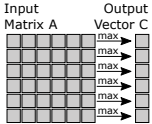
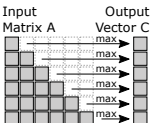
A. Full matrix

```
for (i=0; i<N; i++){
  C[i]=A[i][0];
  for (j=1; j<N; j++)
    C[i] = max(C[i],A[i][j]);
}
```



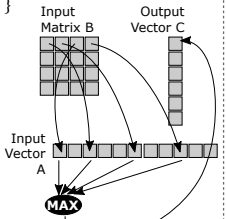
B. Lower triangular

```
for (i=0; i<N; i++){
  C[i]=A[i][0];
  for (j=1; j<i+1; j++)
    C[i] = max(C[i],A[i][j]);
}
```



C. Indirect pattern

```
for (i=0; i<N; i++){
  C[i]=A[B[i][0]];
  for (j=1; j<N; j++)
    C[i] = max(C[i],A[B[i][j]]);
}
```



D. UVE implementation (all cases)

```
  u0 «« input stream A
  u1 »» output stream C
kernel:     config input stream :: u0 «« A[...]
            config output stream :: u1 »» C[...]
.next_line: ; load first block of data
            vectormove          u5,u0
            ; go to hmax if line has no more elements
            b.dim0_complete     u0,.hmax
            ; else sweep through the whole line
.loop:      vectormax           u5,u5,u0
            b.dim0_not_complete u0,.loop
            ; compute max across vector elements
.hmax:      horizontal_max      u1,u5
            ; loop until end of stream/matrix
            b.not_complete      u0,.next_line
```