28
33
75
88
99

Date 9-10-2024

Expt. No. ______ Page No. 28

23) Sorting

```
def    Sorting (arr) :
           n = len (arr)
           for  i  in  range (n) :
                  for  j  in  range (0, n-i-1) :
                          if  arr [j] > arr [j+1] :
                               arr [j], arr [j+1] = arr [j+1], arr [j]
l = [10, 99, 33, 20, 28, 88, 75, 18]
Sorting (l)
print (" Sorted Values are : ")
for   i   in   range (len(l)) :
           print (l [i])
```

Teacher's Signature ______

Output :

```
True
False
```

Date 8/10/2024

Expt. No.

21) mail Sending

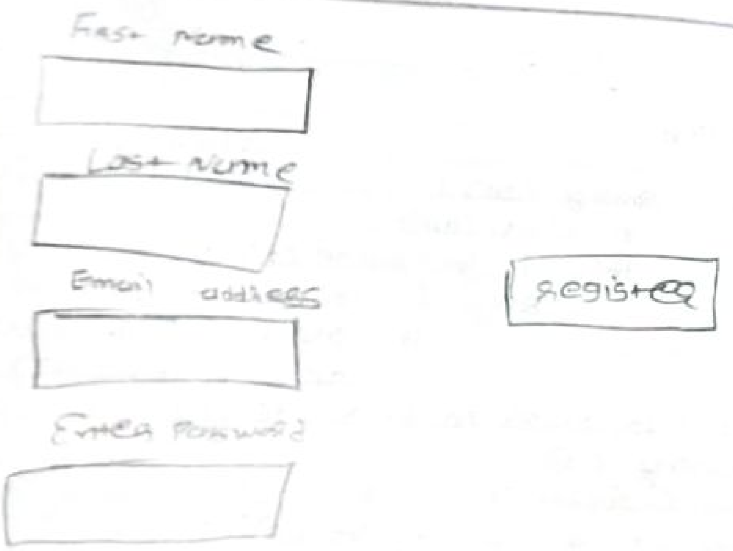
```
import smtplib
s = smtplib.SMTP ('smtp.gmail.com', 587)
s.login ("username", "Password")
message = " Test E-mail"
s.sendmail ("username", "Receivermail", message)
s.Quit ()
```

22) Searching Date = 9/10/2024

```
def search (values, search_for) :
        search_at = 0
        search_res = False
        while search_at < len (values) and search_res is
                if values [search_at] == search_for :
                        search_res = True
                else :
                        search_at = search_at + 1
        return search_res
l = [10, 99, 33, 20, 28, 88, 75, 18]
print (search (l, 18))
print (search (l, 1000))
```

Teacher's Signature

Output -



24) grid layout

```
from tkinter import *
root = Tk()
root. geometry ("250 x 250")
frame = Frame (root)
def on_key_up() :
        name = fname
        frame. pack_forget()
        root. configure (bg = 'green')
        label (root, text = "welcome" + name. get()). pack (pady = 30)
frame. pack()
Label (frame, text = "First name"). grid (row = 7, column = 0, padx = 5)
fname. = Entry (frame)
fname. grid (row = 10, column = 0, padx = 5)
Label (frame, text = "Last name"]. grid (row = 12, column = 0, padx = 5)
lname = Entry (frame)
lname. grid (row = 15, column = 0, padx = 5)
Label (frame, text = "E-mail Address"). grid (row = 17, column = 0, padx = 5
Email = Entry (frame)
email. grid (row = 20, column = 0, padx = 5)
Label (frame, text = "Enter Password"). grid (row - 22, column = 0, padx = 5
Password = Entry (frame)
Password. grid (row = 25, column = 0, padx = 5)
Button (frame, text = "Register", command = on_key_up). grid (
        row = 15, column = 1, padx = 5)
root. mainloop()
```

Output :

```
Enter Any Positive Number: 10
        1
        2
        3
        5
        8
        13
        21
        34
```

Expt. No.

28) Fibonacci

```
n = int(input("Enter Any Positive Number :"))
def recur_fibo(n) :
        if n <= 1 :
            return n.
        else :
            return (recur_fibo(n-1) + recur_fibo(n-2))

if n <= 0 :
        print("Enter Positive Number :")
else :
        print("Fibonacci Sequence")
        for i in range(n) :
            print(recur_fibo(i))
```

output :

```
10340.0
15840.0
```

Date 14/10/2024

Page No. 32

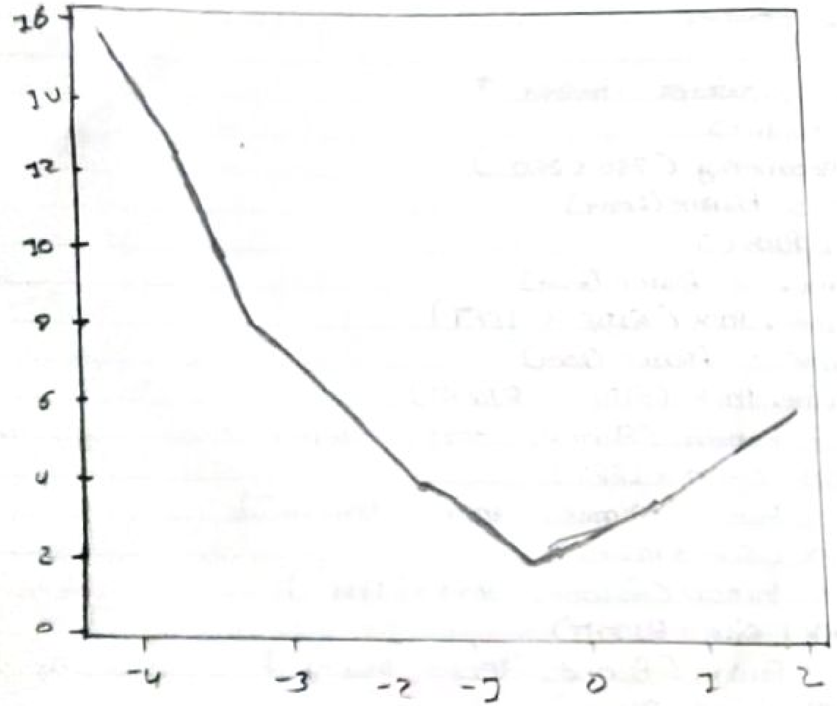
Expt. No.

27) Mortgage

```
import  matplotlib.pyplot  as  plt
import  numpy  as  np
def  plot_mortgage (loan_amount, interest_rate, loan_term) :
        monthly_rate = interest_rate /12
        num_payment = loan_term *12
        monthly_payment = loan_amount * interest_rate * 1 /100
        total_payment = monthly_payment * num_payment
        total_interest = total_payment - loan_amount
        print (total_interest)
        print (total_payment)

plot_mortgage (5500, 12, 2)
```

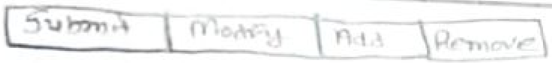
Teacher's Signature



Expt. No.

26) Plotting

```
import matplotlib.pyplot as plt
import numpy as np
~~def~~
from pylab import *
from numpy import *
x = linspace (-4, 2, 6)
y = x ** 2
plot (x,y)
show ()
```



Expt. No.

25) using frame

```
from tkinter import *
root = Tk()
root. geometry ("250 x 250")
frame = frame(root)
frame. pack()
leftframe. = frame (root)
leftframe . pack (SIDE = LEFT)
rightframe = frame (root)
rightframe. pack (SIDE = RIGHT)
btn1 = Button (frame, text = "Submit")
btn1. pack (side = LEFT)
btn2 = Button (frame, text = "Remove")
btn2. pack (side = RIGHT)
btn3 = Button (frame, text = "Add")
btn3. pack (side = RIGHT)
btn4 = Button (frame, text = "Modify")
btn4. pack (side = RIGHT)
root. mainloop ()
```

Teacher's Signature

Output :

```
5
None
```

Expt. No.

3) Divide & Conquer

```
def binarysearch (list, val) :
    list_size = len (list) - 1
    idx0 = 0
    idxn = list_size
    while idx0 <= idxn :
        midval = (idx0 + idxn)
        if list[midval] == val :
            return midval
        if val > list[midval] :
            idx0 = midval - 1
        else :
            idxn = midval - 1
        if idx0 > idxn :
            return "none"

list = [2, 7, 19, 34, 59, 72]
Print (binarysearch (list, 72))
Print (binarysearch (list, 1))
```

Output :

350

Expt. No.

29) Knapsack

```
def   knapsack (w, wt, val, n) :
          if  n == 0   or   w == 0 :
                    return  0
          if  wt[n-1]  >  w :
                    return  knapsack (w, wt, val, n-1)
          else :
                    return  max (val[n-1] + knapsack(w  - wt[n-1],
                      wt,  val,  n-1), knapsack(w, wt, val, n-1)


val  = [50, 100, 150, 200]
wt   =  [8, 16, 32, 40]
W    =  64
n    =  len(val)

print (knapsack (w, wt, val, n))
```