Une image contenant texte, Appareils électroniques, Matériel de bureau, capture d’écran

Description générée automatiquement

Table des matières

[Introduction 3](#_Toc138326039)

[1.1 3](#_Toc138326040)

[1.2 Contexte pédagogique 3](#_Toc138326041)

[1.3 Résumé et contexte du projet 3](#_Toc138326042)

[1.4 Objectif : 3](#_Toc138326043)

[2 Spécifications du projet : 3](#_Toc138326044)

[2.1 Spécification des besoins fonctionnels : 3](#_Toc138326045)

[2.2 Spécification technique 4](#_Toc138326046)

[2.3 Architecture 4](#_Toc138326047)

[2.4 Environnement de développement 4](#_Toc138326048)

[2.5 Technologies : 4](#_Toc138326049)

[3 Mise en place de l’environnement technique 4](#_Toc138326050)

[3.1 Module et fonctionnalités 4](#_Toc138326051)

[3.1.1 Utilisation du module Fortify 5](#_Toc138326052)

[3.1.2 5](#_Toc138326053)

[3.1.3 Fonctionnalités : 1](#_Toc138326054)

[4 Réalisation 4](#_Toc138326055)

[4.1 La base de données 4](#_Toc138326056)

[4.1.1 Création des tables 4](#_Toc138326057)

[4.1.2 Modification des tables 4](#_Toc138326058)

[4.2 Partie front-end 4](#_Toc138326059)

[4.2.1 Le maquettage 5](#_Toc138326060)

[4.2.2 Balisage des pages 6](#_Toc138326061)

[4.2.3 Le langage de Template blade 7](#_Toc138326062)

[4.2.4 Le style 8](#_Toc138326063)

[Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement 9](#_Toc138326064)

[Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement 9](#_Toc138326065)

[Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement 10](#_Toc138326066)

[5 Partie back-end: 11](#_Toc138326067)

[5.1 Architecture MVC: 11](#_Toc138326068)

[5.1.1 Schéma du MVC 11](#_Toc138326069)

[5.1.2 Le concept du MVC 11](#_Toc138326070)

[5.1.3 Exemple pour la récupération de tous les tickets de l’utilisateur 12](#_Toc138326071)

[5.2 Eléments de sécurité 5](#_Toc138326072)

[5.2.1 Requêtes https 5](#_Toc138326073)

[5.2.2 Jeton CSRF 5](#_Toc138326074)

[5.2.3 Classe DB :: 5](#_Toc138326075)

[5.2.4 Version à jour du Framework utilisé : 5](#_Toc138326076)

[5.2.5 Middleware\* : 5](#_Toc138326077)

[6 Conclusion 6](#_Toc138326078)

[6.1 Bilan du projet 6](#_Toc138326079)

[6.1.1 Problématiques rencontrées : 6](#_Toc138326080)

[6.1.2 Solutions opposées : 6](#_Toc138326081)

[6.2 Apports personnels et professionnels 7](#_Toc138326082)

[6.3 Perspectives d’améliorations 7](#_Toc138326083)

[7 Annexes 9](#_Toc138326084)

[7.1.1 Quelques requêtes SQL 9](#_Toc138326085)

[7.2 Schéma de la base de données 11](#_Toc138326086)

# Introduction

## 

## Contexte pédagogique

Ce projet intervient au cours de ma formation de Concepteur Développeur d’Application. Cette dernière se réfère à un référentiel officiel. Il requière la validation de compétences. Celle-ci sont réparties dans trois activités respectivement ‘développer une application’, ‘concevoir et développer une application’ et ‘préparer le déploiement d’une application’.

Ces trois activités en plus de la période de stage en milieu professionnel servent à mettre en application et démonter les compétences acquises au cours d’un ‘’projet fil rouge ‘’. Cette simulation de projet reflétant un projet du métier est partitionné en deux jalons. Nous sommes ici dans le cadre du premier qui doit servir à démonter les compétences suivantes :

* Gérer son environnement de travail
* Développer une interface utilisateur
* Développer les composants d’une application.

## Résumé et contexte du projet

Dans le cadre d’une mise en place de leur solution d’assistance aux utilisateurs l’association AMIO demande le développement d’une application qui permettra à ses utilisateurs de faire une demande d’assistance sous forme de ticket par le biais de leur compte utilisateur en ligne. Ses demandes sous forme de ticket seront prises en charge par des responsables de dépannage par l’intermédiaire cette fois-ci d’une application desktop liée à la première. Les mises à jour de l’avancée de chaque ticket fait par les dépanneurs seront visibles par le réclamant côté application web via leur compte utilisateur. La communication entre les différents protagonistes se fera à l’aide d’un chat.

## Objectif :

Pour ce premier jalon l’objectif est de fournir au client une première application web afin de permettre aux utilisateurs de créer un compte et de déclarer leur(s) demande(s) d’assistance sur la plateforme demandée par la maîtrise d’ouvrage.

# Spécifications du projet :

## Spécification des besoins fonctionnels :

* Pour cette application de ce premier jalon, celle-ci devra offrir la possibilité aux utilisateurs de créer et s’authentifier sur leur compte.
* Depuis celui-ci ils devront pouvoir déclarer une demande d’assistance auprès du service technique.
* Ces demandes d’assistance feront l’objet d’un suivi par la possibilité d’un échange par messages entre les différents protagonistes de la demande d’assistance en question.

## Spécification technique

* Langage de programmation : PHP 10.10.1
* Framework : Laravel 10.10.1
* SGBD : Maria DB

## Architecture

Pour ce premier jalon l’application web sera développée en respectant d’architecture Model Vue Contrôleur proposé par le Framework Laravel utilisé ici.

## Environnement de développement

le développement de l’application s’effectue directement sur le serveur mis à disposition par le client. J’ai écrit le code avec l’éditeur Visual studio. J’ai versionné ce même code avec l’outil git et utilisé également sa solution de serveur distant GitHub pour pérennisé le contenu de mon avancée dans ce projet.

## Technologies :

* La base de données est gérée par le SGBD Maria DB. J’ai aussi utilisé e logiciel Dbeaver pour lancer mes scripts SQL et visionner le schéma ainsi que l’état de ma base de données, celui-ci est renseigné dans l’annexe de ce dossier. Le logiciel Dbeaver m’a également permis de vérifier le bon fonctionnement de mes différentes interactions avec la base de données et donc de valider le bon fonctionnement des fonctions du modèle sollicité.
* Le front-end est développé avec HTML5, CSS3 et animé en JavaScript vanilla.
* La partie back-end est traité en PHP avec le Framework Laravel 10.8.

# Mise en place de l’environnement technique

## Module et fonctionnalités

Concernant la fonctionnalité d’authentification et de création de compte utilisateur elle est gérée côté serveur par La bibliothèque Fortify installé par l’intermédiaire du gestionnaire de paquets Composer.

J’ai adapté l’affichage des pages d’enregistrement et de connexion avec les formulaires que j’avais déjà écrit en les intégrant dans les vues créées à cet effet par l’outil fortify.

Une image contenant texte, capture d’écran, Police, logo

Description générée automatiquement

### Utilisation du module Fortify

Après avoir installé Fortify grâce au gestionnaire de paquet composer et lancé la commande :

php artisan :migrate,

A ce moment j’ai pu m’apercevoir que l’outil avait créé une table users, je la modifierai par la suite avec mon script de modification de mes tables afin d’y ajouter trois colonnes nécessaires à mon application (adress, tel et role). Fortify m’aura aussi créé un modèle et un contrôleur pour les users.

Une image contenant texte, Police, Bleu électrique, capture d’écran

Description générée automatiquement

Figure modification de la table users

### 

### Fonctionnalités :

#### Création de ticket

#### La route

Cette route correspondant à l’adresse/new de l’application **appelle** la fonction ‘store du contrôleur de la classe de la classe’ticketController’

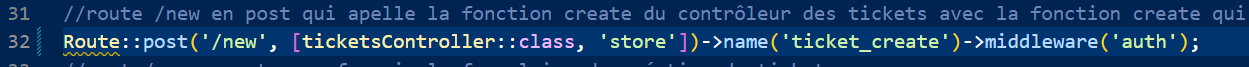


Figure route/new

#### Le ticketContrôleur

La fonction *store* du ticketsController détaillée dans la capture ci-dessous.

Elle prend comme paramètre un objet de la classe request. Cette classe permet de récupérer dans cette fonction les éléments envoyés dans la requête http. Cela m’a permis de pouvoir stocker les informations données dans l’input ‘sujet’ du formulaire. J’ai aussi pu récupérer l’identifiant de l’utilisateur actuellement authentifié visible ligne 76, la date actuelle elle est récupérée grâce au module Carbon qui est une bibliothèque de manipulation de dates. J’ai fait appel à la fonction du modèle qui permet de créer un nouveau ticket en base de données en créant une nouvelle instance de la classe ticketModel afin de faire appel à l’une de ses fonctions, ‘store en occurrence détaillé dans la capture d’écran du modèle un peu plus bas.

L’objet Request passé en paramètre de la fonction représente la requête HTTP entrante. Il m’a permit de pouvoir récupérer la valeur de l’input ‘sujet’ du formulaire et l’identifiant de l’utilisateur authentifié. Ces éléments stockés dans des variables seront passées au modèle qui les insèrera dans la base de données.

Appel de la fonction avec les paramètres lui étant nécessaires

**Le modèle :**Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

La flèche du dessus entre le contrôleur et le modèle montre que la fonction du contrôleur créé une nouvelle instance du modèle puis utilise une de ses fonctions (‘*store’)* du modèle.Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

La fonction attribue à chacune des colonnes de la table ticket des valeurs avec celles des arguments qui lui sont fournis.

#### Le postage de messages dans une conversation appartenant au message

Figure

1. Le formulaire : dans la page affichant le détail du ticket demandé, la conversation lui étant rattachée est visible. En bas de cette conversation l’utilisateur peut trouver un formulaire dans lequel se trouve un input de type tex qui lui servira à écrire son message le bouton type submit utilisé pour l’envoi déclenche le processus pour le postage du message dans la Dans la page correspondante au détail du ticket voulu l’utilisateur à accès à la conversation appartenant à ce ticket, il à la possibilité de rédiger un message et

* Affichage

*Une image contenant texte, capture d’écran, Police

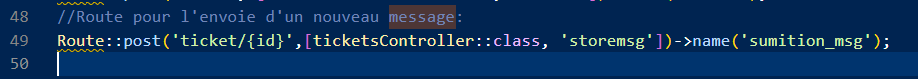
Description générée automatiquement*

* HTML

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

1. La route post : elle est appelée lors de la soumission du formulaire en méthode POST pour l’envoi de message visible au-dessous. Cette route fait appel à la fonction ‘storemsg’ de la classe ticketController dans son chemin aparait l’identifiant du ticket qui servira à lier le message à ce ticket



Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Cette fonction storemsg du Model sollicité par le controller effectue l’insertion du message en base de données. Elle effectue cette tâche en deux temps ce qui justifie le fait que j’ai effectué ses deux étapes dans une transaction.

* La première étape effectue une insertion dans la table message en remplissant les colonnes de celle-ci avec les éléments passés en arguments à la fonction. Ces valeurs sont passées à la requête après sa préparation par un tableau comme le permet la classe DB :: fournie par Laravel. Cette pratique permet de diminuer le risque d’attaques par injection SQL.
* La deuxième étape de cette transaction

# Réalisation

## La base de données

Le schéma de ma base est disponible dans l’annexe de ce dossier

La base de données sera gérée avec le SGBDR\* open source (\*Système de Gestion de base de Données relationnelle) Maria DB. Aussi, pour avoir la possibilité d’avoir un accès graphique sur ma base de données et lancer mes différents scripts j’ai utilisé le logiciel Dbeaver. J’ai scindé la création de ma base de données en deux étapes. En Me référant au Modèle Physique de Données j’ai écrit un premier script pour la création des différentes tables sans les contraintes de clefs primaires et clefs étrangères. Cela m’a permis de ne pas avoir à faire aux problèmes de l’ordre dans lequel je créais mes tables. Par conséquent j’ai écris un deuxième script dédié à l’ajout des contraintes de clefs primaires et de clefs étrangères ainsi qu’à l’ajout des colonnes nécessaires dans la table users créée par Fortify comme nous l’avons déjà vu précédemment.

### Création des tables

Ici l’extrait de mon script de création des tables montrant les lignes SQL qui créées la table type\_panne avec ses des colonnes (la clef primaire l’entier :id et la varchar : label).

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

Ce type d’instructions seront adaptées pour la création des différentes tables de la base nécessaire au projet.

### Modification des tables

Dans l’exemple suivant extrait de mon script de modification des tables de la base créées dans un premier temps. On peut voir dans l’extrait suivant, l’ajout de la contrainte de la clef étrangère id\_auteur qui fait référence à la colonne id de la table users.

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

* Ajout des colonnes nécessaire à la table users créé par Fortify.

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

## Partie front-end

### Le maquettage

J’ai effectué le maquettage des différentes pages après en avoir établie la liste selon les demandes nécessaires à l’application attendue par le cahier des charges. J’ai effectué ce travail de structuration graphique et sémantique grâce au logiciel Canvas. J’ai pris le parti de faire le maquettage en mobile first car cela me paraissait plus facile de passer de l’affichage mobile à celui du format desktop que l’inverse. D’autant plus que je tenais que l’application soit le plus possible responsive. Cela ne changeait pas l’organisation sémentique de mes pages

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

### Balisage des pages

J’ai effectué le balisage en HTML5 en me référant aux maquettes effectuées précédemment et les différents éléments écrit en html seront placés pour beaucoup dans des composants blade.php qui seront inclus dans des templates où extension de Template. J’ai inclus ses composants par la suite selon l’utilisation des pages.

Ici le détail du composant blade du tableau affichant tous les tickets de l’utilisateur :

Le thead contient une ligne dans laquelle apparaissent les titres des colonnes. Dans le tbody on peut voir l’instruction @forelse qui est l’équivalent du en blade de la boucle foreach de PHP. Dans cet exemple la variable $tickets(tous les tickets ressortis de la requête SQL menée par le modèle en base de données) et passée à la vue par le contrôleur avec la fonction view() Cette variable $tickets est décomposée en variable $ticket (correspondant à chacun des éléments de $tickets)chaque élément sera affiché sur une ligne du tableau et chacun de ses attributs dans une colonne dédié.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

### Le langage de Template blade

Avec l’apprentissage du langage de Template blade et la compréhension de son fonctionnement je me suis rendu compte que mon design de nécessiterait l’extension de deux templates (home et sign), respectivement celui de la page d’accueil et celui inhérent aux pages de connexion (sgnin et signup.

. Chacun comprenant un yield(‘content’) correspondant à la zone de la page qui s’adaptera au contenu souhaité.

Le langage blade est un langage de Template conçu pour être utilisé avec le Framework PHP Laravel que j’utilise pour cette application. Il permet de créer des vues en HTML. Il permet aussi de définir des sections interchangeables ainsi que la possibilité de définir des composants réutilisables. Il est aussi possible avec blade de définir des conditions et des boucles

Ci-dessous un exemple dans lequel le tempate [home.blade.php](http://home.blade.php) est étendu. La section interchangeable de ce Template est dans cet exemple remplie avec un premier élément (detail\_table) et un deuxième sous condition que l’utilisateur qui consulte la page est un rôle d’administrateur, auquel cas lui et le troisième composants (maj\_sttatus et msg\_display) seront affichés dans la vue.

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

### Le style

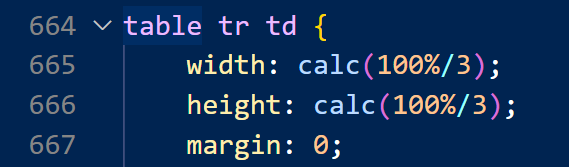
J’ai écrit la stylisation de mes pages en deux temps.  
En effet ayant déjà effectué l’écriture du style pages par pages avec un css dédié. J’ai choisi de me diriger vers la solution du pré processeur Sass pour intégrer mon css une fois l’apprentissage de laravel commencé car en transformant le fichier app.css de Laravel en app.scss j’ai pu bénéficier du nesting\* pour intégrer le css de chaque page plus facilement).

\*Exemple de nesting scss :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Sass m’a aussi permis d’utiliser sa possibilité d’effectuer des opérations mathématiques pour définir les largeurs des cases de mes tableaux (voir ci-dessous :



En ajoutant un id #home au body de mon Template [home.blade.php](http://home.blade.php) et un id #sign à celui sign (servant aux pages de connexion et d’enregistrement il me fallait mettre le css de ses pages entre les accolades des sélecteurs respectifs respectivement body #home et body #sign. Lors de l’ajout des sections dans les yield dédiés le style est appliqué aux différents composants qui se trouvent être maintenant les enfants de ces body.

#### Le responsive

Afin de rendre l’affichage de mes pages un maximum responsive, j’ai utilisé la média querie secreen avec un break point à 640 pixels.

Une image contenant texte, Police, logiciel, Bleu électrique

Description générée automatiquement

Une image contenant texte, capture d’écran

Description générée automatiquement

Au-dessus on peut observer une partie de mon Template home avec sa barre de navigation faisant partie du header. En dessous se trouve l’inclusion de mon composant blade aside ligne 48 puis le main contenant mon yield (‘content’ (zone interchangeable en fonction des pages)) et enfin l’inclusion du footer lui aussi sous forme de composant blade.

#### Affichage de la page d’acceuil :

La page d’accueil est le template ‘home’ sans aucun remplissage de la section ‘content’(correspondant au <main> de la page).Ce template recoit un composant head.blade.php dand lequel est faite la liaison avec le fichier de style sera étendu pour d’autres page de l’application seule la section ‘content’ sera rempli avec le composant necessaire.La page d’acceuil est le template

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

# Partie back-end:

## Architecture MVC:

* Les routes : pour l’affichage de chaque page correspond une route en méthode GET où POST qui appelle la fonction du contrôleur nommé dans cette route. La fonction retourne la vue correspondante. Si cette vue nécessite l’affichage de données, le contrôleur sollicite une fonction du modèle utilisé. Cette fonction du modèle effectue alors une requête en base de données et retourne le résultat obtenu. Ces données feront l’objet d’un traitement avant d’être retournées où le seront directement.

### Schéma du MVC

Une image contenant texte, capture d’écran, diagramme, conception

Description générée automatiquement

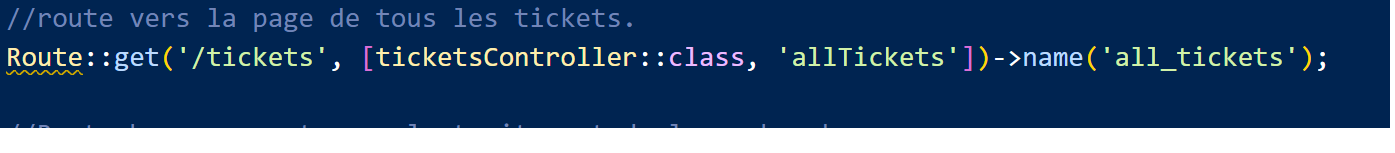
### Le concept du MVC

* Modèle (Model) : le modèle est chargé des interactions avec la base de données. Les requêtes SQL y sont écrites dans des fonctions qui retournent le résultat de leur requête. Le modèle est représenté par une classe.
* Vue(view) : La vue est responsable de la représentation graphique des données récupérées en base de données et transmises par le contrôleur.
* Contrôleur (Controller). Le contrôleur est responsable de la gestion des interactions entre le modèle et la vue. Il traite les requêtes de l’utilisateur. Lui aussi est représenté par une classe. Il sollicite le modèle en lui fournissant des paramètres pour mener des requêtes en base de données et lui retourne les informations qu’il lui faut pour les transmettre à la vue qui l’a elle-même sollicité .Si besoin c’est lui qui est chargé de les manipuler.
* La Vue(View) : Elle est en charge de l’affichage des données qui ont été récupérées en base de données par le modèle et transmises par le contrôleur.

### Exemple pour la récupération de tous les tickets de l’utilisateur

En suivant le processus d’appel dans le MVC :

La route : Dans l’exemple suivant on peut voir la route ‘/tickets’ en GET qui fait appel à la fonction ‘allTickets’ du ticketsController.



1. Le contrôleur :

Voici ici le détail de la fonction du contrôleur appelé dans la route au-dessus :

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Cette fonction créer une nouvelle instance de la classe ticket Model dans laquelle existe une fonction qui sera appelée en tant qu’attribut de l’instance $ticketModel.

1. Le modèle effectue la requête en base de données :

Le modèle : La fonction du modèle getallTickets est détallée ci-dessous s’occupe de faire la requête SQL à la base de données lors de son appel par le contrôleur et lui retourne le résultat afin qu’il la transmette à la vue qui en a besoin pour son affichage:

*Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement*

La requête précédente retourne tous les éléments de la table ticket. On remarque ici comme dans mes autres requête visible dans l’annexe que j’ai utilisé la clase DB :: mise à disposition par laravel, en effet elle sécurise les requête en base de données en offrant la possibilité de préparation des requêtes avec des valeurs de paramètre. Cela peut permettre de se prémunir des attaques par injections SQL. Ci-dessous on observe une requête qui utilise elle aussi la classe DB :: mise à disposition par laravel. J’utilise la possibilité de passer des valeurs par l’intermédiaire de paramètres qui est une pratique permettant de se prémunir des attaquespar injections SQL.

En suivant une requête effectuée par la fonction prend un paramètre qui se trouve être l’identifiant de l’utilisateur dont on recherche les tickets fermés

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. L’affichage après la première requête :

Ici on peut constater que l’utilisateur 30.05.23 qui compte trois tickets lui appartenant : tous ont le statu ‘ouvert’ Ce statu est d’ailleurs donné par défaut au ticket lors de leur création.

Une image contenant texte, capture d’écran, affichage, nombre

Description générée automatiquement

Dans le cas où aucun ticket n’est présent un message l’indiquant est affiché pour prévenir l’utilisateur grâce à une condition écrite avec blade entre les lignes 26 à 28 ici.

Une image contenant texte, capture d’écran, Police

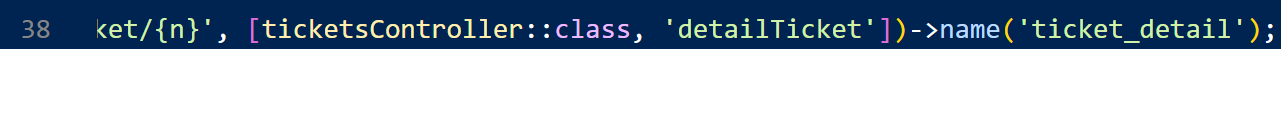
Description générée automatiquement

1. Route paramétrée :

Les routes peuvent prendre un paramètre afin de devenir plus spécifique. Ce paramètre sera l’objet d’un traitement par le contrôleur

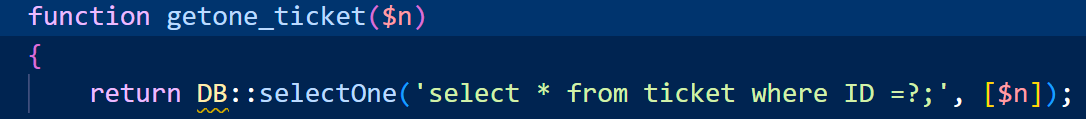
![Une image contenant texte, capture d’écran

Description générée automatiquement](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RncRXhpZgAATU0AKgAAAAgACwALAAIAAAAmAAAIngESAAMAAAABAAEAAAExAAIAAAAmAAAIxAEyAAIAAAAUAAAI6gMBAAUAAAABAAAI/gMDAAEAAAABAAAAAFEQAAEAAAABAQAAAFERAAQAAAABAAASdFESAAQAAAABAAASdIdpAAQAAAABAAAJBuocAAcAAAgMAAAAkgAAEYoc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFdpbmRvd3MgUGhvdG8gRWRpdG9yIDEwLjAuMTAwMTEuMTYzODQAV2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NAAyMDIzOjA2OjA5IDExOjI2OjQ2AAABhqAAALGPAAaQAwACAAAAFAAAEWCQBAACAAAAFAAAEXSSkQACAAAAAzAwAACSkgACAAAAAzAwAACgAQADAAAAAQABAADqHAAHAAAIDAAACVQAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIzOjA2OjA5IDExOjE2OjQyADIwMjM6MDY6MDkgMTE6MTY6NDIAAAAABgEDAAMAAAABAAYAAAEaAAUAAAABAAAR2AEbAAUAAAABAAAR4AEoAAMAAAABAAIAAAIBAAQAAAABAAAR6AICAAQAAAABAAAH7AAAAAAAAABgAAAAAQAAAGAAAAAB/9j/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCAArAQADASEAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDyKF1TeWTdlcDjOD602Vt0rNt25P3cYxXsdTzOg5yFjC+WVJA5I61KJbbaA9scjHIb2/rT5kFhs0sDooSDY+eSDx0qZntSWZbN9gJ5yf1ougI4iimV5LUtgBgOcLyOvt0H405riJzh7ZRnBARcZ6/4/pS5kA4vCYGkFkeDjfztGQcZ/L9KjTZChFxbOTk4yMdRxTugHW8kMcH7y0MhOQH7Z/yac0kC5H2RkZSd2e3II4PToR+NF0FmQq8YMn7ncz/6vI6fhQ7xGPYsBEmBk579+KXMgsx6PFINq2xZsA4UegOT/Wnyy26FkazKSZP3ieOPT680cyCxTPzPhFPJ4ApCCrFWBBHBB7VQWAAsQACSegFG1hj5Tz0460m0gHJFLIpZI3ZR1IUnFDQypjdE65OBlSOaXMgG4OcYPFOaKRRlo3ABxkqafMgsMpVVmYKoJYnAAHWi4CiOQnARs4zjHbGc/lzSYOAcHB6GlzIA2sSQAcjqMUKrMyqqksxwAByafMgH+RKP+WT8/wCyf89jU+lf8hey/wCu6f8AoQpNppgtztbf4TeOrZy8ekx5IxzcR/8AxVNm+EfjmeVpX0lNzHJxcR/41j7ene9zX2U7WsbkfgTx4sEcUmh28qxoEAeWPoFx/eqG7+H3j66tTC2jxHJ5Zp48/cKdd2eh6dK5lyqV+Y05JWtYSH4f+PrexNtFoVsGOQZDNGSQQB3Pt1qpc/DDx7dyiSXSlLbiWxdJ82TnB+b1rolWpu2pmqU10Jx8O/iEFUPpMEoHQPOhA+cN03Y6j8qlTwJ8Ro1ATSLZSCnzCWPd8rbhzuzWbdF9Q9lPsQR/Dbx9Fkpo8IbIKkzxnb8rKe/P3jUMvws8bTTb5NEjZBHsEf2lAAduAevUdapVKad7h7KfYLT4X+PbJCIdKQFlZGP2lOVbHHDewq4vgD4hLnbotmBg8eYhBywP97rkUSnSk7tlKFRdB58DfEcyb20m2ZsYGZk4G4tx83HJ7U1fAnxGQII9It02AAbZkBODxzuzUfue4+Wp2GD4ffELaVbRrZwW3DdKnHH+9/nJ9aG+H3xDNobZNGtY4yc5SSPd90L1LZ6Cnele9xctTsVV+F/jxRIP7HhIcsTmdO+M/wAX+yKtH4ffEHy1QaJaAKWIxImeWDH+L1Fa/WKfcFCouhRufhP42nlSVNFjikAJZkuU+ZixOfvcdQPwrRTwJ8Ro7eKFNJtlWNFQESoGwM992e56VE6lKW7J9lPsRy/Dzx/Ns36HafIu0YlQcbSv970P8qavw7+ISkE6RAzAYDNOhP3Qv970UVPNRta4eyn2Hz/D/wCIk9vJAdJt0STO7y5Y1zn6N+nSqafCzx1HGQNIRnOcs9xG2ckHkE89KanSSsmNU5roWbf4d/EK28oLpFswi+6GkjPcn+971LN4B+IVxF5UmiWnl9DskjU4wVxnPoTVyrU2rXEqU73sVYPhn8QILZrf+yo5I2zw9yp6rt/velWf+EB+ImyRRo9qu8OuRJHld2M4+bjpx6VDlSfUPZT7BH8PfH0V3LdpokBuJm3Ss9wjBvwLdPbmnHwJ8RWLF9HtmLFesqcbW3DHzcc+lJypN3bD2U+xS/4Vd47+0Qy/2NEvlEEKlwgBwSRn5velt/hR4z/tOznbR0jjiePdi5Q8A8nrVKrTWzGqU+x9M0VwHaFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAFFABRQAUUAf//Z/+Ex5Gh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRvclRvb2w+V2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NDwveG1wOkNyZWF0b3JUb29sPjx4bXA6Q3JlYXRlRGF0ZT4yMDIzLTA2LTA5VDExOjE2OjQyPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwADAgIDAgIDAwMDBAMDBAUIBQUEBAUKBwcGCAwKDAwLCgsLDQ4SEA0OEQ4LCxAWEBETFBUVFQwPFxgWFBgSFBUU/9sAQwEDBAQFBAUJBQUJFA0LDRQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQU/8AAEQgAwgSFAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A/Pqiiiv00+ELemaZda1qEFjYwvc3dwwSOKMZLE9h/njFdB4o+F/ijwXZpd6xpMlpauwUTCRJFBPQEoxxntnGa52xuns7yKaOWWBlP+shYq4B4OCOhwSPxr1mRdA1L4X+JLTwZdakiQvFeajb60qmSSNWABjZDsGGIJyMnA5wMHirVJ03G2zOqlGM7pnNab8DvG+sadbX1povnWtzGs0Un2uBdyMAQcFwRkEcEA1y/ibwzqPhHVn03Vbb7JexqrNF5ivgEZBypI5HvW38MfDdtrWuve6mMaJpMRvb5iOGReVjHqWbAA6kZx0rE8UeILnxZ4gv9Xu/9ddSF9o6KvRVHsAAB7CnGVT2vK3dL+u4nGHJzdTKooorsOYKKKfHG0siooLOxACjqSegFGrYzWtvDN1ceGbzXS8MVlbzJbASEhpZGBO1AAQSACTkjAx1ziqej6Xca5qlrp9onmXNzKsUS+rEgDPtz+Vdf8TJE0O30jwjARjSovMvNpyHu5AGkz67RtUemDS/D0f8I1oeteLpBiW1j+w6dnvdSggsPUom5vxFGl2+hVnscdrGmto+rXlg80Ny9tK0TS25JRipIJUkAkZBwcCqdL7nr3NJ1pITFpPWvcvGPijwz4E8XQacPBml3dtLBbyahLNAHfa0a5EC5CxkDJJ6liSSOtYdnNoXg3wa/iaz0C11mTUtTmtrSPWIjLFbW64IDIDguQRznscHg5enyHbY8q/h96MfLmvaI9J0bXvEXw416LRbXT11q6eK906KMG2YxyKu5UOQFYHkdOO5yTzfg3R7a61zxwk1jFLFa6ZfPGskQZYWU4UgEfKR2IwR2pa9fP8AALX/AA/E4u10G/vNIvtThg32FiUW4m3ABC5wgwTk5PoD71n+1e2+DfH/ANn+EniKX/hHPD8n9nGyh8uSxyt1liu6YbvnYYyDxgkmsLwBa2HxM8f3N5qWnaRptpaWL3RsrcG0tGMYAG8gkquSCxGeAeO1O3vW/rYdlZHl1GK9v8e6Tol94J1O4uZvA9vq9qUeyHhe6AaUFgHR4zjdwcgjJBHYZz4hSvqKUbBRmvSPhi2kWPhXxfqmq6Nb6z9hS1aGKcAYZpCB82MgZxkDqBg8GtmXTdN+Jfh/w5qR0ex8PXc2uJpMx0uHyYpY2UNuCZI3DpnJ689gK5eguh4+aCOle36XrGga54/k8Dv4K0qDSWuJLGO5jiIvkK5AkM2ck5XJBHQ4JOOeL1DRbaz+E7zCGF7uLxE9qbtUG8xiDIXdjO0kZxnGeanpcrl6HCe1Bx+Nex6b4f0//hLfhdBLptt5d5Zq9zG8C4nJkk5cEfMSAOTngCsvVfG/hYR67oT+ELO0skilisry3jDXq3AJCO8hI+UkEkAcDgAih6euv4Bb9PxPL6KK7P4U6NNrHig+Vpdjqq28DzSLqcpjtYQMDzJCOqgkZHfPtTSuQcaOldV4R+Hd/wCLrG6vlvdO0nTrdxE99qtx5ERcjIUHBJOOeleifEvQ9Nk+H8esPbeFXv7bUUgL+E5CIXiZSSkgA4bIBB5IHTGTl7+LPBw+FKXP/CDf8S06yYxp/wDa83+t8gHzfMxnpxtxjv1pdH/XY05dUmeVN4Qu5PFi+H7Ge01i7eVYo5tPmEkMhIByrkAEDPJIGMH0rQ8XfD2TwfZrNLr2hajJ5vkva6be+dNGcEksu0YAIwTngkCu2+DPiHwzJ4+toYfCnk3k91NJa3X9pSt9ljMZIj2kYfABG488+1cP4q8ReHNSNsdF8KroUsUpeVnv5LoTDspVgABkc465o2S8w01ZU1Hw5Y2fhDTNXi1u1ur66leOXS0H723AJwzHOcHA6gdRgnnGB716X461K28R/DXRdZXQ9I0e7k1Ge3f+yrQQKyqikZ5JPJPU49qPDZ0zwV8O4fEs2iWevanf3r2kK6ihktrdUUEkpkZYk8ZI4HBGDk6y/rsTa9rHF+FfDN14v1yDSrN4YricOVackJhULHJAJ6KccdcVkevrXqvwx1218RfGTSb6PSbXSEaKYS21gCsRYQSZZFJ+XIxwDjIz3px1fRPG/gnxUkPhbTdHGkwxXFjcWqEXABkCkSyEkyEg8k4+mcENrYdjyelNfRGm+FdJ8MaLotuieBZftVnFdXkniW6IupC4yfL4IjUA4BGckZxkc8z4f8P+F7P4heNoEht9c8P2elT3UCxyrKMARthJOcEZKhgcjnnOaVves/P8A5dE/wCtTxw0V3fiTxxoHibwxNbHwxZaNqsM6tZTaXAEUw4IZJmLZZsYwcHJ9Oc8Kfu0EtWOm8E+BZfHFxNbwavpOmToyJHHqd0YmmZiQBGApLHIwQPUetYmqafLo+qXlhMVaa1meB2jJKllYqSCQDjIOMgVq/Dz/kfvDn/YRt//AEYtb+n+HbfxZ8brnSrpittcavcCXacEqHdiAexIGM+9O2sfP/gB0bPP6K+hta0Pw7qOlavaXsnw/sbWO3kaxk0a+xepIoJQOxA8zOMEHGSehOK+eam+thyjyijuK2dP8K3epeGdW1qKSEWmmvCkyOxDsZGKrtABBwRzkj2zW38K9D0/Vda1G61O2+22mladPqDWm4qJzGBhSRyBkgn6eldjJ4t/4Sz4OeL5BoOk6IkNxZqDpVr5IkzJnDnJ3EYH5n1qmtPP/gjir/15HmnhfQrXW7yYX+rW+jWMERlluJsM5x0WOPIMjE44B6ZJIxVfSdBuvEWtJpukRPezys3kqwWNnABOSCxAOASRk49TXvVn4T0fwvpukWar4EljntIp72TxFdEXshkUEmM4Plrg4UjPTPWsr4b6lpvhn4wz+HtHs9H1PTJbmaW31IxieeNPILBI5gcbRjaeuecnmiyvb1/ALaXPCCNpIPrg0ldb4o8aSeMxb2UfhzQ9LkWbKtpFl5MkhPAUnccjJ6ete26L4Os7y8TQdX0bwPYQPAytY29wX1eI+WSDvxksCATjtk5OOVrythy3lyo8B8HaFaeJNfttPvtWt9EtZQxa9uQCiYUkA5IHJAAyQOfwqjBpq3OsJYJd26o84gW7mYpCAWwHJI4XuTjgV3XwX1i3j8V2Wh3ehaPqttqFyEkm1C0E0sYAIwhJwAfcGuA1BVj1C6VQFVZWAVRgAAnAAp9ULozoPGngC68Ew6dPNqWm6pb6gJGgn0uczIQhAbLFQOpxxnoc4rlv4a73xl/yTL4f/wC5ff8Ao8Vwf8NTs7AwFbFr4Wu7rwre6+kkIs7S4jtpI2Y+YWcEggYxjjnJB9q6D4W6Pp13da1qmp2g1G30fT3vVsmJCzOCAoYj+EE5I9ucjIrqdU8VHxX8FdZnGiaXoqx6nbxhdKtfIST5ScsMnJGRz6GnLSN/63sOKu/67HnmneDL3VPCeq+IYZrdbPTXjjmjdmEp3kAFQAQRk85I6GsH0rStfEeo2eh32kQ3G3Tr1ke4h2Kd5Q5U5IyMH0Iz3zXR+F9Ptrj4aeNbqS2ilubdrHypmjBeINKwbaSMjIwDgjPei27Qjil+9mjjdXp2l6LZt4W+H00lhCZbvWpI5ZHhGZow8YCsSPmXkjByOTXU/b/DmueMPEnhKDwhpVpp8MN55d2sX+lLNGGbcJM8LlThQMAYHTihqyv/AFtcIrmPB6ckZkYKoyScAe9NroPBfiL/AIRnWkuf7M03VfMXyvK1O386NcsDuAyMMMYB9CfWqjq0SZ2taJe+G9Um07UYPs17AQJIiwYqSAw5BI6EHg960fDXhyx1yx1m4u9bttKksbXz4IZwC10wz+7TJHPA6AnkcYyR6L8cvHezxZ4g0T/hHtBP3E/tFrL/AEzmNGz5m7qM4Bx0AFZPw/1a21fwj4j0a50HRpDY6PdXUWoGzBu94IwTISem84wARgc8VEdVc0cfeseYrwwNJXc/CvRdN1C81nUtVtDqNto+nvfCxDFROwIADEc7QTk/TnIyDR8W+N7bxZZRRDw1o+jXEUu5bjSoDDuQggq65IY5wc8EYPHJp9u7I8zO8V+F7vwdq/8AZ17JDLMIo5d1uxK4dQwGSAc4Izx19axvevoT4haTYaLdal4uls4fEF3Bb2lrHYSKHhtC0CgTXCHlgeijBBPU56c78K/DtjdeFtT8SXI8OtqLXotYV8ROIbKPK72IRRgsc4C4AABIxjBq2r8h2va3U8dFaGsaDfaBJarf2/2drmBbqEFgd0bZ2twTjODwcGvTPiE2i+GdQ8P67YQeE9R1NvOS+03TZBdWBIACP5eQVyGPHAyoIzyTq/Fj4jf2fJpFt/wjHhu4+16Jby+dcWG6SHeGG2M7htVf4Rzj3qOl+pXLrZnhtFdB4P8AEVj4buL2S+0O11xbi1eCOO6OBEzYxIODyMdsHngjrWBTMxKKKKYgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA/XT/gkP/wAm3eJf+xsuf/SOzoo/4JD/APJt3iX/ALGy5/8ASOzor4DGf7xP1PrcN/Bj6H5F0UUV9+fJGj4f1j+wdWgv/sVnqXkkn7LqEXmwvkEfMuRnGcjnggHtXReIvihqXiLSX0uHT9K0LT5WDT2+j2ggWcg5G85JODyBnGe3ArjM0cisZU4TakzSM5RVkbcPiy8tfCdz4fhigitLm4W4uJlU+dLtGFRjnG0HkAAc85rF5xijd1o+7VKKjey3Fd6ISiiitCArf8D6xp/h7xJa6nqUElzFaZmigjUHfMATGGyRhQ2CSMnA6GsCijUZYvr2fUr64vLlzJcXEjSyO38TMSSfxJNb3iXxNa3/AIe0HRdOSaKzsImkm81QpluXOXbAJyAAoBPOAeBmua68d6F61PSw79RKKKKok2fFnii78YawdRvY4Yp/Kji226kLhFCg4JJzgDPPX0rV8JfErUvCOnXGnLaafq2mTuJXsdVtvPhDgAbgMjBwB37DiuS9KKWpR1OtfEfWdc8QWOrySQ2s2n7fsUFrEEhtwpyAicjGeuck9DwAK3dQ+OmuX9pqFtHp2jWEWoRSRXYs7QxmcuCDI53ElgCcHOOTkHivOcUlHSw7u9zqPBnxA1DwT9tjtbeyv7S9QJcWeoQ+bDIASQSuRyMnv3pw+Il/a+LE8QabZ6fotyqiP7Np9sI7crjBBQkggjrz7jBArlaM5o13Ffodn4k+JTeJNJmsm8MeG9NMhUm60/T/ACpwQQeH3HGcYPHIJrjKKKNhN3PR/hj4rt/CvhTxjNJFpt5cyR2ohsNTUPHcfvCGHlkgtgHPHTAJrD8T/EjVvFE1huW10u1sG32lnpcIghhbIJZQCTnIHJJx2xk1ynQ0p5FO7bQ7u1j0a6+O2v3EMzpY6Pa6rNF5Uus29kFvXBGDl84BIHUAY7YwKx/CPxK1Dwfplzp6WOm6rYXEonNrqlt50ayAYDqMjBxjrnoK5Cijq7D5mdtffFzXNU8TaNr10lpNf6UNsJ8oqjjczDeAR03YGMcAfU8heXT3t1cTuFEk0jSNtGACSSce3NQUVIrsK6Dwb40v/A+oTXVlHb3CXELW89reReZDNGcZVlyMjgdx+RIrn6Kr0Edr4m+K2p+JtBfRW07StO03z1nS30618lY2UEcAE8HOTnJ4GCKwW8SXR8Jp4f2Q/Y1vTfeZg+ZvKbMZzjbjtjOe/asn2pOtT38yrvQ1/C/iS58I69a6tZxxSXNuWKrMCUOVKnIBB6E9CKoWV19jvoLnyop/LkWTyp13Rvgg7WHcHGCM8g1XoquxJ6VdfG+6utM/s9vCXhMWgLMkS6aQsbMMFlG/AbAHIGeB6Vh+EfiVqfg/T7nT47aw1XTLhhI9hqlv50O8YG8DIwcAd8cDiuRopbO4+ZnWr8StRj8YQeIorPT7e4t4jDDaQ25jt40KsuAikHo5PXOT6cVk6P4lutF0zV7GBIXi1SFYJmkBLKqsGBUggA5A6g8dqyKKAvc7zSfi9qOn6Ta6de6PoevwWq+XbtrFiJ3iTsqnI4HbOfrjFZNp46u9P1TXL21sdPtv7XtZbSW3hhKQxRyYz5ahhtIwMZJHXg1zPal3c0ut+o7sSiiiqIOr8F/EKXwTvaDRNG1KcyrLHcalaGWWFl6bGDArzzx3Gak8RfEq+17WLHVYdN0rRNQtJWnW40m18lpJCQd0hJO4gjPPqc5zXIUUuz7FX0sd9rHxeuNcsruK78L+GHubqNkkv103FxuYEGQPu4bnIOOvauBooo8wcmzb8JeLL/wXrMep6eY/PVSjRzLuSRSMFWHGQR7g+lb2vfF7V9e0G90Q2WmWGlXJjP2WwtfJSEq24lAD1Y4yTnoMYrhqKOmo1dbHeaX8YNRs9JtLC+0bQtfjtEEVvJq9iJpIox0QNkcDtnJ96wbHxheaX4sHiGwittOu1nM0cNrFthjzkFVTJwuCRjPQ9awaKL68wr6WOu8UfEa48TtbSf2JoukXME/2gXWlWfkyu3XLsWORnn6810SftBa7DfC+i0jQItQbie9jsMTXAxgh3DZOeDxjoO3FeX0Uug+Z3udl4J+Jlz4GjP2XQ9EvrnzjOl5f2hknjJAGEcMCAMZAHcn1qHVfiAdW8QWGqt4e0G2a1JJtbex2wTkkkmVdx3dfUVyfSin1TFd2seiaz8ZrrW9FOmS+FvDEVusUkUBh08hrcP8AeaPLkKc85A6gE5rzujNFANtm74Q8YX/gnVv7QsPJdmjaGWC4TfHNG3VXXIyDgdCOlbPiT4sav4l0GfRZLTTbLS3kjkS1sbfyUgK5OEAPAJJJzkk4wQOK4n0pc9aHqgTtsJXS+DPHl/4Je9FtbWd/a3sYjuLLUIfNhkAOQSuRkgk4571zVFMR2+u/FzWfEE2kPcwWKLpd19ptIreAxon3cR4BxsGwYA55OSeMZlj46v8AT/FOoa9HFbteXvn+YjKxjHnBg2AGBGNxxknHGc1zlJ2pb6FXYUqsVYEHB60lFPVbEnd6/wDFy/8AE2ky2mpaLoV1dyRCJtUax/0zAwARJu4OBjIH4VL4b+MN14X0pbG18NeG5h5Bt5bi4sC006E5IkYONwPGQRg4HFcB3pKRV2dVZfEbUtK8XSeINMtrHSp5FCNZ2duEtSmACnlkng4yec5OQQaPFnxCufFlnFaHSNH0i3SXzmTSbMQeY+CAznJJIBOOcc1ytFHZC5jtB8V9aXxS2uiO1EstutpPaGNjbzxKoXa6liSCAM89emKg8M/Ee/8AC0moJa2OnT6bfMHl0q8gM1rkEkYViW4zgfMTjGc4Fcl6UvrT1C50Hi3xh/wlklq39iaPovkBhjSLTyBJnHLjJyRjj0yfWtuP4vag2gW+lX2jaHq629ubWC61Gy82eKPBACtuGMdjjsM5rg6KXS3QOZ3udB4P8aX3gm4vp7GK1le7tXtJBdReYArYyQMjnjvkHuDXPijvRQIKKKKYgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigYUUUUCCiiigAooooAKKKKACiiigAooooGFFFFAgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD9dP8AgkP/AMm3eJf+xsuf/SOzoo/4JD/8m3eJf+xsuf8A0js6K+Axn+8T9T63DfwY+h+RdFFFffnyRoaHb2Fzq1tFqlzJZ6eW/fzxR73VQCeB3JwAPTOT0rtdd8IeE73wTe+IPDN3rEQsbiOCWHWI4wJt5wPLZMDIwSQSTgdBxnj/AA1omo+ItctbDSIPtGoSsTDHvVclQWPLEDgAnk9q9j8aTeI/+ED1eL4lGzN7hTo65g+1eaWG4qIuNm3qT7d8V52IlKM48rt/XbqdlGKlfmX9foczpPw/8L2HhnQ9R15/EF5PrGTCNDhjaKIhioRiwJLnBOBz1GOMnj/iB4Xi8G+L9Q0eG6+2w2zgLKRhsFQ2GA6MM4PuDwOle1eCR4n0jwT4c/4V7BZ6jbSg3GrtcXCFhOeDGysw2KBj7oycA57nyX4t2ukWfj7VI9FdHtNyswjfeiSkAyKrdwGJ+nI7VjRqTlXab01/P8P1NZ04qimlrocbRRRXrnnB6V7L+yj8HtB+OHxZt/DfiO51C10prO5uZJNLkjSf93EXABdGXBxzkfiK8b7Gvp7/AIJ2y+R+0Vay7Vk2aXftskGVbEDHBHcH0ry8yqTpYOrOm7NJm1FKVWEXs5JfiQ+Lvgb8KPFfwH8R/Eb4Ua14sEnhq6gi1LS/FsdvvZJXCK0bQAKDk56tkAggHGeK8A/sf/GD4neF4fEXhzwVc3mjTgmG5mure280D+JFmkVmU9iAQexNe5+LPHU/7S37IOt/8Irp2m+C73whepf6/wCEfDNnHa2N/atyt0qAbhsIJKliP3ZJBO0jrvir8G/iB+0Vqnww8bfCfXLT/hC7PRbO2gni1RbYaDPEpEpePIKkcA7AW4wQAAT8vSzDE4eLhUkk+Zq8tbKyava179D2ZUKVRpxV9Oml9f06nxdZ/BzxnfR+LnTQZ4T4SiE2txXLpBJZKWKgsjsGY5B4UE98YNM8PfCTxb4r0LS9a0rSvtWm6nrkfhu0n+0RJ5moSKHWHazhhlWB3EBRnlga+5/7U0n45fGr9pfwz4Q1KxvdV8QeHbWz05vNVI9QuLeNVlKOSAfmAGc4xz0Ga8m+NXwZ1X4GfsV6R4f1+8sZtdm8drfXtlYzLN9gZ9PdVhkZSQX2orHHGHGCRgnso5xUnKNOokpy5bKz6rX7jOpgoRu4O6V/wF8af8E4fiNY+BfBV14c8OXuo+KLuK5bxDY3Gq2CxWTq6iBYyXUEMpYkhn5A6dD4V4A/Zx+JHxS8Qazovhfwtcapf6NK0GoBZoY4oJFYqVMruqE5U4AYkgEjI5r6J1r4HeLP2kf2W/grJ8OrS08RT+G01Ky1OyW+hgmgeSdWXIldRjCE4znDKQCDkdZ+y/odj4k/Zt8TfDk+Crbxv4s0nxLPLqvhNvFLaLNOojWMSCSIkTKrKU2sQmVznIXPMszxOHw9SUpxnNSatbVe81d67W9PUX1enUnTSi0mlr8ttt7/APDHxZ8SvhX4s+D/AIjOg+MdEn0PVPLEqwzMrq6EkBkdCVYZBGVJGQR1BFW/GnwV8a/Dvwj4Y8U+INAm03QPEkXn6VetLG63CbQwJCsWQlWBAcKSDkA4Ne7/ALcmoeIrfTPhz4e1/wCF0vw3i0m3u105LjxOutvcQs0ZKmQEsApAwGY8NgYAxXm/xk+FfxC8F/Cn4Y+IPFfimLW/DOuWbS6Bpo1aW5bT4tqMVELgLECCmRGSAQASCAD7+CxdTEYenVqWTk7d112s327s561CFOrKELtJX/q6RyPwx+Cfjb4yy6wngzQZtcbSLU3l6IpY4/KiGecuy7icHCrljg4Bwa5zwr4W1bxt4gs9E0Owm1PVbx/Lgtbdcs5wSfYAAEkkgAAk4AJr1P8AZr+FfxD+J0/jFPh/4pi8Ltp2kSXOpmTVpbH7Va87ov3YJcHByGwg43EZGek/YMV5PjuqRshLaVeKbcMEmuAY8eXC5dQsh4IJIAAY8dR6cJc1SUb3sr/mefLSClbrb8jm/Ef7HHxg8I+HtR1vVfCP2TTNPt3urqf+0rN/LjRSzNtWYscAE4AJPYVneFf2Vfit418Kx+JNF8HXd3pEkZlimM0MbyqBnckbuHYHsVU57Zr6I+Pnwx1b4XfAXXPD/hfwF4rHh281JNS1fWPEepWV00ARgV8qO2lchScFpGUEAfMTnI9Q1mwTx3qnhPx14J+EifEWC1sbY6drFj47k01rIoq5hNsSqRlSCCoznB3DJIp8ztf+tg6pep8lfs1/sr678a/E8bajp17aeELW6ms9Tvre4ghuLadIiwj8uQls7igPyEDcemDjj/ih+zp8QPgzptvqXi7QP7J0+5uDbQTG9t5t74LAYjkYjhSckAcV618JfF0c/wC3l/a2tafB4RmuNVvobmwku1ljtrl4JYzGZQArEynGQMEtxmvK/i/+zl4++C8I1DxVo0dnptzdtbW95FeQzJM2GYEKrlwCoJBZR74PFOcnyxltdfiVpeUTfbwvonwq+AdtrmsaVb6t4w8cJLFpUd7GWTTbFCA9yo4xK7Y2t2BBHG4Hnf2bNF8HeIPjV4b07x89onhKZpxfNfXps4RiCQoWmDqV+cJj5hk4HOcHsP2koTq3w2+CniK0Il0pvDEekmROQlzbsVlQ+hyfqcH0rwLHvXymQ8+MwNWtUm+epKd/7urjZdrJfqenmEY0pQpQ2UY/O6Tb+82PGEOnW/i7XItJ2jSkvp1swjl18kSMI8MSdw24wcnPXJ616bofw/0Dw/8As163458R2H2vWda1BNJ8MxvLJGIzH81zc4VwGAHyDcCAw5BzXmHhPwzf+NPE2l6DpcPn6jqNzHa28fPLuwUZPYZPXsAT2r1v9rLxRYyeONP8C6DIH8M+BbNdFtSpG2W4XBupjx95pcg9c7Ae9etW5+alhYS13b8l/m7Hmp3k5W/rp/Xkd54X/YD8ba98FtR199EvP+ExkuLZ9H02PUbMQXNnIFZ5XJfggE4BdTx0NeT6P+zn8RD8XH8B/wDCLLeeK7GNby40eS+twDDhX5k80IQVdcgNnnpkHHrn7PPgvUfi/wDsp/EnwN4Za2vPFbatZ6jHpstwkLywrtyVLED+EjJIAOMkZFeQaf8Asw/EvUviVqPgG38NeZ4u061F7dad9vtR5cJ2EN5hlCH/AFqcBifm6cHHn4fE11VxFOtVSte110std9v16jUVKmnbX/gmn+1lb3Vn8Wnhvfhzp/wuuEsYA2g6XcQTxdGIlLwgJlgeQACNozk5Jq+A/wBlH4r/ABM8OR6/4b8HXN9pEmfKupLiC3EoHBKCWRSwyCMqCCQQDkV7P+3n8E/Gll4oi8cTaNs8LWuk6fYTX/2qA7JwgQr5YfefmIGQpHfOOa6j4xfB/wAeftDW/wAPPFHws1O2vvBtnotpaw29vqqWw0a4jUiRmQsNrDIBK5fjBGAM89HMXHCUfZySu7Nu7S8t73fqVKL50mul/wAjwfwn8K9Ik/Zz+LevazpDx+LPDeoafZ20sksiNas9x5cyGMMFJIyDuBIPTBrgPF3wf8XeA9L0K+17RzpkWuRiXT4ZbiI3MyHo3kBzIoOQAWUA5AHWvqP9m/xRY/DD4T/GHWPFlpD40ttK8SaX9vAnMyXTi8KmdHYfvDuPmKWGGIGcZJGB430O58I/tOeCfip4j10+M/h5rusQXtl4ncAxrGHB8mVQAI2hxygAGEJABDKt0sdXhiakHrG979/di7Lt3MvsN9Vf82eUa9+yL8XvDHg+bxPqXgm8ttGgg+0SyefA8sceMlnhVzIoA5OVGACTgA16B8G/2HfFfxG+F/iDxPeaVexXE2nRXXhdLa/tFj1CRi24SBmJQAAEBjGeevavqfWNJ1/wf8Qte8ceGPgOPE5vI5pD4otfiQzQ31s4Y5MEx27dpyEClV4Ck4Br5a/Y50Z/HXgr41eC9LmtU8Q69osK6bZXEwj88xyOWUE8Ejco59QTgZI4v7TxmJwtSomo8vL+L10UtLd39x0ckYuN+/8AkeHfFD4Q+Lfgzrtvo/jLSf7H1K4txdRwfaYZ90ZZlDZidgOUYYJzx05FdpY/D/w/4Z/Zpv8Axp4gsTdeIfEWpDTvDStK6CGKE5urnargOMnyxuBAYA45rG1j9nPx54V+JegeBNd0X+y/EWtmL7Lb/aIrgFHkKByYnYAAqxOSCApJGOa3/wBrDxdYan8QbbwjoEmfC3gm0XQbAK2VkeP/AF8x4+80mQTzkKDmvedadd0qMJ3b1bjtZfN7uy37mdkpNtbfrt/mbH7Na+HPC/w3+KvxB1nwppfjC/8ADcGnQ6fYa1F51oHuZ3jZ3jPDY2qRkZ4IBBOa888O/Dzxd8evEniK+8JeF7eedXN9cabo6xwQ2ySSYCwxs4OwMwAVckAc8AmvZtL+IC/EL9ivx5ov9hab4ebwlNozG50SP7P/AGp5kpi3Xa8+Y42bixPLEHAxznfsi65f+GvBfxv1TSruWw1C18JvJBcwNteJvMADKexGeCOR1Fcsq1Wn9ZxCj76aSvta0fzvciT5YxS6tfnY8w+KX7PPxD+CtlY3njPw1Lo1reSGKC4+0w3CM4GdpaJ2CnAJAJBIBxnBx6v4guvDvxY/ZL1fxK3grQfDHiXwnqdlZJqHh+0Ft9thlXYfOAyWbuSTyQCMAkHIk1K71H9hW4+1XM10U+II2maQuV3WJZsEk4yxJPqST1Jrof2bfFZ+Fv7OPxW8X/2bZ+Iy1zp+nf2LrUfnafJvZvnli43kZ4GR061liK1aVD2lT44TilbS+q83vexo4qMoOPW/6/5XPFfgPpvhjWPi/wCFrHxp9n/4Ra4vBHftd3ZtYhGVIy8oZSoBwc7h0xnmsr4oWuiaf8SvFVt4b8v/AIR6DVLmLTzDMZkNusrCMrISxdSoBDEnIIOTmuf1C8/tDULm68iG28+VpfIt02RR7iTtRcnCjOAM8ACq/U19FGnJ1VVbtpa3Qx2ugoz81etfEX9mfxX8NPhD4O+JGoXWk33hrxRtW1bTrlpJYHZC4jmBQANhXyFLYKMDg8HyX7vBraE4zvyu9inFxtdbmj4djWfxFpccirLG91ErIwBBBcAgg9QR2r0X4/eIbaXxdqnhuy8NaBo1rpl6yxXOl2AguJQBgCRgcEc5wAOQK898K4/4SbSf+vyH/wBDFdN8df8AksHiv/r+f+lay+GK9f0PZoScMuquO7lFfK0j6Dm/4JyfEb/hTdnqdv4dvJPiFJq5im0htUsRbpp/lEiYNvALl9oxvJwT8vceZfCv9nX4iSfGyTQI/ANv4p1nwncwXmseHbvUbVIpIgysY3kaTYwYEA4LDnkEZFeofDP4W67+0B+xKvhTwOlrqvibRfGTalc6VJdxwTeQ1qYwy+YQuCX4JIB2sASRivNfhZ+yx8UNa+LF9oVn4X8/VPCt7atrNv8A2har9lDnep3GUK+VVj8hbGMHBIr4+jiqy9vCvWimr6NWttr8W2v/AATknTi6UJ06b1/HfTY5P9o6OeD42eKo7jwZZ/D6dZ0D+G9PmimhsSIk+VXiAQ5+8doAyxGBW94c/Y1+MvizwjD4n0zwJeT6NNAbiKWS5gikkjxkMsLyCQgjkYU5BBGQRXp37cPwi8T+Cf2gNa+IXiHQtngjV9ZhS2uzcQSfacQqWXy1cuMiN/vKBx15GfSv2ivgP8V/jP8AF1fiR4C8U2d14Ento59J1yDXRbQ6TAsKh1IDBk+YOT5YYnPODkCIZpKOGw6pyjHmjq3dpNJab3vr1Z0ywqlXqqSbs9lu7vf+kfNemfDfQbz9j/UvGv8AZpk8Vx+Mo9IivFlkz9ma1EhjEYbYSWOc7S3OM44ri/GvwT8a/D3VtD0vX9Cex1fWo45bHS1nimu3EhAQNDG7PGzEgBXVSSCAMggfVv7JfxI8OfCn9l8eIfFGljUdLh+IkcZl2B3sZGsVCXSIQQzRnkAg9yBkCsfwz4Wvfgn+2x4M8ZePPEMfiPwrr15Nfaf41nlDW18ksEiwyNJ91CrPGCOAoAIwoBqoZhXpYitC10rtX62S92PbuXUw8JUKc+tv/bnq/wAjxTx1+yD8YPhr4Tl8S+IfBN1Y6LCoaa4juIJzEp/idI5GdQO5YADuRXq/hH/gnn448RfA3U/EcmhXw8ay3Fs+iaZHqdkLa6s5ArPK5LnDAMcAup4HBr37xpoviL4U6X8SfEFh+zuz6fqmk3i6v4jj+Jj3kF1bNGS8xhnyzEKMj5AwAIUjPPgv7NvgnUfjL+yL8UfAPhY2t74ufWLPUotLmuEheaFduSpYheqMMkgA4yRkV5/9qY3EYb2qajaUddHo2r3tJ2t1vb5GksJQo1ow1d1L8tNbdTwTUP2f/iDpvxQX4cTeG538aMEI0u3mimYBkEgJdHKAbSCSWwB1IxVv4sfs1/En4H6bZX3jXwxJotleTeRBOLqC4VpAC2wmKRsHAJAOMgHHQ16Z+zz8Ofiv8If2lpPD+nWWi6b43sdOkkn0XXr2MwX0EiITbiSNmUuyurDDDG0knAIPf/tM/CrTvhz8JdL8TXugXfws8SjXkmXwIfEi6rZX/UtdRqGYxsMkZ6AErjLCvVqZpVp4mlSUoyUktlu3210X3nKsKpU5yUWmunyvrp/kct4g/Z/+CXwLsdE0n4ueKPF1x451C3iu7rTPCUVv5WmRyDKiVpVO4jBztOT2XBBPjmgfA3Wvix8Rtd0H4U2F74z02zmka2vGVbbdbbiI5JWkKrGWA6MRkg4HGB9P/tHfs0+Jv2pvHVt8VfhVPpvifw94htLY3C/2hFDLp8qRKjLKrkYACjIBLA7gV4BOR8EfAOsa18EfjD8GfCmvaYnxLh1yKWSO1vREuqWcWEeOGY7dyhlcnOB8wBwGzXHSx86eHdf2nNU05k9o3aTut7L8TpnhouUacY+5/N1el/TU+d/HH7NfxK+G/ijQ/DviPwrcaZqmuTpa6cjTwyRXMrMFCLMrmMHLKCCwwCCcA5rotU/Yr+Nei+GtR1688BXkOm6eJWuH+1WzSKsZIdhEJC7KMEgqpBHIJBBr6HGh6r8HPh/8Dvhp461W3u/G7fEOw1a20iO6W5k0mxDLH5bspIALkkAEg7jgkKcP+G/jXXte/b5+JceoaxeXcSW+taekUspKLbxRv5cYXoFUopAAHIJ6kkuWbYxwc6fK+VSd7O0krbai+p0YpOV9XFW6pu6108rnl/wH/YU8U/Fb4Y+I/Fd7pV7F52l/afCotNQtFTUbjcylJAxLKoKj73l9eteWa5+y78T/AAz8RdE8B6l4Z+zeK9aiM9hp/wDaFq3nIA5J8xZSi/6t+GYHjpyM+r/sO6UfGnhv40+B7C5tY/EXiLw0INNtrqURCd1ZsgE8ZG9fwOegJEn7NnwV8Z/A/wDbL+GWk+NdG/sXULpp7qGH7VDPujME67sxOwHKkYJB46YxWv13E0cRiFOotFdRt5X01+/9DCNGnOhFqLvezd/TfT7jy3Xf2SPi74b8Bz+MtT8EXtp4ehi+0S3DzQmSOPu7QhzIFA5JKgAcnA5pPhv+yV8Wvi34bXX/AAr4MuNQ0d2Kx3ktzBbLKQcEoJpFLgEEZUEZBGcgil8afHHx5onxS+I1zbeJ71ptae80e/a623AmtDKw8rEgYKAOFKgFQSFIBOfpT4gfBvx9+0h4T+DniD4Tavaz+GtI0C006S3j1NbU6JfRAiaWRcgg8qCUBbC5AIIJ1qY7G0aUJ1XGKl11stL66/dqP6vQlUlCnd8t9NNdUtP+GPlHwe2jfCP4halY/E/4dXHij7JG9rLoN1qU+lyQXBZSJDJGCxwoIAxghwQTgZ6P9sDwB4e+GPx813w/4W07+ydEt4bSSCz8+SbYZLeN2+eRmY5ZieT34wOK9Z/aS+H+uftRftMa5afDCzh8XXmh6VaW+qXttd28KTzRqEklDyOobBIXIJOVPYZpP+ChnwV8Z6P8VNX+IN3o3leEbw2NlBqP2qBt8wtVUr5YcuOY3GSoHHXkZVDGxqYmhOpO0pxd430v7ttPPW3U1+r2pVoQV+W3TXz18jn9S0H4YeOf2UPFXi7QPhx/wififw9d6bYtqH9u3V79qaVsSSeW5CJu2n5cHG7gjFeMfCv4I+N/jdqlzp/gnw/PrtzaRiS4KSRxRxKSQN8kjKgJwcAkE4OAcGvrzwR+yH8W7H9lH4heE7jwn5fiDWNU027sbP8AtK0PnRRsS7bxKVXAI4YgnsDS/sj6Na6h8DfiN8MLvwpa+LvGVn4jaW/8JSeJX0aW6hSJYiFniyJQkiOCpOw7gSeVzyxzH6tTxMqEue0la7vZNR11e179TKnhnV9iqq5bp9LbN2+bVvM+SfGvwF8ffDvxtpfhPxH4buNL17VJI4rG3lkjaO5Z3CKElVjGfmIBIbAJ5IrsH/Yo+NqaRqepnwBeiz015kuCLm3L5iJDmNBJulAIODGGDY4Jr3742X2vaR4v/Z+8Ja38JJPhpaad4jjfSWn8VrrjSxG4g82MNksqhmjIDMQBgKAAceZftQfG7x14F/bC8X61ovia9tb/AEi6ksrBpCJo7eBolVo1ikDJtIJJBUjOG6gEdFDMMfiuSFNRTak+60dls3vfuXPDYakpSld2t+N+68jxL4V/B3xh8bfEE+h+CtEfXNVgtXvZLeOaOIrCpUM2ZGVerKAAckkAA1x8kTxzNG6skisVZWHIIOCCPXNewfsxfDj4g/Fz4i6lp3w78UJ4U19NNnvLi+fUpdPDwBkEke+EFjuLL8uNuBkkYryCSNo5njcgsrFSVYEZBwSD3HvX1MZNz5Xbb/M8lxXJzLudn8VPgt41+CupabY+NdBm0K51C1F5apJLHIJIiSMgxswBBGCpIYcZAyK0vhz+zj8SPizpD6p4V8L3GpacjmP7U88NvG7DqEMrqGx0O3ODwa3f2nvhb8QvhX4m8P2vxF8UxeK9RvdJiubO4TVpb8wWxZgsRMgDKAQcADacnaTzj13xP8M/F/7QHwZ+Fr/DK8t9T0rQ9Ljs9R0WG/jtntL5WJeZ1ZlBJ6gk5xyAQ2aKT5qXO3fXp8zuw+GhVnJNPRXt1e3+d/Q+evCJ0f4Z+OtR0/4h+BJPEv2ZXtJNHm1OWwa3nDL85kiBJwAwx0O7OeBWn+054N0X4ffHLxR4f8P2X9naPZvB9ntfOeXYGt4nI3uSxyzE8k9fTAqxpfwF+IfxE+I3iPRdKhj8X65o8w/tK8h1OJlJzt3CSZ1L8gjjJ4r0j9tb4N+KdM+L2ueNdT077F4R1S8tbWDVPtETAt9mjU/uw5cYMb9VA+XryM9CtKMNdX/wLfqXKjL2FTlg7Rl21631+48+tf2Q/i9e+G016LwTdtpzQ/aFP2iATFMZBEJk8wkjnG3J9K5T4a/Bfxt8Xry8tvCOgT6vJZgG4YSRwpFnIAZ5GVQTg4GcnBwODX6IeBvhHo/g/wCKmgapY+EZ9UsoreOO3+IereM5JJbsvEQES1DESZyAFIVccgcDPzf8C9D8Y+IrH4j6VpHhfRfiD4UvNbdNS8MXWqCwvQ6sSk8UuV2qcgZJOSpAAwScub3mvX8z0JZZSioXb1369L6af5nzn4z+F/ir4e+LU8M6/otxY69IEMdkNszS7zhNhjLBskEDaTkgjrXVeMv2Xfih8P8AwpN4k1/wnPYaNEEaW4+1QSGIMQAXRZC68kA5AwTzivQPjN4CTwL8d/ANl4W102GuOlpIthruqLepoFyJcpbvMNw8tTggEEgEk5yK9P8Ai58LLq+8B+PPFPjnwuvwy8QtaGc6xo/iNZbDxDcAgiBrQuWAYgEAc55PIwW/4fMnrr+BjHL6ftKlOV9LW+6+rt+iPh3mj+der/s+fs2+KP2lNa1nSvCl7pFvqGmWX21odUumiadQdu2IKjFmzgEkBRkZIyK8rmheCZ45F2SIxRlPUEHBH51KqRcnBPVHg8j5ea2gwKa9m+HuhfD29/Z5+JOp679h/wCE4spLQaGs+otFOweQCXy4BIBKAuSSVbAOeMV4z9aKzrUnWjyqXLt+Dv8AjsKLtJS7HWfDf4V+LPi5rv8AZHhDRJ9b1BUMrxxFUWNem5nchVGSACxGSQBzXs/xu/Zvb4K/s5+DdR8Q+Hv7G8e3mtXFvfSfbfP324V2jGEkaMcAHKgHjk1r/st6PqXjz4EfFvwV4Nu4rXx7qH2Ge3h+1C2mvLWN2MsSOSBjBIIJA+cAkA5rvfG3hRfgh8D/AIHaZ4p1CxvzovjpbjVls5fPjtgG814WYcFlQgMBwCSBnGT8tjcfWWKVKLsoyWnV6Xv6dCo2tzep89Xf7H/xisPCL+JZ/At8ukrB9pZvNhM6x4ySbcOZRgckFMgckV2P7Pv7GfiP4xeA9e8UXGm3iafJpNzL4dltL21QX1/GzIsMisSyKWRgSwQccMAQa+vNa0rXdJ+Kl58QvCHwOXxrNcBpbfxXY/Ehlhu7dgQAbeUhAu058sKyKcbScA18n/seRweM9e+LXhmwa10zUvFPhe/s9Js7mfahkdgViDkckA+mSFJxwawjmeMxOGqTVo8tn3t3Vk7/AD09DScIxtr1R5X41/Zv+I/w98TaF4e17wzNba1rjFdOs4bmC5e4IIUgeU7AckdSOMnoCa0PiB+yh8V/hf4Zm8Q+JvCE1ho8JUS3Ud3b3Aj3EAFxFIzKMkDJAGSBmu5+FX7MviHwH+0N4D0H4l/afA8eqSSzWtzZ6nCJpTGhIRJoXbyyzbVySCdxAOSDX0/4k+Htp4B+CPxsRPh4vgGS80iZ0+0eLJNWudSRCQJ3hZmWMcj5gxbLAEDArXFZxVw7pxhJTva9lo9bae9/mEYKU2ntofL/AMBfhn4r0b4V+LvG03wh0zx74X1LSbhbfVdR1C1jOm+V5gknjjcmQlSp4UKx2Da3PPjHw18ReGPDPiL7Z4t8Jf8ACa6V5LJ/Zf8AaUth+8ONr+bGC3GDxjBzz0r6a/ZB/Zz+IbaJ4w8Rjw/nRfEngvUdP0q6+22/+kTy7BGu3zNy5KNywAGOSMivDNP/AGYfiZqfxK1HwBb+GvM8Xadai9utP+32o8uE7CG8wyhD/rU4DE/N04OO6niKU6+IhOa2vo7aa+elurVg5b0lKPf/ACsdF+1Z4D8IfDn402em6DpMui+HJtNsb2SzguZLiRPNTdJteViS2OmTjIHQV6Dovwj/AGcNd+FviPx9Bc/FNNH0G5t7a5gkk00XDtM21CihCpAJ5ywI7A1b/b0+CXjSz8TxeOJtF2+FrXSdPsJr/wC1QHbOECFfLD7z8xAyFI75xzXIfDrP/DEPxez/ANBfSf8A0ctcEakq2ApVKdV3uk7PvJJ387E6qcbrdfo/1M34LfDOTxt8Z7rWfhf4Du/iH4N0CeGaXR/FN5ZW800ckbALMSRGSXVyNqsAFXI5wfLvi8zf8LS8VB/Dtt4SkTUpo30KzkWSKxZWIMSMmFIUgjKgKewAwK9E/ZT+CPjX4i/ETw74i8PaN/aOi6FrljNqNz9qgi8hFlVydjurN8qk4UE8etdP8c/hnr3wh/aeg8YeNdHFj4U1fxdLf287Twzi5tUulkclEdmA2MDhgDzjGQQPR9tGljfZuabUe+vTfXVv0uUo81OUktf+H/I4df2QfjC3g9vFJ8D3iaOtsbsu88CzeUBuLeQXEvTnG3JHar2p/DDRZP2UfCniuw0qSXxdqXiqXSmuIpJXeaLymKRLFuK5LAYIXcTxk5xXtPx++AXxa8XfFDxH8SNI8U2dz4MuIJr2y8TR64sMFtYlCRAAG3AbRtwqlWPJOSa2f2ePiN4f+GP7MfgbUfEUPlW1z4qvbKDWFjWSTR5pLdgl2iMCrFeQQQcBiRnGD5lTMa9Sgq0ZKTTXuxun10ZnUiozir6XevTZnyL4o+CXjTwf4w0/wpqmiNH4lv1RrfSrW4iubg7/ALoZInYox67Wwcc4xzW58Rv2XPij8J/D413xT4RuNO0nesbXcdxBcLGW4G/ynYoCSBlgBkgdSBXuvwX8K3nwF/bCsZPiHqEOox63BdS6V4onuSttftMh8ucT5O0tnaeSVMg6ggn0D4jaD4l+Ffwv+Jn2L4APomlaxp00Opa9/wALBOpREMMC58mbLOwYgg7QxxjIBNa1M2xEalOEEndJt7J67K7VrfMuMFJtf16nkdv/AME+fiHN8Iv7UHh67PjttXEUelDU7IW504whvPLb8b/MO3G8HA+73r5/1bwLf/Dz4jDwz4wsVsb2yuokv7RrhGCq21sGSNiuCjA5DcZ6givf/g/8M9a+N/7HeteE/B62uo+JbDxiuqyaXJdRwytbm0WIMN5C8tnBJAO1ucjB+b/G3gjWvhx4q1Dw54isv7O1mxZUuLXzUl2FlDAb0LKcqwPBPWvQwVatUr1aNaom10tZ9Nd3p8gqRjKmuU7n9p/RfBHh340a1p3w7azbwpDHb/ZXsL43sLMYUaQiUu5b5ywPzcEEYGK1Ph7r9n4q8A+LdIu/C3h1H0nw/JPBqcOnKLxpFKqGeQk5OGJyADkDmvFvbHNen/Bf/kB/En/sWp//AEYle9hafsqXs272i9X5Lc9TLar+txSWj0/ApfA/XvC+k+NFsvGWj2+p+HdXjOnXVxIMS2AcgC4hYnCshAJOCdoYDrg5nxa+HN58J/iNrfhW8fz30+fbHcKMCaFgGikA7bkZSR2JIzxXMabp1zrGo2thZwtcXd1KsEEMfLSOxCqoHckkAfWvdv2vNfgi/aGk8pYNQk0S0sbO5WZd8c00USl1ccZGTtIz2I+nyspzoZ7CnCV41KcnJdnFxUX5Xu152M4xU8FUbXwtWfrfT8L/ACPAfxr2H4Zfs33/AMYvhrreueENatdY8W6PIZLjwWsbLfS2YAzcQknEpBJBjUZ4HJJVTf8AiF+1J/wsDwfqHh//AIVF8K/Df2wKP7U8PeGvst9DtdWzHL5p2k7cHg5BI71ufsufD2w8OInxp8X+LZfCHhHw1ehLcaTcL/amq3iqGFpboDlQQQGZgBtYjoWZfoZVJ+zlKfutbdf67HBGMeeKi73+R4v4L+HXiP4ha1c6R4e0qbU9Ut4JbmSzQqsmyMZfAYglh02jLE8AE8V0/wAQf2bviT8K/DsWveKPC0+maTK6xi58+GYIzDIDiN2KZ6fMBzx14r6A+Ffxvk+MX7VHjfx/Y6JZeFJpvDuozQW+noAQyRZWWU4w8pIBZsAEgcV5p8P9e1LWv2YfjkdQvri+Z7zRrlmuJS7NK90wdySSSzbVyScnAranKc4XkrOyb+bsRJKMny6q9jmvBv7KPxX8feG7fX9C8Hz3elXCl4Z5Lm3hMqj+JUkkViD2IBB7ZrO8E/s4fEj4iLdnQPC9xeGzvm025V5ooWguFUs0brI6lcAckgDOBnJAr7M+G/wc0PTdL+GGt2XhWXxzaRWdtdyeNNS8ZyWUGlSbgTFFbqxyFOQECgZwrHOTVDxXql3pWm/E1bK4e1Fx8VbGGUxHaXQmFipI5wSoJ9cU5y5KnJ/W6X6mMXzQ5v62b/Q+O/G/7PvxC+HHhmLX/Enhi50rSJLo2a3EskbHzQWGCquWAJVsMQFOAQSCM7Mn7Jvxbj8InxKfBN8NJEP2gt5kRn8vGc/Z9/m9Ocbc45xivqn4keKFj0/4r6l4hmkvNN074jaV5qSZcLbRGPKAegUHgDnnqSa7fWNL1aD4pzfELwf8Fk8bTTgy2niux+IZjjuYWUgAwSEKF2nHlhSoPTOAaiNRyjzf1sn+v4F/at/W58DfAf4YxfFj4hW2l3tz9h0S0gk1HVrwY/c2cQDSEe5yFB5wWBwcYrJ+J/ijR/F/ja9vtB0K18OaGu2CysbVNuIUG1HkOTmRgAWOeST16n1v9ma4TxR4k+LGh2FqllqPiTwxqUOmWYkBHmEhhApwMnbkDAGQpOK+eGUq2CNpBwQRzmvmMFUnic4xDqtr2aiox8mruXnd6X8j0pxVPCLlW8nd+lrLy3PZfjLoXw90n4TfCm88KfY/+Eov7O5fxCLbUWuJVkUxhPNjLsIicsQAq5GeuK734f8A7CfjLxp8E9X8Uf2Nef8ACQ3MdldeG7OLULRYL63lKtJJJuYlSIzkBmQ57E8V8ufjX1P+zP4TvPit+zv8YPAXh57afxXfSabeWmn3E6wtOkU259hYgZwuOSACVyQCDXdjVXweGvTqfaV29dHL12V/uPOi7uKfoeWr+zL8SrX4nweApfC6v4ra2F8ultf2+Hh5OTIJQgBCnjcD6dq1P2trW7sfinDBffDfTvhbcR6ZAG0PS7mCeJ+XxOWhATc3QjGcKMkk5rPi/ZV+KUnxGl8Cp4Xz4qjsRqb6f/aFrxblgu/zPN2HkgYDZ56V7R+3L8DfG9jfaR4zn0TZ4a0vw9pmm3d99rgPl3CjyymwOXOGYDIUjnOcc1jLFpYqhz1IyvfbT9XvshpPmaS/rQ8V+Hv7KvxV+Kfh5dd8MeELi/0liVS6kuILdZMcEoJXUuAQRlQRkEZyCK1vh58LNMuvg/8AHG/8Q6S48S+E10+O0MksiNZzPdNFMpRWCscLghgcEcYPNe8fFL4R+O/2h/Dvwx174Wala6h4T03Q7OzNjBqiWx0i8jUiV3QsAGAKglctx0xgnnP2aNat/hz4U/aC1HxZaQeNotJvdLOpRm4M0d+y38geQSN/rMsC4LDDHGRgkVzVMwrVaM5JrmTXurde8lrr1NKS1g2r7f8ADHzZ4o+E/ivwT4X0PxDrml/2Zputrv083FzEJ50AzvEG/wAwJjGGKhTlcHkZ7S7/AGP/AIw2HhF/E03gW+TSkg+0s3mwmdY8ZJNuH80YHJBTIHJAr034yaPdQ/HLwX8Y73XpPGXwz1nWLO6t9ZZci0hSZWezljAHlsgDgKAAcE43bgPp/WtK13SfipefELwh8Dl8azXAaW38V2PxIZYbu3YEAG3lIQLtOfLCsinG0nANRiM3xFOnTcEryv5JNdNWtf6sNU05Wv2+Z8nfBH9iLxR8Tvhv4h8T3ml3kfm6YLnwx9lvrVU1C43MCsgYkqowPvbOvWvFPil8HfF3wX1q10nxlpH9jahdW4uooftMM+6MsyhsxOwHKsMEg8dMEV7t+yDZp8QNP+N3hTSPs1hrHibQ2TStPnuAoYiRj5YcgZ2hlGcdOTgA48J+KPwd8X/BbWrXSfGWkf2NqF1bi6ih+0wz7oyzKGzE7AcowwSDx0wRXdhcRiHjalKtUVtLRtrt0d+nXclRXJKyvbr9xxnUGvYPhD+y18SPi1a2et6J4SuNR8Om4CPdSXMNskihsPtMjqWAwQSucEEZyK8g43Zr7k8QfCXxF8afBPwS8QeHtd0u0+Hvh/QLSLVbi61BYYdLu4Tm5kkQkHccAAgE5HJUEE9mZYqeGhHkaXM93tt8jJR5rryPEfjv8C5I/wBqHxH8Pfhv4duJo4ZLdbPTbeR5Sga2ikcmSRiQu52JZmwM9QMVzfjT9l74ofDvUtDsPEHhK4sJtbu0sbBhcQSxS3DkKkZkR2RGJPAZhkAnoCR9maRrmi2v7a3xv0a9it59X1zR7e00uK41J9O+0t9lty1utwgLRs4wQVGRtOBXAftBab4h+HP7O/iHw3/woqTwD4bv7+2lbUn8bDV0inWUMGSFizAuAQSNv3snOMV8/QzTFJ0aCS1UdXu77vVr8mb8qlKfk2cf4k/4J6/EW1+HvhK80fw7d3fi66a7/t3T5dUshDaBZALfyyXAO5Mk4dsHj5eleJ+Cf2ffiD8SPFGt+H/DnhubU9V0WVoNQjWeFEt3DMhVpGcJksjAYY52kjIGa94j+Cvir9oD9k34TQ+Ara116+8O3WqwalYi9ihmhM1z5iEiRlXG0A4JBwykAgkix+yf8E7LWfB/xMt9U0W98YeKdMvE0+bwHa+Iv7LiuNhIMksiOFcKxkAJJUbWwCWFdMcwrYejVlVqKUoyatbb3rJ77W229RSimo8vU+efHPwJ8d/DPxTpfhzxL4dm0rVtUZVsonlidLgswUBJFcoSGIB+bjIzgEV6d+0B4B8SfDT4M+CdC8T/AAk0zwdqFvdyq/ii3vrWe51M4ZtkghJYABgcszD5RjA4r0/9s7wPcXHgz4GeE9F0C30nUJrq/wBOttEttabUlgmeWFVg+1zbSTuYAhiFQkqDhQam8dfsq/FKX9k7wV4ZTwv/AMTvRdWv7+/tf7Qtf3MDAlX3+btbI7KSR3FZrMnWp0K1ZpXk01ttdX3/AMyvZ8s1bqjyf9kPTfhv4+8W2XgTxd8Pv7f1TVJrmWHXv7aubb7LHHatII/s8ZCv80THcWB/edwoB+d15r6y/YS+BfjjUPid4W+IdvonmeDoTfQvqX2uAYY200QHllxIfnZRkLjnPQE18+/FD4QeLfgxrlvo3jHSf7H1Ke3F1HB9phn3RlmUNmJ2A5RhgnPHTkV7FGvSeOqU41N1HS99byvp00tsZWbhc7T9pXQfh/oOoeCo/h/9hMNz4dtrrVPsOoteBb1i3mK5MjbGAAygxjPQZq58H/C+h/Gb4e654JXSrWy8c6XBNrGjapAoWTUFUAy2kxJ+Y4wUPbB6AHd4d6GvdP2MLOSP46WGuM/kaZoFjealqE7cLHAtvIhJPTkuo/8A1V5efQnhMqnVhUfPT96L7yT0XmntY7svfNiacHG6dk/yfz63PG/D2g6h4q17TdF0q2a91PUbmO0tLZCAZZpGCooJIAJYgZJA55NavxG+G/iP4S+ML7wt4s0t9I12y2efaNIkm0MgdSGRmVgVYHIJ61F4Vtb7XvHmk2+k3iaRqN5qUSWl5JOYFtpXlASQyDlApIJYcjGR0ruP2nPhT46+D3xWutH+IusL4h8SXNtFevqy3st59qjYFVYySgOSNhXDAEbeOME/TRqO8Lu11t5nLKEfftrY8mooorrOU/XT/gkP/wAm3eJf+xsuf/SOzoo/4JD/APJt3iX/ALGy5/8ASOzor4DGf7xP1PrcN/Bj6H5F8etHHrXpX/DM3xg/6JR43/8ACcvP/jdH/DM3xg/6JR43/wDCcvP/AI3X3Ht6X86+8+X9lPszzXj1o49a9K/4Zm+MH/RKPG//AITl5/8AG6P+GZvjB/0Sjxv/AOE5ef8Axul7al/MvwD2c+zPNePWjj1r0r/hmb4wf9Eo8b/+E5ef/G6P+GZvjB/0Sjxv/wCE5ef/ABuj21L+dfgP2c+zPNePWjj1r0r/AIZm+MH/AESjxv8A+E5ef/G6P+GZvjB/0Sjxv/4Tl5/8bp+3pfzr7xeyn2Z5tjPtXSeD/B0fjD7TGNc0vR7iGSHCarcrbpJGxIdw7EKSmFJXOSGJAOCK6X/hmb4w/wDRKPG//hOXn/xuj/hmf4wf9Eo8b/8AhOXn/wAbrlxE41aThSqqLfXRmlODjJOUWxD4G8I+J/3HhjxItjdWsyxTt4kuIrWK5jOAbiFiAFAbJMTEttIIJIICL4C8J67OJ9G8Uw2GlWbmK/k1qeOO5dVXcbi3hwC6vhlWIFnBAycMMWov2WvjJPaz3C/CvxkEg27lfQrlXOTgbUKBm99oOBycVFH+zJ8YZJFQfCnxsCx2jd4eu1GenJMYAHuTivF+p1FosW/LZ7737+XY6/ad6X5ka+BvCHiVor/RPEi6NpUTOL+28QXEQvIUUMwkiVQon3KMBUG4MQDkENQvgbwh4laK/wBE8SLo2lRM4v7bxBcRC8hRQzCSJVCifcowFQbgxAOQQ1T3n7LnxjsbqSCT4VeM2kjbaTDoNzKn4MqFSPcEj3qH/hmf4wf9Eo8b/wDhOXn/AMbo+pT+zi3pts9PPu+z6ffc9otnS/MRfA3hDxK0V/oniRdG0qJnF/beILiIXkKKGYSRKoUT7lGAqDcGIByCGpIvBXhC+f8At2LxB9k8Kxqzy6fcXcTauGVgoiWMABmkyGDhSqhiGJKHL/8Ahmf4wD/mlHjf/wAJy8/+N0g/Zm+MH/RKPG//AITl5/8AG6PqU9li3b5bdfV9n0H7TvS/MwPFvhvRrXT7TWfD+ppc6ZdERtYXc8f2+0lwSyugwWTgkSKoU5AIB4rlT9K9J/4Zn+MH/RKPG/8A4Tl5/wDG6B+zP8YQf+SUeN//AAnLz/43XsYZwoU1TnV5rddL/wBeZyVIynLmUbHmy47ivS/gP8Zz8FPEuoajJoVv4isdSsJNNvLG4lMJeGQjcFkAJQnAGQDkE8ZwQn/DM/xh5/4tR43x/wBi5ef/ABuj/hmf4w8f8Wo8b/8AhOXn/wAbrrVenr7y+8y9lN6NM7LWP2lvDmk+BfEPhv4dfDO18CDxDD9l1K8k1efUpZICCCieYBsyCRnkYJwAcEeBV6V/wzP8YP8AolHjf/wnLz/43R/wzP8AGD/olHjf/wAJy8/+N0vbUf5l94/Zz7M826d6Pxr0n/hmf4wf9Eo8b/8AhOXn/wAbo/4Zm+MH/RKPG/8A4Tl5/wDG6ftqX86+9C9nPsyv4V+L1zovw18ReBtS0y313QtS/wBItEuHKvp14MAXETAHtkFeAeOQCwbz4frXpP8AwzP8YO/wo8b4/wCxcvP/AI3S/wDDM/xg5/4tR439v+KcvP8A43XHQp4TCynKk0ud3evXubzdapGMZJ2Wxyngfx1rfw38TWviHw7eLYaxahvIumgjmMZZSpIWRWUHBIBxkZ4IrDmmkuJnlldpZZG3O7klmJOSST1JPevRv+GZvjD/ANEo8b/+E5ef/G6P+GZ/jB/0Sjxv/wCE5ef/ABuun2lBS57q/wAjH2c/5WebfjSfjXr9v+yH8a7rS2v0+FvigQKrMUk02RJcDOcRMA5PHAC5PGM5rK/4Zm+MH/RKPG//AITl5/8AG6r21G/xL8A9nPszzXj1o/GvSv8Ahmb4wf8ARKPG/wD4Tl5/8bo/4Zm+MH/RKPG//hOXn/xuj2tHbmX4B7OfZnm340fjXpP/AAzN8YP+iUeN/wDwnLz/AON0f8MzfGD/AKJR43/8Jy8/+N0/bUf5l+AvZz7M8149aX8a9J/4Zm+MH/RKPG//AITl5/8AG6P+GZvjB/0Sjxv/AOE5ef8Axul7Wj/MvwD2c+zOW8C+PNc+GviSDX/Dd4un6vbq6RXTW8cxjDqVYqJFYAlSQCBkZOCKwGZnYsxLMxJLMeST3Nekf8MzfGD/AKJR44/8Jy8/+N0v/DM/xg/6JR44/wDCcvP/AI3UKph4y5k1cfs59mXfgb8crX4S2vifSNa8KWnjTwv4kt4oNR0m5uGty5iYtGyyqGKlSzHIGc4IIIzXnXiK+sNS13UrvStN/sXS57iSS107z2n+zRliVj8xgGfaCBuIycZPWu4/4Zn+MHT/AIVR43x/2Ll5/wDG6P8Ahmf4wf8ARKPG/wD4Tl5/8brKKw0akqykry31/TYrkqWtZnmvcV7V42/aD0rVPg3bfDrwj4FtvB2my3EN5qt0t+95NqE0a4DEugKAt82ASBgAYGQcb/hlj4yfYPtn/CqvGPleZ5W3+w7nzc4zny9m7b/tYxnjOeKgH7M/xg/6JR43/wDCcvP/AI3RWWFruLqNaa7iVOad0meaVe0S6tLHWLC51Cy/tOxhnjkuLLzTF9ojDAvHvHK7gCNw5Gcjmu8/4Zm+MH/RKPG//hOXn/xuj/hmX4w/9Eo8b/8AhOXn/wAbrp9tS/nX3k+zn/Kzqf2gP2nm+MXhnw14O0DwnY+AfAPh4tLY6DYztcEzNndJJMwXcfmbHyg5diSxOR4ax3NntXpQ/Zn+MGf+SUeN/wDwnLz/AON0H9mf4w5/5JR43/8ACcvP/jdZU5UKStGS+80n7WduZfgef6Xff2bqdnd7N/kTJLszjdtYHGcHGcdcV23xS+IWgfEK6e/sPCH9g6xPcGe6vf7TkufOBBBXYVCrzg5A7YxzVr/hmb4wZ/5JR44/8Jy8/wDjdH/DM3xg5/4tR44/8Jy8/wDjdbfWKVkuZfeb06telSlRivdlvpf/AIbfoea/jScetelf8MzfGD/olHjf/wAJy8/+N0f8MzfGD/olHjf/AMJy8/8AjdT7Wj/MvwOX2dTszzXj1o49a9K/4Zm+MH/RKPG//hOXn/xuj/hmb4wf9Eo8b/8AhOXn/wAbo9rQ25l+Aezqdmea8etHHrXrvh39kv4w+ItWisR8NvFWneZnNzqOh3cMK4BPLGI4zjAABySKo3H7MPxgtriWI/Cvxo5RmXdH4eu2U47giPBB9R1o9tR/mX4B7OpbZnmHHrR+Nelf8Mz/ABg/6JR43/8ACcvP/jdH/DM3xg/6JR43/wDCcvP/AI3R7WjtzL8A9nU7M8149aOPWvSv+GZvjB/0Sjxv/wCE5ef/ABuj/hmb4wf9Eo8b/wDhOXn/AMbo9rR/mX4B7Op2Z5rx60cetelf8MzfGD/olHjf/wAJy8/+N0f8MzfGD/olHjf/AMJy8/8AjdHtaP8AMvwD2dTszzXj1pfxr0n/AIZm+MH/AESjxv8A+E5ef/G6P+GZvjB/0Sjxv/4Tl5/8bo9rR/mX4B7Op2Z5rx60cetelf8ADM3xg/6JR43/APCcvP8A43R/wzN8YP8AolHjf/wnLz/43R7WjvzL8A9nU7M8149aPxr0r/hmb4wf9Eo8b/8AhOXn/wAbo/4Zm+MH/RKPG/8A4Tl5/wDG6ftqL0cl+Aezn2Z5r+NHHrXpX/DM3xg/6JR43/8ACcvP/jdH/DM3xg/6JR43/wDCcvP/AI3S9rR/mX4B7Op2Z5rx60cetelf8MzfGD/olHjf/wAJy8/+N0f8MzfGD/olHjf/AMJy8/8AjdHtaH8y/APZ1OzPNePWjj1r0r/hmb4wf9Eo8b/+E5ef/G6P+GZvjB/0Sjxv/wCE5ef/ABun7aitpL8A9nU7M8149aOPWvSv+GZvjB/0Sjxv/wCE5ef/ABuj/hmb4wf9Eo8b/wDhOXn/AMbo9tS/nX3oXs59mea8etHHrXpX/DM3xg/6JR43/wDCcvP/AI3R/wAMzfGD/olHjf8A8Jy8/wDjdHtqX86+9D9nU7M8149aOPWvSv8Ahmb4wf8ARKPG/wD4Tl5/8bo/4Zm+MH/RKPG//hOXn/xun7el/OvvQezqdmea8etHHrXpX/DM3xg/6JR43/8ACcvP/jdH/DM3xg/6JR43/wDCcvP/AI3R7el/OvvDkqdmea8etHHrXpX/AAzN8YP+iUeN/wDwnLz/AON0f8MzfGD/AKJR43/8Jy8/+N0e3pfzr70Hs6nZnZ/Az9qS2/Z98E6xD4Z8C2D/ABCv0lt08Z3l48jW1u+3KR223aCMZDBhyRkEDFeCMxkYszEsxJLMckk9STXpA/Zn+MH/AESjxv8A+E5ef/G6P+GZ/jB/0Sjxv/4Tl5/8brFToRm5qSu/MvlqcihZ2R5rx60fjXpX/DM3xg/6JR43/wDCcvP/AI3R/wAMzfGD/olHjf8A8Jy8/wDjdbe3pfzr7zP2c+zPNvxo/GvSf+GZvjB/0Sjxv/4Tl5/8bo/4Zm+MH/RKPG//AITl5/8AG6n21HfmX4B7OfZnmv40v416T/wzN8YP+iUeN/8AwnLz/wCN0f8ADM3xg/6JR43/APCcvP8A43R7Wj/MvwD2c+zPNfxo49a9K/4Zm+MH/RKPG/8A4Tl5/wDG6P8Ahmb4wf8ARKPG/wD4Tl5/8bo9rR/mX4B7OfZnm340nHrXpX/DM3xg/wCiUeN//CcvP/jdH/DM3xg/6JR43/8ACcvP/jdHtaP8y/Afs59mea8etL+Nek/8MzfGD/olHjf/AMJy8/8AjdH/AAzN8YP+iUeN/wDwnLz/AON0e1or7S/AXs59mebfjScetelf8MzfGD/olHjf/wAJy8/+N0f8MzfGD/olHjf/AMJy8/8AjdHtaN78y/APZz7M81/Gl/GvSf8Ahmb4wf8ARKPG/wD4Tl5/8bo/4Zm+MH/RKPG//hOXn/xuj2tH+ZfgHs59mea/jRx616V/wzN8YP8AolHjf/wnLz/43R/wzN8YP+iUeN//AAnLz/43R7Wj/MvwD2c+zPNvxpOPWvSv+GZvjB/0Sjxv/wCE5ef/ABuj/hmb4wf9Eo8b/wDhOXn/AMbp+2o3vzL8A9nPszzbv1r0vQ/il4f8O+C9S0nT/Bnk6xqWmtp91q/9qSN5gbBLeSUKjkA4BHTGaT/hmf4wZ5+FHjj/AMJy8/8AjdB/Zm+MGf8Akk/jcf8AcuXn/wAbo+sU7NKS+9HXh6lfDSc6a19L/mjN+EfxOHwl8RXOv2+jWuq6wlq8em3F0x22Nw2AJwmCHIUsADgAkHPGDx2oX1zqmoXN5eTyXF5cSNNNPKxZpHYksxJ6kkkk+pr0P/hmf4wbQD8KPHH/AITl5/8AG6P+GZ/jC3X4UeN//CcvP/jdccKeEp154mLXPKybv0Wy9DNus4ezadtzzX8aT8a9K/4Zm+MH/RKPG/8A4Tl5/wDG6P8Ahmb4wf8ARKPG/wD4Tl5/8brs9tS/nX3ow9nPszzb8aPxr0n/AIZm+MH/AESjxv8A+E5ef/G6P+GZvjB/0Sjxv/4Tl5/8bp+3pfzr70Hs59mebfjR+Nek/wDDM3xg/wCiUeN//CcvP/jdH/DM3xg/6JR43/8ACcvP/jdL21L+dfeg9nPszzb8aPxr0n/hmb4wf9Eo8b/+E5ef/G6P+GZ/jB/0Sjxv/wCE5ef/ABuj21L+dfgHs59mcX4V8Ual4J8R6druj3TWep6fMs8Ey/wsD0I7gjIIPBBIPBrY+KPji1+I3jS+8Q2+hWvh5r4LJcWlnIWiafaPMkUEDaHbLbR0z1JyTuf8Mz/GD/olHjf/AMJy8/8AjdL/AMMz/GDp/wAKo8b4/wCxcvP/AI3XE6eD+sfW7rnta9+m9vPy7GydZU3Ts+Vnmn40v416T/wzN8YP+iUeN/8AwnLz/wCN0f8ADM3xg/6JR43/APCcvP8A43Xb7ak95L8DH2c+zPNfxo49a9K/4Zm+MH/RKPG//hOXn/xuj/hmb4wf9Eo8b/8AhOXn/wAbpe1o/wAy/APZz7M81/Gj8a9K/wCGZvjB/wBEo8b/APhOXn/xuj/hmb4wf9Eo8b/+E5ef/G6Pa0f5l+Aezn2Z5r+NHHrXpX/DM3xg/wCiUeN//CcvP/jdH/DM3xg/6JR43/8ACcvP/jdHtaP8y/APZz7M82x713fwr8XeCfCV5fyeNPAH/Ce280arbQ/2zPpv2dgSWbMQJbIIGDwMcdavf8Mz/GD/AKJR43/8Jy8/+N0f8Mz/ABg/6JR43/8ACcvP/jdZVJYerFxlL8bfkP2c+zO2/wCFzfAn/o3T/wAvi/8A/iK8DmZXmdkTy0LEqmSdoJ4GT1xXo/8AwzP8YD/zSjxv/wCE5ef/ABuj/hmf4wj/AJpR43/8Jy8/+N1lRhhqF+Wd/WTf5tl8tRq3L+B5r+NJx616V/wzP8YP+iUeN/8AwnLz/wCN0f8ADM3xg/6JR43/APCcvP8A43XV7Wj/ADL8DP2c+zPNvxo/GvSf+GZvjB/0Sjxv/wCE5ef/ABuj/hmb4wf9Eo8b/wDhOXn/AMbo9rR/mX4B7OfZnmvHrRx616V/wzN8YP8AolHjf/wnLz/43R/wzN8YP+iUeN//AAnLz/43R7Wh/MvwH7OfZnmv40fjXpX/AAzN8YP+iUeN/wDwnLz/AON0f8MzfGD/AKJR43/8Jy8/+N0e1o78y/APZz7M824HfNd/p3xcuNC+Et74K0bTIdNk1W48zWNYjkJnvoVOY4MEfIikkkA/NnnALBrY/Zn+MGP+SUeOP/CcvP8A43R/wzP8Yf8AolHjf/wnLz/43XJiaeExaiqzT5Wnv1W3rbzNabrUW3BNP0KXwX8c+Fvh142TWvFvgaD4g6fFCwh0m6vntIhPuUrKxVGDgAMCjAqd3PQVN8evjfr/AO0J8Sb/AMYeIVhguJlWC3s7UERWtumdkSZ5IGSSTySxPAIAm/4Zn+MH/RKPG/8A4Tl5/wDG6X/hmf4wdvhR43/8Jy8/+N1u5UHNTcldeYkqqi4pOzPNOPWjj1r0r/hmf4wf9Eo8b/8AhOXn/wAbo/4Zm+MH/RKPG/8A4Tl5/wDG629vS/nX3mXsp9mfpT/wSJ/5Nu8Sf9jZc/8ApHZ0Vuf8Et/AfiT4f/ADxBp/ijw9qvhvUJfE9xcJa6tZS2srRm0tFDhJFBKkqwyBjKnuDRXw2KalXm13PqMOmqUUfZFFLRXEdwlFLRQAlFLRQAlFLRQAlGB6UtFACUUtFACUYHpS0UAJgelFLRQAmB6UYHpS0UAJgelGB6UtFACUUtFACUUtFACYHpRj2paKAEopaKAEopaKAEopaKAEopaKAEopaKAEopaKAEox7UtFACYHpRS0UAJRS0UAJRgelLRQAmPajHtS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJRS0UAJgelGPalooASilooASilooASilooASilooASilooASilooASilooASilooASilooASilooATHtRS0UAJj2opaKAEopaKAEopaKAEopaKAEopaKAEox7UtFACUUtFACUUtFACUUtFACUUtFACUUtFACUUtFACUUtFACUUtFACUUtFACUUtFACY9qMe1LRQAmB6UYHpS0UAJRS0UAJRS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//Z)

* Ici la route prend un paramètre appelé n. Il sera défini et traité dans le contrôleur et servira aussi dans le modèle lors de la requête pour conditionner le son résultat. 
* Le contrôleur : Il utilisera la fonction getone\_ticket qui prendra n comme paramètre. Il passera les éléments récupérés à la vue grâce à une variable donnée comme valeur dans un tableau associatif lui-même utilisé comme paramètre dans la fonction view() qui prend deux paramètres, en premier le nom de la vue à retourner et en deuxième les variables nécessaires à la vue. Ces variables sont placées dans un tableau associatif comme dans l’exemple ci-dessous qui montre le retour de la vue ‘detail accompagnée des variables nécessaires à son bon affichage.

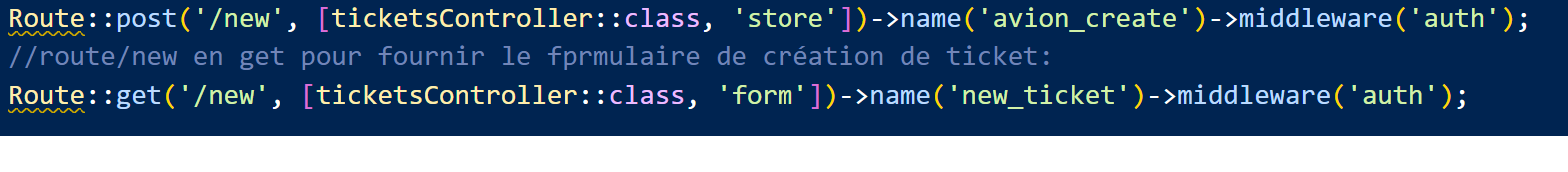
Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

* Le modèle : Il récupère tous les éléments de la table ticket et filtre le résultat en fonction de n. La fonction retournera donc seulement les éléments correspondants à la condition du WHERE. La variable n’est pas passée directement dans la requête mais est remplacée par un point d’interrogation et est donnée dans un tableau en deuxième paramètre. Cela sécurise la requête

3.Les routes en post : elles seront utilisées lors de l’envoie d’une requête post sur le chemin correspondant à celle de la route. Contrairement aux routes en GET elles ne sont pas tant présentes pour délivrer une page mais plus pour récupérer de données d’un formulaire soumis sur ‘url qu’elle indique.

Dans l’exemple ci-dessous la route en post sur le chemin *new* appellera la méthode *store* du contrôleur des tickets.



## Eléments de sécurité

### Requêtes https

Même si pour ce premier jalon nous n’avons pas eu à gérer la configuration du serveur. Je note tout de même ici que le serveur hébergeant mon application exécute les requêtes en utilisant le protocole HTTPS.

### Jeton CSRF

Dans mes formulaires j’ai placé un jeton CSRF (Cross Site Request Forgery) ce jeton est une mesure de sécurité qui permet de s’assurer que les requêtes soumises à un serveur web sont authentiques et ne sont pas falsifiés par un tier malveillant. Cela sécurise l’application et lui permet de se prémunir contre les attaques type CSRF.

Dons l’exemple suivant on peut remarquer que j’ai placé cet élément de sécurité sur la ligne n°7 :

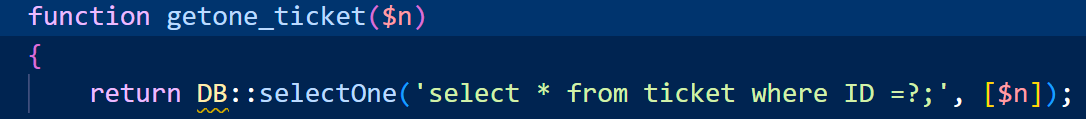
Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

### Classe DB ::

Comme je l’ai expliqué dans un précédent chapitre sur mes requêtes SQL dans mon Model.

J’ai utilisé la classe DB :: mise à disposition par laravel pour interroger la base de données. L’utilisation de cette dernière permet de diminuer l’exposition aux attaques par injections SQL grâce à la possibilité qu’offre cette classe de préparer la requête en lui passant des valeurs par paramètres



### Version à jour du Framework utilisé :

Pour le développement de mon application j’ai utilisé une version à jour du Framework la 10.10.1. Cela diminue le risque de failles de sécurité.

### Middleware\* :

Certaines de mes routes sont équipées du morceau de code pour sécuriser leur accessibilité





Un \***middleware** est une couche intermédiaire entre la requête http et la réponse http d’une application Laravel. Dans le cas que j’expose dans la capture ci-dessus. Le middleware ‘’auth’’ créé par fortify dans le fichier dédié est utilisé afin de vérifier si l’utilisateur qui demande la route en question est bien connecté sur l’application avant de pouvoir accéder à cette route. Dans le cas contraire Laravel redirigera vers la page de connexion.

Ci-dessous le détail de ce middleware construit par Fortify dans le fichier : /home/lab005/s2/www/fil-rouge/app/Http/Middleware/RedirectIfAuthenticated.php:

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Je n’ai personnellement créé aucun midleware avec la commande php artisan make : middleware pour ce premier jalon. J’y ai simplement porté une attention à titre informatif lors de mes recherches sur les éléments de sécurités et après qu’il nous ait été conseillé d’écrire ses instructions sur certaines de nos routes.

# Conclusion

## Bilan du projet

### Problématiques rencontrées :

Lors du développement de cette première application, j’ai logiquement été amené à rencontrer quelques problématiques telles que :

* La compréhension de la transmission des variables entre les déférentes couches du MVC.
* La syntaxe des média quéries en .scss. En effet , je souhaitais que mon application soit autant que faire se peut responsive seulement je ne connaissais la syntaxe uniquement pour les média quéries en .css mais pas en .scss.

### Solutions opposées :

Toutefois malgré ces désagréments liés comme souvent dans les périodes d’apprentissage, j’ai malgré tout su y opposer des stratégies de solutionnement.

* Concernant la transmission des variables au travers des différentes couches du MVC afin de bien comprendre et de renforcer ces concept dans mon esprit, je me suis efforcé à trouver des fonctionnalités qui nécessites d’utiliser la gymnastique du MVC comme par exemple l’ajout de la barre de recherche dans la barre de navigation ou la possibilité de mettre à jour.
* le statu d’un ticket n à partir d’un formulaire utilisable seulement par les utilisateur ayant les droits nécessaires.
* J’ai aussi sollicité l’appuis de l’équipe pédagogique pour m’aider à assimiler ses nouveaux concepts
* Au sujet de la syntaxe de média quéries en .scss, il m’a fallu mener des recherches dans la documentation de Sass : <https://sass-lang.com/documentation/>
* Concernant le documentations, celle de Laravel aura été une aide pour ce projet et la compréhension de sa bonne utilisation. https://laravel.com/docs/10.x/readme
* Dans le futur en continuant de progresser et d’apprendre il me viendra à penser que j’aurais pu et du ajouter telle et telle fonctionnalités. A ce moment-là je reprendrai l’amélioration grâce entre autre à la possibilité d’accès au projet que permet GitHub et à mes nouvelles connaissances et capacités.

## Apports personnels et professionnels

A travers ce projet j’ai pu me rendre compte que toute la période que j’ai passé par le auparavant à travailler le front-end en autodidacte à porté ses fruits et m’a permis d’aborder ce projet avec un peu plus de sérénité. J’ai aussi été très content de pouvoir enfin interagir avec une base de données et d’apprendre l’utilisation d’un framework . Mener ce projet m’a permis de constater que malgré tout je suis capable de réaliser et comprendre des choses. Reste à moi de m’en contenter.

## Perspectives d’améliorations

En vérifiant les déférentes contraintes demandées, j’ai pu m’apercevoir trop tardivement que j’ai oublié deux choses :

* De placer une colonne ‘’date limite » » à ma table des tickets. Cela peut être effectué avec un ALTER TABLE.
* De créer la page tableau de bord. J’imagine cette page comme affichant le nombre de ticket en BDD pour chaque statut d’avancement des tickets.

Pour la faire j’utiliserais ma fonction ‘UserRight pour afficher son contenu uniquement pour les rôles ‘admin’. Je peux créerais des fonctions RES semblables à celles que j’ai déjà pour chaque statut de ticket mais sans la contrainte WHERE cette fois-ci afin de récupérer tous les tickets de chaque statut sans tenir compte de l’tilisateur. Je les aurais compté à l’aide de la méthode count() sur les tickets retournés dans un tableau par la fonction du modèle effectuant la requête correspondante pour indiquer aux admins l’état le nombre d’ assistances des clients par statut.

\_ Je pourrais aussi indiquer aux admins le temps moyen aloué pour le traitements des tickets en utilisant une boucle foreach sur la liste de tous les tickets retourné par le modèle puis utiliser la méthode avg() sur les différences entre la date limite et la date de création des tickets .

\_ Je pourrais faire une chose semblable pour le temps moyen passé à la résolution des tickets par les dépanneurs en faisant la différence entre la date de création et la date fin du ticket cette fois-ci.La comparaison entre ses deux moyennes celle du temps aloué et celle du temps passé peut être effectué afin d’indiquer aux admin les performances du service client d’AMIO. Je pourrais combler ses oublis et ultérieurement sur une nouvelle branche du projet et effectuer un ‘git merge’ lorsque tout sera prêt et entendu avec le client et validé.

\_J’ai aussi effectué quelques recherches pour que lutilisateur puisse metre à jour sa photo de profil via une page dédiée à la gestion de profil.Cette fonctionnalité fera l’objet :

* d’une nouvelle fonction contenant un update de la table users se trouvant dans le modèle( des users).Cela après avoir
* rajouté une colonne dédiée dans la table users,
* placé un input type ‘file dans un formulaire,
* récupérer cette entrée avec le contrôleur grâce à l’objet request en l’ associant à l’utilisateur actuellement authentifié, le contrôleur rangera la nouvelle photo dan un dossier dédié et définiera le chemin ( pour l’utiliser dans la balise <img>).
* Ci-dessous, le stockage dans une variable correspondant au chemin de l’image soumise dans le formulaire si la requête http contiens un fichier provenant de l’input nommé ‘photo’

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

Jai effectué cette recherche sur stackoverflow en écrivant dans la barre de recherche  :’’ *how to update a profile picture in laravel application* ‘’. En cherchant parmi les résultats j’y ai trouvé une proposition de marche à suivre. Je l’ai analysé et grâce à mes nouvelles connaissances j’ai réussi à comprendre les actions qui y sont menées pour réaliser cette feature. Elle aussi pourra faire l’objet d’une nouvelle branche de mon repo Git ultérieurement.

# 

# Annexes

### Quelques requêtes SQL

* La requête *select* suivante récupère les différentes colonnes de la table ticket nécessaire à l’affichage de ces derniers. La fonction à laquelle elle appartient se fait passer un paramètre e para mètre est la variable contenant la valeur de ce que l’utilisateur à soumis dans la barre de recherche. Cette valeur est utilisée dans ma requête afin d’être comparée avec un like. L’éventualité d’une saisie en lettres capitale est envisagée avec la méthode UPPER().J’ai aussi effectué une gestion d’exception. Si la requête venait à échouer la fonction retournera ‘false’ autrement elle retournera true.

Une image contenant texte, capture d’écran

Description générée automatiquement

* Requête *update* : Cette requête est écrite dans le contexte de la mise à jour du statuts d’un ticket

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

* Exemple de requête dans une transaction :

Dans l’exemple suivant puisque j’avais plusieurs requêtes pour l’envoi d’un message dans la conversation d’un ticket, j’ai effectué ses deux requêtes dans une transaction.

* Exemple d’insert

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

## Schéma de la base de données

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement