# 543.https://stackoverflow.com/questions/70189765/how-to-random-mint-a-static-number-of-nft

**T:**How to random mint a static number of NFT?

**Q:**I'd like to mint these amount of tokens:  
  
200 super300 rare500 common  
  
But the mint process needs to be random, you can get a (super, rare, or common) but at the end of the process, it should be minted the same amount of 200 super, 300 rare, and 500 common.  
  
The following code does the random but the final amount of tokens will be different from the beginning:  
  
 function safeMint(address to) public onlyOwner { require(\_tokenIdCounter.current() < totalSupply(), "There's no token to mint."); require(mintCnt[msg.sender] < maxMintCntPerAddress, "One address can mint 1 tickets."); if(mintPrice > 0) { require(mintPrice == msg.value, "Mint price is not correct."); address payable \_to = payable(serviceAddress); \_to.transfer(mintPrice); } uint randomNumber = random(expectedTokenSupply - \_tokenIdCounter.current()); for (uint256 i = 0; i < \_tokenMetadata.length; i++) { if(\_tokenMetadata[i].amount <= randomNumber) { \_safeMint(to, \_tokenIdCounter.current()); \_setTokenURI(\_tokenIdCounter.current(), \_tokenMetadata[i].uri); \_tokenIdCounter.increment(); break; } } } function random(uint maxValue) internal returns (uint) { return uint(keccak256(abi.encodePacked(block.timestamp, msg.sender, \_tokenIdCounter.current()))) % maxValue; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**What determines what the minted token will be - super, rare or common?

1 **Answer**

**A1:**First don't use block.timestamp or any block or blockchain data as a source of randomness, because it will cause the "randomness" be predictable or possible to be manipulated by minners, try with chainlink as a source of randomness, they have a good examples in their docs, if you want to have a fixed supply of each type of tokens you can have 3 variables to know how much of each one have been minted, and when you got the random number and all that you need you just need to apply some math, in this case you want the tokens to be 20% of super, 30% of rare and 50% of common, you only have to do the math you need to decide wich one will be minted, and in case of that type has already reach is max supply what will happend?

**C1:**Yes, I know about the random problem and Chainlink, but it also add costs for this project it isn't really required.