Question 4

To find the square which contain the maximum number of apples. We need to find all possible $n * n$ squares in $4n * 4n$ area and then find which square contain the maximum number of apples. Consider the orchard as a big matrix (**Orchard[][]**) and its element represent the number of apples on each tree.

To do this, the following steps would be applied

1. Compute the sum of the adjacent n numbers on each row. store the result into a relatively small matrix. **(array[][])**. For reducing processing time, instead of doing repeated calculation, the algorithm would simply add the next new number and subtract the old one when compute the sum of next adjacent n numbers.
   Pseudo code:

```
orchard[4n][4n]       // it's the matrix of orchard
array[n][3n+1]        // the small matrix which has the element that
                      //represented the sum of adjacent number of
                      // orchard[][]
for i = 0; i < 4n ; i++ :
    for j = 0; j < n; j ++ :
        array[i][0] = array[i][0] + orchard[i][j]      // init the
                                                       //first element in array
    for j = n; j < 4n; j ++:
        array[i][j-n+1] = array[i][j-n] + orchard[i][j] - orchard[i][j-n]    // add new one the subtract the
                                                                              // old element
```

   The time complexity for generating array[][] would be $O(n^2)$

2. Now we have a matrix whose element represented the sum of n adjacent number of each **row** in orchard[][]. To get the result of sum of square, we just needs to add the calculated the sum of n adjacent numbers of each **column in array[][].** This could be done in the same way as step1 but has a few differences with the index i and j(because now we compute the column). During this process, the algorithm will also record the maximum value and the index of the value.
   The time complexity of this step is also $O(n$^2).
3. Finally, we are able to find the square which contain the maximum number of apples in orchard by the index.