Uma imagem com texto

Descrição gerada automaticamente

![Uma imagem com texto, ClipArt

Descrição gerada automaticamente](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4R+aRXhpZgAATU0AKgAAAAgABgALAAIAAAAmAAAIYgESAAMAAAABAAEAAAExAAIAAAAmAAAIiAEyAAIAAAAUAAAIrodpAAQAAAABAAAIwuocAAcAAAgMAAAAVgAAEUYc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFdpbmRvd3MgUGhvdG8gRWRpdG9yIDEwLjAuMTAwMTEuMTYzODQAV2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NAAyMDIyOjEyOjAxIDE2OjI2OjA3AAAGkAMAAgAAABQAABEckAQAAgAAABQAABEwkpEAAgAAAAMwMAAAkpIAAgAAAAMwMAAAoAEAAwAAAAEAAQAA6hwABwAACAwAAAkQAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMjoxMjowMSAxNjoyNToyNgAyMDIyOjEyOjAxIDE2OjI1OjI2AAAAAAYBAwADAAAAAQAGAAABGgAFAAAAAQAAEZQBGwAFAAAAAQAAEZwBKAADAAAAAQACAAACAQAEAAAAAQAAEaQCAgAEAAAAAQAADe4AAAAAAAAAYAAAAAEAAABgAAAAAf/Y/9sAQwAIBgYHBgUIBwcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQy/9sAQwEJCQkMCwwYDQ0YMiEcITIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy/8AAEQgAeACgAwEhAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A9/ooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKKACuN8e/EPT/A9nGHj+1ahP/qbVWxx3Zj2X9SenQkAHh998ZPGt5O0kWow2iHpFBbptH4sGP60WPxk8a2dwskuoxXiDrFPbptP4qAf1oA958DeN7HxvpL3VuhguoCFuLdmyUJ6EHupwcHjoa6mgAooAKKACigAooAKKACigAooAQ9K+SviPq02sfEDWJpWJWG4a2jG7IVIyVGPTOCfqTQBy1FAHb/CXWn0b4h6eu/bDek2ko/vbvu/+Phf1r6nByKAFooAKKACigAooAKKACigAooAQ9K8C8N2/hPR9P1LWPFlil5Lf6zLpyl0WQQr1L4ONuCTlhlhxjqaAMjSvA1nH8bH8N3AE2n28zTFC33o9nmIpz16qD6812za54e+J+i+INDg02O2k02BpNPlIGSq5AZcD5BkKCvo35AHinhef7P4t0WfGfLv4H/KRTX2SOlAC0UAFFABRQAUUAFFABRQAUUAB6V4toHhMeKtQ16w1R7caH/b9xewRqxE8xVmR9uOiHKAnnpgYJDAA9K/4Q/SPsqW5t4jGk5m/wCPaHJySdudnv8AeGG4+91zxF/4Gt/DFtr0XhC5hS9vbBzLb3StIYouRhJAR5efmxv3bivX5TQB4Ho0iRa5p0sjbY0uY2ZvQBhk19ojpQAtFABRQAUUAFFABRQAUUAFFAGbr+sQaBoN7qtyR5drE0mCcbj2Ue5OAPc14t8IbTWL7RNVvIpWmiivFkjh3BHMu3LtG/8AC+NnDZVhlWGGyAD1T+12a3Aa71MEEKsceluJ2YdQxKlcHpuAUd845psGg3N/Zm3u40sdOlwZrNHMk8/Qfv5iTnKgBgMk4xvIyCAfK+taXLouuX2mTEl7Sd4SxGN2CQGx7jn8a+qPh/4qi8W+FLW93g3caiK7XusoHJ+h+8PrjtQB1NFABRQAUUAFFABWNrfivQ/DskUeq6glvJKNyJsZ2I9cKCQKaTexUYObtEyf+Fn+Dv8AoMf+S03/AMRS/wDCz/B3/QY/8lpv/iKfJI1+rVewf8LP8Hf9Bj/yWm/+IpP+FoeDv+gx/wCS0v8A8RRySD6tV7HkHxb+Ia+JLiPR9Jkf+yosSSSbSvnv24ODtX36nnsDXrPwp0I6F8P9PSSPZcXYN3MOer4259CECD8Kl6GLTTsztaKBHzx8dvDxsfEttrkS/udQj2Snk/vUAHPYZXbgf7JrkPAvjC68E+IRcqJGtZCI7y3HG9c9cdNy84z7jIyaAPq3T7+21OwgvbOZZradBJHIv8SnkfT6GrNABRQAUUAFFABXivxFtrm6+KVhFaWQvZvs8brbtjEm0uxHPHQGrhudGGaU7szrjwz4kli1RE8HIhvHVo2BTMAHYc9/bHJP0qpq3g7xPqU0EkXhb7GI4RGywlQHI/i6/wCP1rXnR2/WKfcof8K98Wf9AWf/AL6X/Gj/AIV74s/6Ak//AH0v+NHOg+sU+5m+HPC0niTxvp2kzRHyo5Ge8GM7Y0I3A46ZPy59WFfUwGBisZ/EcGI/iMWipMTlviD4XHi3whd6cqqbpR51qx7SrnH5glfoxr5edDqWlzSSZF7p6r5m7rJDkIM+6EquOflYdAlAHqPwM8YPDdyeFryQmKUNLZEknaw5dPoRlh05Dete80AFFABRQAUUAFeZap/yXrRv+vNv/RctVE1pdfRnd67rNt4f0efU7tZGghK7xGMt8zBePzp2k61p2t2a3enXcVxCepQ8qfQjqD7GlbS5HK+XmNCkPSkSeV/CaGM6/wCKJyimVJlRXxyFLSEj6HaPyFeq1UtzWv8AxGFFSZBXz34i8Prpfx0S0WP/AEXWgxKqxyFuEeKQ/XcXb24oA8u0rUZ9J1Sz1K2IE9rKkybuhKnOD7etfZ1ndRXtnDdQtuimjWRG9VYZFAE9FABRQAUUAFeZap/yXrRv+vNv/RctVE1pdfRnc+IZ9Kt9CupdaVG04KPODoXByQBwPfFeH+H7ae88bx3Pgy11C30+OVfNaSThU3AsGPTGB90licd+1Q2dzXD6Rk3sfQY6UHpWZynl/wAJf+Qx4r/6+I//AEKWvUaqW5rX/iMKKkyEJwK8H8TfFjSovHsd5F4ejvn0lpIILw3bIzZBViAAVK8nGQ3HIxngA8z1HSNPayk1HQruW5soyBNDcIFntskgb8fKyk4G8dyAQpIz9U+DAV8D6AGBDDTbcEH/AK5rQBuUUAFFABRQAV5lqn/JetG/682/9Fy1UTWl19Gd/quk2mt6bJp98hktpSpdAxXdhg2MjnqKltLK2060S1s4I4IIxhI41CqPwH50r9DPmdrHNS+OXimeP/hFfEj7GK7kscg47j5ulbWi6w2tWck7abf2G2Qp5d7D5bngHIGTxz19jTtZFygkr3R4l4X8e2ngrxbqcWoQO1lf3BEk6ctDtZsHb/EPm5xzxxnoffLa5gvLaK5tpo5oZVDxyRsGVlPQgjqKJbjr/wARktFSZHKeMv8AiYXeieHC7rDqly/2sISC9vEhd0yOQGbYpx2YjvXifjz4Ya5YeLLhtG0ia5028mMlv9ljysW45KEAYQAnAzxjHPBwAdJ4W+CF82kXh1zUZLGe8jERt7YJJhAyv87HjO5FPyn+H7xyQPUfCuozyrfaTfJAt/pUot5Ps4Ko8ZUNG4U525U4IycFT2oA6KigAooAKKACvJvHE+oaD8TtN8QRaXPeW8VrsAjBAZiJFI3AHBG4HpVQ3NqCTk0+qJv+Fuah/wBChdf9/wBv/jdMf4w3se0SeE51LnaubgjJ9B+761Xs/M1+qr+Yd/wty/8A+hQuf+/7f/G6P+FuX/8A0KF1/wB/2/8AjdHs/MPqy/mR5dr3hDWrzQLvxSbOSK2juCXiZTvCNyXGQMqCQM+5PQGvQf2f9Svp9I1bT5t7WNpKht2OSFZ9xdQemOFOB3YnvUy3MazTm7HslFSZHN3jxS/EfSYs5kg0y7lI/u7pLdVP47X/ACNdJQAVyOryt4a8Vprzhf7L1COKzvW6GCRWPlSk/wB0lyh9MqfWgDrqKACigAooAKKACuT8VhLvxH4U005Be/a8LDssMTH9WZB+dAHWUUAMlijmieKVFeN1KsrDIYHqCPSvOv8AhKLP4a6oNA1ppxpEoMml3e0yCGLvC/8AF8hwFI3fKyg9KAOsg8ZeGLiISR+IdKK4yf8ATIwR9Rnj8aw9e+LPhHRI226kmoTgArFYkS5/4GPkH559qAKnw01i48YXOr+K7u3jgaVksYIV+by44wX+93JMvPA+70r0OgCKe4htYJJ55Y4oo1LPJIwVVA6kk9BXF6rrEvjXTrrRvD1o01ndKYLjVZ0K28aHKuY88yuBnG35c4JagDuAMDFLQAUUAFFABRQAVka7oMetJbOtxNaXlpJ5trdwY3xMQQeCCCpBwQRgigDMa08b2tuFg1jRb1173VhJEW+rJIQPwWmt4q1LSCx8SaJJb2yjm+09muoRxklgFEiD32ke9AGjF4x8MzQmaPxBpZjAyW+2RjH154/GuD8V+HD8W9Stn0+8FrpOmK6R37QlxcyORuEYyNyLsHz5wScDOCaAIbP4AaGkeL3V9Rmk/vQ7Ix+RVv51Defs+6Wzf6Frl5CPSaJZT+m2gDpvh3aJ4TF54PurhGureY3Fu5Xb9pgcDDqPUEMpHOMD1Fddqus6doli95qV3HbQJ/E56n0A6sfQDJPagDmrXSJ/GF0uqeIrSSPTlObHR5xwB/z1nXoznsh4UepJx2YUKAAAMDAoAWigAooAKKACigAooAKKAM+fQdIuroXVxpVlNcL0lkt0Zx+JGavhQAAB06UALRQBm6xoOl6/bLBqllFcorbk3j5kPqrDlT7giqWm+CvD2lXovbXTU+1D7s8ztM6/7rOSV69qAN6loAKKACigD//Z/+Ex5Gh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRvclRvb2w+V2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NDwveG1wOkNyZWF0b3JUb29sPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTEyLTAxVDE2OjI1OjI2PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwACAQECAQECAgICAgICAgMFAwMDAwMGBAQDBQcGBwcHBgcHCAkLCQgICggHBwoNCgoLDAwMDAcJDg8NDA4LDAwM/9sAQwECAgIDAwMGAwMGDAgHCAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM/8AAEQgBKwGZAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A/fyiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoooJxQAUVn+JvFGn+DtDu9S1S9s9PsLGJp7i5up1hht0AyWd2ICqPU8V+Sn/AAUA/wCDrHwX8KNXvPDfwL8Pw/ETVLR2ifxDqMrWuhqwznyV+WW4A7t8iH+BpPvAA/Xd5s5VT82OuMgUK7Buv54r+UP4+f8ABdz9qr9ojUJJL/4sa14d0+b7th4a2aPBGv8AdzF+9b/gTtXzf4s+PHjbx1cNNrfjLxZrVw3zGW91a4upG/4E7M1HQNbXuf2nGTatAlBGen1r+JfTfGmtabceZa6rqltcfwvHevHJ/wB9blr2z4F/8FVP2iv2c7y3m8J/GPx1aw27bhbXupNfWP8AutDcb4v++lo1DfY/r+VgwyOR7UjHHYn6dq/Dj9gD/g7BuJNWsfD/AO0J4ct47aR/K/4S3QIXXyh/furTDfL/ALcLBR/c/iH7TfC74paD8ZPAmmeJvDOr6fr2g6xCtxZX9lOs0NzGehDLxn1HYgiiz6hqtzoqKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKr317HpsEk0skcaopctI21UUDkk9gO5qZ32Lmvzp/4OV/23p/2UP2B7jwzo120Pib4uTv4fgKttaOwVd18/8AwKNkg/7eaA16H5df8F4v+C0+r/t1/FPU/hr4B1a4sfg34duntWe3leP/AISy4R2X7VNs+9bMy/uk/iXbL/FsT84pJGm3M33t25v727/a3fMtN3fKvzbt3z5P8TN8zN/6DRQAUUUUAFFFFAB/Cdy5H+dv+z97+9/Ftr7y/wCCIP8AwWD1r/gnb8ZrXw34m1G81D4N+J7tY9Xsy7TNo8z7UGoW/wDdbc371U/1qqdvzKtfBtDSeX83zL/FkNtagD+3LQ9Xtdf0e1vbG6hvrO7iSaC5hdZI7iNlDK6svDKykEEcEHIq1X5if8GvX7cs/wC0b+xXd/DrWrprjxF8H5o9OiLtueTSpd7Wn/frZLD/ALscVfpzHIJAcdjigB1FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBDLPmbb/AHMZ/E//AF/1r+c3/g7C+OMvjz/goRoPgyObdY+AfC8K+T/09XTtPKf+/S29f0ZyKry/p/Ov5Pv+C8Hi5/HH/BW/42XTOr/Z9Ziskd/uxi2sraLbt/i+WKgf/APkmiu/+Af7KnxM/ap1y80/4aeA/FXje4sVR7oaVYPPHaq33fOlX91F/wACZazvjV8A/HP7NvjRvDvxA8JeIvBeuKnmrZaxZtbSSJ/fTd9+P/aXcvzfeoEcjRRU2m6bc6xqFvZ2dvcXV1eOsEEFujSTTOzbURFXczMzbflVWZvu/wC1QBDRX6p/sg/8GpXxd+OXgu38QfErxZpfwnS+iSe30cac2qarCrDd++VZkiiZv7u9pF+6+K4z9v7/AINnfjP+xz4E1Dxj4T1TTfix4V0tPtF6mm2EtpqtjCq/PK1tudpUVfvNHL5i/e27VagD836P73y7/wCHH97/ACtHyN9z7v3V+X+H+9/ut/C38W1t1FAH6Hf8GyX7RE3wU/4Kh6J4ekn2aX8StLu/D84/5Z+akX2q3f8A3me2EX/bWv6ZoZCk2087+/vz/Sv44/2DviY/wa/bc+EnipX2x6D4v0y6l/65LdJv/wDIea/sciXdKD/dJ/qKF5lWt+JYooooJCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBsnAr+Sv8A4KieC9W+J/8AwWC+L3hvRrf7TrXiD4iXOk6dF95ZJ57ryolYf3dzru/ur838Nf1pSj5930FfzZ/EjSLfRv8Ag6ks4bhlEMnxfsrnn+/J5Tp/5E20R3Yj95P2If2PvBv7An7MXh/4e+Fbezt7HQ7VWv8AUDGsMmqXRX99eTP/AHpHBb/ZXao+VVriv+Cq/wDwTw8N/wDBSL9lPXPB9/b2q+KrOCS+8KatsHm6XqKjMXz9onbCOv8AEjN6V+dv/B3f+0Z4i8OaF8JfhZpd5cafoHiFL7XNZiimeGPUfIMUdvDLs6xozyP/AHd/lt/DXpn/AAakftleIPjp+zF44+G/ijU7rVpPhdfWj6Rd3j7p1066R2SHc33kilhl2/3VdV/hoGfz3X+m3Ok6hcWt9bSWt1Yyvb3ETr80Lp8rxM3+wysq/wDAq/Wr/g1N/YK034x/GvxT8bPE1jHdWPw5kTS/DizRfu/7VlTzJrgf3mgiKbd3/P1u/hr4E/4Kj2Gl6B/wUh+O1vpsln/Z6+ONVngMbr5W17ppW+9/tO3/AALbX7if8GoGqaT/AMOy76ytry3k1STxhqVzexLOrXCMUtkVmVf+maR0AfbP7Wv7enwj/YT8M6bq3xV8b6b4Sh1Z3gsYpo5Li8vnXbv8qCFHkfbkbmVdq7l9VrrPgH+0R4J/ai+GNj4z+HviTT/FnhnVGYW1/YsWRirbXRg3zIyt95WVWX0r+Yv/AIL3/tCav+0P/wAFW/ik2pXMjWHgu/XwrpEGdy2dva/K2z/ae581/wDeevsL/gz5+N+sWHx5+K/w0a4kbw/qGgW/iaK0/wCWdrdxTpA7r/tPHcor/wC1AtAHzX/wcUfsG6f+xL+3i+o+GrKOx8GfFKybX9Pt4l2xWd0sm29gT+6nmOkqr/Cs6r/DXwTX7qf8HjVnbp8NfgNdfu/ti6zqsKf3vKaC3Zv/AB5Vr8K/lx/tUAWNL1BtK1K3uo/v27q6/wDfVf2v+E9V/wCEh8NaXqP/AD/WkM//AH0ob+tfxLSfIuf7vzV/aP8As8TfafgH4GkPWTw/p7f+S8dPsEdzt6Adw/8ArUUUgCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBkibnB/Cv5ef8Agu54l1L4Af8ABdDx14t0WRo9Y0HVtC8Q2Thtv76Kws5U/wDHkWv6h5jhffPFfzG/8HDfhLVPi5/wWy8WeGPC9jdax4i1MaHpdjZWqbp7q5ksLXYifMvzMzbfvL7tt3UAj7G/4OF9C8Kf8FBv+CZHwk/ae8H69odra+HEWZre5vIopbi2vliWa1hV/wDWXUFxCq7F+bCS7d2NreR/8Ewf+CNnir4efsl+K/ix8evEXxM8C/CTXrK21HUfAXhJLpdc8VWsHmNC16tuvnrF++LLDF8zK25/K2bq+wv+CSn/AAbx6L+yn4d8P+KPjVcR+PvG2my/2hpfhxp3ufDnhGZ1Te0MLfu5rr5VVpmVV+X5V+Xe36gWn32/+y5/OgD49/YU/Yv/AGc/GnwFtNS0D9lbS/h5ptzIyW+neN/B9h/as8e1SJ2LtcS+U4b5fNff8pyi9/C/iP8AsGfsy/Hj9rrUPh/YfBP4tfs/fEy2imvdE8deDtLfw/YaksWN01rdWErWuNxB23EaM2a/T6igD+ZP/gqJ/wAEGf2jP2evibr3ja3/ALY+POh6xcNf33iXTbVpNYaTJy19bL+8DrhX82PzV2lmbbX0t/waZfDvwb4F8YfEPxVq3jvwnB8Q9eiTQLLweb5Ydas7eKZpJpZbZ9rfvZFiVVXdtWAsWXdtr90rhPMj24+XvXyx+3r+yh+zJ+0V4k8K+HvjFoPhNfGnji9fTvC16jGx8Q3F2kDufstzDtn3RxoW3M2xfl3feVWAPxq/4Onf2ydP+Pv7bWi/DfQbqO60r4Q2U1nfPHLuiOp3DI9wn/bKNIU/2X81f4a/MOvtv/gtv/wSY1L/AIJgfHDT7yz1y+8VeAfiA01xpOqagytqEdwjK01vdP8AxyLvV1lX/Wbm3Ku2viZty/L/AA0ANH8P+8v/AKFX9ov7On/JvPw//wCxe0//ANJo6/i6/ur/ABbvl/8AQq/tF/Zx1G31b9nzwDcW/wDx73Hh3T5Yv9xraMr+lN9A6nb0UUA8n/OaQBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRQW2igCOdQww3fiv5h/2g/wBtjR/hX/wcZ698YNVhkvfD3hH4iGzvPLh86SO1tkbTJpUT+KRFVnVf7yLX9Hf7VPx50/8AZh/Z28bfELVWjWw8G6LdarIHOBKYomZEHuzBV/4FX8afibxFeeL/ABNqGrajI1zqGqXkt1cSbtzSO0u9m/3W3M3/AAKjuC/U/p31jxn4l/YO+J3iD41eFLjV/i5+zL8YLiLxRrcWlSvqOoeB5pYFDapYou77Rpk6IjyxR/PE371dy7lr7L+FHxW8O/GrwFpvivwj4g0zxP4Z1m3W4sdS024S5t7pG/iV14/Dt39vyI/4JP8Aw/8A2nP2NP8Agnh8Ovix8Krr/hcPgfxNp8ur+IvhXqr+VfWK/apd0+h3P3dzxIH8h12yM+5dzOtfTn7Nnwo+Fn7Znhu8+MX7KvxK8UfBHxBqt4//AAkml6Taxy2Kaiv+th1bQ7gNEt0rBtzReVI/3t7/AHqAP0GqO6OIW+bbXyz/AMIN+2Np9v8AY4/iR+z3qUa/KNUuPBGqRXP+95C6n5bNu/h3qtZ83/BN3xF8b2YfHr42eOviZpsg/e+GNFjTwl4auPm+ZJbe0f7TcR9crPcurK3INAFj4x/8FKLe98aX3w9+Avh6f43fFK3Zra6i06fy/DnhmT5m36pqahoINv8AzxRnnb7uzdWl+yT+whffDf4gXPxa+LXiuX4n/G7VrNrOTWHg+zab4ZtX2sdO0e0/5doN23c77p5SoZ2/hGZ+0F+3h8C/+CXuh6D8O9L0lJPE10nkeHPht4D0dZ9UvuOBFZQfLCn+3LsVuxavLdc0/wDbG/4KKWbaZfWlp+yJ8KtSX/TJoNQTVvHmoW5+8iOn7jT2ZeGb5pE/hagD8+v+DqX9tLTf2gPid4W+GnhOOTVtH+FeozjxJrduhks7fWJ0xFpwl+75scUMruv+3j/lk1fkb83yt/DX9HP/AAV6/wCCW3gj4Zf8ERPFXgf4X+G1023+Fs8fjW2TPm3WoTWzH7ZcTyt80krwNcMzH/gPyqtfzjbfvfN8u7dz97+H/wCJoAPm/h+9/n/9n/gVf1nf8EUfj3B+0R/wTA+DuuLLHNcaXoMWgXe3+Cex/wBFf9Ilb/gVfyY/L95vur8zf/FV+yn/AAacft1Q+E/Hnij4B69feXD4mZvEHhkSfd+1xLtu4V/2niRJf+2TUB9m5+9AORRUdu/G3OSv6VJQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRnnFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFAORRQAU2WRY1+b8B6mhnCrmvGf26f2zPCP7B/wCzfr3xG8YTsLHS0MVnZpIFuNWvGDeTbQ5P32Ydf4VV26LQGuy3PzT/AODsD9vqHwr8JtA+AOgXm7VfFhTXfEixSfNDYREm1gZf+m867/8At2r8F5o5HVlhXzpm+WCP/nozNtVf++t1d7+1D+0f4m/a3+P/AIr+JHiu5+1a54qvXu5QW+S1T7sUCBvuIkaqip/CqL/Fur1X/gkP+za37WH/AAUm+Evg+W1+16aNai1nVkP3fsVn/pMqn/ZbylT/AIHQV/dP6kv2Lfg5/wAM9fsifDHwN5flzeE/DOnaZcIv/PaO1RZf/Im5vxrwn9qL/gmHqGt/G68+NH7P/jZvgv8AGi8j8rVblbNbrQPGSJ0TU7PPztwP3y/vF2/3q+yISp+6v3Vxx936VNQSfCWn/tV/tzfD/dYa9+y18P8Ax1eQ/Kmp+FviTDY2dx7+Texeanf7zVR1HRv28v2ukbTNQk+F/wCy74XuW8m6u9KvH8V+KPLP3vJl2raRHb/H95WAr77ooA+df2Kf+Ca/wy/YYhvr7wzp99rnjTXPn1zxl4ium1DxBrkjHczzXL/MqlgP3UYSP5V+XjNfREdOooAyPHHhaw8d+DtV0TVrWO+0vWLSWxu7dxlbiGVGR0P+8rMPxr+Nz9rT9ny//ZT/AGnPHnw51Jme68G63daaspXatxCr7oZ1/wBl4mR/+BV/ZvP/AKs1/Pl/wdn/ALIjfDn9qTwf8ZNPtlXS/iNYf2Nq0qfdXUrP/VM3+1JbFF/7daAPyTroPhT8Utc+CPxK0Pxh4Yv7jS/EPhu9i1DTruL71vMjblb/AGv93+L7v8VYOG+Zf4VpuW+Xb8rblbP8K7W3Ln/gXy/8CoA/ru/4Jh/t/eH/APgo1+yponj7SWtrXWEH2HxBpccm5tL1CNV82P8A65sGV0PdXHcNX0arhxxX8lP/AAST/wCCmmvf8Eyf2mbfxFbvdX3gnXNll4q0eP8A5fLRW+WVE/huYPmZP7y7k+XfX9Uvwf8AivoHxu+Heh+LvCup22seH/EFtHeWF7btujuInG4HPqPuleqsrKcMCAXu7MWzXZnZUUiOHXcOhpaBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUHmgA+6teW61+3B8FfDeqSWeo/GD4XafeQ8SW9z4qsIpUPurSgivjD/g44/a28SfA34DeFfA/hy8fTR8Rri7XUbmEhbgW1oLdjGj/wDLPc9xGSw5+TGcEg/iN4V8L6l488SWOk6VayX+qapOlvbwR/Ksjs237v3f95mZa9HDYFVIc8mfY5PwusXh1iKs7X2SXRaa/M/p/wD+G/fgR/0Wv4Sf+Fhp/wD8eo/4b9+BH/Ra/hJ/4WGn/wDx6v54v2iP+CdvxL/Zn8BJ4i8QWunyaVvWC6ksLjzP7PdvlRZV/h+Ztu5dy/N975qz/wBmn9hXx9+1hpd/qnhu10230qzl8j7ZqFx5EEz/AHvKVtj7mVWX5v8Ax6t/7LXc9X/Umh/z8f3I/ot/4b9+BH/Ra/hJ/wCFhp//AMeo/wCG/fgR/wBFr+En/hYaf/8AHq/mV8d/B3xN8MfihceDdY0y4g8SWtwlq1qV3PI8jbU27V3OrbvvKterfGD/AIJr/FT4J/CuTxbrGl6bNp9mivewW14ss+nqzKrM67VVuo+Zfu/db71H9lruH+pND/n6/uR/Qp/w378CP+i1/CT/AMLDT/8A49R/w378CP8Aotfwk/8ACw0//wCPV/Ld/s7t21mXn73y0VX9lxf2g/1Jof8APx/cj+pH/hv34Ef9Fr+En/hYaf8A/HqRP2/vgRIoYfGr4S8jPPi/Tx/7Vr+W+ij+y49w/wBSaH/Px/cj+pE/t+fAkf8ANavhL/4V+n//AB6kk/b/APgRGDn41fCXOM4Hi/TyT+Hm1/LfRU/2bG9rh/qTQtrUa+SP6ftS/wCCiHwHtrG4uW+M/wALPLtkaZ1XxTZO4VFLE7VlZjjGeB2r+bP/AILLf8FTtZ/4KY/tKXN1ay3lj8NfC8r2nhXSpF8tmi3bXvZkb/lvPtzt/wCWaqq/N95uPj+9833f4q83+OHhNYLOPVYYdrr8tx/8X/n+9WdbL+SPMeTmXCssPRlXpzu4nmvy7dzLuVfmYf3v9mv22/4NEv2Ry8/xL+OGoW7bfk8I6G8n3W+ZLi7cf7XFon/bOvxS0fRbzxRrVtpem20l9qmpTpZ2NvH/AKy6mlbYiL/tMzLX9gP/AATq/ZKsv2Gv2MfAPwxtfLe68O6an9qXCf8AL5qEv727m/4FO7t9MV5Z8f53Pb0+Xjd+dPoooAKKKKACiiigBH+7Xyh/wWr/AGMD+3P/AME6PiB4RsbX7V4m022/t/w8o+82oWoZ0Rf9qVPNh/7bV9YUx/8A61AH8Q25W+593dt5+98v8Tf71FfZX/BeP9iD/hhz/gox4qs7G1+zeEfHTf8ACV6DsT91ClxK/nW/+9Fcq+3/AKZtBXm37I/7N/g/9sD4d6x4H0/UJtB+Pkd09/4Vjvbpf7M8cW6xL5ujLv8A9TfKyO8DN+7l3tF95VoA+fiy/wATbV/iO3dtX7zf+O/8C/u/Ntr9Mv8Ag3z/AOCxEn7FHxQh+FfxC1SRfhT4svF+zXc86+X4VvpWVRKx/htZWwr/AMKblf8AgevzT1LSbrQtUubG+tbixv7OeWC6trhGjlt5Vba0TK3zLIrKyv8Aw7l+X7tQ/d+b+H73C/3fm2/8C+7/AMCoA/tys7tZI42Vh5cgyDjjHYj86tV+Rv8AwbT/APBWR/2hfhuvwJ8eajJceNPBdrnw7eXEm5tY02Lavkb2+/Lb/Kqt/HEV7o1frdBOsyj6ZwetAeRJRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUK24cc9qKAPyK/4OpNwi+BfTbu17Hrn/iW5/pX5dfs/wDxguPgD8ZvD3jK1tY76TQbpZ2t3O3zkZWjZVba21trN821tv3trfdr9Rv+DqVmjt/gWV7HXzx97ppvSvk//gkh/wAEz9A/4KNa348s9c8R61oK+D4rI27aeiP9oWczjc2/5d37qvewtSMMOpS8/wAz9UyPG08Lk9OrV297/wBKZ0X7bH/BVTw58ev2f77wd4V0HXLe615UivJdRSCKO1RXV9kW2VmdmZQu3av3mb+LauJ/wT3/AOCmHh/9mH4X33hPxVo+rXlrDeS3lleadEkkkm7/AFqOjun3dv3v9uvtX/iF9+H5/wCakeMf/AW1/wDiab/xC+eAf+ikeMv/AAGtf/iaI46mVLizAPqfmN8eP2v7z4rftdW/xQsNNjtDpd/aXGm2cr7vktfK8pZXXcu5tv8Ae+Wvpf8AaU/4LAeGfih+zrrHh3w/4e1q11rxRZPYXhvVSKDT4nTZKysrszNt3bfl+6zV9Rf8QvngD/opHjL/AMBrX/4mj/iF98Af9FI8Zf8AgNa//E1X12mH+tmC/mPxd/5aN8u1tzK2PmXdu/vUV+0X/EL/APD8/wDNSPGP/gNa/wDxNH/EL54A/wCikeMv/Aa1/wDiaPr1MP8AWzBfzH4u0V+0X/EL54A/6KR4y/8AAa1/+Jo/4hfPAH/RSPGX/gNa/wDxNH12mH+tmC/mPxdor9ov+IXzwB/0Ujxl/wCA1r/8TQf+DX74f8/8XI8Y9Cv/AB7Wvp/u0fXaYv8AWzA/zH4u1w/x8P8AxSNq3zbo71GXH+49fpp/wVj/AOCQPhf/AIJ6/BLQfFWieLPEGv3Gr64ulPFfxQrHGn2eZ94KfNu3RKv/AAJq/Mv49yeX4Ttf+md0rfKu5vlR/u/5+7uorVFPDuUS8fjqWKyurVo7WPsT/g2S/YZ/4aj/AG8k8cava/afCfwbgXV3aRd0dxqsoaOyRv8AdUPN/vQLX9LULZz64ztP3lr49/4IX/sLf8MFf8E+PCeg6la/ZvGXipP+Ek8Sq67ZYbu5VStu/wDtQRBIfrG1fZFfOx2PycKKKKoAooooAKKKKACmTNtTczbVXk0+igD87f8Ag5D/AOCf8n7Z37Cl14n0CwkuvHHwjaXXtOSNd0t5ZY231uq/xfukWVV/iaBV/ir+ZnT9RmtLq3urW6ktriN0ntbq3l+aFlbcJUf+6rfdf/Z3fL96v7crpFkt3VlRlYYIYfL+PtX8oP8AwW4/YL/4d+/t/eKvD2m2zW3gzxU3/CR+F32/uo7Wdm326/8AXCVXT/rmsVAF79pDR7b/AIKEfsn3H7RWj2tmvxS+HrWulfFzTIl8ttYhf5LLxGi/wyOqLDcL/wA9VVv4tzfHP935ty/Myn+9ub5v/Qa+hP8Agl7+1dZ/sj/tieHtW8Qp9p8AeKkfwp410+dd0F9ot9tiuWdf7qfJNt/i8jb/ABVk/wDBRf8AY3vf2DP2yPGnwzuGkuNN0e6W40O8df8Aj+0ydfNtJd39/wAttjt/z0Rl/hoA87+B/wAavEX7Ofxc8O+OvCd+2m+IfCt6mo2NwF3KsqN8quv8UbfdZf4lZl/ir+uX9gX9r/QP27f2XPC3xO8OkQprluEv7PzfMbTb2P5Li1Y/xbJA20/xKyv/AB1/Hhn/AHdv+221a/U//g1x/b9k+BP7Ut78G9evJIfDPxUfdponf5bHV4l+Rf8Aeni3RN/tpb0B0uj+ixH3rn/Ipar2sDJJu3t7g9/erFAegUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAfkT/wdRJ5dp8CxuZv+Q/yfpptY/wDwa0n/AIrD40f9e+k/+h31bX/B1N/x7fAv66//AC02sX/g1p/5HD40f9e+k/8Aod9XrR/3RfP8z77/AJpyPz/9KZ+xX936UY3UdhTJ/wDV7v7vNeSfArZEmSDQGwa+BfGP/Bcbwf8AAj9tXxp8JfiVplxoem6FexQ2HiO1LTwbZIIpdt3B/rE27/8AWx71buqfeb7b+H3xA0X4oeFrXW/D+rafrWk3yeZa3tlcLPBMnqrqSrVrOjKK1OivhKtKKlJaHQ4+f8KWmg4+tOrBeRzhRRRVAFI/3aWkf7tAH5qf8HOH/JnXgf8A7HBP/SK6r80/+CSP7G1n+2n+3n4H0rWIYbjw34Ln/wCEu1e3k/5eIbUqsUX+611LbK3/AEz31+ln/Bzh/wAmdeB/+xwT/wBIrqvlL/g2a/5P68Uf9iHe/wDpw06vUpf7oz73A6cPz+f5n7qQ/eqWkT7tLXkx2PggoooqgCiiigAooooAKKKKAGv0r8yv+Do/9jNf2iP2DU+IWmWqzeI/g/d/2nuC7mk0yfbFdp/wH91N/wBsK/Teue+KXw/034sfDjXfDGtW/wBq0fxFp9xpl9B/z2gniaKRf++XNAH8Um3zE2J5jsy7VG3buX5tv/fPzf8AfNfpV/wVN8Ot+1R/wSD/AGUf2kYkW41zQ7Bvhz4ou4/vTC381LeX/gMlrM3/AG818BfHj4Nal+zr8cfGHgDVlZtQ8G6zd6JO4/5aNbTPEr/8DVFf/gdfqd+yX4H/AOFwf8GoXxqs7lPOPhXxNfatYb/+Wf2WewvGb/0d/wB9UAfkNJ8rbGX7rMv+0rL96tPwL401T4b+NtH8Q6NdTafrGh3sN/ZXETbZLeeKVXR1/wBpWVW/4DWVndu+8v3f91l/ho+7y33fvY/z/n5aAP7KP2Nf2lNP/a6/ZZ8B/EzS1WO38YaPBqEkStu+zzFds8P1jlR0/wCA16pmvyX/AODSn9ox/HX7Gvjb4c3lx5lx8P8AxB9qs0Z9zJaXyb1GP+u8U7/9ta/WKzbBePOdh4+n+c090VbcmooopEhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFDdOOtAH5F/8AB1N/x7fAv66//LTaxv8Ag1oP/FZfGf8A64aR/wCh31bP/B1N/wAenwL/AN7Xv/cbWN/wa0H/AIrL4z/9cNI/9Dvq9aP+6L5/mffR/wCScj8//SmfsT/d/wB2m3H+r+anJ2oryT8/6H8/3/Bcr9j/AOIngb9sPxv8Srzw1fP4F8S3MMtnrEC+fbQ7bWCJhcbPmibcny+b97b8tfOn7KP7cfxM/Yn8YPqfgPxNdaZDM4+2aZPtn03UNveW23KrN/ddWVlXdtav6fNd0e11nTJ7W8hhuLW5RopoZUDpMjAgowP3lIPSvwt/4LV/A79l/wCE/im5/wCFZax/ZvxGW5AvfDWgBJ9Fh+Yb/N3v5do23PyQbv8Arim7zV9jB1/afu2j9CyPNo4mCwdanzfifa/7Cv8AwX5+HP7SMlnoHxAjh+G/jB2EO66uN2k3znAXyrhtvlM39yZV/wBl3+9X6B2N9FeQJJC6yRzJvRl+ZSPXNfyP/dVV3bvl67drfeb5WVv9nb/vf3q/pv8A+CXn/KPT4N/9ilp//ohawx2FjS1ieXxNktHBtVKStzHvadadRRXnHyYUj/dpaR/u0Afmp/wc4f8AJnXgf/scE/8ASK6r5T/4Nmf+T+fFH/Yh3v8A6cNOr6s/4OcP+TOvA/8A2OCf+kV1Xyn/AMGzP/J/Pij/ALEO9/8AThp1epS/3Rn3mD/5J+fz/M/ddPu0tIn3aWvJjsfBhRRRVAFFFFABRRRQAUUUUAFNbp6fhTq4f9oT4/eEv2XPg5r3j7x1q9voXhXw3b/ab+8nPyoudqqF/iZmKqoH3mYUAfzff8HOPwMi+C3/AAVU8RatHb/ZdP8AiFomn+JFkQfu2n2vay/8CaS0Df8AbWvuX9l/4aL+zz/wabeOdQ1JPJk8aeGNY8QMP+v6dorZv+BRC3b/AIFWT8VP+Dtb4X6n4kvLGz+A/iTxX4dZngS51nULWB7hF+9m38qb/gSO/wAuV6V1esf8Fpv2Qf8AgqH+y5qvwL8eap4y+BOleJLe3silxFb21paCCaN4YormIzwIitEm5ZFRdqN92gD8AI/7zL97a2f975v/AELdSyfNG6/e3Ltr70/b6/4IB/FL9lPwvN488A32n/Gz4UMjXUWueHIllurW3/haa3R33xLtb54GZV+98v3a+DPlflWV/wDbDKyt/u7fl+X+JvvN/FQB+rX/AAaQfEltA/bw8eeGGbba+JPBs1zs/vS215bbP/HJpa/octz+9b3UED09v8+tfzPf8GuLyJ/wVb0nZ0bw1qav9NiV/TJD938TVBsxzZ4x60tFFSAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB+Rf8AwdSZ+y/AvPX/AIn+cfTTaxv+DWg/8Vl8Z/8ArhpH/od9Wz/wdSnNr8C+3/If4/DTaxv+DWg/8Vl8Z/8ArhpH/od9XrR/3RfP8z76P/JOR+f/AKUz9iF+6v0pJDsTjrSr91fpTLjmFucbuOeleSfAx2Pw2/4LS/8ABT74r6n+0R40+EWka1/wivg7w9MtpKmkZgvNU8yCN2aaY/Nt/eH5Y9i8/MGba1fB3wc+CHi79oLxvbeGPA/h3VPEutXn3LWwiZvLXcu52ZtqxR7mXc7Mu1mVt3y1+vXxP/4IN6p+1p+3d8QPiH488Q/2B4I1bU0msdO00iTUdQRYolO52ylurbW/vt/spX31+zh+y14D/ZT8GJ4f8B+GtN8Pad1l+zxfvbpx/HNI37yV+vzMzV631ylShaG59zDP8LgcKoYRe+fnH+wz/wAG49joX2TxB8ctSj1m8j/eR+GdKlZbOHjpczrtaX+L5E2r/tvur9S/AXgzSfhz4XsND0KwtdK0nS4FtbOytoxFDbRJ8qoiL0Wtv5dzetKFzXn1sRKfxnyeOzHEYufNVY5Pu0tGaTcK54nELSP92lpH+7VAfmp/wc4f8mdeB/8AscE/9IrqvlL/AINmv+T+vFH/AGId7/6cNOr6t/4OcP8AkzrwP/2OCf8ApFdV8h/8G2niPT9C/b81mK+vrOzk1Twbd2dok8yRNeTfbbBxFEG+++1HO1fm2ox7V6lL/dGfeYL/AJJ+fz/M/edPu0tQwE/pmpq8mOx8GFFFFUAUUUUAFFFFABRRUc5ZY/l65oAh1F1ggLu3lpHlmbO3bwec1+DH/Bzh/wAFUvhz+078PvDfwd+GHi+HxUmieIH1XxRNpiltPZoIpY7e3S6+5K3mOzMsW5f3fLV9B/8ABwZ+1t45+LHxf8C/sZ/Bu88nxf8AFJo38UTwzbTDYybvLtZWX5kSRIpZZf70SBOVkr5h/wCC5v8AwRk8A/8ABPD/AIJ5fDPVPh/p0l5qWh+IVsvFviSZV+3aw91bsqSv/ciSWJVSJfljWX+Jm3MAfkjJ83zfebc3I+7/AHsf99M1N/hb/wCJ3L/wKjd95W+995sfdXczN/6FuooA+gv2DP8Agp38YP8AgnR4zh1L4f8AiK6/4R+SXzb/AMNX5efR9UHdTEzKyM33VliZXXd/d3K30d/wVK+Bfww/as/ZT0H9sb4H6LH4RsNY1ceH/iJ4STGND1hv+XiLYBGNz7d21VWRZIHVVZnVfzxXb91m27vlz/6F/wCO19KfCX47+LPA3/BOD4nfBu1+GfjC+s/ipr2ma+/iL7FM1nY2tr5cgRUWJlZmZF+fzV+Xd/eoA+u/+DSf4byeJP2+vGXiRo2a18M+DZomc/w3FzdWyr/47FLX9Etsdsjp/dAP+fyr8KP+DUr9pL4PfCFPHHhHxB4mj0P4tfEDWYY7Kw1FGhiurW3Rmit4pmHl+f5ktxui3KzbV4bbuH7q2Sqkj7VI3AH7pHHPH69O1EeoLzLFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH5F/8HU3/Ht8C/rr/wDLTaxv+DWg/wDFZfGf/rhpH/od9Wr/AMHUatj4Ht/D/wAT4AfQadz19/TtWV/wa0f8jl8Z/wDrhpH/AKHfV60f90Xz/M+/X/JOR+f/AKUz9iY/4fpT0+7TE7U5WwK8k/P47DqMYoooGRTLuT057V8Ff8FnP+CmfxA/4J13fw5/4QjSfCOpR+MF1MXn9tWtxN5P2X7Ls8vyp4vvfaH3ZDfdWvvhz8tRywJP95d1VTqJS95G2HqQhPnmro/Cv/iJk+Pf/Qr/AAnP/cL1D/5MpP8AiJk+PP8A0K3wl/8ABZqH/wAmV+6K2EW7/VRj8KX7BH/zzj/75rr+sUv5T2v7WwH/AEDf+TH4wfs7/wDBw98bPjB+0J4D8Jap4b+GNtp3ibxDp+lXUtrpt758cU9xHE7Juu2VW2u33lav2as38xNzfL61MbWOP7yx/wDfNOzj/drCtUjL4VY83HYmhWa9jS5PxPzb/wCDnH/kzzwR/wBjgn/pFdV+Bfxp1a60HR9JvrG5uLG+s9Sint7m3l8qW3dUfa6P8uxlb+JWVl/8dr99P+DnE/8AGHngj/scE/8ASK6r8Bf2gP8AkU7H/r+T/wBAkr0Kf+6M+xwP/IgqfP8AM/Uv/gj/AP8ABzDfaNf6b8N/2lNSkvLGSRLTTvHjL+9t+gEWpqv3l+b5bpVX7q7927dX7kaJqNrrdjb3dndQ3lrdQrNFPDIJI5kYBkdGHysrKeq/Ka/iUVd38O75v9nd/wAB3fLu/wB6v1a/4N6P+C21/wDs4eMND+BnxU1eS8+GuvTJa+GtTu52ZvCt07Mq27O3zfY5JPlXd/qn/wBmvIjsfAH9EVFV7eHZGo2qPl/u7f09ParFMAooooAKKKKACmyfcPGadWb4s1yPwv4X1LUpTti0+1luXP8AdVELf0oA/Hr/AIJAaev7Y/8AwXy/am+Nl839oWfgOebQdEnb7sbNO1hE6/7X2WxlX/trX6i/td/sxeHf2zf2bfF3wy8Vws+ieLrFrGZ14ltX4aK4T/ppFIqSr/tIAflzX5q/8Gh+hPqf7PXxv8aTLuuPFHjVI2b+95Vqsv8A6FctX6+UAfx8/t3f8E4vir/wTz+Kd94f8feH9QTSY5nTS/EttbyvpWsRL0mSZfkWRl2s0Un7xf8Ad21578E/2c/H/wC0h4oj0fwB4J8UeMtSmbY0Gkac1ysbf3mdf3aL/tPtVf7y1/Z/f2cOoWzQ3EMdxDIMNG6hlb8DTNK0y30e2ENrbQ2sPZIoljVfwFAH4k/8Euv+DXG40TWtJ8cftKPZ3X2UpcW/gSwn8+Lcu1h/aEyfLLt/55QblZmXc7ba/a/R9LttDsYLW1tIbK3tYkhhghj2Rwoo2oiqvyhVX+FeFrTooA/O/wD4Le/8EffCH7ZfwF8ReOPCvh+z0f4zeErSXWNL1bT4fIm1ryB5rWdzs/1rSgFVdvnV9mPl3V1P/BAX9ve//bw/YE0e+8RX0l9448Ezv4e1ueV90155aq1rdP8A7UsDpuZvmZ0kr7kkG5cHvX41f8ELpI/2P/8AgsR+1V+z+3+h6bqF1LrOjW7/AHdkFxviRf8At31Bf++KAl8J+ytu5eFc/exg/Wn1Hbn5P1/z/P8AGpKA9AooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD8jP+Dp19y/A3C7vLOuh19Qx01T+ma8G/4IP/t0/DT9iXxB8Srj4keIX0JfEkOnpZMtheXnnGI3LS/6pH/56p/DX6sf8FPf+CfFh/wUN+AS+Gvt1rofiLR7ptS0XUpLcyrHN5LRmOTBBEb5XdjP+rU4OMV+OOvf8ED/ANqTRdUe2tvh3a6pDGCEuLXxJpvkvjuPOuUcFvQpx617GHnTnQVOTSt39T9CynE4GvlSwdeoouN92lu29L7n6qR/8F6f2XVP/JQ7n/wndU/+R6d/w/o/Zd/6KJdf+E3qn/yPX5O/8OH/ANqv/olaf+FLo/8A8k0jf8EIv2qkxu+Fsa7jgZ8TaPyf/Amp+q4f+ZfejL+w8mt/HX/gS/zP1i/4f0/suf8ARRbn/wAJvVP/AJHpW/4L2fsuk/8AJRLj/wAJvVP/AJHr8nH/AOCDv7VjK2PhbGD/ANjLpHzf+TNMl/4IS/tVQFf+LW85ATHibSmwPp9pGPwan9Tw6V+dfeNZDk+/t4/+BxZ+tH/D+z9lw/8ANRLz/wAJvVP/AJHpv/D+j9lz/oot3/4Tmqf/ABivw5+NP7DnxQ/Zy+OHgn4a+NPDq6L4y+I08Nt4esGv7Wf+0HedYEXzYneKL95s/wBa6169/wAOKv2qZEjb/hVw+bLYHiLSPl/8mW/8dpfVcN/OH9iZL0xH4o/Wj/h/R+y7/wBFFu//AAnNU/8AjFH/AA/o/Zd/6KLd/wDhOap/8Yr8mf8AhxP+1R/0S8/+FFpH/wAk0f8ADif9qj/ol5/8KLSP/kmj6rhv5yf7Fyb/AJ//AIo/Wb/h/T+y3/0Ua5/8J3VP/jFL/wAP6P2W/wDoolz/AOE7qn/yPX5Mf8OJ/wBqj/ol5/8ACi0j/wCSaP8AhxP+1R/0S8/+FFpH/wAk0fVcN/OH9i5P/wA//wAUfR3/AAXK/wCCk/wc/bP/AGa/C/h74deKLjXNW03xJHfzRNpV5a7IVtZ0Z980SqfmlRdv3vmr85fhx+xjr/7Z/gz4jWvheO4vNc8BeF5vFlrZIm9tU8i4t4JoFX+95FzM6L/FJEi/xV9IWf8AwQj/AGpLq/him+HENnHI43TS+I9N2Q87gW2XDNt3D+FW/wB2v1Z/4JH/APBLuH/gnz4A1S71jUrPWvHnigRDUri0RvstnCnKW8Duod1+Ysz/AC72Vfl+StKlSjSo8kHc1x2KwOEy54TDVOa/zP5VkaORd6Pv3d93yt/dZf8Avn5m/wBrb/DQ6edGyfL+8+Rfl3fM3yrj/a3NX6ff8HF3/BIKb9kP4sXPxk+H+mqvwv8AHF6z6tbQQfu/C+pysv3k2/LbTt8yt/yzbcn3dlfnx8Jf2bfiJ+0BrFtYeBfAPjDxbcXzrFAmlaTPeLJ820q0qKqKqtu+ZvlVdzN8q14p+fH9M/8Awb5/tk6p+2h/wTX8L6l4gu5NQ8S+C7mbwnql1L80l09rt8qV/wDba3eHd/ebc38VfcFfEf8AwQX/AOCffiL/AIJ3/sNweG/GrQp4y8TarN4g1i2ifzF015Y4kitd6/K7JFEm5l+XezYr7coAKKKKACiiigArzX9snX/+ET/ZD+K2qbtv9m+D9Xus/wB3ZZSt/SvSq+Q/+C7Pxvj+Af8AwSg+M2qtcRw3OraI3h+13Hb5j3zra7R/tbJX/wC+aAPnn/g0n0NdL/4Jb31wysrah431KXn+LZDaxZ/8cr9Qq+H/APg3Z+C0nwT/AOCSfwuhuYWt7rxJFdeI5UcfNtvLh5Yf/IPlV9wUAFFFFABRRRQA2T7hr8aP+C4fg3xJ/wAE5/8Agpj8Kf2zvCWk3GreHZPK0Xxnb24ZfnVHgXc4+59otZXiRm+XzbeJf4q/ZmuV+Mfwu8N/G34Z6x4Q8YaRa694Z8SWzWGo6fcR+ZFdxPwVZf1z/DjdkYzQBk/s2/tC+Ff2o/g14f8AHvgnWIdc8OeJLRbq1uIyu8A8FJAPuSqwZXQ/dZWHavQM5FfiL/wT48SeJv8Agh5/wVnvv2Y/FeqXV98Hvi5cJd+D764YeXDLKzLaSqe0jMrWs4/idYm+7tZv22gl81P880X6B1JKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAEK5YHnj3paKKACq93Humj4z8rcflVimycj8aOa2o0z8bP+C/lynjr/gr5+xh4V0V/M8TWus2t5Kq/ejhfVrbY3/kvNX7JJ1r8bP2S7I/tj/8HR/xe8Wakv2jT/gnpl1Yaajr8trLAkWnIv8A31Lev/20r9kU/dttznqelAiXFGKKKADFGKKKADFFFFAHLfFv4XaD8cPhrrng/wAVaTa654c8SWUun6jYXKgw3EMilWVv/il5VgCK/Fb9m74h+KP+DcP/AIKNSfB/x3qOoah+zP8AFy9F14e1i9YtDpLlkiFz/dWSJmRLz+8jQT/d21+6D9K+Y/8Agq3/AME9tE/4KQfsg6/4EvFgtfEVup1HwxqbpltL1CJG8s7v+ebqzxP/ALMrd8UAfSVhPFewRzxukiSxqyujbtw6/e7j/Perlfzn/wDBOX/g41+I37AENr8Jfjd4ZvvGPhXwfO+h74GVPEHh0QP5RtfmO24SLaUVZXWX5fvv91f2e/ZN/wCCqvwE/bXs7f8A4QD4leHb3UpB82kXtx/Z+qwt/dNtMFkJ7blUr6M1AH0lRUcXQfrmpKACiqesatBo+nyXN1PDa2sI3STSyCNI19WJIxXxz+1r/wAF7v2Zv2Ro7i3v/iFY+MvEFv00XwkyatcM+1vkeVD5EX/bSVW9qAPs64+4v3vvdq/Fr/g5u/aS039pvxj8H/2WfAniCx1jxRr3i+I6/Z2UvmtpszBbe0ilK/Ksm64kl2feXya+UP8AgoH/AMHM3xn/AGs7C88N/DmJvhD4Lul8qWSxuPP1u+Td8yvdbVWHcu7csS7v9tl3Vzn/AAbQ/AFfjz/wVY0DWL2Nruz+Hel3viu6MjN81w222hZtzNufzLpZdzNubZ92gD+lr4Y+ANO+E/w80Hwxo8fk6V4b0230qzT+5DAixIv4KtdDUEDc/eVj/wDqqegAooooAKKKjlOQF2hs+tAElRzv5ac9M15r+0b+1Z8Of2TPAMvib4leMtD8I6LGuY5b+7EUlwy/Nsjj+/I/+ym5jivy6+O//Bwt8VP20/F158Nf2Lvhb4g1zVZ/3D+LtWsv+PXd8vmxW+1ooV5XbLdOq/Mu5FzQBQ/4OYfEml/ET9rz9lnwD4TmjvPi3a6+0qx2/wA09nBPcWa2+/8Au75ULKPRGr9m7bJfP4Gvzi/4JOf8EQrn9lz4q3nxs+N3iJviR8dNczcC9mme6g0WV12ySiR/9bcsrbC/3UXcsWFY1+kUaeWtHW5L+K62HUUUUFBRRRQAUUUUAFFFFABRRRQAUUUZoAKKKKACiiigAoVQo4GOc8UUUAFFFFABRRRQAVXuwSOvGVIHpz/9arFNmQMv09qA30PxH/Z5+Mejf8Esv+Din476f8ULiPw34a+N6vf6Nrd2yrZh7q6W8hLv/AjO9xAzf89EWv2u0u4jvLSOeGSKWOdFdXQ7lYeoPdeeK8D/AG+/+CZXwp/4KS+AY9D+I2kvPdacsn9maxZSCDU9KL9fLkwQUOOUdWUkA43ANX56r/wQE/ap/ZHZl/Z1/as1Kz0OE7rfR9ZnutPgj/2WSPz4H/3vKT/do16hr1P2Sor8b/7L/wCCuvwYOy3vvAvxGtbfnLf2Q3mL7My28jf+hVHcf8Fdv+Ch3wDTzPiH+yTb6/Z2vzS3Gj6Tfrx/11t5rqNf++dtAH7KUV+T/wAFP+Dr74U6j4hj0b4ufDfx98KNUVxHPIYv7Ugtz/edVSK529fuwNX6Nfs5ftUfDn9rTwQPEfw18aeH/GWkN8rzaZdLK1u39yROHjb/AGZFVqAPRqKKKACmyf6v5m206s3xJrVn4X0O61LULqGzsNPia5urieQRxwxINzu7NwqqoJJ/ug0Afy6/8HGHwy0r4W/8FePiXHpMaww+IINP164Qfw3Vzbqs3/fTRb/+2tfDvlrJtZusbblb5VZW/vLu/ir78+OH7Nfxm/4Lv/t2fGT4rfBzwpJ4h8J2+tRaXb39xfwabFHaRRLFaf690ZmaKLzWRN23zfu/PXqXwh/4NM/j/wCMljk8WeKvh34NhfrF9qnvrhf+AInlt/38oA+Cvhn+3h8b/gvp8dr4S+MHxO8P2Mfyrb2fiO8WBf8AtlvaP/x2u3vP+Cvn7Ul5b+W/x++J2zb/AMstbaBv/Qlav1X+GX/Bn54HsFi/4TL4x+LNYY/fXR9Jt9NX/dBlefK/8Br2LQP+DUz9l/TbVUurn4nao3Tdc69FH+kVulAH89HxU/aE+IXx0vGm8bePPHHjB/8AqM69cXn/AKGzL/7NXGxxrGqqi7UX7uPmVa/ot8f/APBpb+zxrsbf2J4p+KXh2b+EJf291Cv/AAFrfd/49XhnxB/4M8X8yR/CfxyVk/hi1Xw4u5frKkzf+gUAfiPu2/w7v4cV+3P/AAZzfD62e4+PPix13XsZ0fRon/upi6ndf++tlebXv/BoR8Yre522/wAUPhpJb9N8i3qSfN8v3fJYf+PV65/wbJ3sn7IP7Yf7RX7N/jCS1tvGFneQ3dttLCO8+wyz28/llvvKyTW8qLj7j0AfthRUEGJRu/vDPTHBqegAooqvdRrJDtbO3P8An8KAG6hcJZ25kkdY44/mZi2OO9flJ+3t/wAHBusa58VpPgn+yH4bb4pfEqaVrKfXobM3Wn2cq8P9njyqztH/ABSs3kL/ALa1xX/BVD/go140/wCCl37Q0P7IP7K+o/b11CV7fxt4ttZfLtfKVts1uky8rap/y1lXd5vESbt25vv7/gm7/wAExfhr/wAEyvg/DoPguwjvNeurdE1vxDcxL9u1aRf7xHyxRK27bEnyr/tNlmndj9Nz4V/Zo/4Nw/E/7Q3j5Pid+2V8Rtf+IXi29k8w6BaakzRW6fKVgmuVCgKvK+VarFGv8LN3/Uz4H/s+eD/2bPAlr4a8B+FNF8J6FZjbHZabbrAjn+82OWb/AGnLMa7a2VjGrMfmI+n6VJT5UmL3urI4Y2+8/Ldh6VJmimo+/seOORimPcdRRRQIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAopEdZEDKQysMgg8EUtABRRRQAHPGPxooooAY8KSt8yjcPzpj2hP3ZGXnvzU1Jty2eemOtPmY7kElq/wDCV/Nh/I0Zk25Mf0HFWKD04p3DTax5r8dv2Vfh1+1H4cbSfiB4H8N+LrNkKY1bTI7iSIescjLujb3X5q/NX9or/g2q1D4SePZviF+yL8UvEHwl8WKxlTSrrUJRayd9iToGk2ttHyTrKv8Au4r9dKCNwqetxeh+MNp/wV8/bg/4J5r/AGf+0X+z/J480Oz+V/EuixG3Zk/vG4tVltHZuvzLB/tba9Q+H3/B2b+zl4gs4/8AhIPDvxQ8K33/AC0ifS7e8jX/AIHFMd3/AHzur9STbBR8v6mvPfHX7I3wr+Jtw0/iT4Z+APENxJ9+XU/D1ndyN+LxtQr9Qu2z8/8Ax/8A8HYf7N/h/SzJoWi/E3xRfKvyQQ6KlrG7f3S8sny/Nj7u6vAvG3xJ/bG/4OCNnhTRfCNx8A/2e9QlQ6pqV8X8zVIVOdrysqNdK3/PKKIQMwXe3y7q/XXwV+xt8JPhrfx3nh34WfDrw/eRHKTab4as7aRf+BpGGr0mGE5HVV9D/h0p9NA16HkX7GP7Ifgr9g/9nrQ/hz4HtPsei6Km6S6nYNdanPJzLdXD/wAUsjD5vZVVcKqqvrwt+eW49AKVoNuNqruzwW/h+n+HFSCktCnsQ/ZBjhpF+hqVF2IFyTgYyeppaKCelgIzSYDZ/KlooArywuH4+Ze3qDz/AI1+V/8AwXK/4J7fEDw98YvDv7XH7PMdxH8Uvh/5VxrdhZQtLLq1vErILjyl/wCPhli/cyxfeeD7u5l2t+rFQ3ETPjb0+uMH1pSu1ZA9tD4g/wCCaP8AwXK+EP7e/hHT7G+1rTfAPxLjQR6h4W1e8WFnm+VWezmk2/aImb7v/LT5vmVa+21nXy1ZTuDc8N/jXxR+3L/wQH/Z6/bl1y617U/Dt14P8W3p3za14bdLOS6kPV5omUxu3+3t3/7X3s/LVz/waizQAQaf+1B8SbXR/u/YXsHZSn91tt2i/wDjn4UwP0J/aw/4KTfBH9izw7NefEb4i+HdDuIkLrpkd2LnVLg9gltCWlb/AL5xX5b/ABk/4KCftHf8F6/FN/8AC39mnwzrXw3+EMkn2LxD4w1B2tp7qBvvedMn+pVl/wCXeLfLJ91mVd619J/s4/8ABrh+zj8FNZj1TxUviv4pakH3uuu3ixWUjdebeHZvXj7rs3/Aq/RHwL8PdH+GPhSz0Lw3oml6Do2np5drY2Fsltb2y+iogUL+FGr2DV6I+ev+CZ3/AAS4+Hf/AAS8+EB0LwrCdU8QaoiPrviG4gSO71SVf4QF/wBVAv8ABEpO3+Jnb5q+nLe285lkkHT7q+lTQwbANx3N/KpKq9ilZbbhRRRUkhRRRQAY5zRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAI6LIu1lDKexGaWiigAooooAKKKKACiiigAooooAAciiiigAoPIoooAKaIVB/xNOooAKKKKACkdd6EbiuRjI6is/xdfzaX4W1G5gby5re2kkjbAO1gpIODxWjT6XK5WoqX9aW/wAwooopEhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH//2Q==)

**Trabalho 1**

**Taxas de Leitura/Escrita de processos em bash**

Sistemas Operativos

Professor Nuno Lau e Professor Guilherme Campos

Trabalho realizado por:

João Nuno da Silva Luís (107403) | 50%

Diana Raquel Rodrigues Miranda (107457) | 50%

**Índice**

[Índice de imagens 3](#_Toc120907485)

[Introdução 4](#_Toc120907486)

[Metodologia 5](#_Toc120907487)

[ Declaração de Variáveis 5](#_Toc120907488)

[ Tratamento de argumentos 7](#_Toc120907489)

[ Tratamento da pesquisa por data e regex 9](#_Toc120907490)

[ Extração e tratamento da informação 10](#_Toc120907491)

[Testes 12](#_Toc120907492)

[Erros 14](#_Toc120907493)

[Bibliografia 17](#_Toc120907494)

# **Índice de imagens**

[Figura 1 - Declaração de variáveis 5](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884152)

[Figura 2 - Código da implementação do getops 7](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884153)

[Figura 3 - Função help 8](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884154)

[Figura 4 - Tratamento das datas para aplicação do filtro 9](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884155)

[Figura 5 - Uso do regex para filtro através do user e do nome do processo 10](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884156)

[Figura 6 - Código para a extração da informação necessária 10](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884157)

[Figura 7 - Armazenamento de toda a informação formatada num array 11](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884158)

[Figura 8 - Segunda leitura dos valores rchar e wchar e cálculo do ReadB e do WriteB. E cálculo das taxas de leitura/Escrita (RateR e RateW) 11](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884159)

[Figura 9 - Leitura dos valores iniciais de rchar e wchar 11](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884160)

[Figura 10 - Print da tabela final 12](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884161)

[Figura 11 - 1º teste. Execução do programa sem nenhum filtro 12](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884162)

[Figura 12 - 2º teste. Filtro pelo nome do processo 12](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884163)

[Figura 13 - 3º teste. Filtro pelo nome de utilizador 12](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884164)

[Figura 14 - 3º teste. Filtro pelo nome de utilizador 13](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884165)

[Figura 15 - 4º teste. Filtro por data 13](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884166)

[Figura 16 - 5º Teste. Filtro por número máximo e mínimo de PID e filtro no número de processos a visualizar 13](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884167)

[Figura 17 - 6º teste. Ordenação da tabela pelo RATEW 13](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884168)

[Figura 18 - 7º teste. Ordenação da tabela na ordem reversa (crescente pelo RATER) 13](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884169)

[Figura 19 - 8º teste. Execução do programa com vários filtros aplicados 14](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884170)

[Figura 20 - Tentativa de execução sem passar argumento numa opção que assim o exigia 14](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884171)

[Figura 21 - Tentativa de execução do programa sem argumentos 14](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884172)

[Figura 22 - Tentativa de execução com passagem de um número numa opção que exigia uma string como argumento 15](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884173)

[Figura 23 - Tentativa de execução com passagem de uma string numa opção que exigia números como argumento 15](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884174)

[Figura 24 - Tentativa de execução sem argumento de tempo 15](https://uapt33090-my.sharepoint.com/personal/jnluis_ua_pt/Documents/RelatorioSO%201.docx#_Toc120884175)

# **Introdução**

Este trabalho foi realizado no âmbito da disciplina de Sistemas Operativos do 2º ano da Licenciatura em Engenharia Informática.

Foi-nos proposto desenvolver um *script* em bash que obtivesse estatísticas de leitura e de escrita de processos, podendo assim ver o número total de bytes de I/O (Input/Output) que um processo leu/escreveu e posteriormente a taxa de leitura/escrita, correspondente ao número de segundos desejado, valor passado como parâmetro aquando da execução do programa.

Para além disto, foram propostos alguns filtros, que complementam a visualização de informação destes mesmos processos:

* -c: Filtro que permite, através de uma expressão regular, filtrar a seleção de processos pelo seu nome
* -u: Filtro que permite visualizar a seleção de processos pelo nome de utilizador
* -s: Ver os processos que iniciaram a sua execução após a data inserida como parâmetro neste filtro.
* -e: Ver os processos que iniciaram a sua execução antes da data inserida como parâmetro neste filtro.
* -m: Através de um PID mínimo passado como parâmetro, filtrar os processos com PID maior.
* -M: Através de um PID máximo passado como parâmetro, mostrar apenas os processos com PID menor.
* -w: Organizar a tabela em função da coluna RATEW, correspondente aos valores de escrita.
* -r: Mostrar a tabela em ordem contrária (crescente) à ordem *default.*

# **Metodologia**

O nosso código está organizado da seguinte forma:

Começámos por fazer a função help() que tem a função de auxiliar o utilizador no uso das opções disponíveis para a pesquisa dos processos. De seguida declarámos as variáveis necessárias e fizemos a sua inicialização. Após termos as variáveis todas declaradas, procedemos à implementação do comando getops para fazer o tratamento das opções de entrada. E depois avançámos para a leitura dos valores e o tratamento de toda a informação.

Por fim, procedemos ao print da tabela com a informação organizada.

## **Declaração de Variáveis**

Uma imagem com texto

Descrição gerada automaticamente

Figura 1 - Declaração de variáveis

Algumas variáveis foram inicializadas com valores *default,* facilitando a utilização dos argumentos de entrada no programa. Estas variáveis vão ser depois atualizadas, conforme as opções selecionadas pelo utilizador e utilizadas para devolver os resultados desejados.

Para o correto tratamento dos dados, decidimos usa um *array* associativo (**final\_info**) já que estes permitem associar diferentes tipos de valores a uma chave, onde neste projeto o PID de cada processo é a chave.

Para as variáveis relacionadas com os filtros, definimos a variável **nprocessos** que é iniciada com o comprimento do *array* **arrPID,** ou seja, com o número de processos existentes, esta variável vai ser usada mais tarde para definir quantos processos vão ser impressos na tabela.

Inicializámos também a variável **sortmethod**, que usa o comando sort em função da variável **$colOrdena**, que corresponde ao número da coluna pela qual queremos ver os valores da tabela ordenados.

Já as restantes variáveis foram inicializadas ou com o valor 0, ou com o valor (.\*), que significa que se não for passado nenhum valor, a variável vai aceitar qualquer caracter em qualquer quantidade.

## **Tratamento de argumentos**

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamentePara o correto funcionamento dos diferentes filtros, usámos o comando *getopts,* que juntamente com as condições de validação e aplicação por nós acrescentadas, vai proceder ao tratamento dos dados e realizar as tarefas solicitadas pelos argumentos passados.

Figura 2 - Código da implementação do getops

Para todas as opções que solicitam um argumento (exemplo: -p 5), definimos uma condição que impossibilitasse a passagem de outra opção como argumento, sendo impresso no terminal uma mensagem caso ocorra esse erro.

No caso da opção **-c e -u**, o valor passado na variável $OPTARG vai ser comparado com uma expressão regular que por sua vez mostra um erro se não for passado nenhum valor no $OPTARG ou se for passado um número.

Já no caso das opções -**m**, –**M** e -**p**, é gerado um erro se o argumento passado não for um inteiro positivo ou se esse valor for o último parâmetro, correspondente ao número de segundos a utilizar para a seleção de processos

No que toca às opções **-s** e –**e,** o valor passado entre aspas tem de ter exatamente 12 carateres (os espaços também contam), e tem de cumprir o comando *date*, ou seja, se for passado, por exemplo o mês “Nev” o programa irá devolver o erro de data inválida.

Por fim, nas opções -**r** e –**w**, em que não são necessários parâmetros extra, é feita a correta ordenação dos processos, com o recurso ao comando sort, havendo na segunda opção uma organização em função dos valores do RATEW, sendo por isso feito o ordenamento pela variável ***colOrdena***, que neste caso vai tomar o valor 7, referente à sétima coluna da tabela.

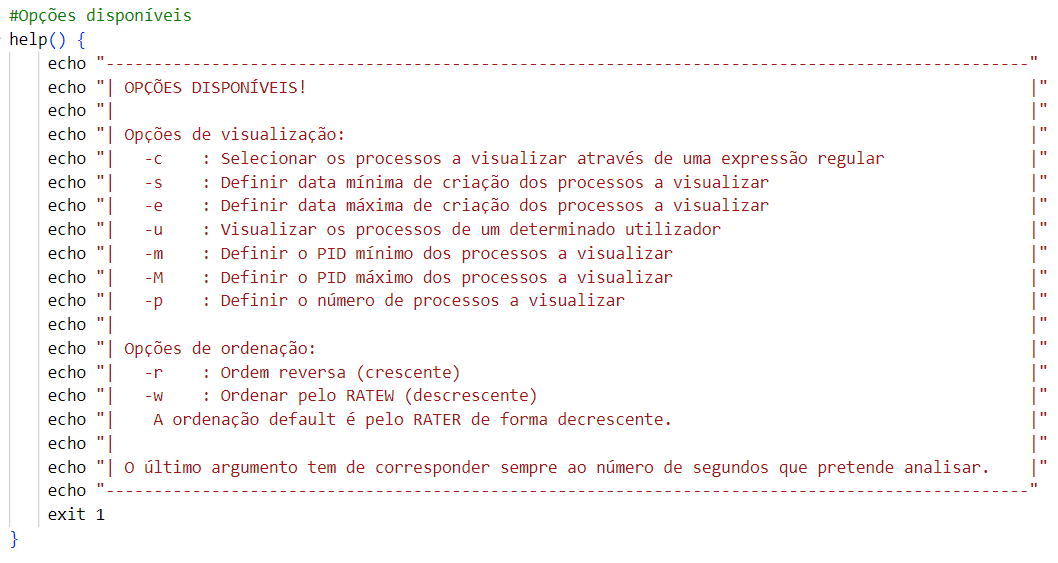
Para além destas opções, existe também a função ***help****()*, que é chamada quando ocorre algum erro, ou quando é utilizada uma opção que não está listada no comando ***getopts****.*

Figura 3 - Função help

## **Tratamento da pesquisa por data e regex**

No tratamento da pesquisa por datas (opções -s e –e) começámos por transformar, com recurso ao comando awk, a data de cada processo (variável **DATE\_Segundos**), a data passada no argumento da opção **-s** (variável **inicio**) e a data passada em **–e** (variável **fim**) em segundos.

Quando é usada a opção **-s** é feita uma comparação que verifica se o número de segundos da data do processo é maior do que o número de segundos da data mínima que queremos. Se for, a data do processo é então concatenada a uma variável que guarda todas as datas que satisfazem este caso separadas por um “|”. Esta variável serve para mais tarde fazer a procura de todas as datas válidas nesta condição através do comando **grep**.

Quando é utilizada a opção **-e,** é realizado o mesmo método descrito acima, com uma ligeira diferença que ocorre na comparação. Neste caso é verificado se o número de segundos da data do processo é menor do que o número de segundos da data máxima que queremos.

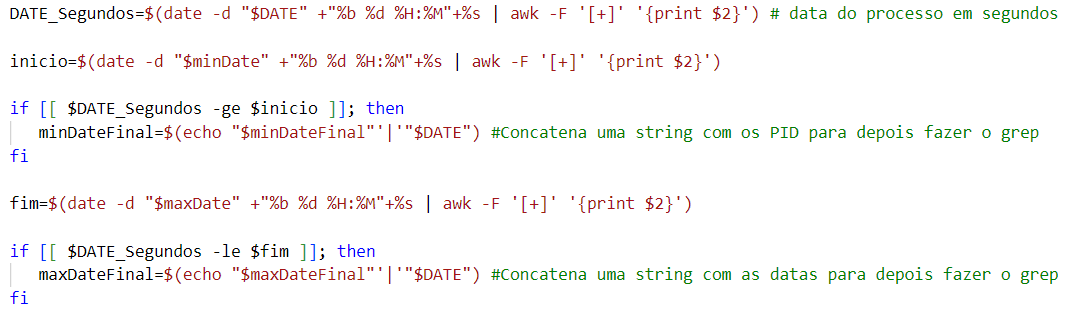
Para o funcionamento correto deste processo foi necessário exportar no início do código a linguagem pt\_BR.utf8 (export LANG=pt\_BR.utf8), pois o nosso computador tem o time local em português e o lang em inglês, o que estava a gerar conflito com o uso do comando **date** na transformação da data em segundos devido a diferenças como, por exemplo “Dec” em inglês e “Dez” em português.

Figura 4 - Tratamento das datas para aplicação do filtro

No tratamento das expressões regex utilizámos o comando **awk** para identificar o padrão passado pelo utilizador na coluna correspondente. Isto é, se pretendesse fazer uma pesquisa pelo nome do processo o comando **awk** iria só procurar o padrão passado na coluna 1, e se a pesquisa fosse pelo utilizador, iria só procurar na coluna 2. Neste segundo caso, como queremos uma procura exata, acrescentámos ^ no início do argumento para indicar que era o início da *string* e que não pode ter nada antes desse padrão, e no fim acrescentámos $ para indicar que é o fim da *string*, ou seja, nada pode estar depois desse padrão (userName="^$OPTARG$").

Figura 5 - Uso do regex para filtro através do user e do nome do processo

## **Extração e tratamento da informação**

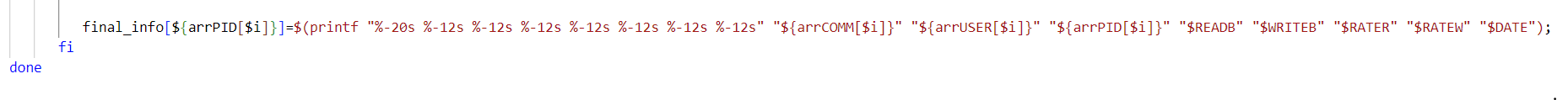
Uma imagem com texto

Descrição gerada automaticamentePara a extração da informação começámos por correr o comando **ps -e** que lista toda a informação sobre os processos em funcionamento e guardámos essa informação nas respetivas variáveis.

Figura 6 - Código para a extração da informação necessária

Em seguida, executámos para cada PID guardado no array arrPID o comando “**cat /proc/PID/io**” para termos acesso ao número de caracteres escritos e lidos por esse processo. Depois, executámos o comando **sleep** com o número de segundos passado no último argumento e após isso voltámos a ler o número de caracteres escritos e lidos por cada processo.

Com esta informação, pudemos então fazer a diferença entre os valores lidos inicialmente e os valores lidos após x segundos. Seguidamente, procedemos ao cálculo da taxa de leitura (**RateW**) e de escrita (**RateR**) de cada processo.

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteToda a informação necessária foi então guardada num *array* associativo (final\_info) que tem como chave o PID de cada processo e como valores todas a informações necessárias a ele associadas, já formatadas.

Figura 7 - Armazenamento de toda a informação formatada num array

Figura 8 - Segunda leitura dos valores rchar e wchar e cálculo do ReadB e do WriteB. E cálculo das taxas de leitura/Escrita (RateR e RateW)

Figura 9 - Leitura dos valores iniciais de rchar e wchar

Para impressão de toda a informação numa tabela organizada, executámos o comando **printf** para imprimir todos os valores guardados no *array* **final\_info** e utilizamos os *pipes* para executar os filtros que fossem necessários utilizando os comandos **awk**, **grep** e **head**.

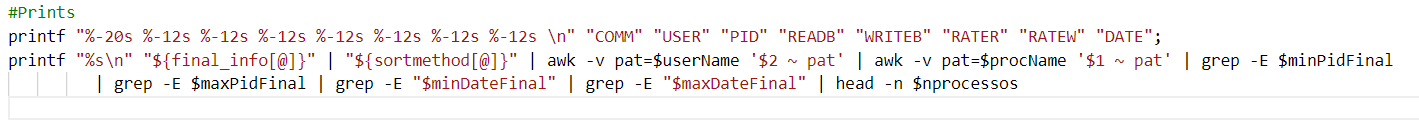
* O comando **awk –v pat=$variavel ‘$x ~ pat’** procura na coluna x o padrão indicado na variável;
* O comando **grep –E $variavel** interpreta o padrão indicado na variável como uma expressão regular estendida;
* O comando **head –n $x** filtra a informação para só aparecerem as primeiras x linhas.

Figura 10 - Print da tabela final

# **Testes**

Mostramos agora algumas combinações possíveis, que podem ser feitas pelo utilizador na utilização deste programa, e que por nós foram também utilizadas, para verificar se obtínhamos os resultados desejados.

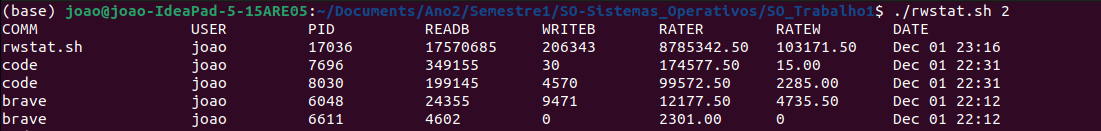
Assim sendo, começamos pela execução mais simples, em que apenas é passado como argumento o número de segundos.

Figura 11 - 1º teste. Execução do programa sem nenhum filtro

Obtemos uma tabela com os valores para cada processo, estando esta organizada pelos valores do RATER.

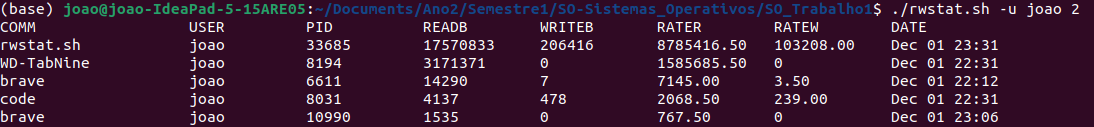
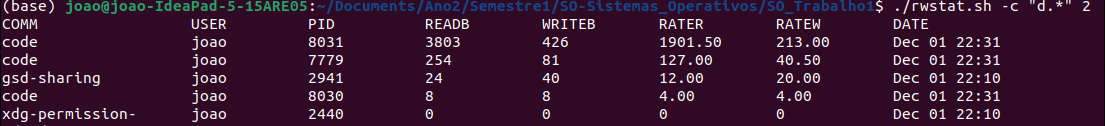
De seguida, usamos a opção **-c,** passando a expressão regular “d.\*”, esperando assim visualizar todos os processos que contenham a letra d no seu nome. Já no segundo caso, testamos a opção **-u**, que devolve os processos cujo utilizador seja igual à expressão passada.

Figura 12 - 2º teste. Filtro pelo nome do processo

Figura 13 - 3º teste. Filtro pelo nome de utilizador

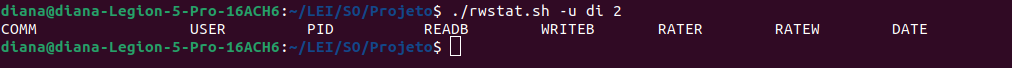
Esta última imagem não mostra nenhum processo pois não existe nenhum utilizador com o nome “di”.

Figura 14 - 3º teste. Filtro pelo nome de utilizador

Agora, testamos as opções **-s** e –**e** em simultâneo, e em cada uma, passamos como argumento uma data mínima e máxima, respetivamente. Estas opção funcionam também em separado.

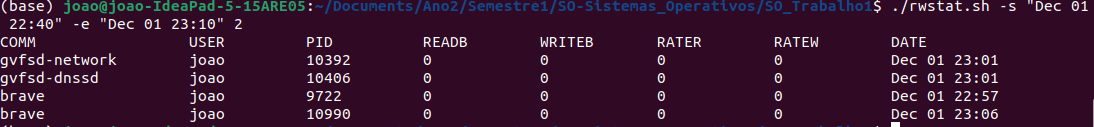
Já quanto às opções **-m** e –**M,** é passado um valor mínimo e máximo de PID. Vamos utilizar também a opção **-p**, de forma a limitar o número de processos a ver. Estas opções funcionam também em separado.

Figura 15 - 4º teste. Filtro por data

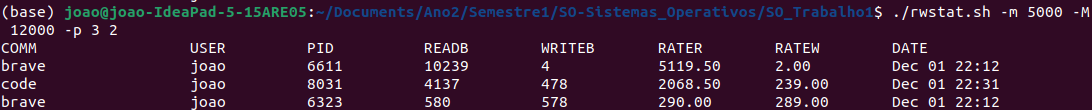
 De notar que, apesar de todos estas opções serem usadas, os processos mantêm-se ordenados pelos valores de RATER.

Figura 16 - 5º Teste. Filtro por número máximo e mínimo de PID e filtro no número de processos a visualizar

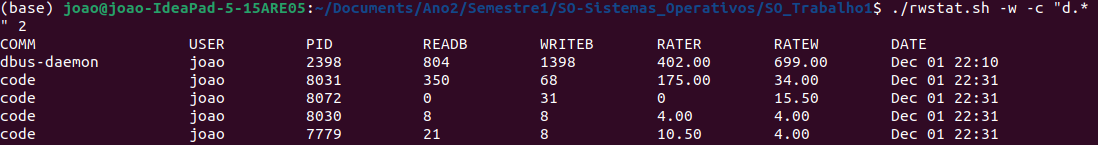
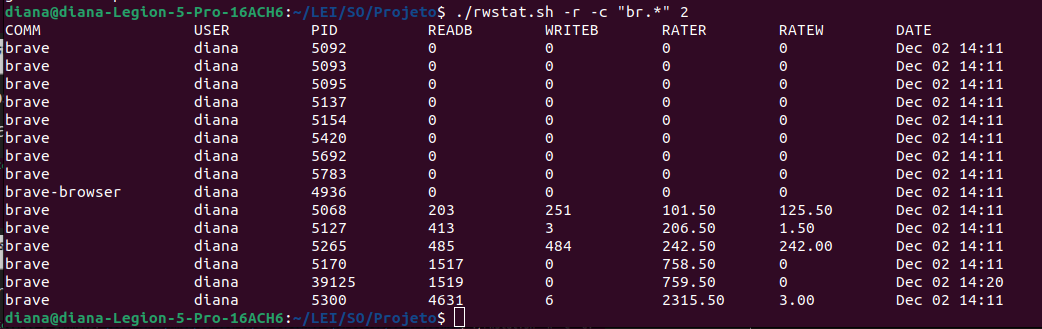
Assim, vamos agora testar as opções **-w** e –**r,** em simultâneo com outras opções, para verificarmos resultados.

Figura 17 - 6º teste. Ordenação da tabela pelo RATEW

Figura 18 - 7º teste. Ordenação da tabela na ordem reversa (crescente pelo RATER)

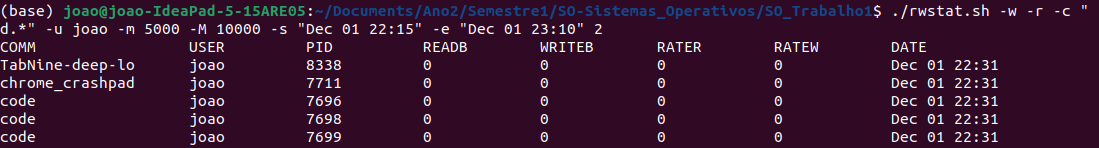
****Por último, vamos testar combinar bastantes opções ao mesmo tempo.

Figura 19 - 8º teste. Execução do programa com vários filtros aplicados

Posto isto, depois de todas as opções de filtragem testadas confirmamos que todas funcionam como esperado, tendo tido sucesso em todos os testes.

# **Erros**

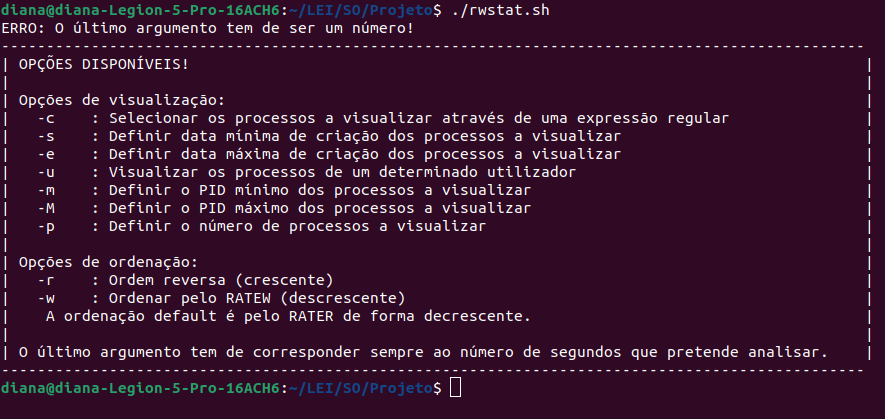
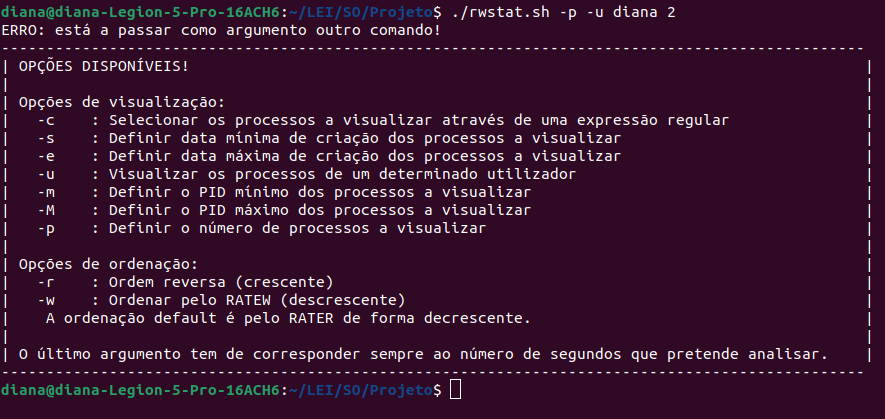
Nesta secção mostramos algumas mensagens de erro que poderão aparecer ao utilizador se não executar o programa ou as opções de filtro corretamente.

Figura 20 - Tentativa de execução sem passar argumento numa opção que assim o exigia

Figura 21 - Tentativa de execução do programa sem argumentos

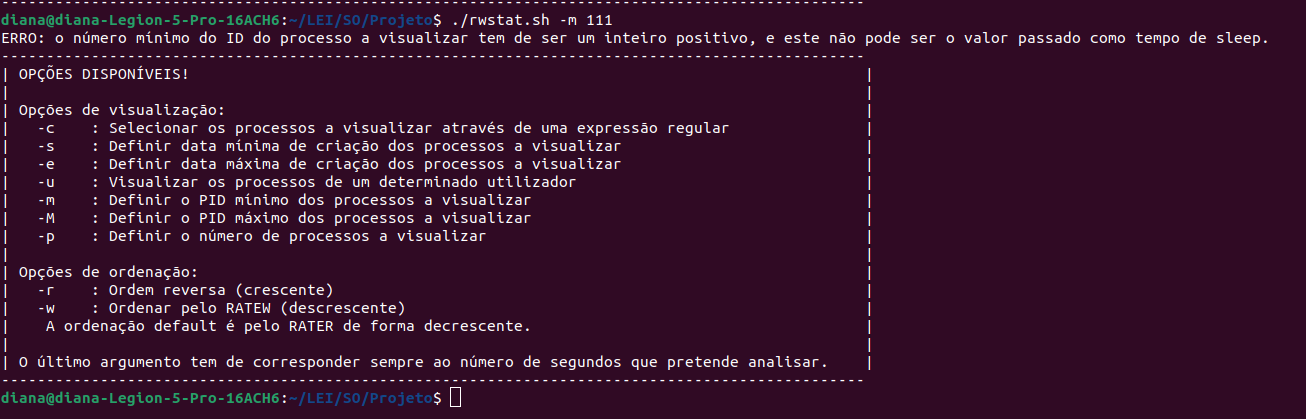
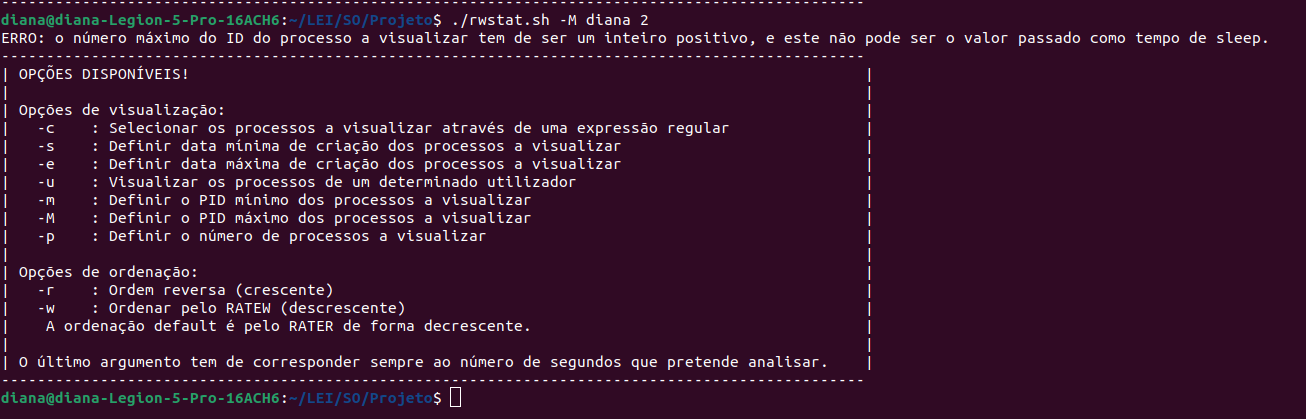
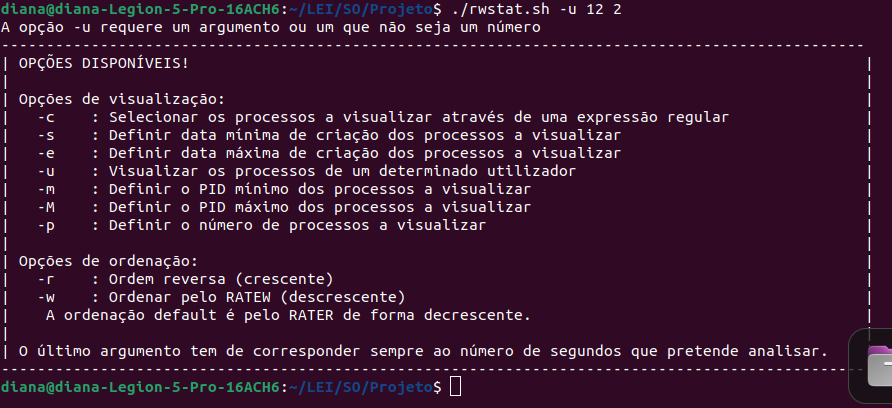


Figura 22 - Tentativa de execução com passagem de um número numa opção que exigia uma string como argumento

Figura 23 - Tentativa de execução com passagem de uma string numa opção que exigia números como argumento

Figura 24 - Tentativa de execução sem argumento de tempo

**Conclusão**

Com este trabalho, desenvolvemos bastante o nosso conhecimento sobre *Bash,* com o uso de novos comandos (por exemplo o getopts()), e também de algumas estruturas de dados, como os *arrays* associativos.

Para o uso correto desses mesmos comandos, surgiram algumas dúvidas, que foram maioritariamente respondidas com recurso a fóruns na internet ou com recurso ao comando “**man**” do terminal.

Para além disso, aprofundámos também o nosso conhecimento sobre Git e GitHub, visto que utilizámos ambos para o controlo de versões e partilha do trabalho entre os dois elementos do grupo.

Assim, concluímos que os objetivos deste trabalho foram cumpridos com sucesso, construindo um script conforme o solicitado, respeitando corretamente as opções de filtragem desejadas, e com visualização de mensagens de erro apropriadas para apoio ao utilizador.

# **Bibliografia**

* Imagem de Capa adaptada de: <https://www.hostinger.com.br/tutoriais/como-gerenciar-processos-no-linux-usando-linha-de-comando> consultado a 24 de novembro de 2022
* <https://stackoverflow.com/>
* <https://unix.stackexchange.com/>
* <https://www.computerhope.com/unix/bash/getopts.htm> consultado a 15, 16 e 19 de novembro de 2022