

Questions d'Évaluation

4.1 Compréhension Technique (40 points)

1. Gestion des limites d'exposition et mesures de sécurité (8 points)

Mon smart contract gère les limites d'exposition à l'aide d'une variable `limiteExposition` assignée à chaque contrepartie. Lorsqu'une mise à jour est effectuée via la fonction `mettreAJourExposition`, le contrat vérifie si la nouvelle exposition dépasse la limite définie.

- **Mesures de sécurité :**
 - La validation des données garantit que l'exposition est positive.
 - Les rôles et permissions sont contrôlés pour empêcher toute modification non autorisée.
 - En cas de dépassement, un événement `LimiteDepassee` est déclenché, fournissant des alertes en temps réel.

2. Implémentation des calculs de risque (10 points)

Les calculs de risque sont effectués à partir des éléments suivants :

- **Scores de crédit :** Un score est attribué à chaque contrepartie et utilisé pour pondérer les risques globaux. Un score bas augmente significativement le niveau de risque.
- **Exposition courante :** Cette variable est comparée à la limite d'exposition pour identifier les écarts. Elle est intégrée dans les calculs pour déterminer le ratio de couverture et le risque global.
- **Historique des transactions :** Bien que le contrat ne stocke pas directement l'historique complet (pour des raisons de coût en gas), les positions longues et courtes sont conservées pour analyser les variations de risque.

3. Efficacité en termes de gas et optimisations (7 points)

Plusieurs optimisations ont été mises en œuvre pour minimiser les coûts en gas :

- **Types de données optimisés** : Les variables comme le score de crédit utilisent uint16 au lieu de uint256.
- **Simplification des calculs** : Les fonctions évitent les calculs redondants, notamment lors du calcul de l'exposition nette.
- **Éviction de données inutiles** : Le stockage de l'historique détaillé est évité pour limiter l'usage de gas, tout en conservant les informations critiques.

4. Gestion des cas limites (8 points)

- **Congestion du réseau** : Le système repose sur la confirmation des transactions blockchain. En cas de congestion, les utilisateurs peuvent attendre ou soumettre à nouveau leurs transactions.
- **Transactions échouées** : Les transactions sont validées avant exécution. Toute transaction non conforme (données invalides, insuffisance de gas) est automatiquement rejetée, évitant les corruptions.
- **Entrées invalides** : Les contrôles au niveau des fonctions (par exemple, vérification des adresses Ethereum) assurent que seules des données valides sont traitées.

5. Stratégie de test et résultats (7 points)

Les scénarios de test incluent :

- **Ajout de contreparties** avec des valeurs valides et des entrées erronées.
- **Mise à jour des expositions** au-dessus et en dessous des limites autorisées.
- **Calcul des indicateurs financiers** dans des cas extrêmes (valeurs très élevées ou négatives).
- **Résultats** : Les tests ont validé le bon fonctionnement du contrat dans les scénarios standards et les cas limites. Les erreurs sont gérées proprement avec des messages explicites.

4.2 Compréhension de la Gestion des Risques (30 points)

1. Comparaison entre gestion traditionnelle et blockchain (10 points)

- **Gestion traditionnelle** :

- Avantages : Flexibilité humaine, systèmes éprouvés, ajustements manuels possibles.
- Inconvénients : Risques d'erreurs humaines, faible transparence, lenteur des processus.
- **Gestion via blockchain :**
 - Avantages : Immuabilité des transactions, transparence totale, automatisation grâce aux smart contracts.
 - Inconvénients : Dépendance aux frais de gas et à l'infrastructure blockchain, complexité technique.

2. Gestion des scénarios de risque spécifiques (10 points)

- **Augmentation soudaine de l'exposition** : Le contrat détecte et alerte sur tout dépassement des limites via l'événement `LimiteDepassee`.
- **Détérioration du score de crédit** : Le calcul du risque est ajusté dynamiquement en fonction du nouveau score, augmentant le collatéral requis si nécessaire.
- **Transactions multiples simultanées** : La nature séquentielle de la blockchain garantit que chaque transaction est traitée dans l'ordre, évitant les incohérences.

3. Améliorations potentielles au modèle de calcul des risques (10 points)

- Intégration de **prédictions basées sur l'intelligence artificielle** pour anticiper les risques.
- Ajout de pondérations sectorielles pour différencier les risques par industrie.
- Incorporation de **l'historique des transactions complet** pour une meilleure analyse comportementale.
- Surveillance en temps réel des tendances économiques.

4.3 Implémentation et Innovation (30 points)

1. Fonctionnalités supplémentaires et valeur métier (10 points)

- **Calcul du collatéral requis** : Une fonction pour vérifier automatiquement si le collatéral est suffisant, réduisant les risques pour les contreparties.

- **Automatisation des alertes** : Des événements comme `LimiteDepassee` permettent une gestion proactive.
Ces fonctionnalités augmentent la sécurité et la réactivité du système.

2. Applications potentielles dans le monde réel (10 points)

- **Cas d'usage** : Gestion des risques dans les banques, automatisation des prêts garantis, ou même traçabilité des dettes inter-entreprises.
- **Modifications nécessaires** :
 - Déploiement sur un réseau principal (Mainnet Ethereum).
 - Intégration des réglementations locales via des couches KYC et AML (Anti-Money Laundering).
 - Amélioration des interfaces pour une adoption facile par des utilisateurs non techniques.

3. Réflexion sur le processus de développement (10 points)

- **Défis rencontrés** :
 - Intégration de Web3.js avec l'interface utilisateur.
 - Gestion des coûts en gas sur la blockchain.
- **Solutions** :
 - Tests réguliers via Remix IDE et Sepolia Testnet.
 - Optimisations de code pour limiter la consommation de gas.
- **Améliorations futures** :
 - Automatisation des tests via Truffle ou Hardhat.
 - Adoption d'un processus agile dès le départ pour mieux suivre les étapes.