Practical 2: Interpolation

(a) Program for Newton's Forward Interpolation.

Problem Statement: Write a Scilab Code to find f(8) using Newton's Forward difference interpolation formula for the following data.

x	1	3	5	7
f(x)	24	120	336	720

Scilab Code:

```
clc;
clear;
x=[1 3 5 7];
y=[24 120 336 720];
h=2//interval between values of x
c=1;
for i=1:3
   d1(c)=y(i+1)-y(i);
   c=c+1;
end
c=1;
for i=1:2
   d2(c)=d1(i+1)-d1(i);
   c=c+1;
end
c=1;
for i=1:1
   d3(c)=d2(i+1)-d2(i);
   c=c+1;
end
d=[d1(1) d2(1) d3(1)];
x0=8;//value at 8
pp=1;
y_x=y(1);
p=(x0-1)/2;
for i=1:3
   pp=1;
   for j=1:i
   pp=pp*(p-(j-1));
```

```
end
disp('pp:',pp);
y_x=y_x+(pp*d(i))/factorial(i);
end
printf('Value of function at %f is: %f',x0,y_x);
```

Output:

```
 "pp:"

  3.5

 "pp:"

  8.75

 "pp:"

  13.125
Value of function at 8.000000 is: 990.000000
-->
```

(b) Program for Newton's Backward Interpolation.

Problem Statement: Write a Scilab Code to find $\sin(38^0)$ using Newton's Backward difference interpolation formula for the following data.

x (in degrees)	15	20	25	30	35	40
f(x)=sin x	0.258819	0.3420201	0.4226183	0.5	0.5735764	0.6427876

Scilab Code:

```
clc;
clear;
x=[15 20 25 30 35 40];
y=[0.2588190 0.3420201 0.4226183 0.5 0.5735764 0.6427876];
h=5//interval between values of x
c=1;
for i=1:5
   d1(c)=y(i+1)-y(i);
   c=c+1;
end
```

```
c=1;
for i=1:4
   d2(c)=d1(i+1)-d1(i);
   c=c+1;
end
c=1;
for i=1:3
   d3(c)=d2(i+1)-d2(i);
   c=c+1;
end
c=1;
for i=1:2
   d4(c)=d3(i+1)-d3(i);
   c=c+1;
end
c=1;
for i=1:1
   d5(c)=d4(i+1)-d4(i);
   c=c+1;
end
c=1;
d=[d1(5) d2(4) d3(3) d4(2) d5(1)];
x0=38;//value at 38 degree
pp=1;
y_x=y(6);
p=(x0-x(6))/h;
for i=1:5
   pp=1;
   for j=1:i
      pp=pp*(p+(j-1))
   end
disp('pp:',pp);
y_x=y_x+((pp*d(i))/factorial(i));
end
printf('Value of function at %i is: %f',x0,y_x);
```

Output:

```
 "pp:"
 -0.4
 "pp:"
 -0.24
 "pp:"
 -0.384
 "pp:"
 -0.9984000
 "pp:"
 -3.59424
Value of function at 38 is: 0.615661
-->
```

(c) Program for Lagrange's Interpolation.

Problem Statement: Write a Scilab Code for the following problem:

If y_1=4, y_3=12, y_4=19 and y_x =7, find x using lagrange's interpolation formula.

Scilab Code:

```
//Lagrange's Interpolation Formula
clc;
clear;
y=[4 12 19];
x=[1 3 4];
y_x=7;
Y_X=0;
poly(0,'y');
for i=1:3
   p=x(i);
   for j=1:3
      if i~=j then
```

```
            p=p*((y_x-y(j))/(y(i)-y(j)));
        end
    end
disp('p:',p);
Y_X=Y_X+p;
end
disp('Y_X=',Y_X);
```

Output:

```
 "p:"
  0.5
 "p:"
  1.9285714
 "p:"
 -0.5714286
 "Y_X="
  1.8571429
-->
```