFw3zbugHpf9vaR/0FbH/wACU/xo/t7SP+grY/8AgSn+NHJP+Vhzw7nA+NYLuH4taLqSya5ZWcekzxPe6TprXZDmRCI2HkygZAJ+6Dx1FU7geI7j4tZQeIx4QmvkSQK08ZW5+ztyvR1tfmGTwokGeBXpX9vaR/0FbH/wJT/Gj+3tI/6Ctj/4Ep/jRyT/AJWHPDueM+FfD2sWXhvwposUfiSxZdduo9ZWOW7iURZmKkPkARsCh3IQCxzndzWp9p8W/wDC2oGtbbV7e0XU54JYHjvZbd7cQHy5WleQwYZ8YCIrKeC2evqX9vaR/wBBWx/8CU/xo/t7SP8AoK2P/gSn+NHJP+Vhzw7njOoaNqut/CbbqEHiq51yC5tJ9ThuDdYMq3H7wwr/ABAKWIEWUACkDcAa1NS/4SdNW8QDTv8AhIv7CF1owTH2kzfYvL/0jyd37zf93ft/eZzn5q9S/t7SP+grY/8AgSn+NH9vaR/0FbH/AMCU/wAaOSf8rDnh3PKraz8T3t1pdskviVNCm8STiJpZrmK5Fj5B4lckSqnm7tpchsEcj5a7r4bHVv8AhB7ePxB9s+2RXFzFuvg3nNGs7iMsW5PyBcE9Rg89a2/7e0j/AKCtj/4Ep/jR/b2kf9BWx/8AAlP8aXJP+Vhzw7l+iqH9vaR/0FbH/wACU/xo/t7SP+grY/8AgSn+NL2c+zH7SHcv0VQ/t7SP+grY/wDgSn+NH9vaR/0FbH/wJT/Gj2c+zD2kO5foqh/b2kf9BWx/8CU/xo/t7SP+grY/+BKf40ezn2Ye0h3L9FUP7e0j/oK2P/gSn+NH9vaR/wBBWx/8CU/xo9nPsw9pDuX6Kof29pH/AEFbH/wJT/Gj+3tI/wCgrY/+BKf40ezn2Ye0h3L9FUP7e0j/AKCtj/4Ep/jR/b2kf9BWx/8AAlP8aPZz7MPaQ7l+iqH9vaR/0FbH/wACU/xo/t7SP+grY/8AgSn+NHs59mHtIdy/RVD+3tI/6Ctj/wCBKf40f29pH/QVsf8AwJT/ABo9nPsw9pDuX6Kof29pH/QVsf8AwJT/ABo/t7SP+grY/wDgSn+NHs59mHtIdy/RVD+3tI/6Ctj/AOBKf40f29pH/QVsf/AlP8aPZz7MPaQ7l+qF9zqumr6SO3/kNh/Wj+3tI/6Ctj/4Ep/jUMd3b6jrkDWVxFcR28Ehd4nDBWYqFGR7BquMJRd2uj/ImUotWT7E2ucaNO39za/5MD/Sr9V7+2+2abc2wODNEyA+mRiqyaheCNfM0e8L4G7a8OM+37ypS5oWQ27Suyj4703UdZ+H+t6dojbb+6spIoPn27mK/dyememfeuD8TFdf+GlhZ6H4Z1WC6s201Hhl0qWFoEWaMmJdyguFCkkpuUBc56Z9M/tG6/6A19/33B/8co/tG6/6A19/33B/8coUGu33oOdf0meUw3fi9/itHJFb63a2j3WoRXECwXcsIiWFzBJ5kshtzvcKVCKmD8pPPNONfFC/D6WBW8QGew1JcXrpqZfV0EJOGjLfaLcElctHmPeDj5c49i/tG6/6A19/33B/8co/tG6/6A19/wB9wf8Axynyvy+9BzLz+5njfifS/EniDSfFMV5puvJCY9BlsbH7RcSiN9w+0KrA/vCoJ3nnlQ55AI0dRh8T2mp6rbf8VG2hR6/ZxyNBJcyT/YRbDcYnUmVl8wLvMZLH5s/xV6n/AGjdf9Aa+/77g/8AjlH9o3X/AEBr7/vuD/45Ryvy+9BzLz+5nkfiy58UDXLNdAg8Rwx2qae1pK0V7M12hlJl83Y4hQqgG4TIztu5wekN9ous6N/wnQ0aDxBHqdxrdrPbSW7XbJLbM8BdlYEoxxvDYJYKCDhRivYv7Ruv+gNff99wf/HKP7Ruv+gNff8AfcH/AMco5X5feg5l/SZ5nAPFD+NIix18asPEj+du+0DTv7L2nGP+WH3NuMfvN/XnNM8B23iq21Lwheao+vu96NTj1ZL6Wd40VJP9HLI52xnCjacAsCeTmvT/AO0br/oDX3/fcH/xyj+0br/oDX3/AH3B/wDHKOV+X3oOZef3Mv1Q0rlr5/712/6AL/Sj+0br/oDX3/fcH/xypNLgkt7ECddsskjyuuc7SzlsfhnH4UW5Yu4X5pKxHpfyyX8f9y7b9VVv/Zqv1QksrtLyaewuoIhPtLpNAZPmAxkEOvYD8qPJ1f8A5/rH/wAAn/8AjtElGTvf8/8AIUW4q1vyOUsfhxbv491/X9dVLuK8vLe6sYFupdiNFEqbpIsiNmDDKkhsdcg1nJ8NtUh8L31pFNY/bX8UHXoVLMIp1E4kWOVtuQSqgEgNjA64rtLy5vNPiSS/1jS7VJJFiRprYoGdjhVBMvJJ4A70tpcXt/G72OsaXcpHI0TtDbFwrqcMpxLwQeCOopWX8y/H/Id32/L/ADOAufhx4hmvoNZUaONQ/wCEj/tqWxaeQ26qsHlqiyeXkuSAxcoME5wdvNO8+F3iw+GF0y01K0KXKag11aJqFxawxS3L7kZWjTdKiAsNjhVbOSOePUvJ1f8A5/rH/wAAn/8AjtULzVzpzzJqHiLRbVreNZZlni2GNGbarNmbgFuATwTxTsv5l+P+QXfZ/h/meean8KfEV3DqNpBJo7Q6potjpUs00km+38n/AFkiKI8N04UlcnBJGMHTf4a6ufFX9pLd2bQDxLDq3zyOZDClt5RB+XG8tzjOMd+1d/5Or/8AP9Y/+AT/APx2jydX/wCf6x/8An/+O0WX8y/H/ILvs/w/zPNNI+GPiXRdP8Orp95Y2upabLIJr+C7lH7iS6kleIxGMpOrI44PllXQEMRVqX4c640N5ZpLp/2d/Fqa/FMZ33tGZfMeNk8vAYYwMMQc87e/dWtxe33nfYtY0u48iVoZfKtmfy5F+8jYl4YdweRU/k6v/wA/1j/4BP8A/HaLL+Zfj/kF32f4f5nPeBfDOq+GIG0/UE02a2t5bp7e+idmuJhNOZfnBUBMcAgM27C9NvPYVQ8nV/8An+sf/AJ//jtHk6v/AM/1j/4BP/8AHaXKv5l+P+Qcz7fl/mF382t6en91ZZPyAX/2ar9UrazuBe/ar64imkWMxoIoTGFBIJ6s2ScD8qu0p20S6DjfVsoeFf8AkTtG/wCvCD/0WtFHhX/kTtG/68IP/Ra0V0T+JmcfhQyL/kcdQ/68LX/0ZcVpVl3VrqkOvz32nW9ncRT20UJE9y0TKUaQ9o2yD5g9OlO83xD/ANAzTP8AwYyf/GKznByd0VGSSszSorN83xD/ANAzTP8AwYyf/GKPN8Q/9AzTP/BjJ/8AGKj2Uv6aK54mlRWb5viH/oGaZ/4MZP8A4xR5viH/AKBmmf8Agxk/+MUeyl/TQc8TSorN83xD/wBAzTP/AAYyf/GKPN8Q/wDQM0z/AMGMn/xij2Uv6aDniaVFZvm+If8AoGaZ/wCDGT/4xR5viH/oGaZ/4MZP/jFHspf00HPE0qKzfN8Q/wDQM0z/AMGMn/xijzfEP/QM0z/wYyf/ABij2Uv6aDniaVFZvm+If+gZpn/gxk/+MUeb4h/6Bmmf+DGT/wCMUeyl/TQc8TSorN83xD/0DNM/8GMn/wAYo83xD/0DNM/8GMn/AMYo9lL+mg54mlRWb5viH/oGaZ/4MZP/AIxR5viH/oGaZ/4MZP8A4xR7KX9NBzxNKis3zfEP/QM0z/wYyf8AxijzfEP/AEDNM/8ABjJ/8Yo9lL+mg54mlRWb5viH/oGaZ/4MZP8A4xR5viH/AKBmmf8Agxk/+MUeyl/TQc8TSorN83xD/wBAzTP/AAYyf/GKPN8Q/wDQM0z/AMGMn/xij2Uv6aDniaVFZvm+If8AoGaZ/wCDGT/4xR5viH/oGaZ/4MZP/jFHspf00HPE0qKzfN8Q/wDQM0z/AMGMn/xijzfEP/QM0z/wYyf/ABij2Uv6aDniaVFZvm+If+gZpn/gxk/+MUeb4h/6Bmmf+DGT/wCMUeyl/TQc8TSorN83xD/0DNM/8GMn/wAYo83xD/0DNM/8GMn/AMYo9lL+mg54mlRWb5viH/oGaZ/4MZP/AIxR5viH/oGaZ/4MZP8A4xR7KX9NBzxNKis3zfEP/QM0z/wYyf8AxijzfEP/AEDNM/8ABjJ/8Yo9lL+mg54mlRWb5viH/oGaZ/4MZP8A4xR5viH/AKBmmf8Agxk/+MUeyl/TQc8TSorN83xD/wBAzTP/AAYyf/GKPN8Q/wDQM0z/AMGMn/xij2Uv6aDniaVFZvm+If8AoGaZ/wCDGT/4xR5viH/oGaZ/4MZP/jFHspf00HPE0qKzfN8Q/wDQM0z/AMGMn/xijzfEP/QM0z/wYyf/ABij2Uv6aDniaVFZvm+If+gZpn/gxk/+MUeb4h/6Bmmf+DGT/wCMUeyl/TQc8TSorN83xD/0DNM/8GMn/wAYo83xD/0DNM/8GMn/AMYo9lL+mg54mlRWb5viH/oGaZ/4MZP/AIxR5viH/oGaZ/4MZP8A4xR7KX9NBzxNKis3zfEP/QM0z/wYyf8AxijzfEP/AEDNM/8ABjJ/8Yo9lL+mg54mlRWb5viH/oGaZ/4MZP8A4xR5viH/AKBmmf8Agxk/+MUeyl/TQc8TSorN83xD/wBAzTP/AAYyf/GKPN8Q/wDQM0z/AMGMn/xij2Uv6aDniaVFZvm+If8AoGaZ/wCDGT/4xR5viH/oGaZ/4MZP/jFHspf00HPE0qKzfN8Q/wDQM0z/AMGMn/xijzfEP/QM0z/wYyf/ABij2Uv6aDniaVFZvm+If+gZpn/gxk/+MUeb4h/6Bmmf+DGT/wCMUeyl/TQc8TSorN83xD/0DNM/8GMn/wAYo83xD/0DNM/8GMn/AMYo9lL+mg54mlRWb5viH/oGaZ/4MZP/AIxR5viH/oGaZ/4MZP8A4xR7KX9NBzxNKis3zfEP/QM0z/wYyf8AxijzfEP/AEDNM/8ABjJ/8Yo9lL+mg54mlRWb5viH/oGaZ/4MZP8A4xR5viH/AKBmmf8Agxk/+MUeyl/TQc8TSorN83xD/wBAzTP/AAYyf/GKPN8Q/wDQM0z/AMGMn/xij2Uv6aDniaVFZvm+If8AoGaZ/wCDGT/4xR5viH/oGaZ/4MZP/jFHspf00HPE0qKzfN8Q/wDQM0z/AMGMn/xijzfEP/QM0z/wYyf/ABij2Uv6aDnieN+PtevtR8S3GnajrLW5sPE+nR2uiiGP95bZjb7QTt8zl2xu3bR93GSKSTxpq+j6XNcxahHZ2kfiLUorm30+K2gurlFmVUEIkiMcrjcxZeJHyCCcGvZfN8Q/9AzTP/BjJ/8AGKPN8Q/9AzTP/BjJ/wDGKr2cu34oXOv6TPGNU1nUfCN/8UNQ03XpI9UiurSeC0uI4DujcwDzdmwMQA3lg5xg85b5qb8VXvrc+JtInujqDR+HrCQ3MlpCJ5nN/jlo0UkeiDj2yST7T5viH/oGaZ/4MZP/AIxR5viH/oGaZ/4MZP8A4xRyS/poOZf0meVJ4/1i3a8mt/Ea6noFn4h06A668UAja3mQefGzogQKrFfmABG/BPFZOp69e+Jf7H1jUPEUtpp9n41uraC+jSBI4bfY3kvueMj7vALZBD85OCPa/N8Q/wDQM0z/AMGMn/xijzfEP/QM0z/wYyf/ABij2cu34oOdf0meT2fi3ULXVru0udWg8P6TP4s1O2utVit7eHyhHErxIWdDHudt2XcFm24zk0y08ZeMNVtdJjk1qawabw9fagZ4bOENdGGUCGba6MFDptYgAAhjjHGPW/N8Q/8AQM0z/wAGMn/xijzfEP8A0DNM/wDBjJ/8Yo9nLt+KDnX9XIvCWpz634K0TVbsKLi+0+3uZQgwNzxqxx7ZNa9Zvm+If+gZpn/gxk/+MUeb4h/6Bmmf+DGT/wCMVPspf00HOjSorN83xD/0DNM/8GMn/wAYo83xD/0DNM/8GMn/AMYo9lL+mh88R/hX/kTtG/68IP8A0WtFWNFspNN0DT7Gcq0ttbRwuUOVJVQDjPbiit5ayZnHSKLtFFFSUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB/9k=)

* A ConvNet architecture is in the simplest case a list of Layers that transform the image volume into an output volume (e.g., holding the class scores) There are a few distinct types of Layers (e.g., CONV/FC/RELU/POOL are by far the most popular)
* Each Layer accepts an input 3D volume and transforms it to an output 3D volume through a differentiable function
* Each Layer may or may not have parameters (e.g., CONV/FC do, RELU/POOL don’t)
* Each Layer may or may not have additional hyperparameters (e.g., CONV/FC/POOL do, RELU doesn’t)

The data that is used for this assignment is about the images of natural scenes around the world. This is challenge dataset that can be used for more classification of images. All the images are of size 150x150 divided into 6 categories. The categories are buildings, forest, glacier, mountain, sea, street. So, when it is provided to me, it is in the directory format which is the images in the folders, in total 3 folders for train, test, and predict of the classes. When I tried to import that file into the python, it interpreted some metadata and unnecessary information that is causing issues to the read images function provided to read the data. So, I have uploaded the zip file into the drive, and then unzipped into folders into a directory in drive. Then, for reading the data and checking the labels, I have created a custom column.

The bigger picture in this assignment is predict the which category of image based on the features such as edges in the images provided by the user.

### **PART II: Exploratory Data Analysis (Data Pre-Processing, Data Visualizations)**

As this assignment is dealing with images, I made use of Google Colab Pro for activating GPU with High-RAM. To check the GPU in system, I executed this block of code…

Graphical user interface, text, application, email

Description automatically generated

To start, I decided to use the “Google Colab” as my programming IDE for the tasks. Also, I decided to store my dataset on the google drive. So, the first step is to import various python modules which can be used to better understand the data. Then, I mounted the google drive for accessing the dataset.

Graphical user interface, text, application, Teams

Description automatically generated

To read the contents of the folders as per classes in the folder, I have used the glob module in python to count the number of images in each folder. For training and testing folders we have classes folders, but the prediction folder has only images.

Text

Description automatically generated with medium confidence

For reading the images and loading into the model, I have used the OpenCV module in python. I have created the classes and labels variables that stores the categories and the label encoding for the classes, then opening each folder, I have read the image then converted that by resizing the 150x150 to 50x50 size. Then, as the machine reads only numbers, I have used the NumPy module to convert the images into array of integers. So, the total number of images in train data are 14034 and test data are 3000 images.

Graphical user interface, text

Description automatically generated

As we are dealing with images, there’s nothing much to do for EDA in the model, so I have plotted some sample images in train and test data. Then, I have plotted the division, percentage of all classes in train and test data. Also, the pie plot for all classes in training data to check the imbalance in dataset.

For train data…

Graphical user interface, text, application, email

Description automatically generated

A picture containing window, indoor, surrounded

Description automatically generated

For test data…

Graphical user interface, text

Description automatically generated

A picture containing indoor

Description automatically generated

For number of images in train and test data…

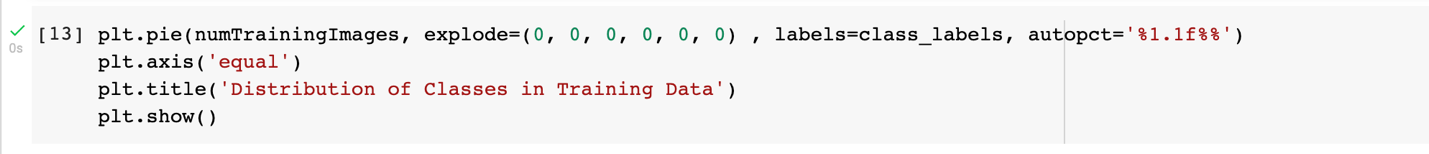
Timeline

Description automatically generated with medium confidence

Icon

Description automatically generated

Now, we will go through the class imbalance in the train data...



A picture containing chart

Description automatically generated

For scaling the data into similar numbers in the arrays of images, I have divided the data by 255 as we will be having the 0 to 256 values in RGB of images.

Graphical user interface, text, application

Description automatically generated

As we can see that from the images of the both the train and test data, there is not much imbalance in the classes, so we can proceed to model training and evaluation.

### **PART III: Model Training (Splitting the Data, Applying the Model)**

For model training regarding the CNN, I have used TensorFlow and Keras modules in python to import all the sub modules to build the network.

Sequential - A sequential approach is suited for a simple stack of layers with one input tensor and one output tensor for each layer. In Keras, the simplest technique to build a model is sequential. It allows you to layer-by-layer construct a model.

Conv2D - These layers make up our initial two layers. These are convolution layers that will deal with our 2-dimensional matrices as input images. The number of nodes in each layer is 200 in the first layer and 150 in the second layer. Depending on the size of the dataset, this value might be modified higher or lower.

Kernel Size - For our convolution, the kernel size is the size of the filter matrix. A 3x3 filter matrix will result from a kernel size of 3.

Activation Function - The layer's activation function is called activation. The ReLU, or Rectified Linear Activation, is the activation function which is used for our first two layers. In neural networks, this activation function has been shown to be effective. In neural network models that predict a multinomial probability distribution, the SoftMax function is utilized as the activation function in the output layer.

MaxPooling2D - For 2D spatial data, the maximum pooling operation is used. It down samples the input along its spatial dimensions by obtaining the maximum values for each channel of input across an input window. Each dimension of the window is adjusted by steps.

Flatten - Flattening is the process of turning data into a one-dimensional array for further processing. To construct a single continuous feature vector, we flatten the output of the convolutional layers. It's also connected to the final classification model, which is referred to as a fully connected layer.

Dense - Dense Layers are used to identify images based on convolutional layer output. Each layer of the Neural Network contains neurons that compute the weighted sum of the input and send this through a non-linear function known as an "activation function."

Adam Optimizer - Adaptive Moment Estimation is a technique for optimizing gradient descent algorithms. When working with massive problems with a large dataset or parameters, the method is quite efficient. It is efficient and takes minimal memory.

Sparse Categorical Cross Entropy - Categorical cross-entropy is computed by both Sparse Categorical Cross Entropy and Categorical Cross Entropy. Only the way the targets/labels should be encoded differs. The targets are represented by the category index when utilizing Sparse Categorical Cross Entropy (starting from 0).

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

For visualization of the model, I have used the Keras utils plot model functionality to export the model along with layer names, activation functions, and also the shapes of images into that layer of model.

Table

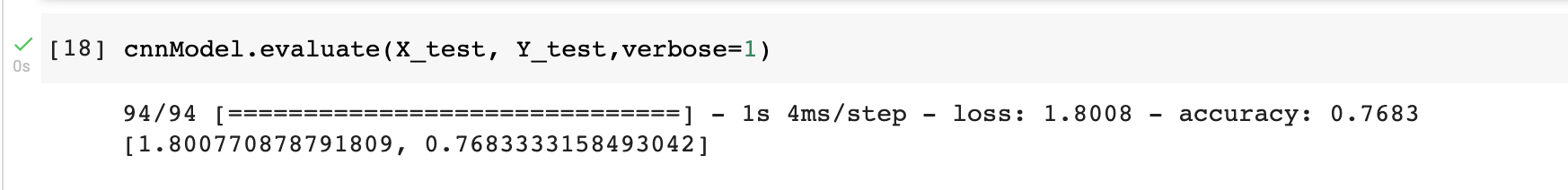
Description automatically generated

For model training, I have trained the model created and compiled using the train data files and images with batch size of 64 images at once into the network. I have run this model for 30 epochs which resulted in 98% training accuracy.

Table

Description automatically generated

For Testing accuracy, I have evaluated the model with the test data, as it achieved 76% accuracy.



**PART IV: Model Evaluation Metrics (Prediction Results, Test Scores, and Metrics)**

Machine learning model performance assessment is just like assessing this course, how are used to evaluate our schools in high schools and colleges for the meeting the eligibility criteria for getting the best courses or getting selected into the campus interviews for companies, etc. so apparently the good school recognizes the fact that the candidate is always good. The same is being expected in the machine learning model and that should achieve the expected result in predictions or forecasting or any other required automation in the problem statements.

Accuracy is just a number, for getting a better understanding of a prediction-based problem that corrects the predictions which are made by the model built by the team with the available number of records. So, we need to train the model across different combinations of data to get better accuracy.

For this aspect, I have created a function and using the K-Fold validation function, I have returned the percentage of correction predictions in the total number of predictions which is accuracy and loss. I have trained 10 folds, each time leaving out one for testing and other for training the model. Also, for early stopping criteria, to stop the epochs on certain condition, I have monitored the losses, when the loss is almost same for 2 epochs, the execution will halt with early stopping.

Graphical user interface, text, application, email

Description automatically generated

Although the model seems to be achieved 94% accuracy with the test data, for betting understanding the incorrect predictions, I decided to perform some error analysis. I collected the all the folds predictions and created an average of evaluate using the test data, it achieved almost 75% test accuracy.

Graphical user interface, text, application, email

Description automatically generated

For plotting the accuracy and losses in the history of all the training purpose, I have created a function to plot the accuracy and loss at each level in history of training the model.

Graphical user interface, text, application, email

Description automatically generated

A picture containing chart

Description automatically generated

As a part of visualizing the model I have used the Ann visualizer code to export the model into a graph file.

Graphical user interface, text, application, email

Description automatically generated

This results in a pdf file with the network parameters…



For Predicting the images in the prediction folder, I have created another load data function as this data does not have the labels, then using the NumPy functionalities, I have retrieved the label for each prediction, for sample I have printed some predictions…

Graphical user interface, text, application

Description automatically generated

A picture containing window

Description automatically generated

As a part of metrics, I have plotted the confusion matrix using the sklearn functionality…

Graphical user interface

Description automatically generated

From this we can see that buildings have 64% accuracy, 96% in forest, 73% in glacier, 72% in mountain, 67% in sea, and 82% in street.

Now, as a part of the assignment, I have performed the analysis using **AlexNet** model…

AlexNet supports multi-GPU training by placing half of the model's neurons on one GPU and the other half on a different GPU. This not only allows for the training of a larger model, but it also reduces the training time. AlexNet was the first convolutional network that employ the graphics processing unit to improve performance.

* AlexNet has five convolutional layers, three max-pooling layers, two normalization layers, two fully connected layers, and one SoftMax layer in its design.
* Convolutional filters and a nonlinear activation function ReLU are used in each convolutional layer.
* Maximum pooling is achieved using the pooling layers.
* Due to the presence of fully connected layers, the input size is fixed.
* The input size is typically stated as 224x224x3, although due to padding, it is 227x227x3.
* AlexNet has a total of 60 million parameters.

Diagram, engineering drawing

Description automatically generated

As the model is dealing with tensors, we need to use the data loaders from Keras and transform functions from torch vision. I have resized them to original form, then jittered their color, then converted into tensors and also normalized the tensors.

Graphical user interface, text, application, email

Description automatically generated

Then, I have taken the data locations of train and test data using the transformed declared, I have sampled the data and loaded the data. Then we can check for usage of CUDA for running the pretrained AlexNet.

Graphical user interface, text, application, email

Description automatically generated

So, the input data will be clipped to a valid range for all the images with RGB data, in the following way…

Graphical user interface, text, application

Description automatically generated

Now, we can import the pretrained model, make necessary changes to the model and save it to the device. Using cross entropy loss and stochastic gradient descent optimizer, I have called the imported model on the transformed data to train the model based on transfer learning.

Graphical user interface, text, application

Description automatically generated

For, evaluating the model trained, I have calculated accuracy with the test data which achieved a 90% accuracy in the test data and accuracy per class…

Graphical user interface, text, application

Description automatically generated

For the individual classes we can see that buildings have 92% accuracy, 98% in forest, 79% in glacier, 94% in mountain, 92% in sea, and 91% in street.

Graphical user interface, text, application

Description automatically generated

### **PART V: Summary and Conclusion**

In this assignment, I have not used the sklearn much, but have introduced with TensorFlow, Keras, PyTorch and much more advanced functionalities in machine learning field. Also, we are dealing with images which are quite complicated to deal with, in training and validating the model.

We have trained a classic CNN with 2 convolutional layers and 4 hidden layers which achieved a 74% accuracy in the test data, but the AlexNet has achieved 90% accuracy with the data that the model has never seen before to that moment. The reasons are…

* AlexNet was the first large CNN model to train with GPUs. This resulted in faster model training.
* AlexNet has an eight-layer architecture, which implies it can extract features more effectively than LeNet. It also worked well with color images at the time.
* This network's ReLU activation function provides two advantages. Unlike other activation functions, it does not limit the output. This indicates that there isn't a lot of feature loss.
* It is the accumulation of gradients' negative output that is negated, not the dataset itself. Because not all perceptron is active, this will enhance model training speed even further.

Also, some particular categories such as buildings, mountain, sea, street have improved much better accuracy. This is because of the architecture of the AlexNet has approach in dealing with the images. Convolutional Neural Networks (CNNs) have long been the preferred model for object identification because they are powerful, easy to control, and even easier to train. When applied on millions of photos, they don't exhibit any worrying levels of overfitting. Their performance is nearly equivalent to that of similar-sized traditional feedforward neural networks. The only issue is that they're difficult to apply to high-resolution photographs. At the ImageNet scale, an innovation that was optimized for GPUs and reduced training durations while improving performance was required. But AlexNet is a sophisticated model that can achieve high accuracies on even the most difficult datasets. AlexNet's performance will be severely harmed if any of the convolutional layers are removed. AlexNet is a leading architecture for any object identification task, and it could have a lot of applications in the artificial intelligence field of computer vision. In the future, AlexNet may be used for picture jobs more than CNNs.