Chapitre 5 : OpenCL

5.1 Comment faire la transposition des éléments d'une matrice b[y][x] = a[x][y] tout en évitant les accès mémoire non consécutifs?

5.2 Suggérez différentes optimisations pour la fonction OpenCL suivante qui effectue la multiplication matricielle Y=AX où X et Y sont des vecteurs et A est une matrice formée uniquement d'éléments non nuls dans certaines diagonales. Le nombre de ces diagonales est diags et la position de départ (dans la première rangée) de ces diagonales est fournie dans offsets. La position est -(n-1) pour la diagonale qui ne contient que l'élément de première colonne dernière rangée, 0 pour la diagonale qui va de (0,0) à (n-1,n-1), et (n-1) pour la diagonale qui ne contient que le dernier élément de la première rangée.

La matrice A est stockée de manière compacte, une rangée par diagonale non nulle. Pour accéder les éléments de la rangée 0, il faut prendre l'élément 0 de chaque diagonale, donc les éléments de la colonne 0 dans la matrice compacte contenant une diagonale par rangée.

Un work item est le calcul d'un élément dans le vecteur de résultat.

```
__kernel
void dia_spmv(__global float *A, __const int rows,
              __const int diags, __global int *offsets,
              __global float *x, __global float *y) {
  int row = get_global_id(0);
  float accumulator = 0;
  for(int diag = 0; diag < diags; diag++) {
    int col = row + offsets[diag];
    if ((col >= 0) && (col < rows)) {
      float m = A[diag*rows + row];
      float v = x[col];
      accumulator += m * v;
    }
  }
  y[row] = accumulator;
}
```