Name: SRADHA KEDIA
Date and time of Examination: 9:30am - 12:30pm ; 27/03/2021
Examination Roll no: 20234757053
Name of the Programe: MCA
Semester: Ist
Unique Paper Code: 22340111 01
Title of the Paper - Object Oriented Programming
Email-ID- 200083@cs.du.ac.in
Mobile no.— 8840502121
No. of Pages— 4

2023475053

Question 2 →

(a) Input: Giving the function insert a list [1,2,3,4] / user input & the number to be inserted (50, taken)

Enter a list separated by space: 1 2 3 4 5
Enter the number to be inserted: 50

Output: [1, 50, 2, 50, 3, 50, 4, 50, 5, 50]

```
def insert(_list, x, newl=[]):
    """ Function name - insert
        Arguments - _list, x, newl ;
        newl is taken for convinience to be noted that
        driver function contains only two arguments
        as mentioned in the question
        Returns : a new list where element is inserted
    """
    if len(_list)==0:          # if list is empty
        return newl            # returns the empty list
    else:
        newl.extend([_list[0], x])      # extend the list by adding x
        return insert(_list[1:], x, newl)
                               # recursive call to give
                               elements from 1st index again
```

```
if __name__ == "__main__":  # drive function
    _list = list(map(int, input("Enter a list separated by space: ").split()))
    x = input("Enter the number to be inserted :")

# taking input
# Assumptions : User inputs as the format specified correctly
in order to run the function accurately

p = insert(_list, x)  # calling insert function
print(p)  # print p list
```

(b)
```
'''
Input: Enter a list separated by space: 1 2 2 3 5 2
Enter the value of k : 2
'''

def remelt(_list, k, i=0):
    '''
    name : remelt
    arguments : _list, k, i=0
    Assumption as (a)
    returns : the updated list after removing occurences
    '''

    if len(_list) == i:  # comparing length of list with i
        return _list  # returning the list

    else:
        if _list[i] == k:  # checking ith elt with k to remove
            return remelt(_list[:i] + _list[i+1:], k, i)
                # recursive call
```

2023475053

```
    else:                    # if not matched
        return remelt(_list, k, i+1)
                             # passing with next index

if __name__=="__main__":     # driver function

    _list = list(map(int, input("Enter a list separated by space").
                                split()))
                             # taking input

    k = input("Enter the value of k:")

    remelt(_list, k)         # calling remelt function.

# assumption: User must input as the format
specified to run the function accurately
```