|  |  |
| --- | --- |
|  |  |

**Faculty of Technology and Engineering**

**U & P U. Patel Department of Computer Engineering**

Date: 02 / 25 / 2022

**Practical List**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Academic Year | : | 2021-22 | Semester | : | 4 |
| Course code | : | CE259 | Course name | : | Programming in Python |

**Note: Practical List is for Students. We need to cover concept require to implement respective practical**

|  |  |
| --- | --- |
| **Sr. No.** | **Aim** |
| 7. | Lapindrome is defined as a string which when split in the middle, gives two halves having the same characters and same frequency of each character. If there are odd number of characters in the string, we ignore the middle character and check for lapindrome. For example ***gaga*** is a lapindrome, since the two halves ***ga*** and ***ga*** have the same characters with same frequency. Also, ***abccab***, ***rotor*** and ***xyzxy*** are a few examples of lapindromes. Note that abbaab is NOT a lapindrome. The two halves contain the same characters but their frequencies do not match.  Your task is simple. Given a string, you need to tell if it is a lapindrome.  **Input:**  6  gaga  abcde  rotor  xyzxy  abbaab  ababc  **Output:**  YES  NO  YES  YES  NO  NO |
| Src | n = int(input("Enter no. of test cases: ")) print('Enter',n,'strings:') lst = [] # lst = ['gaga', 'rotor', 'abcde', 'lol', 'papa'] for i in range(n):  a = input()  lst.append(a)  for ip in [i for i in lst]:  ip1 = ip[:len(ip)//2]  if(len(ip)%2 == 0):  x = len(ip)//2  else:  x = len(ip)//2 + 1  ip2 = ip[x:]  lst1 = sorted(list(ip1))  lst2 = sorted(list(ip2))  # print(lst1, lst2)  if lst1 == lst2:  print("YES")  else:  print("NO") |
|  |  |