**THIS CHECKLIST IS NOT COMPLETE**. Use `--show-ignored-findings` to show all the results. Summary - incorrect-exp (1 results) (High) - divide-before-multiply (9 results) (Medium) - assembly (52 results) (Informational) - pragma (1 results) (Informational) - dead-code (5 results) (Informational) - solc-version (1 results) (Informational) - too-many-digits (1 results) (Informational) ## incorrect-exp Impact: High Confidence: Medium - [ ] ID-0 Math.mulDiv(uint256,uint256,uint256) has bitwise-xor operator ^ instead of the exponentiation operator **: - inverse = (3 * denominator) ^ 2

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

## divide-before-multiply

Impact: Medium Confidence: Medium - [ ] ID-1 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division: - denominator = denominator / twos - inverse = *2 - denominator* inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-2 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - denominator = denominator / twos
  - inverse = (3 * denominator) ^ 2

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-3 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - low = low / twos
  - result = low * inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-4 Math.invMod(uint256,uint256) performs a multiplication on the result of a division:
  - quotient = gcd / remainder
  - (gcd,remainder) = (remainder,gcd - remainder * quotient)

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L315-L361

- ☐ ID-5 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - denominator = denominator / twos
  - inverse = *2 - denominator* inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-6 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - denominator = denominator / twos
  - inverse = *2 - denominator* inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-7 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - denominator = denominator / twos
  - inverse = *2 - denominator* inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-8 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - denominator = denominator / twos
  - inverse = *2 - denominator* inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-9 Math.mulDiv(uint256,uint256,uint256) performs a multiplication on the result of a division:
  - denominator = denominator / twos
  - inverse = *2 - denominator* inverse

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

## assembly

Impact: Informational Confidence: High - [ ] ID-10 SlotDerivation.deriveArray(bytes32) uses assembly - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L64-L69

- ☐ ID-11 Math.tryMul(uint256,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L73-L84

- ☐ ID-12 ERC1155Utils.checkOnERC1155Received(address,address,address,uint256,uint256,bytes) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/token/ERC1155/utils/ERC1155Utils.sol#L25-L50

- ☐ ID-13 StorageSlot.getAddressSlot(bytes32) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L66-L70

- ☐ ID-14 Arrays._swap(uint256,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L170-L177

- ☐ ID-15 Arrays._castToUint256Comp(function(address,address) returns(bool)) uses assembly

- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L194-L200

- ☐ ID-16 Math.mul512(uint256,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L37-L46

- ☐ ID-17 Arrays._begin(uint256[]) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L142-L146

- ☐ ID-18 SlotDerivation.deriveMapping(bytes32,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L107-L113

- ☐ ID-19 SlotDerivation.deriveMapping(bytes32,bytes) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L144-L154

- ☐ ID-20 Math.add512(uint256,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L25-L30

- ☐ ID-21 Arrays.unsafeSetLength(uint256[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L477-L481

- ☐ ID-22 Arrays.unsafeSetLength(bytes32[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L466-L470

- ☐ ID-23 SafeCast.toUint(bool) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/SafeCast.sol#L1157-L1161

- ☐ ID-24 Strings._unsafeReadBytesOffset(bytes,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Strings.sol#L484-L489

- ☐ ID-25 StorageSlot.getInt256Slot(bytes32) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L102-L106

- ☐ ID-26 Arrays._castToUint256Comp(function(bytes32,bytes32) returns(bool)) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L203-L209

- ☐ ID-27 Arrays._castToUint256Array(address[]) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L180-L184

- ☐ ID-28 SlotDerivation.deriveMapping(bytes32,int256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L118-L124

- ☐ ID-29 Arrays.unsafeSetLength(address[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L455-L459

- ☐ ID-30 Math.log2(uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L612-L651

- ☐ ID-31 Math.mulDiv(uint256,uint256,uint256) uses assembly
  - INLINE ASM
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L204-L275

- ☐ ID-32 Panic.panic(uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Panic.sol#L50-L56

- ☐ ID-33 SlotDerivation.deriveMapping(bytes32,bytes32) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L96-L102

- ☐ ID-34 StorageSlot.getBytesSlot(bytes32) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L129-L133

- ☐ ID-35 Arrays.unsafeMemoryAccess(bytes32[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L433-L437

- ☐ ID-36 Strings.escapeJSON(string) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Strings.sol#L446-L476

- ☐ ID-37 SlotDerivation.deriveMapping(bytes32,string) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L129-L139

☐ ID-38 StorageSlot.getStringSlot(string) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L120-L124

☐ ID-39 Arrays._mload(uint256) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L161-L165

☐ ID-40 StorageSlot.getBytes32Slot(bytes32) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L84-L88

☐ ID-41 Math.tryMod(uint256,uint256) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L102-L110

☐ ID-42 StorageSlot.getBytesSlot(bytes) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L138-L142

☐ ID-43 Strings.toString(uint256) uses assembly
- INLINE ASM
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Strings.sol#L45-L63

☐ ID-44 ERC1155._asSingletonArrays(uint256,uint256) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/token/ERC1155/ERC1155.sol#L380-L400

☐ ID-45 Math.tryDiv(uint256,uint256) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L89-L97

☐ ID-46 Arrays.unsafeAccess(address[],uint256) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L383-L389

☐ ID-47 Strings.toChecksumHexString(address) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Strings.sol#L111-L129

☐ ID-48 StorageSlot.getBooleanSlot(bytes32) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L75-L79

- ☐ ID-49 Math.tryModExp(uint256,uint256,uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L409-L433

- ☐ ID-50 StorageSlot.getStringSlot(bytes32) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L111-L115

- ☐ ID-51 Arrays.unsafeAccess(uint256[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L409-L415

- ☐ ID-52 Arrays.unsafeMemoryAccess(address[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L422-L426

- ☐ ID-53 ERC1155Utils.checkOnERC1155BatchReceived(address,address,address,uint256[],uint256[],bytes) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/token/ERC1155/utils/ERC1155Utils.sol#L60-L87

- ☐ ID-54 SlotDerivation.deriveMapping(bytes32,bool) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L85-L91

- ☐ ID-55 Arrays._castToUint256Array(bytes32[]) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L187-L191

- ☐ ID-56 Arrays.unsafeMemoryAccess(uint256[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L444-L448

- ☐ ID-57 Arrays.unsafeAccess(bytes32[],uint256) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L396-L402

- ☐ ID-58 SlotDerivation.erc7201Slot(string) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L45-L50

- ☐ ID-59 SlotDerivation.deriveMapping(bytes32,address) uses assembly
  - INLINE ASM

lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L74-L80

☐ ID-60 StorageSlot.getUint256Slot(bytes32) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L93-L97

☐ ID-61 Math.tryModExp(bytes,bytes,bytes) uses assembly
- INLINE ASM

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L449-L471

## pragma

Impact: Informational Confidence: High - [ ] ID-62 2 different versions of Solidity are used: - Version constraint ^0.8.20 is used by: -[^0.8.20](lib/openzeppelin-contracts/contracts/interfaces/draft-IERC6093.sol#L3) -[^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/ERC1155.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/IERC1155.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/IERC1155Receiver.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/extensions/IERC1155MetadataURI.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/utils/ERC1155Utils.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L5) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/Comparators.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/Context.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/Panic.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L5) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L5) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/Strings.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/introspection/ERC165.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/introspection/IERC165.sol#L4) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/math/SafeCast.sol#L5) -[^0.8.20](lib/openzeppelin-contracts/contracts/utils/math/SignedMath.sol#L4) - Version constraint ^0.8.30 is used by: -[^0.8.30](lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L4) -[^0.8.30](src/AchievementNFT.sol#L2)

lib/openzeppelin-contracts/contracts/interfaces/draft-IERC6093.sol#L3

## dead-code

Impact: Informational Confidence: Medium - [ ] ID-63 Context._contextSuffixLength() is never used and should be removed

lib/openzeppelin-contracts/contracts/utils/Context.sol#L25-L27

☐ ID-64 ERC1155._burn(address,uint256,uint256) is never used and should be removed

lib/openzeppelin-contracts/contracts/token/ERC1155/ERC1155.sol#L334-L340

☐ ID-65 Context._msgData() is never used and should be removed

lib/openzeppelin-contracts/contracts/utils/Context.sol#L21-L23

- ☐ ID-66 ERC1155._burnBatch(address,uint256[],uint256[]) is never used and should be removed

lib/openzeppelin-contracts/contracts/token/ERC1155/ERC1155.sol#L353-L358

- ☐ ID-67 ERC1155._mintBatch(address,uint256[],uint256[],bytes) is never used and should be removed

lib/openzeppelin-contracts/contracts/token/ERC1155/ERC1155.sol#L317-L322

## solc-version

Impact: Informational Confidence: High - [ ] ID-68 Version constraint ^0.8.20 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - VerbatimInvalidDeduplication - FullInlinerNonExpressionSplitArgumentEvaluationOrder - MissingSideEffectsOnSelectorAccess. It is used by: - [^0.8.20](lib/openzeppelin-contracts/contracts/interfaces/draft-IERC6093.sol#L3) - [^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/ERC1155.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/IERC1155.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/IERC1155Receiver.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/extensions/IERC1155MetadataURI.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/token/ERC1155/utils/ERC1155Utils.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/Arrays.sol#L5) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/Comparators.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/Context.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/Panic.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/SlotDerivation.sol#L5) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/StorageSlot.sol#L5) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/Strings.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/introspection/ERC165.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/introspection/IERC165.sol#L4) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/math/SafeCast.sol#L5) - [^0.8.20](lib/openzeppelin-contracts/contracts/utils/math/SignedMath.sol#L4)

lib/openzeppelin-contracts/contracts/interfaces/draft-IERC6093.sol#L3

## too-many-digits

Impact: Informational Confidence: Medium - [ ] ID-69 Math.log2(uint256) uses literals with too many digits: - r = r | byte(uint256,uint256)(x » r,0x0000010102020202030303030303030300000000000000000000000000000000)

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L612-L651