Jared Hogan u1350778

Q1.

a.
```
for(i=0; i<512; i++)
for (k=0; k<i; k++)
for (j=0; j<512; j++)
C[i][j] += A[i][k]*B[k][j];
```
b.
```
for(k=0; k<512; k++)
for (i=k; i<512; i++)
for (j=0; j<512; j++)
C[i][j] += A[i][k]*B[k][j];
```

Q2.

a.
```
for (i=0; i<512; i++)
for (j=0; j<512; j++)
for(kr=0; kr < j mod 3;kr++)
        C[i][] += A[i][kr]*B[kr][j];
for (k=3; k<j-3; k+=3){
C[i][j] += A[i][k]*B[k][j];
C[i][j] += A[i][k+1]*B[k+1][j];
C[i][j] += A[i][k+2]*B[k+2][j];}
```

b.
```
for (i=0; i<512; i++)
for(jr=0; jr < 512 mod 3;jr++)
        for (k=0; k<jr; k++)
                C[i][jr] += A[i][k]*B[k][jr];
for (j=3; j<512; j+=3)
for(kr=0; kr < j mod 3;kr++)
        C[i][] += A[i][kr]*B[kr][j];
for (k=3; k<j-3; k+=3){
C[i][j] += A[i][k]*B[k][j];
C[i][j] += A[i][k+1]*B[k+1][j];
C[i][j] += A[i][k+2]*B[k+2][j];
C[i][j+1] += A[i][k]*B[k][j+1];
C[i][j+1] += A[i][k+1]*B[k+1][j+1];
C[i][j+1] += A[i][k+2]*B[k+2][j+1];
C[i][j+2] += A[i][k]*B[k][j+2];
C[i][j+2] += A[i][k+1]*B[k+1][j+1+2];
C[i][j+2] += A[i][k+2]*B[k+2][j+2];}
```

Q3.

a. j and k loops can be unrolled but no others as there would make the dependency graph lexicographically negative.
b. The valid permutations of tijk are: tijk, tikj and tkij as other wise one of the dependency graphs will be lexicographically negative.
c. Full tiling is not valid but partial tiling is possible with the j and k loops.
d. The j and k loops can be parallelized. The others can not because that would lead to a data dependency error as they carry the dependency.

Q4.

a. Suppose there are two ways to write to the same element. (i1, j1) and (i2, j2). By the code we know that the loop will access are A[i1+1][j1-1] and A[i2+1][j2-1]. Thus to access the same element i1+1 = i2+1 and j1-1=j2-1. Simplified i1=i2 and j1=j2 thus a contradiction so we have no output dependencies.

b. Consider the first possible vector let the write in each iteration be (wi,wj) and the read to be (ri,rj). For the same element to be accessed iw+1=ri and wj-1=rj. For a flow dependency the vector is (ri-wi, rj-wj) = (iw+1-iw,wj-1,rj)=(1,-1). For the other possible vector let the write in each iteration be (wi,wj) and the read to be (ri,rj). For the same element to be accessed iw+1=ri+1 and wj-1=rj-2. For a flow dependency the vector is (ri-wi, rj-wj)=(wi-wi,wj+1-wj)=(0,1). So there are two flow dependencies of (1,-1) and (0,1)

c. Consider the first possible vector let the write in each iteration be (wi,wj) and the read to be (ri,rj). For the same element to be accessed iw+1=ri and wj-1=rj. For an anti-dependency the vector is (wi-ri, wj-rj) = (iw-iw-1,wj-wj+1)=(-1,1). For the other possible vector let the write in each iteration be (wi,wj) and the read to be (ri,rj). For the same element to be accessed iw+1=ir+1 and wj-1=rj-2. For an anti-dependency the vector is (wi-ri, wj-rj)=(wi-wi,wj-wj-1)=(0,-1). So there are no anti-dependencies both possible vectors are negative.