In this exam, you are asked to complete the function definitions to sort the given array **arr** with **descending or ascending** order with respect to given boolean parameter **ascending** by using **multiDigitRadixSort**.

multiDigitRadixSort is a derivative of radix sort that uses multi digit counting sort as a stable sort function. In each counting sort, you should use the number of digits given by **groupSize** parameter.

For example, arr[4] = {4367, 1234, 7890}. If groupSize is 1, then the function is exactly equal to classical radix sort. That means, the first counting sort will use the last digits of the numbers which are 7, 4 and 0. If groupSize is equal to 2, then the first counting sort will use the last two digits of the numbers which are 67, 34 and 90.

If groupSize is not a multiplier of maxDigitLength, you should use maximum remaining digit count in counting sort function. For example, groupSize is 3 and maximum number in the array is 65789 , that means maxDigitLength is 5. In the first counting sort, you should use least significant 3 digits, in the second counting sort you should use most significant 2 digits.

You should return the number of iterations during counting sort.(The counting sort algorithm in our book has 4 different loops. Different than the algorithm for Counting Sort in your book, initialize count array as int C[k] = {0} and use the fourth loop for copying the array back. Otherwise the return value of the function will not be evaluated as correct.)

Do not forget to initialize your count array to zero. Otherwise, you may see some unstable behaviours.