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: 223401101

Title of the Paper - Object Oriented Programming

Email-ID - ~~[illegible]~~ 200083@cs.du.ac.in

Mobile no. — 8840502121

No. of Pages — 4

2023475705 3

Question 3 →
encrypt details are mentioned after the end of encrypt func

```
def encrypt(message):
    encoded = " "       # null string to store encoded
                          message

    for i in message:   # for loop to traverse each
                          character
        if i.isnumeric():   # checking if character is a
                              digit
            s = ((int(i)**2)+5)% 26 +65
                            # getting the ASCII form
                              of a digit
            s = chr(s)      # convert ASCII to character
            encoded += s + " " # storing the encoded
                                 message
        elif i.islower() # checking if character is
                           a lowercase alphabet
            s = (ord(i) - 3)% 26 # returning ASCII value
            s = str(s)  # ASCII to string form
            encoded += s + " " # storing in the encoded
                                 message
        else:
            encoded += i + " " # storing in the encoded
                                 message
    return encoded # returning the encoded message
```

2023475 7053

```
''' The previous one was to Encrypt
   ~~def ( da~~ Returns : encoded
             Argument : message   '''
def decrypt( encoded)

   ''' Argument : encoded ( encrypted message)
       returns : decoded
   '''
   # Assumptions : for Alphabets to be lower case &
   # upper case it is not possible to decrypt correctly
   # Encryption function should ~~to~~ be invervitble .

   # We take as our decryption function will always
   produce upper case alphabets, even if they don't correspond
   to the original text. Because it is not possible to
   uniquely determine the original text from the encrypted
   text.

   decoded = " "
   ~~cap~~ alphabet = (65+6) % 26 # Encrypted value for
                                    character 'A'

   for i in encoded :
       if ord(i) in range (0, 26):
           decoded += chr((ord(c) - 6) % 26 - alphabet) % 26
                                                   + 65)
                                                 # chr of
                                                 calculated
                                                 ~~result~~
       elif ord(~~c~~ i) in range(65, 65+26):
           remainder = ((ord (c) - 65) - 5) % 26
           j = 0                        # flag variable
           while True :
```

2023475 7053

```
    if (j **2) % 26 == remainder:  # check if equal
                                                  to remainder
        decoded += chr(c + 48)   # append in decoded
        break  # break
     j += 1
  else:                    # if not equal to remainder
     decoded += i          # appends as it is

return decoded          # returned decoded message

# driver function

if __name__ == "__main__":
        message = input ("Enter the message")
  encoded = encrypt( message)         # indentation expected
                                        # error at writing time
                                        # kindly consider
  decoded = decrypt (encoded)
                                           # called function encrypt
                                                       & decrypt
  print (" Encrypted message:", encoded)
  print (" Decrypted message:", decoded)
                                   # printing messages.
```