Justifications Détaillées des Réponses

Examen Solidity

Question 1 : Fonction Modulo

Réponse correcte : C. 1

```
contract ModuloFunction {
    function modulo(uint a, uint b) public pure returns (uint) {
        return a % b;
    }
}
```

Calcul de `modulo(10,3)` :

— $10 \div 3 = 3$ avec un reste de 1

— Donc $10\%3 = 1$

Conclusion : La réponse correcte est **C. 1**.

Question 2 : Fonction isEven

Réponse correcte : A. false

```
contract BooleanLogic {
    function isEven(uint number) public pure returns (bool) {
        return (number % 2 == 0);
    }
}
```

Analyse de `isEven(7)` :

— $7\%2 = 1$

— $1 \neq 0 \Rightarrow false$

Conclusion : 7 est impair, la fonction retourne **false**.

Question 3 : Fonction Factorielle

Réponse correcte : C. 120

```
function factorial(uint n) public pure returns (uint) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}
```

Calcul de `factorial(5)` :

$$5 \times 4 \times 3 \times 2 \times 1 = 120$$

Question 4 : Fonction Fibonacci

Réponse correcte : B. 8

```
function fibonacci(uint n) public pure returns (uint) {
    if (n == 0) {
        return 0;
    } else if (n == 1 || n == 2) {
        return 1;
    } else {
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
}
```

Suite de Fibonacci : $0, 1, 1, 2, 3, 5, 8, 13, \ldots$

$$fibonacci(6) = 8$$

Question 5 : Contrat Vote (Après déploiement)

Réponse correcte : A. 2

```
contract vote {
    uint public nb_vote;
    mapping(address => bool) public voters;

    function ajoutervote() public {
        require(voters[msg.sender] != true, "vous avez d  j    vot
           ");
        voters[msg.sender] = true;
        nb_vote++;
    }
}
```

Deux votes sont possibles uniquement avec deux adresses différentes.

Question 6 : Même Contrat Vote

Réponse correcte : A. 2
Deux appels avec deux adresses distinctes incrémentent correctement `nb_vote`.

Question 7 : Fonction calculExact

Réponse marquée : A. 9

```
function calculExact(int8 a, int8 b) public pure returns (int8) {
    int8 resultat = 5 + a * 2 - 4 / b;
    return resultat;
}
```

Calcul pour `calculExact(2,2)` :

$$5 + 4 - 2 = 7$$

Remarque : Il existe une incohérence entre le résultat réel (7) et la réponse marquée (9).

Question 8 : Valeur de nb_vote

Après deux votes valides :

$$nb\_vote = 2$$

Question 9 : Adresse du propriétaire

Réponse correcte : A.
L'adresse ayant déployé le contrat devient automatiquement le propriétaire.

Question 10 : calculAvecFonction

Réponse correcte : A. 11

```
function calculAvecFonction(uint256 a, uint256 b) public pure
   returns (uint256) {
    return a + b * 2 - div(b, 2);
}
```

$$6 + 6 - 1 = 11$$

Question 11 : Tableau statique

Réponse correcte : C. `int[5] myArray;`

Question 12 : Mot-clé view

Réponse correcte : A.
Une fonction `view` lit l'état sans le modifier.

Question 13 : Mapping

Réponse correcte : A.

```
mapping(address => bool) myMapping;
```

Question 14 : Retourner un string

Réponse correcte : C. `returns (string memory)`

Question 15 : Tableau utilisateurs

Réponse correcte : A.
Le tableau stocke les adresses appelant la fonction.

Question 16 : Variable public

Réponse correcte : B.
Permet l'accès externe via un getter automatique.

Question 17 : Mot-clé memory

Réponse correcte : B.
Stockage temporaire durant l'exécution.

Question 18 : Syntaxe require

```
require(condition, "Message d'erreur");
```

Question 19 : Struct Coordonnees

Réponse correcte : B. (4,6)

$$(1 + 3, 2 + 4) = (4, 6)$$

Question 20 : GestionVotes

Analyse basée sur la gestion d'un tableau de votes et l'ajout contrôlé des entrées.

Conclusion : L'ensemble des réponses est cohérent avec les règles fondamentales de Solidity concernant la logique, la mémoire, la visibilité et les structures de données.