```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts/utils/math/SafeMath.sol";

contract MyToken {
    using SafeMath for uint256;

    string public name;
    string public symbol;
    uint256 public totalSupply;

    mapping(address => uint256) public balances;

    constructor(string memory _name, string memory _symbol, uint256 _totalSupply) {
        name = _name;
        symbol = _symbol;
        totalSupply = _totalSupply;
        balances[msg.sender] = _totalSupply;
    }

    function mint(address _recipient, uint256 _amount) public {
        require(_recipient != address(0), "Invalid recipient address");
        totalSupply = totalSupply.add(_amount);
        balances[_recipient] = balances[_recipient].add(_amount);
    }

    function burn(address _sender, uint256 _amount) public {
        require(_sender != address(0), "Invalid sender address");
        require(balances[_sender] >= _amount, "Insufficient balance");
        totalSupply = totalSupply.sub(_amount);
        balances[_sender] = balances[_sender].sub(_amount);
    }
}
```