**PRACTICAL NO – 2(B)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name :

Roll No :

Aim : Program for Newton’s backward interpolation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function [**yest**]=newtonback(**x**, **y**, **xest**)

n=length(**y**);

for i=2:n

d(i,1)=**y**(i)-**y**(i-1);

end

for i=2:n-1

for j=(i+1):n

d(j,i)=d(j,i-1)-d(j-1,i-1);

end

end

h=**x**(2)-**x**(1);

p=(**xest**-**x**(n))/h

e(1)=p;

for i=2:n-1

e(i)=e(i-1)\*(p+i-1)/i;

end

**yest**=0;

for i=1:n-1

**yest**=**yest**+e(i)\*d(n,i);

end

**yest**=**yest**+**y**(n);

printf('\ny is %g x is %g',**yest**,**xest**);

endfunction

**OUTPUT :**

-1->x=[1,2,3,4,5];

-1->y=[1,4,9,16,25];

-1->xest=2.3;

-1->newtonback(x,y,xest)

y is 5.29 x is 2.3 ans =

5.29