2066年、株式会社VTIジャパンは創立50周年を迎えました。会社の取締役会は過去50年間で最も従業員が多い年とその年の従業員数を把握したいと人事チームに伝えました。

しかし、人事側に聞いてみたら、毎年の従業員の人数の集計データはありませんでした。50年間で記録されたものは各従業員の入社の年と退社の年のデータだけでした。

そのため、あなたはこのデータを計算する関数を作成することを指名されました。入社退社データ量は数十万レコードを上る可能性があるため、できるだけ処理が速いアルゴリズムで結果を求めることが考慮の課題です。

関数の入力は（inYearArr、outYearArr）の二つの配列とします。  
社員xはinYearArr[x]の年で入社し、outYearArr [x]の年で退社します。  
年は2桁の数字として保存されます（2016は16として保存されます）。計算時点で会社に滞在している従業員は、OutYearArrでの値は99として保存されます。

関数の出力は、2つの数値の配列です。最初の数値は今までの従業員最大の数であり、2番目の数値は従業員数が最大の年で、16〜66の範囲内の数字とします。

※注意：  
・複数の年の従業員数が同じく最大数になる場合は一番小さい年のものを出力してください。  
・同じ年で入社退社する従業員は、その年の従業員の総数にもカウントしてください。

例1:  
入力：  
inYearArr  
[16,18,23,33]  
outYearArr  
[99,28,23,99]

出力:  
[3,23]

例2:  
入力：  
inYearArr  
[16,16,51,16,15,32,52,33]  
outYearArr  
[99,99,99,23,99,99,99,34]

出力:  
[6,52]

例3:  
入力：  
inYearArr  
[16,16,17,23,23,18,39,24]  
outYearArr  
[99,99,23,24,99,24,52,99]

出力:  
[6,23]

**[execution time limit] 3 seconds (java)**

**[input] array.integerinYearArr**

入社年の配列

**[input] array.integeroutYearArr**

退社年の配列

**[output] array.integer**

