Questions d’Évaluation

➔ Compréhension Technique

1. Gestion des limites d’exposition
Le smart contract gère les limites d'exposition grâce à une vérification dans la fonction mettreAJourExposition. Lorsqu'une exposition (longue ou courte) est mise à jour, le contrat recalcule l'exposition courante et vérifie si elle dépasse la limite spécifiée pour la contrepartie. En cas de dépassement, un événement LimiteDepassee est déclenché pour alerter les parties concernées.
Mesures de sécurité :

1. Les fonctions vérifient que la contrepartie est active et existante avant toute mise à jour.
2. Les calculs d'exposition sont encapsulés pour éviter les erreurs logiques.
3. L'émission d'événements fournit un mécanisme de journalisation pour suivre les dépassements.

2. Implémentation des calculs de risque
Le calcul du risque est implémenté dans la fonction calculerRisque. Voici les aspects pris en compte :

- **Scores de crédit** : Le risque est proportionnel au score de crédit, qui amplifie ou atténue l'impact de l'exposition courante.
- **Exposition courante** : Le risque est directement calculé comme une fraction de l'exposition courante par rapport à la limite d'exposition.
- **Historique des transactions** : Bien que non explicitement intégré, l'exposition longue et courte cumulées offrent une vision des transactions passées.

Limitation actuelle : Le modèle ne prend pas en compte l'évolution temporelle des scores de crédit ou des transactions.

3. Efficacité en termes de gas
Le contrat est conçu pour minimiser les coûts de gas :

- Utilisation de mappings pour un accès rapide aux données des contreparties.
- Calculs simplifiés pour les ratios et les risques, réduisant la complexité algorithmique.
- Les événements sont utilisés uniquement en cas de dépassement ou de mise à jour, limitant leur coût.

Optimisation possible : Réduire les écritures sur la blockchain (opérations coûteuses) en externalisant certaines vérifications (via des scripts ou oracles off-chain).

4. Gestion des cas limites

- **Congestion du réseau** : Le contrat repose sur la gestion des transactions via le gaz ; si le réseau est congestionné, la priorité des transactions peut être ajustée via des frais plus élevés.
- **Transactions échouées** : Les vérifications en amont (existence de la contrepartie, activité, etc.) empêchent les erreurs d'exécution.
- **Entrées invalides** : Les require statements garantissent que seules les données valides passent.

5. Stratégie de test et résultats

Scénarios testés :

1. Ajout de contreparties avec des paramètres corrects et invalides.
2. Mise à jour des expositions (longues et courtes) et déclenchement des limites.
3. Calcul des ratios et risques dans différents cas limites (exposition nulle, score élevé).
4. Désactivation et réactivation de contreparties.

Résultats : Les tests ont validé le bon fonctionnement des fonctions principales, avec un comportement attendu pour les cas limites.

➔ Compréhension de la Gestion des Risques

1. Comparaison : gestion traditionnelle vs blockchain

- Avantages de la blockchain :
  - Transparence accrue via un registre immuable.
  - Réduction des coûts opérationnels (pas besoin d'intermédiaires).
  - Automatisation via les smart contracts.
- Limitations :
  - Dépendance à l'infrastructure blockchain (coûts de gas, congestion).
  - Moins de flexibilité pour les ajustements manuels.

2. Gestion des scénarios de risque

- Augmentation soudaine de l’exposition : Alertes immédiates grâce aux événements déclenchés.
- Détérioration du score de crédit : Pas de mécanisme intégré ; une mise à jour manuelle est requise.
- Transactions multiples simultanées : La nature séquentielle des transactions Ethereum garantit une exécution ordonnée.

3. Améliorations potentielles

- Incorporer des oracles pour surveiller les changements de marché en temps réel.
- Ajouter un historique des scores de crédit pour modéliser des tendances.
- Introduire un mécanisme de pénalités automatiques pour les dépassements répétés.

➔ Implémentation et Innovation

1. Fonctionnalités supplémentaires

Ajout du ratio de couverture pour évaluer la solidité financière des contreparties. Cela permet de fournir un indicateur de liquidité utile dans la gestion des risques.

Valeur métier :

- Identifier rapidement les contreparties à risque de défaut.
- Faciliter les audits financiers.

2. Applications potentielles dans le monde réel

Applications :

- Gestion des risques pour les plateformes de prêt décentralisé (DeFi).
- Systèmes de gestion de garanties pour les échanges de dérivés.

Modifications nécessaires :

- Sécurité renforcée pour résister aux attaques malveillantes.
- Conformité avec les réglementations financières locales.

3. Réflexion sur le développement

Défis rencontrés :

- Complexité des calculs d'exposition dans des cas extrêmes.
- Gestion des coûts de gas lors des tests sur des réseaux simulés.
- Après avoir progressé jusqu'à l'étape de calcul du risque, j'ai rencontré une erreur persistante. Cette erreur, liée à la logique initiale défaillante du contrat, a renforcé la nécessité de corriger la méthode de calcul. Malgré mes efforts pour résoudre ce problème

dans le cadre de l'intégration front-end, il est devenu évident que la base même du calcul nécessitait une refonte complète.

- **Solutions :**
  - Simplification des fonctions et modularité accrue.
  - Utilisation de réseaux de test pour optimiser avant le déploiement final.
- **À améliorer :**
  - Intégration des tests automatisés pour les mises à jour futures.
  - Optimisation des coûts d'écriture sur la blockchain.