PROGRAM:

```
pragma solidity ^0.5.0;

contract Election {
    struct Candidate {
        Uint id;
        string name;
        Uint voteCount;
    }

    Mapping (address => bool) public voters;

    Mapping (uint => Candidate) public candidates;

    uint public candidatesCount;
    uint public votersCount;


    event votedEvent (
        uint indexed _candidateId
    );

    constructor () public {
```

```
    for (uint i = 0; i<15; i++) {
        addCandidate("Candidate"+uintToString(i));
    }
}
function uintToString (uint _v) public view returns (string memory str) {
    uint maxlength = 100;
    bytes memory reversed = new bytes(maxlength);
    uint i = 0;
    while (_v != 0) {
        uint remainder = _v % 10;
        _v = _v / 10;
        reversed[i++] = byte(48+remainder);
    }
    bytes memory s = new bytes(i + 1);
    for (uint j = 0; j <= i; j++) {
        s[j] = reversed[i - j];
    }
    str = string(s);
}
function addCandidate (string memory _name) public {
    candidatesCount ++;
    candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
```

```
    }

    function vote (uint _candidateId) public {

        require(!voters[msg.sender],"you have already voted");



        require(_candidateId > 0 && _candidateId <= candidatesCount,"candidate
dowsn't exist");



        voters[msg.sender] = true;

        votersCount++;



        candidates[_candidateId].voteCount ++;



        emit votedEvent(_candidateId);

    }

}
```