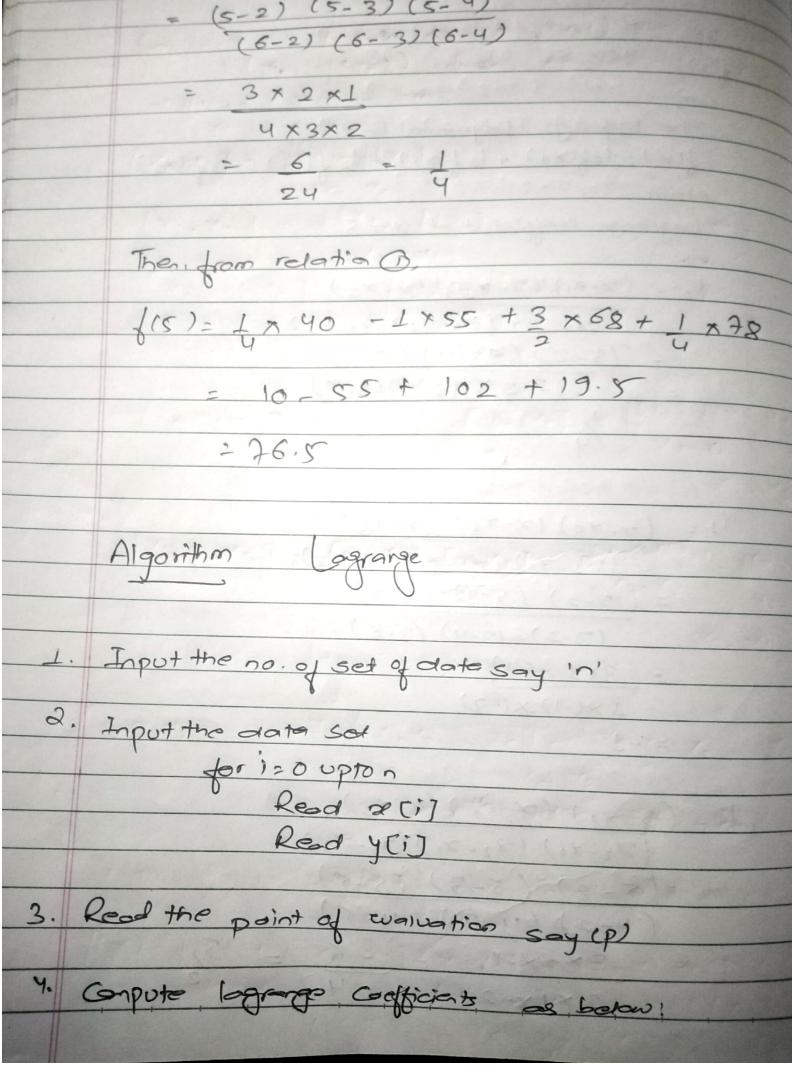


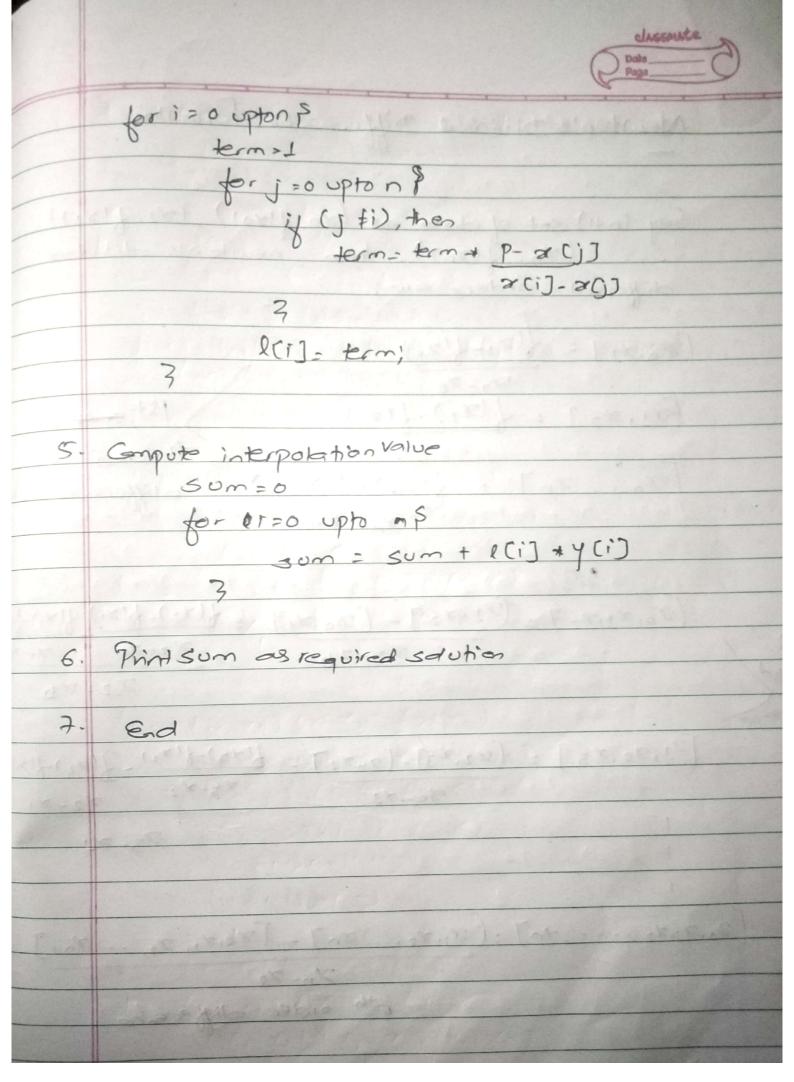
	elassmate Pore
	Algorithm of Newton Rahpson Method
<u> </u>	Define function f(x), f'(x) and specified error (E)
2.	Input to as initial guess such that f'(x0) to else
	Print message-input error and goto step 2.
3.	Compute the next approximiated root
	\$1 = 20 - f(20)
	and also the error of computation
	$\frac{ \mathcal{R}_1 - \mathcal{R}_0 }{ \mathcal{R}_1 }$
4.	Replace to by 81
5.	If f'(xo)=0, then print message there is no convergence of root and exit from the
	program
	continue
6.	If error > specified error, then goto skep 3.
	print or, as rot
7.	End (ad)

	6=0
	x3= 82- \$(x2) (x2-x1)
	$x_3 = x_2 - \{(x_2)(x_2 - x_1)\}$ $= \{(x_2) - \{(x_1)\}$
	$\frac{1.73 - 32 - \{(32)(32 - 21)\}}{\{(32) - \{(31)\}}$
	6 9
	This is called as approximiate formula for secont method.
	Secont method.
	ALCO ASSESSED CONTRACTOR OF THE PARTY OF THE
	Algorithm: Secant Mekad
	of this: Of and I peptal
1.	Define function f(x) and specified error (E)
2.	Take two initial guesses so and st.
Acres acres	Company of the second s
3.	Compute next approximiated root and error 83 = 72 - {(72)(82-71)
	f(82)-f(81)
	$\frac{\text{error} = \left \frac{2}{3} - \frac{2}{3} \right }{2}$
	6 Transport Tran
4.	Replacement X1= X2
	22 = 23
5	Use print 83 as a root.
1	
· Le	- 2 Mint - 1 - 12101-10 1121-12 1201-13
6.	End
1 -3 - 1 CA	aceter of aceter of the series

	Where,
1	in Joueses, Ind
	approximiated root can be evaluated as
	x1=3(x0)
7	Similarly,
	82 = 9(8)
	23-9(2)
	2; +1 = 9(81)
	Charles de la
	2it1=g(xi) This is the next approximinted
	root expression to fixed point
	iteration method
	Algorithm: fixed Paint
	Add the same of th
1.	Define function 9(x) and specified error E.
2.	Take x
	Take vo as initial guesses
3.	Con to soul
	Compute next approximiated root and error as
	31= 3 (20)
	erra = 21-20
	261
4.	Peplace to by 71
5.	If (error > E), then go to step3
	Clse print x1 as result.
6.	End

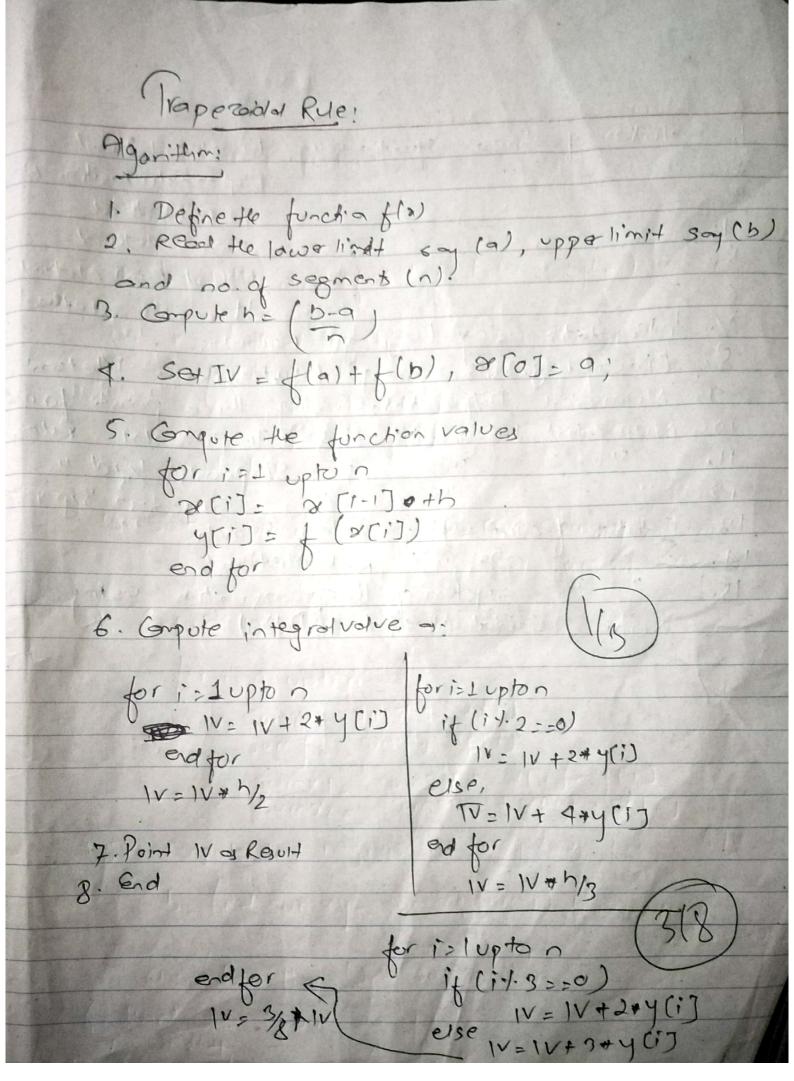
hind the value of
using Herner's formula.
Hese, 70-1, 92=0, 90=1, 93=1, 91-4
in have
b3: 03:1
b2 - 90 + b380
= 0 + 1 01
5 1
b1 = 9, + b2 ro
= -1 + 1-47
-0
pod spido
bo= do + b1 80
= 1 + 0 + 1
- 1
*
Algorithm: Verners
1. Input the order of polynomial say a and pain
at examption of
of evaluation say so
2. Read the point of roefficients of polynomials.
te 1 2 0 to n
Read a [i]
3. Set b[n]= 9n
4. De following appoination for n>=0
Drn-17 = arn-17
5[n-1] = a[n-1] + 5n + 80
5. Print bo as a functional a value
0. End





Unear Regression Mgontem 1. Input the no. of set of data say(n) D. Read fledata as for i=0 upto n
Read & Ci] Read y (i) end for 3. Set sum x = 0 , sum k = 0 , sum my = 0, SUM no = 0 4. Conput the values on: de l'i=0 upton Suma = Suma + a Cij Sumy = Sumy + y [i] Sumay = Sumay + a[i] + y[i]
Sumaz = Sumaz + a[i] + x[i] 16 11 11 1 6 1 1 1 5. Compute the regression coefficients as a 6 b 5 = (n# sumay - suma # sumy) /(n#suminea = (Bumy) - (b * summ) /n; 6. Print y = a +bm as fit line.

Bounday Value Prester Nymerica Differential using Three Paint Formant forms f"(x)= f(x+h)-2f(x)+f(n-h) Q. Marian Marian Marian Marian Algorithms 1. Define f(a), h = 0.001 2. Input the point of evaluation, p 3. Complete $f_1 = f(p+h) - f(p-h) | h$ $f_2 = f(xp+h) - 2f(p) + f(p-h)$ h + h4. Print fl, f2 os first derivatre de second derudu 5. End



Solution of ode Algorithm. 1. Define function of my, 2. Read initial caditions 20= a of you = 5 3. Read the stp size (say (h) and point of evaluation sy (p). 1. Compute the solution.

While (xo<p) & Euler's

Compute P.K while (xo<p) \$ m= f(x0, y0) m1 = f(20+h/2) do = Joth 3 you you man yo+m, +1/2) 5. Print you as root Ms= f(no+1/21 Yo+ M2+1/2) 6. Ed M = f (x0+h,
y0+m3# 2) EUKELUS A A ELLIA PLANT Id a to the property of the said m= (m1+2m1+ 2ms+, m9) estanted (Kan a consider of so かっこめのより To= yorman The contract pateins

```
Algorithm:
Readtle dimension of system of equations (say n),
RHS vector, accuracy limit (error)
Set initial guesses s:
   for i=1 upton
new-x(ij=0
  end for
 Compute the following
 for i, + upto n
    sum = b(i)
    for 1 = 1 upton
       it (ifj)
           sum > sum - a [i] (j) * new - 2 [j]
   enderfor
  old-x[i]; new-x[i]
  new-& [i] = sum la [i][i]
   Ecij= new-xcij-old-xcij
 end for
Check for error level
   for is 1 upton
  if (E[i]>erro)
go to stp 3
Display new 18 vector of regult.
End
```