NAME : SANA ARORA
ROLL NO. : MAT/21/60

Practical 10
Newton interpolation

P-1 (Computing Divided Difference)

```
In[1]:= NthDividedDiff[x0_, f0_, startindex_, endindex_] :=
         Module[{x = x0, f = f0, i = startindex, j = endindex, answer},
          If[i == j, Return[f[[i]]],
            answer = (NthDividedDiff [x, f, i + 1, j] - NthDividedDiff [x, f, i, j - 1]) /
               x  j   - x  i    ; Return answer ; ;
```

Q1

```
In[2]:= x = {0, 1, 3 };
        f = {1, 3, 55};
        NthDividedDiff  x, f, 1, 2

Out[4]= 2

In[5]:= NthDividedDiff  x, f, 2, 3

Out[5]= 26

In[6]:= NthDividedDiff  x, f, 1, 3

Out[6]= 8

In[7]:= Clear  f, x
```

Q2

```
In[8]:= x = {-1, 0, 1, 2 };
        f = {5, 1, 1, 11};
        NthDividedDiff  x, f, 1, 2

Out[10]=  4

In[11]:= NthDividedDiff  x, f, 2, 3

Out[11]= 0

In[12]:= NthDividedDiff  x, f, 1, 3

Out[12]= 2
```

In[13]:=
```
NthDividedDiff x, f, 2, 4
```
Out[13]= 5

In[14]:=
```
NthDividedDiff x, f, 1, 4
```
Out[14]= 1

In[15]:=
```
NthDividedDiff x, f, 3, 4
```
Out[15]= 10

In[16]:=
```
Clear f, x
```

P-2

In[17]:=
```
NewtonDDPoly[x0_, f0_] := Module[{x1 = x0, f = f0, n, newtonPolynomial, k, j},
    n = Length[x1]; newtonPolynomial[y_] = 0;
    For[i = 1, i c n, i++, prod[y_] = 1;
     For[k = 1, k c i - 1, k++,
      prod[y_] = prod[y] x (y - x1[[k]])]; newtonPolynomial[y_] =
       newtonPolynomial[y] + NthDividedDiff[x1, f, 1, i] x prod[y]];
    Return newtonPolynomial y  ; ;
```

Q1

In[18]:=
```
nodes = {0, 1, 3 };
values = {1, 3, 55};
NewtonDDPoly nodes, values
```
Out[20]= 1 + 2 y + 8 -1 + y y

In[21]:=
```
Simplify %
```
Out[21]= $1 - 6 y + 8 y^2$

In[22]:=
```
Clear nodes, values
```

Q2

In[23]:=
```
nodes = {-1, 0, 1, 2 };
values = {5, 1, 1, 11};
NewtonDDPoly nodes, values
```
Out[25]= 5 - 4 1 + y + 2 y 1 + y + -1 + y y 1 + y

In[26]:=
```
Simplify %
```
Out[26]= $1 - 3 y + 2 y^2 + y^3$