⑤ SORT DATES

- Hume array given hai (Dates) ka [DDMMYY] Formate me!
- Hume Radix-Sort technique use krengj!

5
12041996
20101996
05061997
12041989
11081987

- Sabse phele Days pe sort krengy.
- Fir months pe sort aur last me year pe sort. KYUNKI Year ki prelecdence months se jyadahai aur month ki Days se jyada hai!

SORTING ON Dates

12 04 1996
20 10 1996
05 06 1997
12 04 1989
11 08 1987

→ DAYS

SORTING ON MONTHS

05 06 1997
11 08 1987
12 04 1996
12 04 1989
20 10 1996

→ MONTHS

SORTING ON YEAR

12 04 1996
12 04 1989
05 06 1997
11 08 1987
20 10 1996

→ YEAR

FINAL SORTED ARRAY

11 08 1987
12 04 1989
12 04 1996
20 10 1996
05 06 1997

∴ HOW TO GET DAY OF THE MONTH

12041996 / 1000000 = 12 % 100
= 12

∴ (date / 1000000) % 100

HOW TO GET MONTH OF THE YEAR

12041996 / 10000 = 1204 % 100
= 04

∴ (date / 10000) % 100

HOW TO GET YEAR FROM THE DATE

12041996 / 1 % 10000
= 1996

∴ (date / 1) % 10000

- SORTING will be based on RADIX SORT, which is based on COUNT SORT, which does STABLE SORTING!

```
public static void sortDates (String [] arr) {
    countSort(arr, 1000000, 100, 32);   // (0-31) ] Date sort
                   divisor  mod  Range of freq array
    countSort(arr, 10000, 100, 13);     // (0-12) ] Month sort
                   divisor  mod
    countSort(arr, 1, 10000, 2501);     ] Year sort

}
```

```
public static void countSort (String [] arr, int divisor,
        data array will be passed in string!        int mod, int range).
{
  int [] farr = new int [range];
  String [] ans = new String [arr.length];
  // collecting frequency
  for (int i=0; i<arr.length; i++)
  { int val = Integer.parseInt (arr[i]);
    farr [val]++;   string to integer
  }
  // convert the farr to psa (prefix-sum array)
  for (int i=1; i<farr.length; i++)
  { farr[i] += far[i-1];
  }
  // fill the answer array
  for (int i = arr.length -1; i>=0; i--)
  { int val = Integer.parseInt ((arr[i] / div)% mode);
    int pos = farr [val];
    ans [pos-1] = arr [i];
    farr [val]--;
  }
  // fill the original array
  for (int i=0; i<ans.length; i++) {
      arr [i] = ans [i];
    }
  }
public static void print (string [] arr){
  for (int i=0; i<arr.length; i++) {
        System.out.print (arr [i]);
  }
  }
public static void main (string [] args) {
  Scanner s = new Scanner (System.in);
  int n = s.nextInt ();
  String [] arr = new String [n];
  for (int i = 0; i<n; i++) {
        String str = s.next ();
        arr [i] = str;
    }
  sortDates (arr);
  print (arr);
}
```

Time complexity = O(n)
Space complexity = O(n)