```
---------------------------------
            PROGRAM
---------------------------------

Program Elevator

    do while (power == "on"){

                do while (calls == 0){
                            updateCalls();
                }

                calculateNextFloor();
                moveElevator()
                in-&-Out();
                updateCalls();

                bringElevatorBack();
    }
EDN
```

```
---------------------------------
            VARIABLES
---------------------------------
var calls = [0, 0, 0, 0, ....., 0, 0, 0];
var currentfloor = 0;
var nextFloor = 0;
var floorsBtween = 0;
var direction = "up";
var power = "on";
var upCallRequest  = "no";
var downCallReqest = "no";
```

```
--------------------------------
            FUNCTIONS
--------------------------------

function updateCalls(){

    if(upCallRequest   == received) {
                var Request = floor wher this call came from;
                calls.UpRequest = 1;
    }

    if(downCallReqest == received) {
                var Request = floor wher this call came from;
                calls.UpRequest = 1
    }
}




function calculateNextFloor(){

    // we will keep the Call wit a FIFO system

    if (nextFloor > currentfloor){
                direction = " Up "
                floorsBtween = nextFloor - currentfloor;
    }

    if (nextFloor < currentfloor){
                direction = "Down"
                floorsBtween = currentfloor - nextFloor;
    }
}




function moveElevator(){

    if((nextFloor == currentFloor ){
                in-&-Out();
    }

    if(direction == " up " && floorsBtween <   3){
```

```
            goUpSlow(n);
            slowSpeed(n);
            stop(n);
    }
    if(direction == " up " && floorsBtween >=  3){
            goUpFullSpeed(n);
            slowSpeed(n);
            stop(n);
    }

    if(direction == "down" && floorsBtween <   3){
            goDownSlow(n);
            slowSpeed(n);
            stop(n);
    }
    if(direction == "down" && floorsBtween >=  3){
            goDownFullSpeed(n);
            slowSpeed(n);
            stop(n);
    }
}




function In-&-Out(){
                openElevatorDoor();
                deletethisFloorfromCalls();
                waith(13);
                obstaclesonDoorsCheck();
                closeElevatorDoor()
}




function wait(t){
    if (close door button is push): break;
    if (open door button is push): restart the wait time;
}




function stop(n){
```

```
    var n = floorRequested;
    checkifinbtweenfloors();
    if (currentfloor == floorRequested){
            break;
    else: adjust and then break;
    }
}



function bringElevatorBack(){
    if (calls == 0){    // no requested call on hand
            if(currentfloor = 0){
                    stop();
                    break;
            }

            else{
                    var Request = 0;
                    calls.UpRequest = 1;
                    moveElevator();
                    break;
            }
    }
}



function obstaclesonDoorsCheck(){
            if (there is something between doors){
                    openElevatorDoor();
                    speakers = "Object obstructing doors";
                    wait(3);
            }

}



function deletethisFloorfromCalls(){
            calls.currentfloor = 0;
}
```

```
function openElevatorDoor(){
    open elevator doors;
}



function closeElevatorDoor(){
    dowhile(doors are closing){
            obstaclesonDoorsCheck();
    }
}



function goUpFullSpeed(n){
    n = floorsBtween;
    move full speed "n" - 3 floors;
}



function goUpSlow(n){
    move slow speed reminder floors;
}



function goDownFullSpeed(n){
    n = floorsBtween;
    move full speed "n" + 3 floors;
}



function goDownSlow(n){
    move slow speed reminder floors;
}



function slowSpeed(){
    went entering the final flour:
            comence deceleration to speed "0";
}



function checkifinbtweenfloors(){
```

```
    check building marks
    if(BuildingMark = 0){
            break;
    }
    else{
            adjust towars "0";
    }
}
```