

CLASS: CSE7345

NAME: Hanspal, Randeep

SMUID: 47812509

QUEST: basicPythonQuest

CODE:

```
# Basic Python Quest  
# When returning lists of values, order is not important unless  
specified
```

```
__STUDENT_ID__ = "47812509"  
__CODING_NAME__ = "RonShell"
```

```
def isSorted(list):  
    isSorted = True  
    #Create for loop to check if list is sorted  
    for i in range(len(list) - 1):  
        if list[i] > list[i + 1]:  
            isSorted = False  
            break  
    if not isSorted:  
        return False  
    else:  
        return True  
  
def isSortedAndUnique(list):  
    #Check if both functions returns unique & sorted list  
    if hasUniqueValues(list) and isSorted(list):  
        return True  
    return False  
  
def hasUniqueValues(list):  
    isUnique = True  
    #for loop to check uniques values  
    for i in range(len(list) - 1):  
        for j in range(i + 1, len(list)):  
            if (list[i] == list[j]):  
                isUnique = False  
                break  
    if not isUnique:  
        break  
  
    if not isUnique:  
        return False  
    else:
```

```

        return True

def genSortedArrayUniqueValues(list):
    # for loop to sort the list
    for j in range(len(list) - 1):
        for i in range(len(list) - 1):
            if list[i] > list[i + 1]:
                list[i], list[i + 1] = list[i + 1], list[i]

    #list created to store unique values
    uniqueList=[]
    #for loop to create unique list from the sorted list
    for li in list:
        if li not in uniqueList:
            uniqueList.append(li)
    return uniqueList

def listToMapTwoByTwo(list):
    # Create a new dictionary
    newDict={}
    bool = True
    for element in list:
        if bool:
            # first element of list is stored in var k
            k = element
            bool = False
        elif not bool:
            #first element is stored as key & second as value
            newDict[k] = element
            bool = True
    return newDict

def wordsInStringToDictWordCount(s):
    # Create a new dictionary
    dict = {}
    # Split the string in list
    splittedList = s.split()
    # for loop to iterate each element
    for i in splittedList:
        if (i not in dict):
            dict[i] = 1
        else:
            dict[i] = dict[i] + 1
    return dict

def reverseWordsInString(string):
    #split the string into list

```

```

splittedStringList = string.split()
# create an empty string
reversedString=""
#for loop to iterate each element from reverse list
for i in splittedStringList[::-1]:
    #append each element
    reversedString+=" " + i
return reversedString.strip()

def genListOfOverlaps(list1, list2):
    # Create a new list
    newlist = []
    # for loop to iterate each element from list1
    for i in list1:
        # check if each element from list1 is present in list2
        if i in list2:
            # if present append the element in new list
            newlist.append(i)
    return newlist

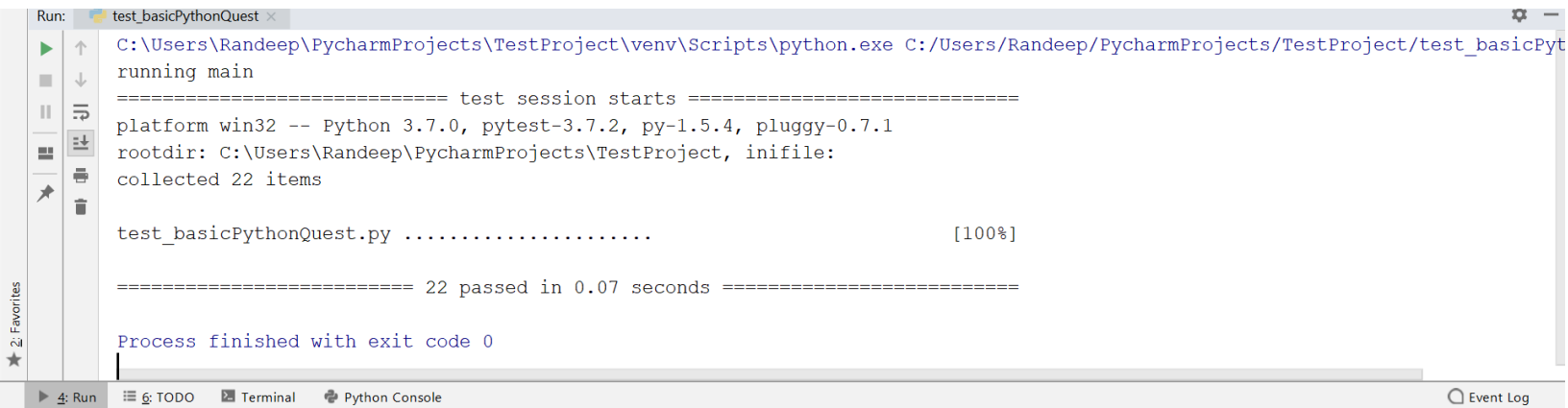
def removeDupsNoSet(list):
    # Create a new list
    newlist = []
    # for loop to remove duplicate
    for i in list:
        if i not in newlist:
            newlist.append(i)
    return newlist

def removeDupsUseSet(mylist):
    # mySet stores the unique values
    mySet=set(mylist)
    # uniqueList stores the values in list
    uniqueList=list(mySet)
    return uniqueList

if __name__ == '__main__':
    print ('ready to go')

```

OUTPUT:



The image shows a PyCharm Run window for a test session. The title bar indicates the file being run is 'test_basicPythonQuest'. The output text shows the execution path, the command used to run the tests, the environment details (platform, Python version, pytest version, etc.), the root directory, and the number of items collected. The test results show that all 22 items passed. The window also includes a sidebar with icons for Run, TODO, Terminal, and Python Console, and a bottom status bar with an Event Log icon.

```
Run: test_basicPythonQuest x
C:\Users\Randeep\PycharmProjects\TestProject\venv\Scripts\python.exe C:/Users/Randeep/PycharmProjects/TestProject/test_basicPyt
running main
===== test session starts =====
platform win32 -- Python 3.7.0, pytest-3.7.2, py-1.5.4, pluggy-0.7.1
rootdir: C:\Users\Randeep\PycharmProjects\TestProject, inifile:
collected 22 items

test_basicPythonQuest.py ..... [100%]

===== 22 passed in 0.07 seconds =====

Process finished with exit code 0
```

Run TODO Terminal Python Console Event Log