Total sheets used in this question- (6)

4/4/2022, 9:00 AM
MCA
SEM-I
OBJECT ORIENTED PROGRAMMING
Course Code- 22340110)
Exam Roll no- 2123475700 9

---

(1) ⟶
(a) ⟶
Ans-

```
def SSD(n):
    """
    It is a recursive function that returns sum of square
    of the digits.
    """
    if n>0:
        digit = n%10    # Last digit
        square = digit**2    # square of the digit
        return square +SSD(n//10) # If n= 6384
                                  then n//10 = 638

    else:
        return 0    # If n>=0 we return 0 as the sum
```

```
def checkHappyNum(number):
    """
    It will return True if number is happy
    otherwise False.
    This process can go to infinite loop, That's
    why we are iterating for 6 times to
    avoid infinite loop.
    """
    for i in range(6):
        number = SSD(number)
        if number == 1:    # The number is happy
            return True

    return False    # Number is not happy


if __name__ == "__main__":
    # Driver Code
    num = int(input(" Enter the number"))
    res = checkHappyNum(num)
    # Here res can be True or False

    if res:
        print(" The given number is HappyNumber")

    else:
        print(" The given number Is not happy number")
```

Output

Enter the number 19
The given number is Happy Number

(b) →

Ans -

```
def mergeLists(lst1, lst2):
    """
    It will return dictionary
    Contains merged number list and
    merged string list.
    First we seperate number and
    string from both lists, sort all of them.
    Then merge number lists in a single
    number list and two string.
    list into single string list.
    """
    res = {}
    lst1_num = []
    lst1_string = []
    lst2_num = []
    lst2_string = []
    for i in lst1:
        if type(i) == str:
            lst1_string.append(i)
    else
```

```
        else:
            lst1_num.append(i)

    for j in lst2:
        if type(j) == str:
            lst2_string.append(j)

        else
            lst2_num.append(j)

lst1_num.sort()          # Sort list 1 of number
lst2_num.sort()          # Sort list 2 of numbers
lst1_string.sort()       # Sort list 1 of string
lst2_string.sort()       # Sort list 2 of string

merged_num = []
i = 0                    # pointer for list 1 of number
j = 0                    # pointer for list 2 of number

while i < len(lst1_num) and j < len(lst2_num):
    if lst1_num[i] <= lst2_num[j]:
        merged_num.append(lst1_num[i])
        i += 1

    else:
        merged_num.append(lst2_num[j])
        j += 1

while i < len(lst1_num):
    merged_num.append(lst1_num[i])
    i += 1
```

```
while j < len(lst2_num):
    merged_num.append(lst2_num[j])
    j += 1

merged_string = []
i = 0
j = 0
while i < len(lst1_string) and j < len(lst2_string):
    if lst1_string[i] <= lst2_string[j]:
        merged_string.append(lst1_string[i])
        i += 1
    else:
        merged_string.append(lst2_string[j])
        j += 1

while i < len(lst1_string):
    merged_string.append(lst1_string[i])
    i += 1

while j < len(lst2_string):
    merged_string.append(lst2_string[j])
    j += 1

res['mergedStr'] = merged_string
res['mergedNum'] = merged_num

return res
```

Driver Code

```
if __name__ == "__main__":
    lst1 = ["Morning", 100, "hey", "Hi", 23, 78,
            "Good", "Evening", -9]

    lst2 = [89, "Rain", "Sunny", "Cloudy",
            91, 107, "Humid"]

    result = mergedLists(lst1, lst2)
    print(result)
```

Output

```
{'mergedStr': ['Cloudy', 'Evening', 'Good', 'Hi',
               'Humid', 'Morning', 'Rain', 'Sunny',
               'hey'],
 'mergedNum': [-9, 23, 78, 89, 91, 100, 107]}
```