**Slot 1:**

1. **Sort Values in Range**   
     
   The program must accept N integers and sort only the integers that are in the range (inclusive) X to Y. Other integers (which are not within the range) positions must not be changed. In case Y is smaller than X, consider the range (inclusive) as Y to X.  
     
   **Boundary Condition(s):**  
   2 <= N <= 100  
   1 <= Each integer value <= 10^5  
   1 <= X, Y <= 10^5  
     
   **Input Format:**  
   The first line contains N, X and Y.  
   The second line contains N integer values.  
     
   **Output Format:**  
   The first line contains N integer values.  
     
   **Example Input/Output 1:**  
   Input:  
   5 99 10  
   56 500 25 20000 44  
     
   Output:  
   25 500 44 20000 56
2. **Add K to Multiples**   
     
   The program must accept N integers and add the integer value K to the values that are multiples of K. Then the program must print the revised N integer values as the output.  
     
   **Boundary Condition(s):**  
   2 <= N <= 100  
   1 <= Each integer value <= 10^5  
   1 <= K <= 10^5  
     
   **Input Format:**  
   The first line contains N.  
   The second line contains N integer values.  
   The third line contains K.  
     
   **Output Format:**  
   The first line contains N integer values.  
     
   **Example Input/Output 1:**  
   Input:  
   5  
   56 500 25 20000 44  
   100  
     
   Output:  
   56 600 25 20100 44  
     
   Explanation:  
   Here K=100. As 500 and 20000 are multiples of K=100, K is added to them.
3. **Remove Alternate Repeated Characters**   
     
   The program must accept a string S and remove alternate repeated characters. Then the program must print the revised string value as the output.  
     
   **Boundary Condition(s):**  
   2 <= Length of S <= 100  
     
   **Input Format:**  
   The first line contains S.  
     
   **Output Format:**  
   The first line contains a string value.  
     
   **Example Input/Output 1:**  
   Input:  
   queenbee  
     
   Output:  
   quenbe  
     
   **Example Input/Output 2:**  
   Input:  
   quenebepe  
     
   Output:  
   quenbep

**Slot 2:**

1. **Numbers with Two Digits**   
     
   Two digits A, B are passed as the input. The program must print all the distinct positive integers that can be formed with the digits A and B which are less than or equal to N in ascending order.  
     
   Boundary Condition(s):  
   1 <= N <= 10^7  
     
   Input Format:  
   The first line contains A and B separated by a space.  
   The second line contains N.  
     
   Output Format:  
   The first line contains integer values separated by a space.  
     
   Example Input/Output 1:  
   Input:  
   5 6  
   100  
     
   Output:  
   5 6 55 56 65 66  
     
     
   Example Input/Output 2:  
   Input:  
   5 1  
   9999  
     
   Output:  
   1 5 11 15 51 55 111 115 151 155 511 515 551 555 1111 1115 1151 1155 1511 1515 1551 1555 5111 5115 5151 5155 5511 5515 5551 5555
2. **Largest Two Digit Odd Integer**   
     
   The program must accept an integer N and print the largest possible two digit odd integer I that can be formed using the digits in N. If no such odd integer can be formed, the program must print -1 as the output.  
     
   **Boundary Condition(s):**  
   11 <= N <= 10^15  
     
   **Input Format:**  
   The first line contains N.  
     
   **Output Format:**  
   The first line contains I or -1.  
     
   **Example Input/Output 1:**  
   Input:  
   120  
     
   Output:  
   21  
     
   **Example Input/Output 2:**  
   Input:  
   6018529  
     
   Output:  
   95
3. **Interchange First Last N Characters**   
     
   The program must accept a string S and interchange the first and last N characters. Then the program must print the revised string value.  
     
   **Boundary Condition(s):**  
   2 <= Length of S <= 100  
   1 <= N <= (Length of S)/2  
     
   **Input Format:**  
   The first line contains S.  
   The second line contains N.  
     
   **Output Format:**  
   The first line contains a string value.  
     
   **Example Input/Output 1:**  
   Input:  
   abcdefgh  
   3  
     
   Output:  
   fghdeabc  
     
   **Example Input/Output 2:**  
   Input:  
   mango  
   2  
     
   Output:  
   gonma

Slot 3:

1. **Pattern with D Digits**   
     
   A string S is passed as the input. A pattern string P containing only alphabets is also passed as the input. The program must print the non-overlapping count C of pattern P in S with the condition that at least D digits must be embedded within the pattern P in the string S.  
     
   **Boundary Condition(s):**  
   2 <= Length of S <= 10^5  
   2 <= Length of P <= 10^4  
   0 <= D <= 50  
     
   **Input Format:**  
   The first line contains S.  
   The second line contains P.  
   The third line contains D.  
     
   **Output Format:**  
   The first line contains C.  
     
   **Example Input/Output 1:**  
   Input:  
   ma91ngopluckedbymanaskedbym9a78777n7agercay12ma56444n  
   man  
   2  
     
   Output:  
   3  
     
   Explanation:  
   The three occurrences are  
   ma91n  
   m9a78777n  
   ma56444n  
     
   man in bymanasked is not considered as there are no 2 digits between m and n.
2. **Return Student Comparator**   
     
   Please fill in the missing lines of code to return a Comparator which will sort N Student objects based on marks. If the marks are same for a given pair of students, then the sorting will be done based on the name.  
     
   **Boundary Condition(s):**  
   2 <= N <= 100  
     
   **Input Format:**  
   The first line contains N.  
   The next N lines contain the name and marks of the students separated by a space.  
     
   **Output Format:**  
   The first N lines contain the sorted student details with each line containing the name and marks of the students separated by a space.  
     
   **Example Input/Output 1:**  
   Input:  
   4  
   Rahul 50  
   Mitra 60  
   Arun 50  
   Suresh 55  
     
   Output:  
   Arun 50  
   Rahul 50  
   Suresh 55  
   Mitra 60
3. **Largest Second Largest Digits Sum**   
     
   The program must accept N integers and for each integer value must print the sum of the largest and the second largest digit. If the integer value is a single digit value or does not contain at least two distinct digits, then just print the largest digit.  
     
   **Boundary Condition(s):**  
   1 <= N <= 1000  
   1 <= Each integer value <= 10^9  
     
   **Example Input/Output 1:**  
   Input:  
   5  
   7056 6 999 9995 800  
     
   Output:  
   13 6 9 14 8

Slot 4:

1. **K Reverse Count**   
     
   The program must accept an integer N and another two digits value K. The program must print the number of times the reversed value of K is present in N.  
     
   **Boundary Condition(s):**  
   1 <= N <= 10^16  
   10 <= K <= 99  
     
   **Example Input/Output 1:**  
   Input:  
   1959699 90  
     
   Output:  
   4  
     
   Explanation:  
   Reverse of K=90 is 9 which is present 4 times in N.  
     
   **Example Input/Output 2:**  
   Input:  
   545554545546054 45  
     
   Output:  
   5
   1. **Define function getLastTwoDigits**   
        
      Please define the function**getLastTwoDigits** so that the function returns an array of size N where the values are last two digits of the elements in Array arr.  
        
      **Boundary Condition(s):**  
      2 <= N <= 1000  
      10 <= Each integer value <= 10000  
        
      **Input Format:**  
      The first line contains N.  
      The second line contains N integer values separated by a space.  
        
      **Output Format:**  
      The first line contains the original array element and the last two digits separated by a hyphen.  
        
      **Example Input/Output 1:**  
      Input:  
      5  
      1054 50 75102 1500 205  
        
      Output:  
      1054-54 50-50 75102-02 1500-00 205-05

**3. Merge Adjacent Same Unit Digit**   
  
The program must accept an array of N integers and merge non zero adjacent integers having same unit digits by replacing the left value with the sum and replacing the right value with a 0. The array elements with zero values are considered non-existent and the program must continue merging by comparing the remaining adjacent values. The program must print the revised array as the output.  
  
**Boundary Condition(s):**  
1 <= N <= 1000  
1 <= Each integer value <= 100000  
  
**Input Format:**  
The first line contains N.  
The second line contains N integer values separated by a space.  
  
**Output Format:**  
The first line contains the integer values separated by a space.  
  
**Example Input/Output 1:**  
Input:  
10  
16 54 54 48 72 52 18 34 42 92  
  
Output:  
244 0 0 0 0 52 186 0 0 0  
  
Explanation:  
The merging occurs in the below order.  
54 54 --> 108 0  
108 48 --> 156 0  
16 156 --> 174 0  
174 72 --> 244 0  
Now the array is 244 0 0 0 0 52 18 34 42 92  
42 92 --> 134 0  
34 134 --> 168 0  
18 168 --> 186 0  
Now the array is  
244 0 0 0 0 52 186 0 0 0

Slot 5:

1. **Sum Divisible First or Last**   
     
   The program must accept an array of N integers and print Yes if the sum of the array values is divisible by either the first value or the last value.  
     
   **Boundary Condition(s):**  
   2 <= N <= 1000  
   1 <= Each integer value <= 10000  
     
   **Input Format:**  
   The first line contains N.  
   The second line contains N integer values separated by a space.  
     
   **Output Format:**  
   The first line contains either Yes or No  
     
   **Example Input/Output 1:**  
   Input:  
   5  
   10 50 75 15 25  
     
   Output:  
   Yes  
     
   Explanation:  
   The sum of the array values is 175 and is divisible by 25.  
     
   **Example Input/Output 2:**  
   Input:  
   5  
   10 50 75 15 22  
     
   Output:  
   No
2. **First Value Decreasing Sequence**   
     
   The program must accept N string values and check if each of the string value contains sequence of integers in decreasing order. If yes, the program must print the first integer value. Else the program must print NO as the output.  
     
   **Boundary Condition(s):**  
   1 <= N <= 10  
   2 <= Length of each string <= 10^4  
     
   **Input Format:**  
   The first line contains N.  
   The next N lines contain the N string values.  
     
   **Output Format:**  
   The first N lines contain the starting integer value or the string value NO.  
     
   **Example Input/Output 1:**  
   Input:  
   3  
   999897969594  
   99989799594  
   1000999998997996995994  
     
   Output:  
   99  
   NO  
   1000  
     
   Explanation:  
   999897969594 - Contains the integer values from 99 to 94. Hence the starting value 99 is printed.  
   99989799594 - Does not contain integer values in decreasing sequence.  
   1000999998997996995994 - Contains the integer values from 1000 to 994.

|  |
| --- |
| 3. **Alternate Sorting of Numbers** Given an array of N integers, rearrange the array in such a way that the first element is first maximum, second element is first minimum, third element is second maximum, fourth element is second minimum and so on.  **Input Format:** The first line contains the value of the N integers separated by one or more spaces.  **Boundary Conditions:** 4 <= N <= 100  **Output Format:** The N numbers alternately sorted as per the given instructions.  **Example Input/Output 1:** Input: 1 2 3 4 5 6 7  Output: 7 1 6 2 5 3 4  **Example Input/Output 2:** Input: 10 99 44 22 56 63  Output: 99 10 63 22 56 44  **Example Input/Output 3:** Input: 9 5 6 9 3 2 5  Output: 9 2 9 3 6 5 5 |