Practical 10 : - Newton Interpolation

Find the approximated polynomial using Newton Interpolation.

Q1 : Data = {(-2,39),(-1,3),(0,-1),(1,-3),(2,-9),(3,-1)}

In[ ]:=
```
ClearAll["Global*`"];
n = 6;
Array[x, n];
x[1] = -2;
x[2] = -1;
x[3] = 0;
x[4] = 1;
x[5] = 2;
x[6] = 3;
Array[y, n];
y[1] = 39;
y[2] = 3;
y[3] = -1;
y[4] = -3;
y[5] = -9;
y[6] = -1;
Array[dd, n, 0];
For[i = 0, i ≤ n - 1, ++i,
 Array[dd[i], n - i];
]
dd[0] = y;
For[i = 1, i ≤ n - 1, ++i,
 For[k = 1, k ≤ n - i, k++,
  dd[i][k] = (dd[i - 1][k + 1] - dd[i - 1][k]) / (x[k + i] - x[k]);
 ]
]
poly[t_] =
 Simplify[Sum[dd[i][1] * Product[(t - x[k]), {k, 1, i}], {i, 1, n - 1}] + dd[0][1]]
```

Out[ ]= $-1 - 3\,t^3 + t^4$

Q2 : Data = {(−1, 5), (0, 1), (1, 1), (2, 11)}

In[ ]:=
```
ClearAll["Global*`"];
n = 4;
Array[x, n];
x[1] = -1;
x[2] = 0;
x[3] = 1;
x[4] = 2;
Array[y, n];
y[1] = 5;
y[2] = 1;
y[3] = 1;
y[4] = 11;
Array[dd, n, 0];
For[i = 0, i ≤ n - 1, ++i,
 Array[dd[i], n - i];
]
dd[0] = y;
For[i = 1, i ≤ n - 1, ++i,
 For[k = 1, k ≤ n - i, k++,
  dd[i][k] = (dd[i - 1][k + 1] - dd[i - 1][k]) / (x[k + i] - x[k]);
 ]
]
poly[x_] =
 Simplify[Sum[dd[i][1] * Product[(x - x[k]), {k, 1, i}], {i, 1, n - 1}] + dd[0][1]]
```

Out[ ]= $1 - 3x + 2x^2 + x^3$

Q3 : Data = {(-3, -23), (1, -11), (2, -23), (5, 1)}

In[ ]:=
```
ClearAll["Global*`"];
n = 4;
Array[x, n];
x[1] = -3;
x[2] = 1;
x[3] = 2;
x[4] = 5;
Array[y, n];
y[1] = -23;
y[2] = -11;
y[3] = -23;
y[4] = 1;
Array[dd, n, 0];
For[i = 0, i ≤ n - 1, ++i,
 Array[dd[i], n - i];
]
dd[0] = y;
For[i = 1, i ≤ n - 1, ++i,
 For[k = 1, k ≤ n - i, k++,
  dd[i][k] = (dd[i - 1][k + 1] - dd[i - 1][k]) / (x[k + i] - x[k]);
 ]
]
poly[x_] =
 Simplify[Sum[dd[i][1] * Product[(x - x[k]), {k, 1, i}], {i, 1, n - 1}] + dd[0][1]]
```

Out[ ]= $1 - 10x - 3x^2 + x^3$