```
// define the SpMV wrapper interface
define_ops(SpMV)
{
    // obtain A, x, b from struct Array
    swFloat *diag = accessArray(vertexData, 0);
    swFloat *x = accessArray(vertexData, 1);
    swFloat *b = accessArray(vertexData, 2);
    swFloat *upper = accessArray(frontEdgeData, 0);
    swFloat *lower = accessArray(backEdgeData, 0);
    swInt ivertex, iedge, idim, vertexNum, edgeNum;

    // Computation on diagonal elements
    dims = getArrayDims(vertexData, 0);
    vertexNum = getArraySize(vertexData);
    for(ivertex=0; ivertex<vertexNum; ivertex++)
    {
        for(idim=0; idim<dims; idim++)
        {
            b[ivertex*dims+idim]
                = diag[ivertex*dims+idim] * x[ivertex*dims+idim];
        }
    }

    // Computation on upper triangle
    dims = getArrayDims(frontEdgeData, 0);
    edgeNum = getArraySize(frontEdgeData);
    for(iedge=0; iedge<edgeNum; iedge++)
    {
        for(idim=0; idim<dims; idim++)
        {
            b[owner[iedge]*dims+idim]
                = upper[iedge*dims+idim] * x[neighbor[iedge]*dims+idim];
        }
    }

    // Computation on lower triangle
    dims = getArrayDims(backEdgeData, 0);
    edgeNum = getArraySize(backEdgeData);
    for(iedge=0; iedge<edgeNum; iedge++)
    {
        for(idim=0; idim<dims; idim++)
        {
            b[neighbor[iedge]*dims+idim]
                = lower[iedge*dims+idim] * x[owner[iedge]*dims+idim];
        }
    }
}

```