Name: Krishna Aryal

Student ID: 12843013

License Plate Recognition

And

Character Detection

Contents

[Introduction 2](#_Toc11707032)

[Dataset 2](#_Toc11707033)

[Workflow 2](#_Toc11707034)

[Models 3](#_Toc11707035)

[Stage 1: Model1 for locating the number plate 3](#_Toc11707036)

[Preprocessing 3](#_Toc11707037)

[Experiment with SSD: 4](#_Toc11707038)

[Experimental Settings for Faster-RCNN: 5](#_Toc11707039)

[Conclusion and improvement of work: 6](#_Toc11707040)

[Stage2: Model2 for identifying and detecting the characters 7](#_Toc11707041)

[Experimental 1: 7](#_Toc11707042)

[Experiment 2: 9](#_Toc11707043)

[Conclusion 11](#_Toc11707044)

[Personal Contribution to the project: 12](#_Toc11707045)

# Introduction

Object detection is carried out with the help of Faster-RCNN. The model was used to detect the number of plates and identify the characters inside the number plate. The report will focus on as follows: Section 2 will focus on the Dataset followed by Section 3 which will focus on Pre-processing the dataset. Section 4 focuses on all the models and respective steps that have been used to process the license plate while detecting the plate and characters. This section will present you with both success and failure models during the experiments. After this Section 5 will present you with the conclusion of overall experiment and planned future work.

Object detection is carried out with the help of Faster-RCNN. The model was used to detect the number of plates and identify the characters inside the number plate. The report will focus on as follows: Section 2 will focus on the Dataset followed by Section 3 which will focus on Pre-processing the dataset. Section 4 focuses on all the models and respective steps that have been used to process the license plate while detecting the plate and characters. This section will present you with both success and failure models during the experiments. After this Section 5 will present you with the conclusion of overall experiment and planned future work.

# Dataset

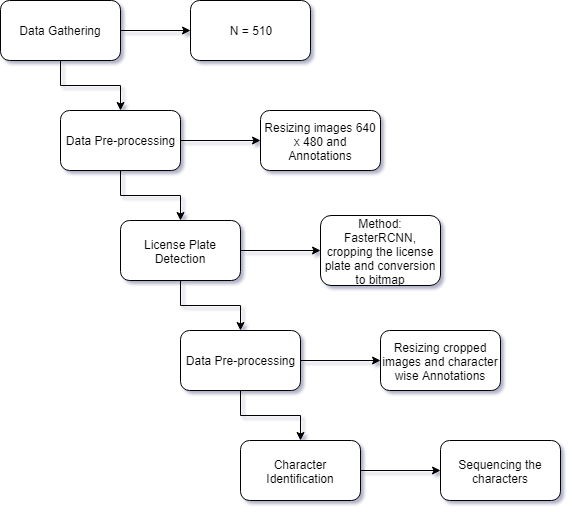
The dataset for this project was acquired from University of Zagreb, Licence plate detection, recognition and automated storage. The dataset consists a total of 510 images of motor vehicle number plates. The dataset was further split to 80-20 for train-test purpose. Below you can see sample images present in the dataset.

![A car parked in a parking lot

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDcRXhpZgAATU0AKgAAAAgABAE7AAIAAAAGAAAISodpAAQAAAABAAAIUJydAAEAAAAMAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGtpdG5hAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAM5MQAAkpIAAgAAAAM5MQAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAxOTowNjoxNyAyMjo0NToyNQAyMDE5OjA2OjE3IDIyOjQ1OjI1AAAAawBpAHQAbgBhAAAA/+ELGGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTktMDYtMTdUMjI6NDU6MjUuOTE0PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPmtpdG5hPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAPEC4gMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APojUdQttJ0u61HUJPKtLOF555NpbYiKWY4AJOADwBmuJg+OXw6uYme38RCQIMsq2dxu/wC+fLz+lbvxC/5Jj4o/7A93/wCiXr4PVP3qPCxjcdCpxzQB9nv8efhtEoL+I8A9P9Buf/jdMHx++Gh6eJf/ACQuf/jdfH1xKl0gjv3VJ+0wHB/3sfzFUJoHtn2S8cZGDwfcHvQI+0P+F/fDTOP+El/8kLn/AON05vj38NV6+JPf/jwuf/jdfFYIHJJz2HrSAkt81LUZ9p/8L/8Ahn/0Mv8A5IXP/wAbpR8ffhoeniX/AMkbn/43XxWOMfXpXfeA/CKXKjWdWQeQpzBG44bH8Z9vT1qkmxN2R9aWXxG8Lahapc2upM0UgyjNaTJn8GQVaTxpoEgyt/n/ALYyf/E18x6x8RVtpPsmlQido+N7cKD7DvWOniTxHfvn7XJGCwG2CL17Zqmooz5pH13D4n0i4YCG6ZifSF//AImrE+tadbQ+bPdKqfQk/ljNfNvhaw1S4nS4v2n2RnDfaixDe2016Pb3OmW6BUsLX3PlLz+laUaaqK70MZ13B2O5m8deG7dsTamiH3jf/wCJqFviL4UT72sRj/tm/wD8TXKrqlkE2LZ24X+75QxUajQizN/Y1hlhhv8ARk+b68Vt9WXRkfWvI6o/E3wcvXXIR/2zf/4mm/8AC0PBv/Qdh/79v/8AE1zUUfh+OJo10SwCN1UWyYP6UybSPCd0cz+HtOYkY/49wv8AKl9WD62ux0g+K3go3JgGuJvC7v8AUS4x9duPwqUfE7wcemtxf9+pP/ia4t/B3giflvD9qp9V3D+RrLvfhf4Tv5Axie3AIyLclcj05Jqfq7K+tRPSh8SvCLHC61GT7RSf/E1J/wALE8K4z/ay4/64yf8AxNeLeIPhF4bsrB7my1Ka2KjhZ2BH5jnP51ysHgK7ufJa1028uFY8OXIX659KzlTadrGkaykrpn0mnxC8Lyfc1Pd9LeU/+y09vH3hpD82okHGf+PeX/4mvnu78B6zZJi6tbkxsrAGwBkaPHQ8EHP1NcVf/bNPnWNL3U7WTJDLeDZjHuSeelS4qO6KVS+x9bH4h+Fx11P/AMl5f/iacPH/AIZYfLqe72EEhP5ba+QTr+v2UiqNQSQ4BIcqdvsT61etviBqdm6G9s1cH5gyZUkeopWiVzSPprUvjN4D0iYRajrbwORkBrG4OR7ER4qqvx4+G7sFXxHknp/oNz/8brw1td0fxvYGzuQqvjhWG11PqD6155quizaDqogmG5MFkk7OM/p9KmStsOMrn2Pa/FfwZeLut9XZhtDc2c68EZB5Tpio7r4v+B7KSJLnWypmXdGRZzsHHqCExXydbX8mo29vKqESRfuwsUgX5Qfr7+lNv7fWftVpPIjEQRqkK+ZnaM8Kozzj2rmVSfNrY0sz6zn+L/ga2nEM2t7ZiQBH9jnLZPTgJmq03xw+HdvcyQTeIgskZ2uPsdxwfr5eK+Tru7vYdROoYYsxLNv3cZ5+91/Gs9JptQ1CS+lP7yWQuSecknJPv2q4yk9WI+wE+OXw7kJ2eIgcDJ/0K4/+N0Q/HH4eTuUh8QFivXFjccfj5dfISMsMMj3A2IDxITk59AO5qO0uJZ7qMqhit1JKp/eODyT3NaRu3qK59in41eAB11//AMk5/wD4iop/jp8ObZQZvEW0E4H+hXB/9p18oMc9aw9alzPGn90Z/OtZRSRKlc+x1+PXw2cfL4kH42NwP/adD/Hr4bR43+JMZ6f6Dcn/ANp18UhyfunFLI5KLkgkZyRWZZ9rxfHf4cT58rxHux1/0G5/+N1Zl+M/gGCNXm17YG6A2Vxk/hszXxHbO0bbgcDvzVv7Qp6tz3Of60C1Psk/HX4cg4/4SLn2sbj/AON0H47fDgf8zF/5I3H/AMbr40NxHnGeP9mmPJI/yxqxU9gMUAfZv/C9/hvz/wAVIMD/AKcrj/43Sj46/Dg4x4jBz6WVx/8AG6+LVs7lukLVPHpl52QjPcmnZhdH2Qfj18NhJs/4ST5v+vG5/wDjdOX47fDh1LL4jBA6/wChXHH/AJDr46TQ7t+ScVZXRRbqZbyRimfuLwZD6D+po5WF0fYH/C8Ph4UR/wDhIflc4U/Yrjn/AMh/rUTfHj4bqcHxH/5I3H/xuvkKZ5XbcwC7htVV4CqOw9qitLZr26VOiryx6cUgPslPjZ8P5NmzXyd/3f8AQrjn/wAh0H41+AFbB15s/wDXjcf/ABuvmXR3gt7iSXy4wIk8tBjPPc1BfXZu7ppW9NoAGMCtOQjn1sfUn/C5/AP/AEHv/JK4/wDiKiPxw+HgznxB0OD/AKFcf/G6+VJ2AiP05rNQD7REpHAO4g0nEpSufX4+Nfw/MBmGvExjgt9huP8A43UX/C9PhyP+Zi/8kbj/AON18wW01tJpz2gkVVeQscjoecc+nNc7NG0UrI45U84OaOVApXPsH/he3w4/6GL/AMkbj/43VzT/AIweBNTZhZ+II2KDJ8yCWP8A9CQV8WfhWhYxRNCZJEDsGwiE4B4zz7UrDufYd58X/AthGslzroWNmKq62szqSOoyqEVUHxz+HTEAeIhz/wBOVx/8br5DutRvLy4UyStKY+I0x8qAdgvQCtC41PWzpxFx5JhZdvMEWcfXGaLIVz6xHxq+HzYx4hU5/wCnWf8A+Ipy/GbwCxwuvg/S0n/+Ir4uLEYI4P8AOhpT1FKyGfav/C3/AANn/kOA/S0m/wDiKqS/HP4dQSBJfEWGPT/Qrg/+06+M/Ofsf1qORzlWPY0AfcbfFHwcpw2sf+S03/xFUbj41fD+0j33Gv7FyRn7FcHkfSOvnnT7j7To+n3HUvAEY+6/L/QVka7ZrMksYHEi71/3h1/Sm0rEqR9MH45fDoKGPiLg9P8AQrj/AON1dvfGukeJfhj4k1nwtqEtzDZ2d0hmhEkDpIkO/wCUsoIIDKQwHB+lfE0cxQmN/m2HjNfRnwrEa/s0eNWikWXcl+zfKRg/Y045qSzyq/8AiT4sCf6H4k1eCBowuyTU5Hk3c85yD+I4qKbxp43itI3m8V61DhS23+0Jiz/+PVx0jGaYyiIRrnb5UYJCfTOaurb6jphiu/MKSJh4ywPHsc1LA0pviL42kZX/AOEr1xQqgYTUZhkepG7rTrXx146v3Edt4s14soLN/wATObp/31VO40S7aaO48xBLcNl0QcAk5/8Ar4rXi8M3Gm2dzLFK80koIVI0wePfsT6Ck5pIdiKT4g+OVvVW+8V63AGABKX0oAHqBuGTWlo3jHxa+sW0eseJvFItFYeeLe7mZ9hPB5fg9OKzNV8OarZ6jaPZ207yR26zYERJTjLbgfTp+FWrnxBqV9bEzOI3nCmQWx2CTnIBUYGQTVRTkS2kaD+MPGdle3RtPEHiGWFSVCXGoSM59wN3FKvjLxzJeLLceJtUghZf9X/aTqFOOB1yea53e4UBm6dSWq1bXVrBy6Qk9y7c1fs1fcnmZ0a+MfFs0TRP4u1LkcPBdzlh68Z5qVfEfi6SSPzPFXiBol7LcSoWOO53jisH+2bEcOoP+67Y/SlGs6OB81p5n/An/qaFCKI55djuIfG/iNFRG1fVFjUYw99liPdi+TTovFurqp3+I9U3MMEnV8Y98bjXEpr+iDrpKt/wNv8AGpl8R6ADzoUbe+9v8afs4C5pnfW/jS+Ro9+vXrbDkhtZPz/XmtnTviJd2dssIvhcEEkvcaqXY5OepNeWp4m8PIefDsR/4G3/AMVVlPFfhoH5/DSH6TP/APFU0l0QrzZ7Bb+PdSuo5JY7qzVY13FTqK5b6fNWXc+PfE8pP2S8t4+eAsySH/0bXE6Z4y8ERlhfeFXKlTjZcN1/OqUniLwVJIWXQp4ueMXDGna/QNbHXz+L/HL5xrMyD/plbbsfkxrndS8V/ERd4j8RXQ9CxaL+grJfWfBzZI0++j/3ZzTl1rw4eYrrUoj73B/wo5ULmkV08WfEhsiTW9SkfPWLUWx/6HVG98e+PLJsXOua3Ceo33kgB/Hditf+1dGk6alcv7SFG/mKcb/T9h8m5l3f7O3H4jOKXJEr2j6o5c/FLxcOD4n1jP8A1/y//FVDL8SPGU8TFfFeuJgZyNRmH/s1a2qT29+my4t4pB2cRBSfrj+lcvf6WkFpJJabtqj54852+49R/Kh07ao0U0y8nxM8ZLGqnxVrjEDr/aMx/wDZqU/Erxo3H/CUa4P+4hN/8VWBDM6xiNdoAw2dtOMkrsxLn5utZlm5/wALH8a52r4p13I7f2jNn/0KmSfEnxquP+Ks1wE9v7Rm4/8AHqwypaVpCTuPUioJgrOQcllGfrQB0Q+I3jWRW/4q7XeB21KYf+zVGvxH8bYA/wCEv13/AMGU3/xVYARVjfZnrgU4W+z7wIboQe1AH3n4EuZ7z4deG7m7mknuJtKtZJZZWLM7GJSWJPJJJzmio/h6MfDHwuP+oPaf+iUooAX4g8/DLxP/ANge7/8ARLV8K7SOiH64r7w8aAN4B8QAjcDplyCPX901fD2oW0VjcYaRwjfMuBnj0p2EzPlRXjwwcN64pkbmNPJuB5lvn8U9x/hWnara3DKI5ZmLEA7lwAa1rzTtPS28gSnzChLSGPo2OmM9M8UhXOYTS2mm227+aCcKVB5p8+j3FrMUuo2icclSORVvS5XgAQhlmB3R8cE10kVzdTXs1xOhxKq7mfHJFO10Ddjn/Dfht9b1mKDkQA7pmx0Udfz6V669tFeTJpMaMlnGmJTEwXYAOB/SodEsk0zSXupUAuJwNiqOWY/dH9avWtnBbXBWL5ppAPtEpOc+w9BXRTgYzmLpXhHRbO1kRrRZlkcMPOGSAOgz1rZQxWybLeNIkHQIoFQNMMYqB5vm611JJbI43Jvc3riQmA/TNZ/nHsahN/mEqByeOtRKxzg01oSXlmYkc1YjnIPWs9Scc1Kr4/8Ar07k2NVZQasLMD1FZSTY/hBq1FIh6pigTL3m+wNP80iqm+PHal3pwBigkw7yX+2/Fq2UhJtbFd8i9mY//rA/Ouk+0kYCnAHAA7Vy/hlVluNT1AD/AF9yVB9h/wDrroCQamHcqXYupdv/AHj+dVtS0zTdajVdVs4rnZ91mGGX6Ec1EkmDirCyVpdNakK61R5jrfwgu1uWl8OSwTQfM4huDhsk/dz0PB68dK0rr4f6Pb/D9JvEL3mn3UQcxguJEgcjp6bTjPr9a79JijUahDBq2kXVhef8e9zEY39QD3HvWPsYa2Rt7aelz5TlL2F95trKC0TcOnRv/rV3sTw+MvDXkuwW5jGY2PVW9P8APYiuf8XeGpPDurNatPFcRADbJGRyD2I7Gs/w/qb6PqindiJ2AbPT61w7OzPR0kroZBHfaXdu0avBLGCrnbyvNdd4b8XQ6eySXWq6jGRyyrFGVPHYnkdBnBq3rOj2Op2r6g1wLbcgJJTIZhwM46en4VykelKyy5fCngMB/KuadNSbiy1JWuel67rP/CR6FPp2jWNw19cqFluL1zgR5zgDJ56jPUV53Zak/hiz1GOaAxXjObd5G6qoPzIg9Scc9gPWte71q4ms2Uaf5sjMCZVudpwB0xiuCnmuL3UFlnVtpfID5IAJzz7c1ao04xskJSlLc6DQtDk1qZdQ1GGR7YHbBBCuRn3HXH6mtnUdEe3sZbiWymhdCAjeVtXBOOleo6TYa/J4HX/hDNUtrWaFiC9okbbxjgkAd/8AOa8n1jxN4xm1K40/xRq1xcNG+ySJyMZHI6Ae1JJymncjWSuZv2aZ+iGszUdEnnuBIHjUEAYZq02unPVjTTKxHJ4JrraTHqY6aCw+9cL+Ck/zxVmPQICu553POCAAMVbLkUizYbA59qmyQ7sij0S02jasj+uW71YTSbZR/qUH15pBcMjlQcA8017llj3Nvcd9oo0FqWVtIEXhVH0UUboEOAuaoC4ab7sbIO2TThvPGaVx2LhughwqKPrUbX5/h2/gKrFRn5ufrTkQySBEALE8UrisTxTyy5LMVjX7xx0+g7n0FbMfhe6v2j+13dvaXMq/6PZOxaRVxn5gPu+5NV/DV9b2/iCKWe1kurO2DMPLwC0mMB+fTt6dasa1rkQM0Hh3TTZGRWWSYuXkYN1Ge1Q3LoUrHJXpeJhGxXI6471r2Fulnp/myFQzH05NUo9HupUBmRgD2retY4Rbx212DGkY4O4fNWsVrcGVrabNrtVupJb61GzheCwrpY57NYwI7K2K47IOaYw06TO6xiBz/CoH8qvUjS5yk7q0ZVTj1zxVaZdtycYPTGOc8V2UumaTJEWe0C8ZyGI/rWZZ6bbSXoubVjFAhIG87sn2qSk0YbpcIQ3lMOOpHWqskcgJJQgH2rsn0hGufO+1qRjCgp0qld+H5biUsl3DjsDkYoaBM5XB/wAitNbdmhiFvGWbGQw5wc1cbwvec7ZoG/4FUlv4X1ErzdRQr2/eE5/KpsMoNs075XCNOwxhedv196ynkZcjJwT0zXUnwnMdoN7CQP8AZNO1Hwp5rqbExqAoBDOeT3NFmGhyDH5B0BqLqTW9N4U1VDxCj/7jis640u8tf9dA6D1IqdSiiRilyPIIJ5zkU5kYHBUg+hFL5Em3PlsPfaaQHdeEpvO8LhM5NvcEfQMAf5g1Y1A7oFYcvE2Tn071leBZd1nqMB/upKPwbH/s9adw4W7+blXGD71XQjqcLq1uLTVG2/cJyMjqDzX0H8HUeX9mrxokQJdjfKgB7mzjxXhviS3b5S3WMGP+or2/4Osx/Zk8amPO8fb9uOufscdQaI8UivLfRFCzwTGTJOxlG3cO+SOR24qn/alzqUmZ8uWbD7GIZlJzj3qO8F9NcC2u/MJGAFbkr6deaW3gFrNOYWkSSA4DbcbuvODyPpzU2Qy/BdCykaaaZFML7Y7faSxPqR/Wumtp53RPNkgWZfm+8Tx3yOg61xsVhqE0kXlWsxZRuDbSwbvnkYq5Jpl073JbCyypuCYIY59ARkd6hxTC5r3uoySXU8V5Gi3aYZZkzllPfI6HH9RWPdTGOHKnBb5V9vU/59asBla0tLeNHQxxBXDtn5v4j+JrMvJPMuCFPyr8oreMeSJnuyEE96Xd0FCjilHWgoUUDmpzbskauwwp4oUhemKYhmxu6sOfSlC7eTxUjyYJJJZvrUD7pTljj2pXAlFyoUBYySO+KPtDlRtUj1wtVVhyTzVnzpIVATBHuKBNCeaf4lf34phmUdVbP1qYXsvdVP4UG+c/eijb8KeoalbziejMPrS+ew6kH8KJGWQghFT/AHaay5PAoGSrMG+8MVIrZHBq3f2enw6Lp9xY3LS3Mqt9qiZceW2eMeoxWWCQwK8U7sNy4JJE+47D6E1Lb3khmHmNu7c/yqqj7uDwf50u3B4qhbm0PDVtqJ8+wl8hmADREZUH27itGTwPpcelXsx1eRLuFC1vAYQRPg4xuzxxWZp12+3CuVLDBIPQ1t6SLefy1k8wzZwSQP3ZHfntkda46zlTfMtiU5I5YeHtQ3Y+znkZBz1plv4Z1R78hYFifI2mVgF6ZrsJNKv7S7e4kWW7JJKpJIoUZ9809Jr+BFNzbRyFhuEgmXaPY981z/WG1pYrmZwx0HVLe5gN5atGZZQckjnLfWr0vh3Ub7UbiRIPKjeViHkO0YJJFdhdawkduhntjJNn/VwHcAe3bNZNzq0kzYe3MJwSWkf7o9lH9aqNaclsF2fWvgWIwfDvw5CSCY9KtVJHfESiijwO2/4e+HWJJLaXbHJ7/ulorrWxY3x7I8Pw38SyRHEiaTdMp9CIWxXxzdWn9sWSS28f3hvXPAU91zX2Z4xCnwLrwcZU6bcZHqPKavk95wqgRIq46E8kfT0rSKuiJHP2fhmUOs0rLHt5yOefqeP51sJaWdupZ08+XP3m5/nx+lVr7VlhXdOWZz2pftTBAeVJGSHHIqtFsT7xXv18y8jdV2lentWz4a01tW1ZBJloosM+e/oKwppjK4UZYk8V6Rotuvhrw21w4H2ggHnu56D8KSvcHoiS8lCai4HKWg2j0Mh/wotHKxlz95zn8KyVkZwkJbJJ3OfUnkmr/mbRjsOldsFZHJMtvOd2M0wz5zVVC8km2NWYk8ADJqyNNv8AG77JNjryhqrozsPWX8qmWWqJV4m2yoyH0YYNSo1O4jRWbjnBqUS57VniTkVMj0gsX0kx0OKlEuF4OKpg/L7U/IxRcVi2Jvekurv7Np085PEcTN+QqmH5qnr85Hh64QdZdsY/4EwH9aTdkJK7NDw3mDw/ahvvSL5jcdS3NawkAqhCBFDHEOiKF49hU6tTjsJ7lpZAakEgC1UDAUvmVRNi20oP1rivFniW+03VntrJS8TWTNMrkKF54dSTyR3FdVvyK8w+KkLNqVnKZQF+zsNrE8YYdB+P6VlWbULo1pRTlqcFcXK3Mj5UgsASc5z6mqEy5jyDkp/KphIFmZo8AdMZ7Uz/AFkjsBkKR8pPUV556S0O18Iamt/Yvp11huNpB78cfmP5VQvYZtOu2t3clE5QN0IrntNvG0zUlmTKx7sN7DP9K7/VoV1TS0vYlDOgyQP1H9ataonZmVHdW8sYjdfL4x6g10ek20IsAktvBdxZztdc/kRyPwrjMf3T+dTQ6rJpcMkyMw2DO0HrTTE4nai0trTUFvfDV5d6JeL1wd6n2J6kexBrmdT0bUUjv9S1ZxdTTSbvPDA7iWGWPpVrTPFtvdIr3EJxj5ih5H4Voa3PZ3Xhi5ltbhXb5CUzg/fHUVVo7iu7nE4Qd8n25pG5XhfzNLupM5NSWMEOep/WgxAd8mn7cCmnPPNADWTHI6jmkG0jPXNOJp1tZXFzLstYmkz6DgUgIScfSnJveTEYYsegUcmuhs/CpOGvZf8AgEf+Nb9ppkFouLaFY/Ugc/nTsFzkrTw3fXTBpQIE9X6/lW5H4UsRbNHI8rFxhnVsE+1b0dqXPHPvVlbZF+8cmnyiuczF4UtljCQ3N3Gvs4/wqZPBcW3Kajc/Qqprodin2FSBCDheQaOVCcjmH8KTDgak49A0QP8AWqz+EboHK30TH/ajIrs1GcgikaBSfSqsLmOOXw/qsH+qls3P/AhmmtpmtL/ywtn/AN2XFdgYPfNMMWKNQujjXtdYUHdphb/clU1Eft0Yw+lXQ/3UyP0rtPL96TY3alqGhw7X7R/622uY/wDeiNINWth952T/AHgRXdYb1qNo1b70at9VBouw0OLXULV/u3CD8amW5iZcCZT6YYV08mnWUp/eWcD/AFjFVZPD2kyfe0+If7uV/lRdhZGEeR8sh/A0NM0KjMrg+xrUfwrpJ5WOWP8A3JSKgk8IWr/6q8u0+rhqWoygL+Q8NJKw+tQXM0c4/epu/wB8VefwdcjmDVWHs8X+BqrJ4U1lc7Lq1mHoSQf5UtStCuly5LTeUu4HAKqCce3pUE+qzSRlG83bjGNlWxouu2ikLYQyZ67JRzVK6tdVUZl0e4yOm35h+lF3YBngiTZrE8XaSCRfyGR/Ktm/XLfSsPwtFPB4mt2uIXj3FgQ6kfwmusurBnydyj6ml0Hpc5zUwl3Ykqcuq4OevtXtnwAMcPwJ8TtcECFb27Zyem37NFmvHbi3hhZVQl5Cecelev8Awztzafs6eOlHQm/Zfp9jSs/IsyJNf8Lq/wA8lqwByMIKwZYfBX2qWeWXfNI/mqzuDsOc8CvKlumTHyDI9eac17IzDqB0wKy5ZdyHzHucfjXQ4lTypYtqjGN4FcH4v1qDU/EwutNkAWSAK7Ic4Iz/AEriGnP8IwD75rS05We2LE/fbAH86cKb5tWL3upJI/l28k3f7q/WspVz71par8qxQDjHzGqaJ0GOa6XuNbDQlPSP5lxyc1KkWRTxHtcE+tFhjp5fMiRAMFSSaqsPLXJ69quJFnGeaheMSOcDgHAosLYgt03li3WpxbD3/OnIjROHiwCPUA/pS+ZIzMTjJPOBily3E79ASBeeO9RyKmcBS3rzViMNjvS/ZmOKaiIplEz9xh6c0wRdeK0fsuO2aBAT0FUkO5n+Uf1pzQEKD61eW2O7kVL9m+XkdKLC5jK2fLjFMMZzWo1uADxVd4irdOKVh3KWw4qWJs/K/Xsal8vk8Uwx8+4osO5Ytn8qYDs3H412Phy5j89lur22trdlwzXUuxc9seprjFG9cnqOtacUcdzZ4kXcG4I96mpCM4uMiX3PXbbwVJqFvG9he2X2Y87ojvz9DTz8JWubhZJdUBjXkxrEOvrnrXkXhXxbqfg7U2ubdmax3gXNozcMPYdmFfR+h65ZeINGS4tJ2a1uo8B43KsufccgiuWOHpLZFeTOLuPhDMI5NmvTW8cmOFiAA/M1zetfC7S7UkzeK4YJGTneFYk5+9gGq3j3w7rHhjVPOuru5vdNuG/dXLOSf91vRv51xn9pQLP5coaNmPysTlW/GuqNCHcL2eqPsnwfbLZ+BtCto5luEh063jWZRgSARqNw+uM0U3wWc+AfD5HT+zLb/wBFLRUvRli+NDjwFr//AGDLn/0U1fIEkoA9a+vPHBx8PfER9NLuf/RTV8YSXFXF6EsknlUyqzKrEdMjoahluGbljyevNVJbjJHPQ1G0haRUUFnY4Cil1A6rwZpn9payJ5RmGA7j7nsK67xJd7r6O0Q/LCN7/wC8am8L6ami6E0sw/1MfmS+7en9K5uS4e4mkmk5eZyx/Gt4R1MZO5o2IJLSHnsK7jwz4PXV4Fu72RliYnbGp5YD1NcXB+6hVB2HNdb4U8SWui2sv226lyzfJAiE498+9dD2OWV3sdVbxW+mCUadpCkiYW8M5UEE5AO45zwc/lVyG+1C4uogsSJby3LIreSMeWoPJ5zzg/pWG/xIsEjxBZTsO2cLmqj/ABK8xhjTyF/66/8A1qVkY8re5q+O5rE2Itx5b3TEFBkZQdzXACMr/Gg/Gobq8e7u5biUndIxY1EJfWhKxcY2ReUAH5pF/Cp0HcPkeoU1liTn3rf0ycRWK5JV8knLDFNuw0iINheWI/4CaXzl2kZJ+orUTWdvDSQt/wACFSrrUJ+8lufU7hU8yEzE8zPRqytR1Kzm1GysWvLdCs4km3yBQoUEgHPHJxXaLrNkV/1cH6Vh6nY+Gb2Zp7qK2hkkPzMJFXcf8aTeg46PUtJKGUFWDAjIIOc1OjEr61WttQ0K2hjghu7dUT5FHnA4I7VWn8a+GrNmEl9DkNtJAJGfr0q+aPcnlk9kapY4oDnFYkXxG8M3Lsou3+XqfIIH54p6/EDws2CNRVcgn54yMfpS549x8kuxsqx3d/yrhvidYLc2trKpAmTeuCcZXj+tdbb+M/Ds2BHq9nkDOGfaf1rl/Huu2GpadPb74neFlMMkUgbPqD6is6s48li6cZKR5XPYFUBEij2NR21ufOMK4dpOEK+vpViWEtHxyAeRTIFMMwlP3VNcR362Kc3ELKVO8HofyrsvBGp+fCbGducYGfUdP04/AVy8kxFx5hjKAEsM9Spq1psbWetAQykS7zgAH5h1BHHUGqjuJ6o1das30+/cAERN8y+3tWW0u7IIBBGCD3ruNVgGqaQt1AoLldwX/a7r/T8q4cxpMpaDIYdY26iqkrBF3RBBMf7QEafJHGn3R0Jq+9wWgK9v/r1jxyEXku7IwQBVsS5U9T9KSGSAk/WrMUXc1UjDHnAH1NMv7420JG/n0UY/Ci9gJ7y9htlOSDWfaXF7q2oxWWl25lnmbai9zWNJI88hZzn29K9Z+BWlxtql/q0qBjAgijJHQnlv0x+dK9xpWL2j/CbU0VZdVPnSH/lkgwq/j1NdNF4Iv4ECRWu1R0CrgV6Xb6gM7cjOOnetKC96Vd0RueSnwnqCDm1b8qBoV3H9+2cfhXtEV6hwCqn6irKPbyfft4z+FHMB4h/Z8yLgwt+VRPZvnmMj8K91Nvpsn+stU/AU06Jo0/W3UUcyFyng/wBnK/w0CLFe4yeD9Fl/5Z4+hqnL8PtJl+4zCjmQcrPHSAKTbXqc/wAMLZv9TcY+orOn+GNyufJmVvxqudC5WeeFfWmfhXZ3Hw91SLOI9w9qy5/CWpwZ3QN+VF7isc6Vz0HFNKEVqS6PeQ/fhb8qrNaTLw0bD6ipGU+lJznpU7RMOoxTdpA/+tTEREcdKYR7VM3JxTGB54oAhIpChx04qXb+FLjt1pgV6OnNOZTjNMBw3IoAUnHeo55Y4ITLO2xF/Mn0A7mi5uY7WHfJ3OFUDJY+gHc1WSB5JFur7Acf6uLPEQ/q3v8AlQhiRJLNILm6GzAxFFn7gPr6k1Q1a78uMBe/61Y1C/W3Xnlj0WsCSdriQtIcgdKU5KKsioxbZHukDbx1I5Jr2r4fnP7OPjMdSIr/AD/4CrXicje/Pavc/hUEn+A/ilCAyl7xWB7/AOjJXInrc22R8w+TIpw0TqcZxik8s98575r0ltGtZFwkTxsepBz/ADqq3hG2kDETuAxyzEYA/AVksRHqZ8xxNvZXM8oW2Qs4GeDwB9a6K1j/AHoVzkxrgn371umO10jTJIbKIZ2kNK4O5vX+VYtlLHHaXMswZiy/Lg4wc559RgEfjXTQlzpyJbuZF43n3zn0OKlCLCQWXnFV1J35Pc5qwGLY3c/hWxQK2WBAxzT1Qu3NKgz0q0kefamIiZNkRIHJ4pkcPtVmaM7Vwe/NEcRPRhTJuReVn+GmLbktwO9aEdvn+Kp0gUdMGqSJuUI7dugHeri2uFBxzV+2hDHp+laUOn+awAA/KqshXMGOzLN0qb7BtU5HSu107wxNdsogjzIThRj71dPN8NLizszPdxZBO7HoPQ0m0hK7PHxbYOMU0we1dzqOhwqxURiM+o4rmbuwe1mKMeOx9aasxXOZvbuO1faYpCc9SMA/jVSG7S6kMbARv/CM9a6d7GSVOImcE/3CQaoyeHWN4vkWEjTZz5Sxnd9QPaoaady01Ywp7mOC5ETdiN3HSpmiBAZSCD0I71ZvtOinJbeqyJxuyDj2NItl9iBEjYYgEx9B9QKWtyuhRDiK4EcnG77p7H/69aum8s8J/iGRmoJ4FKjO09wDTbeWSw1e2llViiyKSjjqM/ypPQe4T+F9c1aaaXSbCe7t42+cxc7GI9K6n4bahrvg/WRp+radeR6deOAS8LYik6A/Q9DS6J4ivNF127XSmCR3HzBGG7IHI/HBrsrX4k38JxeWMUo6EoxU/wBaya10He6PRXW01bTpdP1OBZ7aZdrxuOv+B968P8d/DK68OTNdWiPfaMzZEi/fi9m9Pr0r0yx+Iul3BVbiCaBj6gMKTVPiv4Z0m9NjcPczSFQWWODcuCM4OTTUuUuKctEj1DwX5X/CBaB9nQpD/Zlt5ak5Kr5S4GfpRVrw/c2954Z0u5so/KtprOKSFNu3ahQFRjtwRxRUjtbQo+PDj4ceJT/1Cbr/ANEtXxDJLwSTgepr7c+IJ2/DPxOfTSLs/wDkFq+Ffnk5c0wHyXBbiPP1711vgrw5cXF9HfXMRES8x7v4j2pnhLwob+RLy9jxBnMcZ/5ae59v516pZ2W1VjhXp14rSMerM5Stoip4k3W3huK2jBBuJfnPqFGcfn/KuNDiGVSwOAeRXqn9mRTw7bsCRP7rdKoT6Do5OBar6fKxGK1U7aGVjhv7Qt8DMpX8KDqVuPuyLXZp4S0uUgLav+DE0yXwbpWSqxTZHU7hx+lX7QjlRxx1CJh/r1H40LdI33ZlYk9Aa6s+CdKbGDLz6leP0rh/Gmn/ANi6hHDo5lOUzIfL4B7YPel7QagnoaHn/vBG0ihjxhjihp1U5e5iXHq9UNB8DXGv2SXl/rMtq7HAjMJz7dxXTv8AA+zjjSS41y4cOMg+SuPzyaXtJPZBywWjZgtq1lF/rL+H8GzQPGWnwLgXjSeyR5ref4V6NZbVkuUkP94o7Z/75bFPTwBpsRHlXUCDGciCT/GneYrQMBvEz33y6dpl5Kx6MIeP1IpnleIZl3GyljB7NPGh/TkfnW/e+EXRFWx1AEHqY0ZSPzqgfAt3OmyXU7kqTkgyMP5VDcikomFe2GseWfNgi2kYxLd7v8Kxf7NvQ2PNsIscj51Y/rk13Y+HiLzvhLerRlj+ppU8DTR/duYl+kdK19yk0jiDpt80RVLvzi3/ACzjQ4P6AVLa+B9UvJlhsVhaR+iFhn/CuzPgq473o/75NC+CJFOReEH8aOWPUOZ9GY5+HviF4/JSSRFXiVGmGAfpUZ+G+vOi5uIyMZGXGM10S+DJcf8AH0x9etH/AAh+D/x8n8qrlgRefc57/hWWtMyNcX9qApzkyAkfrU83w61SSIv/AGnZyPg7fMZAT7ZzW2PCXrc5/Cnf8Ikne7cH2UUcsOwXnfc4a68KazYyMk6wFl5+R1YEfUdaxZJIt2xnjVWP3v7pr1aLwirtgXUnXsBTT8ONJ3ET+ZuY7j8ij+lZuHYtS7nlcQtXQlpvnAZWycgZHy/rVZGaO2861ZluLZ/nJGeD0YfSvYZPhvpNsoYo4V+BlgM9+RtzUkXgTT1YNHbswcZOJcEj3xilyjUkc14VZ3s3tJxg+Wsg5zjsSP0Nc14o0yTTdR+1QoRDMc5XordxXp6aZFZblt4VTPBJ5J/Gq1zZw3ET29ygeKTrkdDVWurCUrM8kV4bg/vcJIf4wOv1pskLxNyOvQ9jWr4i8MzaPIZ7fMloT1H8H1rG+0ubdoicqeme3NZvTRmvoOE2FPIyOBisS9mM1we4U4FX5nEUDMOw4rJB4qSrDlBLAKMnoBXpXw48Yr4Xs7iyuLTzonk3s8RwykjGPevPLcKFyW2O/CH0re08GORdwBdgN+O5qW2ldGlOKnJRfU9kXxHofiJo5BrElhcBQuG+X/JrbifW4oTJpepxXQH3F3hgRgY614XfWb2motFcIybgGAYY4IrsPhV4cXxZe3ti+rXGnT2sXmRyQ87sEZzz0waUaikro0rYZ0up7FoGr6pNHMuswLDIjgIV6OMdfzrfjv8Anhq4GXwL4205j/ZPiSz1EL0SU4ZvYAg5+ua4b/hbt9pl7Ja6nbRPJExVyoK8jr61dzlse/JfDPWrK6gvGDzXilh8ZNJnwJ45I2/2WDf4V0Vr8RNAuMAagsRPaVStFybM9Tj1MBcE1OmoA964C08RWd0ube8hlz02SA1ox6njo1AanaJfY6HNTpeg+lcdHqfT5qspqfTmgd2jrVuVPWn+ZG/XB+tczHqYK981ZTUlI5NIfMbElnaTf6yGNv8AgNVJvDelz/etlH0qKO+XrnNWUvCACeh6U9QumZNz4E0qf7qshrHuvhjA+fImA+tdmL5QOTThfRHqcU+ZhaJ5fefDK9jyYSHHsawLvwZqdrndCx/CvcftcfY0GeJxhgD9afMxcq6HzzNpF3C3zxHj2qo1vIh+ZCB9K+hrjT9NuR+9hTnvise78G6Zc5MOAapTRPKzwwg9CKpX11HaRgsC7udsca/ec+g/x7V6rr/g+00vT7i8uWCwwIXdvYV5+/hma1jN/fLtu513bD/yxQ8hB/X1NVe5NjBgt3Ev2q9Ie5IwqqfliH91f6nvVa/1NYFIUhn/AJUuo3MkHmLzx3Fc5NIzuM8mplK2xajfVkjymeb5/mLdT3FMlAT7vToKciiNMt171BPMEVi3U9q5JO7NkV7qfy8AH5jwPavffhEDb/AbxM5x9+7fBOP+XZK8BsYftM7TyjKLwoPevoD4bI03wF8WxxNtd/tiqT2JtUxTWwM8xTW7OeQfvWU/xfL8o9iacLqLapguY2iBIKqc4rlzpl9Bqqx+aoSNAhEY3b/UduvWppZdF0vUliKyPMYzu2twTngcHFcfIumpi0S65e2xj8qHJlz87bsgjHBHNZczeXpoUcFv6/8A6jUF/IPt0MIj8oJHll3Zwc55P5Vs2dvHPEVdQwyOD7D/AOvXpUVaFkLYwI17npUyg11EenW7KB5a4qwuk2p+8uPpW1mHOjlY/rVlAe5rqotIsD95W/A1aTQ9MYjIkH0Yf4UC5kcksHnYy1SR2pA4euzj8P6Z/fmH4j/CrcPh3Sjz5swP1H+FFyeZHFxWLMPv/SrSabIP41rt4/Del5B8+4/76X/CraeHNK4/e3PH+2v/AMTT5ieZHFW+lyqdxdcegNbNhp0yMpDJn610kfh/TSuPMusf9dB/hVyHw9pwORJdZ/66/wD1qXMxcyJvDUl1pl5FO8ibVbkBuorsNX8e2ttYSNBGssoBHlnndx/jxWFY6Np6lTJJcsmeR52M/pUt34f0N5X2m5XPbzjxUPVlKSSPN5vGGmX+jF5dP1SfVjJmRoFTyEG7pjORx7dakvNd8OTKy2/hDWLog/K016tuMfQE12TeG9FgYmMXH/f2oZNF0kfwSAe70x+0ieZ3rW91JmHwhb26Y587VZXYe/y4qF4y1lJaKiWkEv8ArY7V3Bk/3nYlmHtnHtXpEui6QRjy2/7+GqMuh6RzmJj7eY3+NCSFz3PLYdD06CaaJLfAYZ+8eR+dJJoemAgi0j/HJr0mXQ9JX5hbDPTO8/41UfTtPK4+zR5XjnNP5Bznm8trawyMq20QBHB2dKqXkMm0FVPC/Lx0xXf3lvZJlYreIe+2s1/KQHbGgP8AuipaZamc5b3LWuq6ddsCCGUMCOcH/wCscV6fawWWpRgSRoSevHNeYa2+FLrgFSGAA7j/ACK1fC/iR/7duLG4YB0ctG3TenUfjg1DLW1ztbzwwbbE1ryvpXH+M9HR9Qsrpl2CWIq7lsDcvb8iK9c0udby3UY3cdMir1z4Os9aszbzfuwx3KQASp9ayqXcdNztwlWNGtGc1ddTuvA//JPfDuP+gXbf+ilorQ0Wz/s7QNPsgwf7NbRw7gMZ2qBnH4UVa2MZtObaMv4gf8kz8T/9gi7/APRLV8f+EvC0uszC7niY2kZ4HQOf8K+yfF9qL7wPrlox2ifTriIn03RsP614lNZwWPht7a0jEUUaogwMZywFXGy1ZnuSWujrb26KdqMOpJxj2ArSjhSJBtmQfSp7Q2iZEoR3J9M1Mb3TcsHMSFXEZDIB83p696HUbJ5EVTEkn3roU9LS2/57KfxxS3ktm/y2pi3ofm2jiqgZSRyuD6UvaSDkiajrHIiDz0XaMZVsE01bWHGPMQ+5YVSVUI4bP0pSnHBpe1kHs4mjHbpHnbIvP+0tV7zSrO+YG42kjoQyiq20+tKEPrS9rIPZxHpoOnrjaxGP9sVKuj6cgwSxHtIv+FQAHPWlO7BFP2sg9nEvRaZp4XPlMw9fMFS/Y9NHSID0/e1nRmTbtBIFMYNnij2sheziaog01FOYVPuZahZNMJPyOD6K+f6VlsxXrxTRMFb1470e2mHs4mqV00/LiQf8CH+FJ/Z9k/KSN+LrWK13GrfNIo+rVJFMJT8jKfxqfazH7OJrf2XbH/lo3/fa0v8AY9qcZuj9OKzSzdxTct7Ue1mHs4mq+mW4XaJgR65GagOk23/Pc/iwqjlqa25ee1L20w9nEuNpVv8A8/P/AI8KQ6PBtDG4bB75GKpBiRkEY9zSmV9mzf8AL6Z4p+1mHs4lr+y7cdLrofUU+O1gjdT5ysVO75sf41nbvcfnTSf9pfzp+1mHs4mjcWcE0jP5qIWOcA8fzqubFEHy3YH5f41SdgvV+faoTMuep/On7SQeziXpLaJs5uE+h/8A11UksInHFxH+NQG4UH19qha8iDYJQEngE01OQuSI2fR3cMoaGZG4ZC3X868z8V+E30iV7m2Qi3z86/8APMn+lelLeQv9zacccHvXN+Mrox+FtY/i+W3I3HofMAque61Go22PKtQ+W1x6kCs3sKluLh5wA+MZ6AVHUFmjbKxaEFUMeByeua1EkK3cZzzuGTWPbAFonRC8o4HPA966TRI7W51u0S7AeCSQI4zjIPFG44uzujV8TW9/cWlvd3MLkRL5fmjkY7VV8J6/c+FfEUd7BJIglj2sY2AJ5/Lt0NenXugaguiz6daOl7aTR7VSUhZFx0w3Q/pXlWtaPq2nog1CzmhMP3CV4P0PQ1EI+ysobHVKpGvGTqPU9l0/4raOQkib7aaFcxidDtLYwAxXPH5V4d4ykil8W389syPFNK0ilOhyc8e1dnpVvpup6PbxLDbhDHmWYnbJE2Bz9OprgdZj+dHHPUZqovocnU774QeE7TxN/aAv0DRxqCMqOSTiuz8SfDHQLNFd82Q2/wCuhR9g7ZYjgfpVH4BlYV1CN2VSYY2AJ65Jr0LxJeRTatp2hySxxJdMZ7hncKDEhHyjPUs2Bj0zQS27nl0vwmuVuUi03V1WUKGdJHBYjPVQOcY9TUN/4V8caBcbdMvpruEAYZWK49sZNekrNBBqt9PJZi4J1eKPzhIFeIlECY45Xk5HpW7DcTyaxLbuyPESw2lBx+PejQaTlsjxMeKvHOl3n2e7tHmYdMxhg2Bng8Zq7afFy9GRe6UTsHzFQy4H4g13WnXrC6SOe7+2XdxBPJcWLqFMDp0VcDgduc560ljqFtb2tzNdW9s062SXZfT1HIZsCNs/xZx1PvxTFoYVh8YNIlwJ454j7YfH5GugtPiNoF1gLqKRk/8APVSn8xVDVvBmk+IpLiTU9Lmj1C2VBLDCod9rfdYEZ3Dr78Vh23wk0nU7i5fTprhViJR4V3xNE2MgYbrx7UCsj0a08SWVzj7PewSD/YlBrUj1UjBST8jXhuo/CjVrSST+zLySUoT8jbdyj36Vx91qHiPw7rTaa13LHLGgdgJGXAPtk0C5T6E1y71691aM6aWWCOIkt5oQO3YHv39O3Ws8an4zsosKHuW3YGdhGMcZ5z16+1eU23j3xFaBdmqFxjJEqBv1rWtvivrScTQ2VxjrglD/ADoHY9Df4halp9xKl9FHshVyxdSrEgcYHvW3L4+t7IxQX8bpcmJJJVT7sZb+HJ715inxaV8fbdC3c5yrhv5in3PxT8K3AD6tpVxvJAz5QYn07igVj1U+NNPXTLe+kmeOG4OI9y8tz6fhV6z8QRXtuJ7WQtGSQCQR0+teRjxZ4Fu0ClprcDoDHIAPyyBW5pvi/wAL2tnHa2erW6RoMKruQf1pi5To/EmsNqesabo27dDu+2XYPQpGfkU/V8fgprK8RTeerE8+prC0TxFpt7ealqsuoW6vdz+VCjSgFYY8qvB9Tub8RWldXVvcRHy7mFh7SA00xNHnOtR7ZmJGVbg1geRtlJ/hA4rqPEU9lbhjNeQJjqDIM/lXn2oeIgMpYDcM8yMP5Cpn5GkTTuZgpA64rNkLXdwsSdO5qJroyQqR95hn6Vo2Fv5Me9vvt19qysWW4o1hjAXgKOK9s+GTPP8AAXxcsA3SE3ioM4yfsqY5rxC4lVIWOeR1r2X4GzG7+BXixiSubu7UH0H2WKq6AeSW3hLXZpFaeSGMIuNhy4kPqecZq4ngK8nuDJf3VpblWwnljcSR/L86344ZLSSTbd74uo3ZypqxBBc3dustv5p778cE/wBa5VKT2Cx5he2Ew1i53XBkMUnlh3Gd2O9atv59nbRvOp2OW2yDox7j9RT/ALDLc3uoZZU2ytnP1qpcsyWMSk52sRn/AD+H5VtTre/ynu1spUcEq6XS5pR3+CNoP4mp01F8/NWAs5wOtTLP712cx8zynSxX8eRuyKuQ30ZbiQVykdwTiphcEU7kuJ2KahGoyXBqQavEoyWwPyrkEumHU1nT6mi6h/pqnylPyqRwaLgoXPRIfEMCnmWP/vsVZ/4SC2bGLiNf+BisLTPF/hgJHD/ZUUsp4wsOSa6KHxPoSct4cb/wF/8ArVNyvZklv4hs1OHvIR7mQVfj8SaepGb+2H/bUVFD4u8PYGfDh/8AAT/61Wl8Y+HQuF8OZP8A16H/AApcwezXcnj8Uab0OpW3/f0U2XxRpobjUbc/SQGo4/G2hhzt8NkgH/nz/wDrVL/wnelBf3fhiQ/9ulK4ezXcqy+LNPJ4u4j/AMCqlL4ptCTidD9DWm/ju2P+r8Kz/haiqN78Qvslu00nhi4jiQZZzbjA9zT5g9miifE1o/H2hFOeATjNMl1wZ4GfxrmvEHxKtNdtTbQ6cUzwTsFZlndSpZqk7fMvA5zx2ppkyhY66TWz6D86pSarJuJBXnrxWE93z1zUL3f+c07i5TVmvd7fMeTVCe8A5HA75qC71QXCxhYo49iBfkGN3ufes2afdmk2UkWbyGOe1jd5CXfcTGOMDtk981iasraR4gtb6NiRiNsE5JXaAf6j8K6GyNrJaH7UDux8pHasfxeI3a3MaNmO3VGfPBOcj9DXLGo5VGux9Ji8HChgaU1a8t+/zPRtJvmVkeKRgrAEFTjivQdG1CZsH7RLkDj5q8h8G3yXWh229xvi/dtz6dP0xXpWizIceW4P0NWzxD27T2L6bbMeS0KE/kKKbpRzo9l/1wT/ANBFFWBV8TTpbeEtXnlIWOKxmdiegAjJNfOGreMdPutLkSC7hIyp2K45wwNfQvjrj4deI8gMP7KuuD3/AHTV8aJb2k6ZaFAe4wRUtpbm9KjKp8J6tFrFu7bkPfORTJrw7ZPs21Wmdd20lcDcMnp+P8q8xi062J/dzPGf9mQ1ZW2uYlHkatMmPWQ0cyNXg6p6RHIispUQSRxfMsallG4Z55HuRS20hkhRmdSIpGxk8scnJ9we1eXPrGqWuoi2bVn2hNxOA3fpV+PxBqSjI1GN/wDfiFEmmjKOHm21Y9StZVhhCqwBPJwasC8Yf3TXmEPinUF+/wDZJfxKVpW/iuPcovYWtw3AkVt6fiR0paEypVIq7R6Et6O8YNO+1RnOQfzrmUudygq+QeQQad9qf++350WMjpBdIjgr/Oni5iY5bOT33Vw154ptbKRo5J2Z16hBnFQL44tv+mv6f40coHoy30MfTf8AnmrMOq2URLhZFPdQQVP4EV5n/wAJvAenm/8Ajv8AjQfGsA6iX8h/jRysDutU1GG7uzJBAsCYA2r/ADrIvppGhzEjyhclo422s3HH4ZxnHOK5Z/GtuSR+8/T/ABqL/hMbfOf3n6f40KLAh/t/WbZ2F1YQq+47Vjg3AD055P5itGz1bU726gWLTI2jcEySRRmIxnsF5P4k8VVbxnC2M+afbg/1oHjaEjAMhHplf8aq7JsjvY7nZAizSK0gUBiO5pjXwBOMfWuDPjO3U87+fdf8aT/hMrf/AG/zX/Gp5BndfbmPcflQ90Noy5LZ7HiuFPjGD/b/ADX/ABpp8ZW/cv8Amv8AjRyjO4a6H8PAP41Gbk+tcT/wmVtuwWkH4j/GtC01eK/jL20+4A4IzyKfKB0ZuSO5ppuRg5JrENw398/nVHUNYhsEXzmZnc4SNeWanYDpHuQe9QtcjPLfrXEXXiW6VsLFBF/10l3H8hVCTxFePnN7GnskOf5mloaqjUfQ9Ba6UD7wqhdpbXUivKx+XHA71wF3rl0E+TUbgucYwoUfyqZpi4zJd3Un+9KR+goukVHD1JOyO3intbKNhGSqk55PSud8V6lDdeH9UjSVCz+SoXcMnDgn+dYMhgBOYyx/2mJz+tUdTKnTJNkSLnHRRnqO9HMi3hKkU2+hglB0yCaCvFMAxT9/qKo5SW1kdJPLXJ8z5cDrk17R4X8EQ20SPcYW72BVDfdX1APr714ojGOZJF6qwYfhX034d1/T9Y0m2mcxkyRg/MdjDjpu6H8cVUbEyvbQltrS80xAsiyLH2P3kqwdSiH7q9t2EbfxqvmIfqK2rVQmBaXWwt0im+Xd9M8H8DS3FqoJ+2WAUn+OP5D/AIGq5U9iLnKDwx4b1VpoxaW+4NuV7Y7Gweeg98jpWJqvwisrxCLO/kTuEmTOD/vD/Cu1l8P2V2+6K52P284FCP8AgQpraJ4gsl3Wk8k0Q6BwJl/Mc1HKxpo87j8F+KNAVzpUQmVkCubaUEsB/s8Gq8niXXdOHkarBIVHHl3URYf+PA16PHqmoQPtvdNDkdWt5P8A2Vv8a0E1a0nTZcB0BGClzF8v9RQO555Z/EGCTUIry+sIZZoSCrcryBgEgHaSOxPSuztPiXoU0jPLE0UrjBdef5VLc+EvDeqqXk0u2JP8dsdhH/fPH6VgX3wo0ufJsNQubdv7syiRfz4/kaQ7mlDcxaqzE6/aNLDZzW1rIQyyu0gHzSE9wABx9amXSbj+z799P01bO0aK1R7eF0dpmSTMjgKTn5ePU46VxV18M/EFpzYT292oPASXYfybArOdPF2gsDLa31uo/iVWwfxH+FGgadD1qLWYrK61/XtjvbyNBb2nyEGdwuNqg8n5mx+dXtPsrzRNLi2rHd3tzP5l48svllmbrjg9OAB6CvIbb4k65AQLiVpAjBlEqhtpHfnvXR2nxfZ9n9pWME4QhlJXbtPr9aLE2Oi0CztLjxJrNxe6XI0sWpStHf8A8ChQvy5Bz+BGK8f+JMy3XxY1d16RRRx/T5BmvS/DfjGwOqt5F/NDbXU7yzW86IyFm5JD8Fecdc15J4svYtR+IWv3dscxSTgKfUBQKCkYtz5yhBbruZjjntU9xA9rIqyYLMoJwMYPpTS5R1ZCQRyCDyKWSZrh2eZ2d25LOck0DIXcAcdayrx2nvUiySF+Y5PftWlI4UEAV1nh7wzpepeH7e4u4N0s+5zMhIYDJwPwqJSUVdlxV2ccSFReAWxSNI2z/wCua6668E2+4i0v5OD/ABAH/Cs6Xwler0uIWAPHGM1kqi7noOcGtUYq8pghzx1DUhQMp+aRfYkf4Vv2nh3UfOCMIdufvZ4qbW/DtxZW++P7POcZxC7MfyxWMsQozUb7m0MPCUHLscNe4jbAZifc1VVjtA981Yvra5ikzcwyRFuQHXGapjO7Fdq2PKla+h0WkRCVRIw4Xp9a2JJBFCzt0X9TWboa7rYgdm5NOvJTdTeXGxEKdW9aXUgiSR54yHP7vdknux9K9++A7Z+CPit+Bm+uzyOP+PaKvBSVhgaZxtjjHyj+le6fAlm/4UD4rdRl/tV4QPU/ZYqBlBpgGJRhKyjBC4A/GqetaxqemWKtoumi9kZ8MHk2gDqCcelZdxLqOARFLBgAkZAA/Kqn258ASr8xJwS5Pas9SjkXuLt9UvDcr5UzykyRJ91WJyQPalmYtpvz8FZen1FXrw3Nvq01xFCWjmIJkKZGSP0qrdwSCxmaQjJYNx2rF+7PU+vw69rgXytu0XddFoZ+MjrilQDdgHjFQKOeealDY7Cu4+KsWIpOx/MVYV8dzVMNT1fA60CsXlfjqajuFWVEDgHBPWq6y4HNSI28YHrmgVjtvhhZQw+Kknht4WlWymZfNj3qDt6kV7Hc6dBqmkuiyRtdvci3s5bFPKDMygksMchRlvwrxbwnrl14T1e01i3tln8sMhRwcMGGD/Ou0vvjBa3qwJceF1xbsxi8qd49hPBxtx1qGnzXNo1IeycGte53lpbxw+IUsW00zwxXc6eYUjxKFhBCdc5zz2HvSTBZte1SO10pEIiktdP2qu0yogd8r/e54PTj3rjdP+MStfweZ4bjgPnl1l813IZl2lsdyQMVfm+KGjW+sT3L6dsvX3RSyLDLleME9cA8AZHPHWmZ6Fy7DT6WPsUcrxQT3EgeZBHL5Uapl24+ba5Ybe/4V0Gt6baW+rS3enoVa1sJLyRB/q2JXbGNnQdGb8K4c/FPR5o47SfQ42trMj7Kfs7tx3yCfXHXOauyfGCCSG4urXSGlnkRYZN8BAbrhT82cct2oDQ35bR4fC9+sU0kzafdKy3Il2uqFEZjx9/buOFP0q14rvTJ4Q1y2cEPHA/8PBQjivOW+I13GkMB8J2CwQMZIV8pvlJ6nBNLrHxK8R+I9LuNO/stYI7hdkrpFyV9M54qk1Z3RnNXaadrHkN8irfMFAHI4AqbzGye3Jpb63khviZVIIOar7yq+9A90SlzuqJmOetN30wtQOwOzY96gbcaez9KbnJpFJam1Y2LSWQlEoGBjae9YOvCRdfMBBCbUGPXjk16fofgC8utGi1S5mjhtigfYOWI7ewrza60fWptau7mXS78oJG2fuGOFycdq5qUZKTcj6HNMRQnTp0qW8dzM03VJ9HvHaPLRMcSR5xnHf611Nl41sFdS00ls/qQcfmK5K6sbrzn823ki3Ho6kH8qrGwYdv0rZpM8G595+EZxdeCdDuFfeJdPt3D/wB7ManNFVfh8u34Z+GF9NItB/5BWiqAXx8cfDbxKf8AqE3X/olq+KIr8x8Yr7Y8eDPw58SADJOk3XA/64tXxKQB1H51EjvwkpRT5XYmW8wKlN9mi2ggmjwyAEHk5qc6dbNkqWUeuc1PKdvtmtDFvJx/aCP1ypFWhOHjUAGodWsxbrFMpYruwc9qIlAjxn6Gh7GEHepI0obOWVQQQB65pLoS2lqwL5+YZGeDVNZpkXCykD0qKeaWRQhYuSRgY5NZpSuelKthY0mlF39Tu/CN88+gqshJMTsgJ9O1bZm4rA8P2j6dpEcUoxIxLuPQntWmX610HzMt9DhZrG/1LW7iDTo3nlLsdg5q0PB/in/oFS/mv+NdV4EP2f4lFenmBv5E17RjvTJufOH/AAiHivA/4k8/5r/jR/wiXiljgaRN9Ny/416t4z0iynuoLGwjmGsatNhZFupR5MY5eXaGwMDgcYyasapbRTeKrHRr1rmXTbfSXuPs8cjhpXVwoJ2kFjjoPWi4XPHj4Q8Tlvm0uQfV1/xpG8HeJz/zDH/77X/Gva9Evof+EbvDpl5dzi0naIfboyskJ4PlndycZGM81e0a7urqZ1uJVlATPEewg5x6/wD6unXNK6KUZNXS0PC7Lwv4ntLoTDSZJOCCvmKP1zUL+EfE0kryf2TINzE4Dr/jXovjjUL7WpGfTLh4NL0q8hhkkjYj7TOzgFQR2UH8zWhrtvv8Sao3iGz1O4tFt0Om/ZVkKKQp3/c6Pu7ntRcm55R/wh/iU9dLk/77X/Gg+DfE4/5hUv03L/jXrmmeJLuH4d6PNlrzWdRTybSN+WkfJG5vZRyT/jWHZpqOj+CfFKNqsst2mpeULhnIeQ4TcqdcMckAD1p3C558fBnifP8AyC5v++l/xo/4QnxQemly/mP8a9Y8NSw23iy9Gn2l9p1rBYK8un3JYyzOTxIqEnoBjIPJqbxHrjSav4Vmhs9QCyXE0rWoiKzEImOUz75oFc8ek8E+J0jLvpswQAknAP8AWtHwbEI7e4lJbczBTzxXo9pq73eoeNNTMdxAlvbRxJDcAqyFY2J+XtniuB8PR+VpYOOXYmgZueZXEateSXHiGeMNgbhED6AV1+41xmvWc1pq7XKgiOVgwcDoe4qZbG1CSjUTlsOTTfNQvuAA96qXVqLdhg7hTPNl5/eEA0nOfmYn61glJO7Z9DVq4ecLRhZ97kTHfIM+oFaQkwvFZ9tGJtRjjB4+83sK3orO2/iDE+hNa20PMhPlk2Zskm4VEkT6oVsYiFaTozdOOf6VqXQtooSyRr6Yx3qLQ+dag2DH3ucdPlNK1mXKq3CWvQzZvB+qxE7YVlA7o4/rWfNpN5bn99azJ9UNep8Be4z2zTWt5rpxFAGIJ5A7VoeVY8o8jHUYPuK9H+HWq7LWXTriPciHfGTkYB6gGt1NCt7WNXvhGznnavP4VOtx+7KQQxwxEc4Xk0XCx0NpdyQj/Q7sqp6wzYKn8+D+lbln4qmsgI7uGWBT3iO5D/wBsj8iK4NbhosAkdOmeauW2qlRhJMA/wAJ6H8DTuTY9LttY0rUcCRLd3PeJ/If/vl/l/I1eWwslYNFqE1mx+6J1Mefo3Q/nXmC3ds5HnQbT/ehbb+hyK0LPUri1x/ZuqMgP/LOQlM/zU1XMybI9Jk07UfLDZS+jxwXRZQfx61lzxWaki702SE92tpCP/HWzWDb+JL6zIe7shj/AJ7W5MRP/Ak+U/iK27Xxta3Khbi4JH928hEo/wC+1wf0o5u4W7FVtJ0u6kza6gkUvYTqYW/76HFDaLr9qu63lkni7ZAmX8xzWrt0jVB8kCsx72Vyr/8Ajj4aoP7GW2kH2DWDaydkuA0DfrxRoxamV9v1C3Yrd2CPjvE+0/8AfLf41Imu2ynExntT3EiED8xkVtOfEsEWZok1CH1ZFmBH1HNUJNTsWO3UdE8pu7QOU/Q8UrAVJbfSdXX99b2N7nqSilvzHNZN58PvDV2CRYyWjE9beUjH4HNbD6b4Zvm3JcS2r/8ATeHOP+BLUkfhm6I/4lGspMOyrcBv/HXpWA4a7+E9scnT9ZZD1C3EP9QT/SsC7+FWsWwka2it7ssdxMMw3Mfo1eqzaf4jshme1SdR3MZXP4jIqsdSli/4+rCeM9yhDj+h/SgdzwnUvC2t6cc3ulXkC9i8DAfniscxsvByCOua+lItbgHyi6aHPVZcp/PikubTT9TTN1ZWN6D3eFHP54zQO58v3pZLdtvU8D8a9E8KPInhO1ilQLGikAheSMnvXWXfgrw5qfikWqaMkcVpb+dcLFK6gu5wgxk44DH8qsj4fadFAYrG81CyQ/wLKHUfgwrOpFzVi4TUXqclcFn3mE7QSRVeOKRpSzM/+yuRXa/8INcqmI9WimUdBNa7T+an+lZk/wAPtXluWnh/s+4O3au+V0IH1x71gqMr6mzrRsYsMF07FvKZE/hJ4J/Oq9yZdzDOCvB3etbEnhjXNMR3vrFYUbhZIrsSKT6Y6/pVCe1uAfmgdyVBwOazlDlkawnzI5S+0mG9ke4lgZnI5O8jGO1cfqdokFynkoVRsg855r0zUi1tYyNLGIsDAyeT26Vwr2our5ZZpAyLzgVvSbZnVsNsHkS08hQVLtlj7VdihGMdEXkmhEzISAAWPPt7VDe3IVDFGeO59a3Ocq6teCSNY4+Fz09BX0J8AQT8B/EwBwTd3eD/ANusVfNNw++4I7LxX0t8A0EnwG8TpkgNd3YJHUZtYqYGVHZx3LAXMskuONqk4P5U6fw/azRhVgVCp4YkkircNt5YVUcgKMAGrwjwu4nA9+tZFM4691VPD+roZ7RZIpE2lA2Dkdx68GsrxRq1hqult5FqLaVQTkqFYj0962PiFZrdaPCx+VopfvHtkf8A6q87NjqJGIT5i56LIP603FyVjpw+JjQnzON/m0Q6dax3lx5ctzHb/ISGfGCfTkgfrV9NGt2Rj/bVku0ZAbdz7dP/AK1Y629zx/o8pPQ4WtTTvDOu6spOnaPfXIXvHAx/pWpxMtReGppbWSaLUdOfy3CbBP8AMxPTAxzTk8LXpt4ZZbqxh80HCSXIBUg4wfQ1l6hpGp6TN5Wo6ddW0n92aFlOPxFUGMg42MAP9k0CJ2k2EjqQccVNZTj7Sm8qOeNxwD7E9qzskZyp/KnA/L0oA9ysvEeg6d4bitxYSi9kHzrPbiWMDHUFcg1ltrVqxPlRWi/XTz/8RXkiyYI+YgegNWFnOBh2A9jVXM/Zo9TXWZMYUWX/AILz/wDG6k/tW+bHlSWa/wDcPb/43XlYmYg4duPc1qWxhCgyLI5xn5pSP5UXD2Z6TDqOrMQPtcIyeNlgw/8AaYrRu9Q1fSWiE2pyBiobC2irj88V5Y91aBT/AKOu7/rq3+NQXkEUoiaxDzsyZkVFZth9KVxqNj1O58a38rAzarcnaMDakSf+z1Anj9IZALyeS6jzlo5LqPB/ImvLRp1++NthdH6Qt/hSnSdUYcWFwP8AeTH86LsXKnuza8Xara6hqbvaCFUP3UgLMB9WIGT9BiubZuv1q2NG1LGWgRB6vPGv/s1MOk3A+/PZp/vXA/pmlqUrIrKwVwSNwByR6j0rS/tHSUhVY9HEjhuXlmb5h7gHFUzp6J/rNRtF/wB3c39KZ5FkrfvdROM8+XAT/MigehqR+JILZj9k0SwT5gwMqGQjjGM8cVRutTl1S6ha78tBGNqiKJVAHXoOv1NUmaxTpJcyfQKv+NPsTbzXybIGKKCW3uTn0pWGnbVHe+E7671LUIbbzp5LS3+chidq46DFdy67gfJdSfauW8EwKqzTlWWPiJRGvc//AKq694XuMKm1D2Yqc49aysk9DadSdR3m7lXyop023UMcgzj5lz/Oo30+0ViIbZS4wP8AVKR9Dn/GrUdkUkPzmYA/dk559f8AIqRnMP8ArUIGewz1+lFiD13QV2eHNNXAXbaRDAGAPkFFO0U7tA08jGDaxnj/AHRRWoFTxdn/AIQnXMZz/Z1xjH/XNq+YV0y01CLeYopgf4sDP519SeIxu8LaqCM5sphj/gBr5o1HQ91w1xZhonY5PlsVIP4VcY8yFzWZmnwnbKxaJZI89lPH61C/hVtuI7mRf95Af8KtG51qzOFuywH8M8Qb9eDUqeItSQYnsbeYf7DlT+oNS4MtVH3Ma78Hz3MLRNdIVYd4yP61n2ng+QtNDJdAtAwU7YzzkAg9a7BfE8BH+k6ZdR+6hXH6HNVrLxDpI1q8abzoEkWPYZIWAJAIP9KlxZXtJX3MWPwbGP8AWTSN9ABWha6BbWTB4bf5/wC+3JrpE1jQpAMX8K/7wK/zFWY59Km/1Wo2bfSZf8anRClKUt2c/wCQ46qaQxN6GuqWxhkXKTRMPZgaX+ygegB/GquRY461Mej+I7PWJm2RxSASk9MHjP616fD4lt7mya7tQk9spIM0cmV4684rnX0beNrRhg3bGajXRXhXbFujX+6pIH5UXQdBsg1KXxdeaxpmowrLJAI1iuLMyeTEvUKdw78mrV6b2e4tNYt7lYtSsoTDLP8AZ98MiNg8pnI7EYNVBpMkP3GdQeuCRmk/s6ZBgM4Hpk4qrx7E6mxoAXTbW5hkSS8ubuaS4upCAgZ+A2F7AcDFW4tVsrONlitigcEt8+Dj/wDUfyxXNtaXHzZkfnr8x5qP7LN0Dt6dTS93ew7u1hLvwfpN7pa2mlfbbaJLhJWH2tnTO7JG0nGSe/rWpf2GsSaje3GlahNaQ30YWeGSMTBMDG6PJG04+orLa3nHBduTn7xoS1n6q7j6Gi6FZmiPD+iiDTon0+7LafB5EEiXbIwXv90jk96pReF4LXRbqyia6WSW+GoQzs4YxOCCn3idwGO/WkFrc/8APST/AL6NH2Of+KST/vo0XQWZqaZFPaaxNrGqyNfX8kK24ZQkaRx5zhVBPUnOSalu3N34m0/VjBJus45IkjEi7WL8ZJ7dKxfsMhzln/M0HSiRkknn1o5kMr65f2emw67prSMdR1mcSPFuy0YYAdhjAUHv3rBgtPJhVEUhVGBXTf2Sqtkp83qRzTvsKL14+tK4znBbueimkexMiFXQMCOQRnNdC62kQ/ezwp/vOBVd9Q0hMg39sT6LIG/lRcRysvhe1kORCYz/ANM2IH5VSuvC0MFtJL5ko2IWxgHNdg+t6OB8s5f2SNj/AErO1PVoLjTpo7O1uJZGACgx7c8jPX2pWuWpyWzMK28HrC5f7Q5ZsZOwflV1fDoz80srfkP6VpHVbt1/d6cqH/ppL0/ACmefqk3R4Yh/0zjLH9arlYvaPuV18PwAfNFu/wB4k1KlpFFMkUIRXY4CrgE1Zi0fULw/vDdT59TtX9MVfg0h9KX7VJHGnl/w5yTnj+tVy2J5rjbbRsgyXT7FHvyama8gt9yW4C54yKgnu5pep+gHSogqgYfGR1qCgZy5MjuPUse9PYjcB/DS4RVAwdvAORkimOwALbcHHU560APdY0yZCOnpzUf7t+T97PAHQCoVkuGnDMuA3YCrAIySyqozwx4NIQuVGfLc56gZyMUqyTAkldwHcf8A16YNiuAhUKRz9aC+9m3/ACjsM0XAuW+pSQPmKZ4yfQ4zV0asZObmCKbPViu1vzGP1zWGdoZcjocEe1B4/wBX8vPY0XFym/8AaLGT7rz27fhIP6Gr1rreq2g22Wq+ZH/zzZzg/wDAW4rkhI68j5h15qcXIVcMp9adwtY7e38Y3lowa709Af8AnrBugJ98ocH8q27fx7aXChbiaZR023UCXC/mNrV5rFetGcwzFPUBsVN/aOf9fFFL7ldp/MYoEeopeaFqPWCydj3trkxN/wB8vgfkaWXQ9KI3rcXdl6G4gJT/AL7XIry8TWcn/PaE+zBx+uP51atb26tG3adqhjOeAJGjP+H607sLI9KttK1VOdH1iOcekNz/AENOuLnxFa8X9nHcr3MtuGz+Iri7Xxdrlp89xDFeD+9JCr/+PLz+tXYfiNiTLw3Ns3/TrdED/vlwwouFjcOqWMny32hovqYZCv6HNNNj4XuuTHdWrHv5YP6jmoIPH1lc4E14je17YA/+PIf6Vdj1jS7rpa6ZOfW2vvKP5SAVWgrMgtfC+lRX091p+sRrJcBQ/mbkY46c5qzL4a1ZlzZXsE3p86t/PBqXZp5Xc2narEp/ijjWdfzQmoTNoyttGrLbP/duoXiP6ijQWpnT6f4lsv8AXafDKv8AeAK/yJFVZdbvbaIC50eaPj+Fuf1Are8qSb/jx1a2n9PLuh/LNPnj8RCPrNIo9w4/rRYRxs/i60RcXVpdIP8Aai3D9DWNf+K9BlBLqVP/AFwYfyFdbf2+ovnz7RW9d0AH9K5XUbP726xiU/7mKOW41I4HxDrNtdb0062nnZuP9WwH61zAtpYObldsjchM9K7bWbyPT0YyBEPQIo5NcbcXBd2lkOHb/wAdFJpLRFp3IriYW8ZUfePX/CsWef5jnk9fpT7285yef7orMYs8nOfmpDJEO4knnPNfT3wAZU+BXiRnO1VvLok+g+zRV8yKuBX018A03fAjxMvGGu7oc/8AXtFQBVOs2sf3dz8cADGfxNWLaea9UyBBEpOF5z+WarWmmWMGCsQL46tzWmqK64Y5X0zisy9DI8T28b+HbvzXY4j4JI5OeOK8sBbaApr0PxndLHb29snSQl2A9uP61xymPtGorSC0M5PUx9PhtYPFumvrO/8AsuS4Q3GM/cz8wr6tMllb65B9n1L7BpRsUks/s8oSA4chxj7rZDJ7jtXzrvjks/szxI0Rbdhl71saDe32ixqlhfzRQhtywEh4wfUK2QD7iqFfQ9j0zPih00rxNamW3msGRTI4LNNDKUkdSOVPIP4V81+LY5vD/i7U9Ls9Qknhtbho0k3dQD39x0/CvSNT8WeItQtzENZeAEMCYIUjb5vvfMBkZ7+tefT+G7YykvNISTk/MDk/WgnQwhq1+D/x8ufrUq6xfY/4+G/ECtVfDVuTxLJx6kf4UL4bhDY8xz+VIehnR6zfHkXJI/3R/hUn9sX563B/If4Ve/4RyFhjzXUe2KlHh2Ef8t3PXg44oFoZ66vesuPtLY79Oa1ND1qz0mRnudLhvS3/AD0x/hQug26873PpyP8ACrDaVZuEB3gqMcMBn9KBOz0LuoeM9MvoDGvhmyiOMBhjI/SuVF20DubVmgD9QjEfyrdGkWHq5+rj/CmnRNObOV3D3c/40agkkYD308hwbiRj05cmo5HkwCWbrjk10yaPpaHIhAPrvJ/rUg07TsEGFWHbPOKBnJAsYzkmoyDtzXYrZWSZ2op9itIxt43yEBPsMUBc49LeWXGyKRv91SakGnXTYAgf64rq/PQH5IyMjmmiZjgKn6UBc5238O3MzfvisKerHk/gK3YtHsrNFS3kLfKN7OOS3+FWgtxKv7uNs/Sla0uWXLoEHcnimkJs6Twpqa2e2yUI3mvkFmxiuwztHynD9s8Z9q8lVxayAiQM6ngL616paF3s40lT5mQEg9Scc1ElZlxLDZaMNGMgcggYOfao1X5lYlssuAc9fwpk0flB/K5YDIXqD/gKiJlkAaUFck58pv8AJqCj2LRc/wBg2GRg/Zo8j/gIoo0U50DTz1/0WPr/ALoorQCp4vfyvBGuSDqunXDflG1fONl4m+QCZVce4r6M8Zf8iJr2P+gbcf8Aopq+R5EZAWjOPY1UdgPQYtV0y4AEo2fhmpDY6VcjKSQkn14Nebi+kj+8CKmTViP4qd2LQ719As2b5JFHsHqJ/C6Nyr5/DNcjHrbg8SH86tw+Iplx+8NPmYrI2n8J9cCM/UYqB/CG7rbxt+Oahj8U3C/8tfzq3H4tl/i2t9RRzeQ7IqHwdGDn7IQfUUo8LtH903af7kpH8q008WDjKLVlPFcR6px6g0uZdgsYo0i/h5h1C/jx6TP/AI00rriN8mrX59y+f5iujj8T2h++p/nUy+JbFvvKP++aLx7BZnMLd+I4iNur3I/3kVv5rVmPWfFUXKa45/3raM4/Na6H/hINMb+7/wB8/wD1qP7c0094h/vKf8KXuhqctLeeIZ5WeXVQzMckm2Qf+y1E91rSKWk1KNQOpMC/4V1Mmu2KH5Vhcey/4iql1rVhdQNDLbxsjcEAY/Wi0RanHv4nn3Hy9XaUA43JZrg/nipbfxJdu+1NXaNyDtElooDY9xkVPJoHhksSLaYZ7C5bip7PSvDFrOs62kpdem+diOnpU6D1JfP19ump8e0Cf/E00Nr7H5tUl/CJB/7LXRL4j0xetuCPQYH9KcfFGkjpZN/38/8ArUe72D3u5zhh1phzqd5/wHaP5LUZ07VpD82oX5/7aN/SukfxXYf8s7TH1aoG8V2w6QAfjT93sKz7mIPD1zKP3s9631mb/Gnr4RjYfPDI/wDvyE1ov4tX+GFPxqF/F8g+7HGPwp3XYLPuQx+D4E+7aRD3OKsr4XQL9yJR9M1Rl8W3J+66r/uqBVSXxPdOOZ2/PFHN5BY308ORKPmkAH+ytP8A7J06HmabP1cCuRl12Zj80rH8aqPqzE/fp8zCyO5LaNb9ERz7sTUT69ZQcQRov+6grhH1Jj3qu1+x6HNK7HodtceKGIO3P4tWb/bEt7dJBIxKMTlR34zXMGaaToDj3rR0yB4rqKWQjPJ57cGlcEdBtUjcc4BP4CjaNh6A/wAI659/zqF2O47VbJGAcDoe9PH3cnJJ6YOcUiiVfnmHJII5OaUfIzDc2M9DTORjDDjuR0pVPljc3JC85GaBD1fZHlRkAcYFNJHy4XGARyc0gDrIvK9CeOBjpTWyDzz8vB9DSGAXG0tnjjJHWgxgSFwN3OPpU0fRfMzzxkc8+lR3c62m4THyyFDBGGD7cd80gEkOVwRk46g4zTFUgMWQjjjmlRy8SsilWYcgnGO/vSLG7koUyc/ex/WgCCcXQTECgj1ApLRLppA052oDyTV9YBtALkH1z0pdpEhbKkcE+3vSAY438Fe2MGodu0/xLx61M8eHykjAryfcU7cqqpZiTg44z+v40rARgTdgGA701pWHVT+FSpIGO1l256D0pyoHALleTnA6/wD1qd2KxAl60ZBR2U+xxUx1ecjErLMP+mqB/wCdSPbqFIcLnOOvTFVzYQS/MvyjuUbHP8qdwsSi/tX/ANbahfeGQr+hyKkEtk/3LmaI+kkYb9QRWdLpzrkxz854V1zkfhVZoLxV3iMOucZRuv50XFY6GGa4hcG0v4Sf9iYxn9QK1YfEniOBNq3N5In93d5wP4c1wT3vlSbZso3owxUqX/dGx9DVAdx/wmLlgt/ZWErek1mqn9ADU8XiywUlhpiRn1tbyWL9NxFcUmtXIXaLiTb/AHS2R+VB1dm/1kcEn+/Cv+FAjuJfGVuI22T6xBx0W8Dgf99LXIa54nhmDBr3UpPZnUf0qjLqUDKd1jB/wEuv8jXP6ldW5J22qJ/wNz/M00wMzU9R86QmGMqT/HI25v8A634VjXl5hcZz6CpLy5GWbgDooA6msoku2W60MaGNukfcx5qZE4yTSAVIBzxSGAXmvpv4A8fA3xIQf+Xy6/8ASaKvmYD1r6Y+APHwM8SegvLr/wBJoqAKVvNM0pTC5HGTmrbqzJ+9dsAfwcZqCYhVEynC98Vbt3Eke4sT6CoKOP8AHFkYY7O+gTCIWSTr35BJP0NcxFdwP98YPtXquo2K6hYy210pKSLgnHT0IryPU9Ol0nUJLWfqp4I7jsauLtoQ0X1e2PIlI+oq3FNHgBbtB9RXN7x3Jpwb0Y5q7omx1AfjK3MR/Go2tTK2fPh/77rnBIwHDZH1pwlcdG/Wi4WOi/s6YjCvGfpIKcNIvDjYFP8AwMVzYnk/vt+dOFzL/fb86V0Fjo/7D1Bukf8A48P8akXQdSJx5Wf+BD/GuaF3P2lb/vqnfbLj/nq3/fVArHSf8I7qjfdhz9XX/GlXwtq7f8so1+syD+tc6L25/wCer/nTTeTk/wCtY/8AAjQFjqV8I6oeWe0T/euk/wAaevhSaPma/sFP/XyP6VyRuZT/ABt+dN85/wC+aB2OuOg20fMusWK/SQt/IU02Wkx/6zWYj/uRsa5PzGI65pfMPrSuKx0zDQ4z/wAfs8n+7Fj+ZqJr3Q4/uQ3Ep/2mArnS+KQOD1p3DlN86zZR/wCpsE9t7E1E/iCTH7mGGP6JWLu7Cnokj8KjMfRRmlcfKi9JrF7J1mZV9uKqPcSSn55GP1NXYPD+q3Kjy7OQA93G3H51v6b4AuGcPqU4RQfmSLk/meKnmQ7FDwlo41DVFmuATBCdxH949h/jXoyRF5jK87GPn92nAFRQ6daWlhFALVQsTHYScFfenhXDZVvMTpsJIb8Mdai9ykrEqbVLMD8u7qOcU1iNzDORjHTrSh9xKsWVsfdfjGf/AK1MC5z3x95DjP8A9egZ69ogx4f08df9Fj7f7Ioo0T/kX9Ox0+yxf+gCitBFTxlz4F17/sG3H/opq+TioVfm/PFfWPjH/kRde/7Btx/6Kavks7mJ9fXpVxERlgvVdy+hpywWk45G00EEA+3U1WdSGypINDAtnRYn5ilxTDoNz/yycN+NVRfPE2M9KsR6s6fxGpAY2lahH/AT9KiaK9j6xv8AlWpDrzADLfrVtNbif7wU0aj0Oe+0XCfeVh+FKL+Veua6Zb6yk+/Eh/Cnj+ypPvwj8DRqBzA1Nx/+unjVG966Y6fo0vRWU/hTv7A0aRcrMQfdaVwszmBqjd807+1Tjqa6T/hFdLf7t2o+oNSL4HsJcbdTgXPqcYouGpy/9qn1NB1U+pro9Q8DW9k4Eeo210CM5gcnHtVAeF4d2BL+tFwszL/tQ0h1M+tar+FolbBkP505PCkTkgyhcHuaLhYxzqh9TTTqZ963v+EStl63C0f8IzYqfmnWldBZnPnUm96adQf3roxoOmp1kz9BUn9l6VGO5P0ouFjljfSE9KT7RO3RSfwrqDDpydEB/GmtNZp92Nc0wsc0PtUnRG/KnrY30o4Rq3W1CBM7VWoJNZC52nFGoGcui3jffIUfWnrom3/WTflTpdYY9DVV9RZj/wDXoEWxp9rH99i1Nb7PGMRoKht1lvGwGCqDySa2rbSoYl37tzj+JulAFS0s2lbzJgFVeVQ9/rVoEi5V5Mbfz7U25nLXB4wexHGaijlZpVBGcGgZqK4K7/ulid2RQFXeeo7btuCc84qERErlkGxjjAbBOOvTtTlKxkBmZtzd+ccZxQUWSQyhQWxxhg3al3ZADBtmMMMfrSQ5kUSjlMYJ7H1qVU/ebsjaOvvQIaiIQHGBjpk9eOtN2u11JIJCIyAI4wOFA6k+pzmnSL+5DKBt5zilXKYzyW7gdaloY1JWhdXY4VHXpkdPpzWNdX0lvfS6nqRae4mfy7ONhuwgOMA/XNasiytKpO1kByYicbqjuWTVf3ZBjS0kwjBfmL98fhxWUruVrGsbKN7kroroLkbVVV5jDdPXmpIYyMLu+XOVH93jpUBssyuYpGTJyVHPsDVm2j8iExM5fnO404QcdxTmpbDmY/L8pO4np2qOVmCnHIx1x39KlKsVVUBbvjP3fpTiyqqguFbPTqDWhkVISeFcBgcZxn6VaxuGQOO9OVcLuUDPoBzio1kY7Ttzu4GOufegBXKqqk84Od2MUqOCDg5J4xjuKZ5HmSHehO4Y45FTAxBRuXDZPekBHhnO1RkNznNOVTkq209io7U03SQktnqcAk4rLvdXEEZK7SO+PSiwGjJMFYYONuQfm9arSX1tGrZkwcZ2Zwfyrm21lpJBt+X/AGQe1Zs3hqbUtX+1R3BUyEbtxOR9KLAamnxJrWvG9nH+h27YjU/8tH9foK6VrC1nyZYFJbjpgj34qG2t4tPsoraNAAq4Qd6uRMflBBAxnJHSkBn3GhQLnyZJImzwA+R+tUzo12D8s4wem9P8K32c7hjGexJ6UwPtAART6kDpT1DQ5HWFn0m1E100RXIUbCcn8Kwol1DW0mbTbVpRHjd8wHX69at+N9Qa71KKyi+ZYBkgDqxrq/CuknTNCjWcbJJP3kgI5Gf8BVX0FbU8yu9Nv7Yk3lpNHjjLIcCqoUfSvcG2rnIBUcAnvWTfWWnXSkT2Vu3f7gBx9aLjseTgY9aeBXd3Hh7Q5nO1JIWP9yTgfnUJ8G2DMdl7MFHbaCTTEcZjB4r6X+AfHwL8Sf8AX3df+k0VeRReFtJg2vJ50y9wzgZ/AV7z8Lo4F+EOvR2sKxRiW4G1Rj/lglAHLoExhskGoI5ZreVljjL4GN27AwenWrJQqo4IpHQ7VlxwOv0qSi0yXcsa5lSIbuQqbiR9a4D4h6LPDMup2ryS/KEmJOcY6HHTFeiW77owhzx39qdJDHI22QBlYHJI60tmDV0fO51SeN8NGjfpU6asvG+Fl91YGvX7/wCHuhXshka1EUp5PkSFR+XSsib4VaYwcxXl1GRyOFYU+ZE2POhqsBxv81ee65qQanZ5wZiPTKGu1b4Qb1Pk6rgjkB4Oo/A1mN8LZpJnjj1e0dkAyNrZ5+maOZBYw0vrMjm5Uf8AATUn2qzwCLmPr3NXl+GGteb5JltFOPl3yEZ/TrTj8KdazxJav7iY4P6U+ZCsZ5u7XgfaYc5/vCn/AGi373EQ9vMFXT8K9fUgbLbnp+8z/ShPhhrZkbc9ogA6mU/4UXQWKP2m25xcxf8AfYNIbq14zcxf99Vpp8KtZdc+faE9T8zf4Vaj+EmogKZr+3UEZJVGbH8qXMgsYH2+yDEG6TPtmmnUrFT/AMfHfsprql+D4cp5mr4OOscHI/Wkg+Gun2twItVu7n5j8kihQh5/Q0cyHynJtrFmvRpG7cJTDrVvjCxyH8hXoVz8MtKbT5Y9O81LpxmOWZ8gfgO1T+GvCNnpunhNRsbO8uldj5iRlyF9DkfWlzoOU8+sLs6lfR20cflhursd2Pwq8+ga+948dhA08APyzFVCkfU161DAlt8sdpFaxkYAijX09e1PZWjWQSc7DkEc4B9KnnHy6HneneA9ZmmWTVbtbaLPMSY3Eflx9a9Et7a2gt4ooIyhQDggAj0PvSBXBVyyOCu0DBz/AJ6VMHRiIhvViQD7nvzUttjtYkZSoYFcqcFuhI9qlDkx7wRjGNg55pvmDaUkBIIxxz+Y70kpjDK6fJ1Byfpg46U0hDi2dpPCdM8UDCKrEY/hBAxk/wCe9Rr5rf6wAfLz7Hp1p0UkgHlugcgkBkP3hnANUASxLIpVkVl3dCf8elQfvYHK2xMyswyjDnB64P8AQ1Yij8pmJyeT3zkemaje7VcrExJPDcYX8TTA9f0Tjw/p+Bj/AEWLj0+QUUaJn/hH9Pycn7LHyP8AcFFWIi8S2k9/4U1azs08y4uLKaKJMgbnZCAMngcnvXz1/wAKl8bk86Jxjtdw/wDxdfTFFNNoD5lf4ReN/L2ponXuLuH/AOLqB/g/45I40LJ97yD/AOLr6hoouwPlOX4MePZGP/Eg/wDJyD/4uq5+CnxCH3NBP4Xtv/8AHK+tKKQHyQfgx8SdvGgc/wDX7b//ABynH4L/ABLEW5NDiLAfdN5CCfp8+K+taKAPkiL4Q/E4N+88NYHqL+3/APjlXoPhD8RC377Q2Qe15bn/ANqV9U0UAfMS/B/x+FyunKD6NdRf0c0h+E/xFTpo6N9LyH+rV9PUUAfMI+GHxGU/8gEkf9fsH/xdPHw0+I//AEAT/wCBtv8A/HK+m6KAPmYfDT4jd9B4/wCvyD/4ul/4Vp8Rf+gD/wCTkH/xdfTFFAHzQfhr8RT10I/+BsH/AMXTT8M/iL/0Af8Aycg/+Lr6ZooA+ZD8MviN/wBAH/ycg/8AjlRn4X/EYn/kAn/wNt//AI5X0/RQB8un4WfEY/8AMAb/AMDbf/45Sf8ACpfiKwGdFA+t7Bx/4/X1HRQB8wj4MePmXL2UC+wuoyf500/Bvxsm7fpM0xx8vl3VuvP4yV9QUUAfJ7/Bz4iEnb4fYj3vbf8A+OVAPg18SWdg3hzaM8H7db8/+RK+t6KAPkeH4K/Egs4m0AYB+Vhe2/I/7+VP/wAKU8fk86Dgf9fsH/xdfWNFAHy5ZfCDx7bZJ0LnqP8ATIP/AIutSL4XeNgvzaHg/wDX3D/8XX0fRQB82zfCXxjJ00XOOgN1D/8AF0xfhN41Q5XRMH2uof8A4uvpWigD52i+FXjFNznSlyeifaIv576B8LvGBkJbRSQw5zdQ8/8Aj9fRNFAHz5/wrLxgM7dHHHT/AEmH8/v0qfDTxl5rE6RtTHH+kw5z/wB919BUUDufPx+GXi4/8wfHGP8Aj5h4/wDHqRfhh4vWXeNJH0+0xf8AxdfQVFAjwE/DXxcwJOkEN2zdRHj/AL7pR8NPFhAJ0gq3vcwnH/j1e+0UDPAx8NPFiZ26VnI5/wBIiH/s1JL8NPFjK2NJYnt/pMPT/vqvfaKBHgkXw08VxxlRpIBPUm4iP/s9OPw28WMAv9lL7sbiL/4qveaKAPBh8NvFQXjS23dc/aIuf/Hqcvw58VqxJ0rPPT7RD+B+9Xu9FKwHhjfDzxX/AA6SOnI+0RYz/wB9VA/w48XEkjS2YHjabmHj/wAer3qiiwHzde/C7xw4xBoe4HqPtcH9XrEufg18Q7mQ50Tah7fbYOn/AH3X1ZRTA+WbL4LeNo/ml0Paf+vuAn/0Ot63+Fni+JN/9jlXHb7VD/8AF19EUUAfPY+GPjBef7I3EnJBuYf576jl+G/jY8R6Fge91B/8XX0RRSsB86J8NPHBxv0Q/wDgXB/8XUyfDDxlu3PpA56gXUX/AMVX0LRRYD5/X4WeJll8waHH5n/PQ3EO4/jupW+G/i8sSNGJz1P2qHn/AMfr3+iiwHznd/DPxzLGyxaKRkY4u4OPf79ZVz8KviG3MWgZP/X5B/8AHK+oaKYHyj/wqH4jlgT4f78/6bb/APxyrQ+FHxFSMhNBAPbdeQH/ANqV9SUUAfMI+FXxA8v59Ay4HBW8gxn/AL7r1X4a+F9d0P4c6zputWRtr65uJ2hi81H3K0KKpyrEDkEcntXpNFAHkbeCvETtzpx+pnj/APiqengrxAODp/Hp50f/AMVXrNFA7nk0XgvxDGwxp+Mf9No+R/31Vr/hEdc25+wEHoVE0fP/AI9Xp9FKwXPMT4P1sMStjkDp+9T/AOKpG8JeICfksh6ZaZOPwzXp9FLlQXPLV8E60ZG820Zhj/nsgB/XNLH4J1eADydPA5zhZYxnjvzXqNFLkQXPLpPBetTKQ9jgdRmZMg+vWkg8G67vPnWRC4wQJY/m9/vV6lRRyIfMzzKTwdrL7yLNlbHB89Ofb71RjwZrX2dFGnqrKOf3seD+teo0UciFc8s/4QnW13bLRkOMhlmTr3zzSN4N16R43exO7Pzfv04/8er1SijkQXPLX8Ha62M2LZU5yJ0G7nn+Kop/BeuThg2mZDfeVpo8Een3q9Xoo5EO547aeAPEcCtHNbM8BbKIs6ZjGOmS3NXE8Fa8gHl2PlnaM4mj5P516tRRyIVzy1/BmtsvOn5JULkTIDj/AL6qF/BOvvskFl+8U/MDNH8w/wC+sV6xRRyILnlKeCtefb59iQUO4Ymj+Y+n3qe3gvXAf+PDIJ52yx59+S1ep0UciC55YvgzXRIM2RO4kktNGQp/PvS/8IVrcjc2ezcME+ahx+tepUU+ULnmDeDtcZ8mz2nJIKzJge+M0N4V18MV/s1nwfkcTR8D8WzXp9FOwXPKT4O8RSsjSWJAA+756YH5NUr+DddVcJZAjOcCZAf516jRRYLlTSoJLbRrKCZdkkVvGjrnOCFAIoq3RTEFFFFABRRRQAUUUUAFFY/i7X/+EW8H6prv2b7V/Z9u0/keZs8zA6bsHH1wa0rO4+12MFzt2edGsm3OcZGcZoAmooooAKKKKACiiuOtfHst38RIvDX/AAj99a280E8keoXhEXmtCyq2yIjcU+YYc7c9gRzQB2NFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFYFn4n+1ePtS8NfZNn2Gyhu/tPm53+YzDbtxxjZ1yc5rfoAKKKKACiiigAorN1i71a2SFdD0uG/mkYhzc3f2eKJQM5ZgjsSTwAEPuRVDwl4pfxF/alrfWH9nanpN41peWyzecmcBkdHwu5WRgRlVIOQRxkgHQ0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRWP4u1//AIRbwfqmu/ZvtX9n27T+R5mzzMDpuwcfXBoA2KKhs7j7XYwXO3Z50aybc5xkZxmpqACiiigAooooAKK4vV/Gus+HpbK613w5Db6Pd36WRmi1HzbmDzGKxyPCIwm0ttyFkYgNnBxiu0oAKKKKACiiigAooooAKKKKACiiigDjvi3/AMkg8T/9g6X+Vc7f6RY638UPCNpqtut1a/8ACO3DvbyEmOXDwYDr0dc87WyMgHGQK9Nu7O21CzltL+3iuraZSksMyB0dT1BU8EfWmDTrIXkN0LO3FzBEYYpvKXfHGcZRWxkKcDgccCgDxOPwto7eAfiZLLZrK2hXuoppAkYsumhIFmX7OM4h/eMTlMZwoOQBjo59QtvEcmgaTd+HNP8AEWptoEOqTnW7ox2sSNhdwUxygyFt2SEBA6tyBXow0jTRbXluNPtfJv2ZruLyF23JZQrGQYw5KgA5zkDFVtR8K+HtXt7WDVtC0y+hs1KW0dzZxyLApAGEDA7RhV4HoPSgDwyz8QQRfCPwZDq7xX2mDXZ4JbGS9jEFxawmbZGZp2RHRMR7VcgNtUVeit4tJ+GXjzxV4Kh07Tvt15GIE0ueF5LG3UxLKGlgLqh2l5NqFtgYEc16b4g8M341DRb7wpHp8aabdT3EumTkwQXLSqQZC6I21wWZs7Dks3TrS+HPDeoweLNa8Sa6LKC51SGC3FlYu0saJEG+Z5GVDIxLn+AYAA560Ac5oHhe60nxl4fvdJ0vwz4c05oZY7iHStTeVtTj8omP5TBGJCrYbeSTgtzzzraz/wAlz8Mf9gi//wDQ4a6TSfC3h/QJpJtC0LTdMlkXa72dnHCzjrglQMiqc/w/8G3NxJPc+EtCmmlYvJJJpsLM7E5JJK5JJ70AdDRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHmp/wCSv+M/+xdtv5zVxkHhnTLb4U/DjVLSKS21S6v9Ngm1GCZ0umim/dvH5wO8JsYqFBAUYxjAr3T+zbH7ZNd/Yrf7TcRiKabyl3yIM4VmxkgZPB45NR/2Npn2K1s/7NtPstm6PbQeQuyBk5QouMKV7EdO1AHm6rpXgj4peILbT7MadoaeE11K6tNOXygXSWRS6KpGHKLjIwTgc96xbR1s/iF8P9S0/wAP6P4e/tkzkmwv/Nuby3eDf/pCiJQ/zBG3F5Pm785Ps/2CzGpHUBaQfbTCIDc+WPMMYO4Ju67cknHTJrO0/wAH+GdJmWXSvDuk2UiuJFe2sY4yGAZQ2VUcgOwz6MfU0AeEFtA8Zz3GlXlppE2u3niZmTWdRvbMyNbpc/LH5TSGflEESw7Npyp4Brsr3TLjxB8VvEemXvh3Q9ct9OtbVNNtNXvGhS2t3jG94YxBIOZAVLjBGxR0ArQm8EeLbnwHe+C7t9GuLS9eRZdalnkadkklLmQ2/l7TJgkA+bjOD2xXcaj4W0HWYbaPXNHsNV+ypsha/tknZBxnBYHGcDPrigDltL1+fwP8L9NGvXceu6uzva2cVjcmdr2QyMIolkYAsQu0MxHG1ia2PAvhy80HSbm41uZLjW9WuWvdRkj+4JGAAjT/AGUUKo9cE9609S8MaBrFvbwavoem38NqNtvHdWkcqwjAGFDA7eAOnoKXSPDWheH2lOg6Lp+mGbAlNlapD5mM43bQM4yevqaANOiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK474t/8AJIPE/wD2Dpf5V2NQ3dnbahZy2l/bxXVtMpSWGZA6Op6gqeCPrQB5lf6RY638UPCNpqtut1a/8I7cO9vISY5cPBgOvR1zztbIyAcZArnY/C2jt4B+JkstmsraFe6imkCRiy6aEgWZfs4ziH94xOUxnCg5AGPbBp1kLyG6Fnbi5giMMU3lLvjjOMorYyFOBwOOBTBpGmi2vLcafa+TfszXcXkLtuSyhWMgxhyVABznIGKAPOZ9QtvEcmgaTd+HNP8AEWptoEOqTnW7ox2sSNhdwUxygyFt2SEBA6tyBXE2fiCCL4R+DIdXeK+0wa7PBLYyXsYguLWEzbIzNOyI6JiParkBtqivc9R8K+HtXt7WDVtC0y+hs1KW0dzZxyLApAGEDA7RhV4HoPSsfxB4ZvxqGi33hSPT40026nuJdMnJgguWlUgyF0RtrgszZ2HJZunWgDzKK3i0n4ZePPFXgqHTtO+3XkYgTS54XksbdTEsoaWAuqHaXk2oW2BgRzXV6B4XutJ8ZeH73SdL8M+HNOaGWO4h0rU3lbU4/KJj+UwRiQq2G3kk4Lc889H4c8N6jB4s1rxJrosoLnVIYLcWVi7SxokQb5nkZUMjEuf4BgADnrWvpPhbw/oE0k2haFpumSyLtd7OzjhZx1wSoGRQBwHxJAg0Ox8XWmvNrtvBqVtcafpM7RNa3TtIFVYjCquzjcWUs0gBXJU9R6pWPa+EfDdlq/8Aatl4e0q31Hcz/bIrKNJtzAhjvC5yQTk55ya2KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAP/2Q==)

# Workflow

Following steps shows the overall workflow that have been finalized for the project:



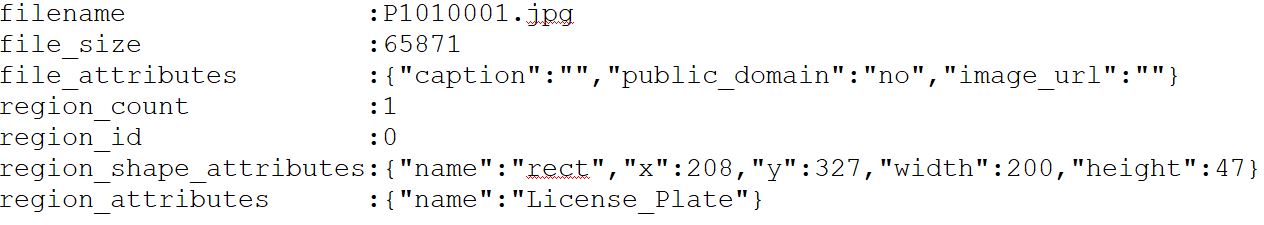
Overall workflow in the project

# Models

## Stage 1: Model1 for locating the number plate

### Preprocessing

As different approaches will be applied to both the models in different stages we have pre-processed our images and generated supported files accordingly. The images that we had received from the source were of different sizes, example: 640\*480, 1024\*768 and 1600\*1200. We resized all the images to 640\*480 first and carried our the annotation process with all the images. We have used VGG Image Annotator online tool for the annotation of our images. The annotation details were then received in csv file with respective values as shown below:



As we had not received xml files, our next substep, to generate csv and pvtxt file in the format our model required, we used the script from xml\_to\_csv.py (Github resource file) and modified as per our requirement. This generated the csv file in the following format:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| filename | width | height | class | xmin | ymin | xmax | ymax |
| P1010001.jpg | 640 | 480 | License\_Plate | 208 | 327 | 408 | 374 |

and the generated .pbtxt file consists of only one item with the class name “License\_Plate”.

After this with the help of generate\_tfrecord.py file using generated csv and .pbtxt file we generated .record file. By this phase, we had all the supportive files that we require to to train our model.

During the first stage, we had worked to identify the location of the number plate in the image itself. We had carried out this experiment with two standard models: SSD and Faster-RCNN. Following are the experimental settings and respective results for each of the standard model:

### Experiment with SSD:

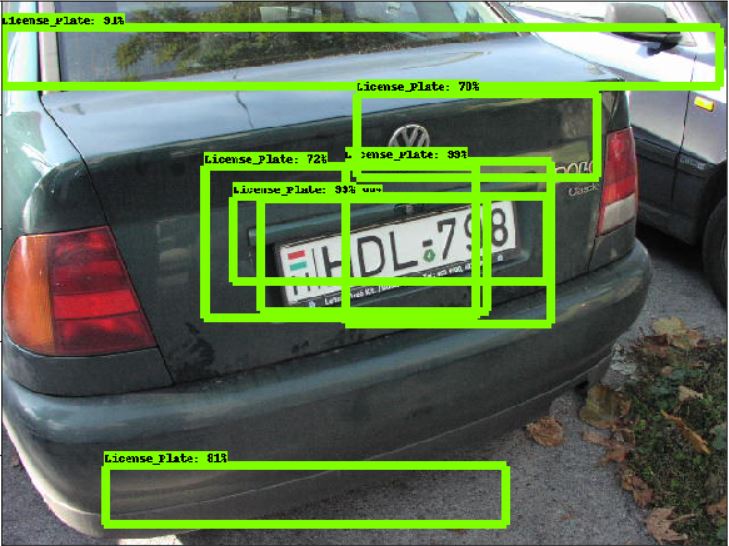
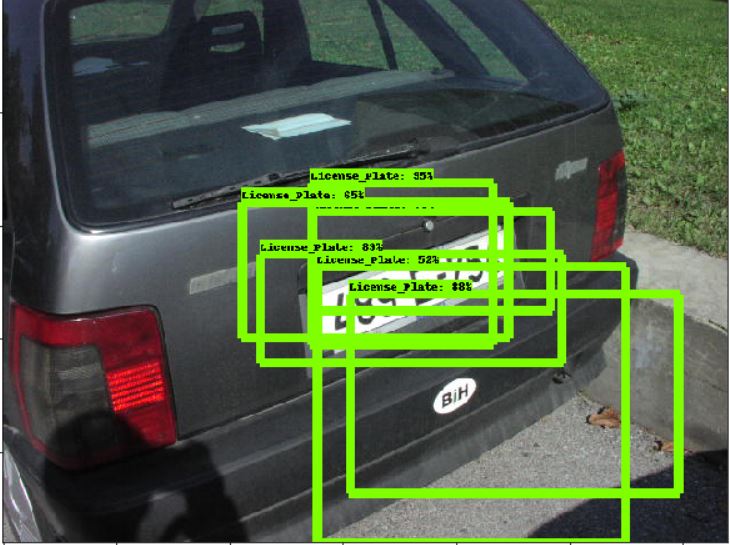
With our previous work and projects we learned that SSD is quite faster in comparison to other models so first, we tried to work with the SSD model.

Experimental settings: The configuration and pipeline we used for the SSD model is as follows:



num\_steps = 1500 and num\_eval\_steps = 10.

Results: As a result, we have received the images with multiple bounding box with higher accuracy. Some sample images are as follows:

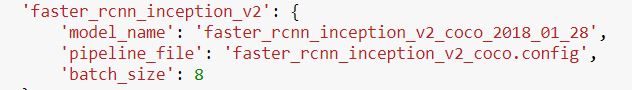


With the total of 1500 steps total loss and end learning rate are as follows:

Loss/total\_loss = 13.958896, global\_step = 1500, learning\_rate = 0.004, loss = 13.958896

Experiments with Faster-RCNN: While working with Faster-RCNN, we had the same preprocessing steps and use same annotations with same csv, .pbtxt and record files and implemented Faster-RCNN configuration as per the requirement of the standard model.

### Experimental Settings for Faster-RCNN:

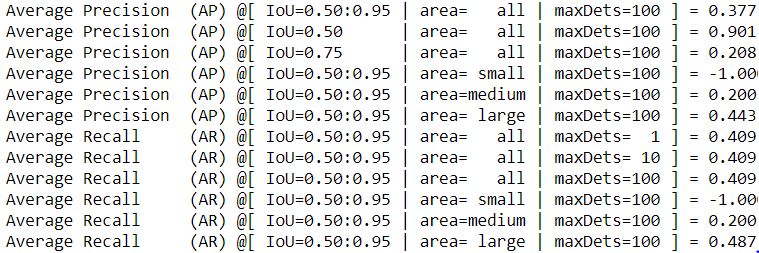


with num\_steps = 1200 and num\_eval\_steps = 16;

Results: With this configuration and training Faster-RCNN gave us quite clear result. The images from the experiments are as follows:



with this model the average precision and average recall received are as follows:



### Conclusion and improvement of work:

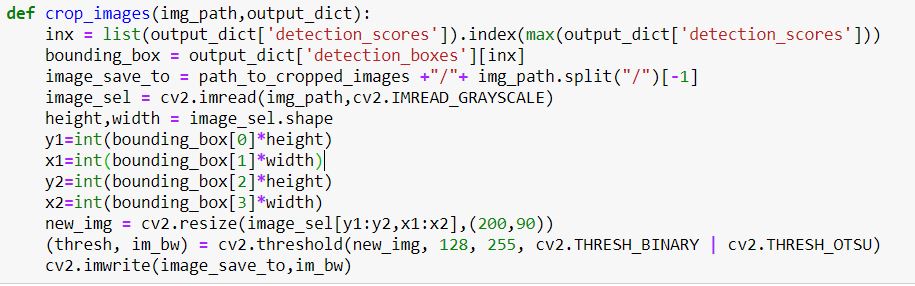
As we received better results from faster-RCNN we used the model for our first phase to detect the license plate. For the second phase of work, only that license plate is the area of our interest, so insted of processing all the images, we cropped the images only including our area of interest and planned to proceed with that. As we can see (images given below) the section of the number plate is not clear so to decrease the noise we converted the images to bitmap, thus the image will include either 0 or 1(255) no intermediate pixel value. In addition to that, the area of our interest (the license plate) is in the image varied in size, so we resized the cropped image to 200\*90.



output of Model 1: While predicting the number plate if the predicting image(on left) is as follows then the output from model one to feed to model2 will be the image on right.



The part of the code that performs this crop, conversion to bit image and resizes it is given below:



## Stage2: Model2 for identifying and detecting the characters

Post detection of the license plate the intention was to locate the characters in the license plate and then identify those characters. Unlike simple object classification such as distinguishing cats from dogs, identifying the characters on the license plate is a challenging task, as there were many other characters on the plate but our intention was only to identify the legitimate characters on the plate. For character identification task we did a separate type of pre-processing. These cropped images were used with customized VGG16 architecture initially to classify the character images and the results have been analyzed. Identification of the characters was one of the important tasks to be performed on the images.

**Architecture Used**

Experiment -1: The first architecture used on the classifying the characters was VGG16 architecture. Our intention here was to train the model on individual characters which included alphabets (A-Z) and number (0-9) which added up to 34 classes in total and then use the model to classify or identify the characters in the image.

### Experimental 1:

The base model VGG16 is taken as the reference architecture and is further customised for the image classification task. The customised architecture consists of 4 blocks of convolutions layers. Similar to the base architecture, 150 x 150 RGB images are passed through 4 blocks of convolution layers and each block consists of 3 x 3 filters. Each block of convolution is followed by a max pooling layer with a stride of 2, furthermore, customisation is added by adding a dropout layer at the end of each convolution block. The images are flattened and then the input is given to the fully connected (FC) layers. The final layer is the given 34 output neurons with the activation function used is ‘softmax’.

As 4 blocks of the convolution layer is used for the customised architecture and the top four layers were frozen and the total number of trainable and non-trainable parameters achieved are as follows:

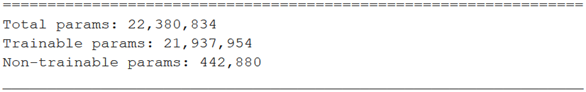
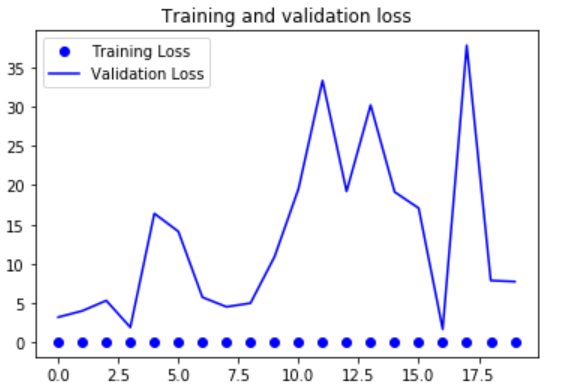
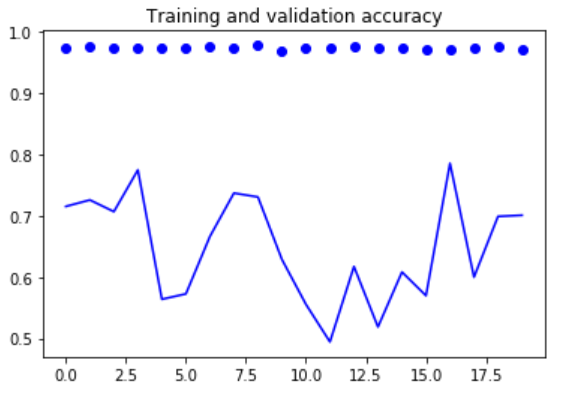


Figure 2.0 Total number of parameters for the VGG16 architecture

The experiment was performed with both RMSprop and Adam optimisers and have plotted the graphs for accuracy/loss vs iterations respectively. As the dataset consists of several classes the loss used in this experiment is ‘categorical\_crossentropy’. The activation function used is ‘relu’ and padding used is ‘same’. Rectified Linear Unit (ReLU) is the most used activation function it is one of the important activation functions as it does not saturate, and it also avoids the problem of gradient vanishing. The total number of epochs given was 100.

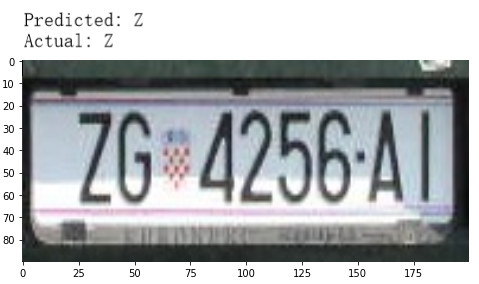
Results:

As the training accuracy and the total loss was very good as showcased in the figure but the model did not perform on the test set the example is shown in the Figure



Training and Validation Accuracy Training and Validation Loss

Below are some of the classifications of the characters done using the VGG16 architecture



### Experiment 2:

As observed in the above experiment the model did not appropriately identify the images in a license plate as shown in the figure, with this we had to change the model and make use of the standard Faster RCNN model to identify the characters in the image. Changing the model and also changes the train and test dataset images, the images cropped from model 1 were then decided to be changed to bitmap images and then use Faster RCNN to identify each character. Following the pre-processing done to prepare the train and test dataset:

Preprocessing: As our model 1 results gave the cropped license plate bitmap images, we had to prepare a different data set that included these cropped bitmap license plate images which can be fed into our new model for the character identification. Following are the steps included in the preparation and pre-processing of the dataset:

Step 1: Cropping the actual images just to the license plates

Step2: Resizing all the cropped images to a size of 200 x 90

Step 3: Converting these cropped and re-sized images to bitmap images, some of the image shown in table 1.0



Table 2.0 Sample train set bitmap images

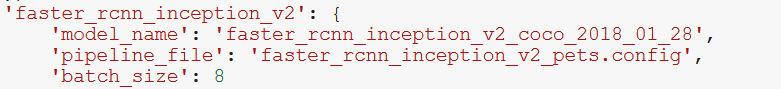
Step 4: Performing the character-wise annotations on the pre-processed images

Step 5: Splitting the entire dataset into train and test with 80% and 20% respectively and then creating .pbtxt files for the train and test images

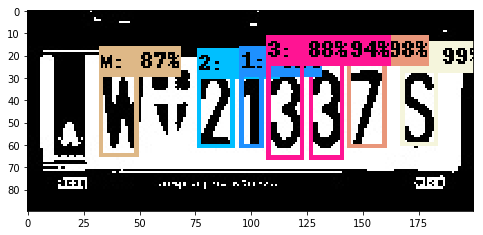
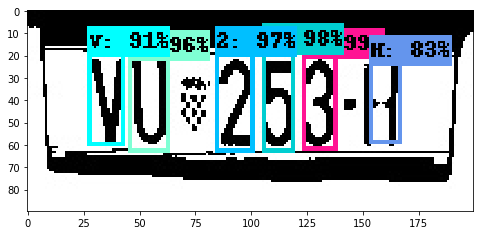
After completing the pre-processing, the train and test data were fed to our model in the below step for character classification.

**Experimental Settings**

Object detection task the model Faster-RCNN has some of the defined hyperparameters such as the number of steps provided are 2000, the number of evaluation steps is 50 and the config model is selected as ‘faster\_rcnn\_inception\_v2’. The batch size selected for this model is 8.



**Results:** Results obtained with this configuration was as follows:



As observed in the above table the model performed well and detected all the characters correctly on the license plate.

There were some challenges faced in getting these bounding boxes and identifying the characters in the images. As the standard architecture only accepted RGB images to generate the bounding boxes we specifically converted the bitmap images to RGB get the bounding boxes.

Now after detecting the characters with the help of bounding boxes our main intention was to print these characters and show the exact number printed on the license plate. Our model generated around 300 bounding boxes for each image, so we only selected those bounding boxes where the threshold was greater than 0.5. Bounding boxes with the threshold value greater than 0.5 gave us all the characters in that image in an unsorted format. As we have the values for the coordinates we sorted the characters based on the coordinates in ascending order.

Following is an example of a number plate and sorting out the characters in the correct sequence:



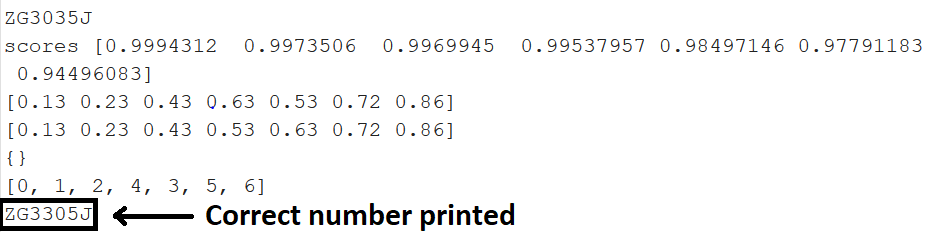


Figure 3.0 Sorting the order of the number on plates and printing the correct value

### Conclusion

As observed from the above results the Faster RCNN had better and accurate results as compared with the customized VGG16 architecture. Characters were more detected and the accuracy was also good in the detection of each character. Initially, the loss and the accuracy on the training set for the VGG16 architecture was considerably high but it miserably failed to predict the other characters where Faster RCNN only with standard setting outperformed the VGG16 architecture.

# Personal Contribution to the project:

First phase of work was to do the annotation for the license-plates. We divided it equally among the group mates and we completed the task. I had to do the annotation for about 170 images and did that.

With the initial phase to decide, whether the generated supportive files will work or not, I had initially tested on Faster-RCNN with small data size of 30 train images with 6 test images.

As the annotation file generated was not on xml format, it was on csv with inner details on directory (hash map) format, I changed the script from file xml\_to\_csv.py as per our requirement to generate module accepting csv and .pbtxt file.

We together with the groupmates did research about the second model input as that was something to get output of characters from the images and in the sequence as shown in the image. Based on the conclusion of our discussion we choose to reduce the image size to feed to second model, so I wrote the code to select the area of interest which is only number plate and convert it to bit image with the concept to reduce the noise in image and feed to second model.

When we received the output of second stage model, we got image classified to characters or digits we need to order the sequence of the character based on the images we got. I did the logical part to get the result based on the coordinates of bounding box and the accuracy score for each of the classified character class.

And as we have to build the user interface and implement the models locally, I did research to implement object detection on local machine. As my machine is windows which supports Linux terminal (Hybrid system) I setup the local environment as per the requirement.

We are still working together to implement and integrate the models and logical parts to the User interface.