```python
def NewtonPolynomial(xi, f):
    table = [0]*len(xi)
    for i in range(len(table)):
        table[i] = [0]*(len(xi)+1)

    for i in range(len(f)):
        table[i][0] = xi[i]
        table[i][1] = f[i]

    for i in range(2,len(table[0])):
        for j in range(i-1,len(table)):
            table[j][i] = ((table[j - 1][i - 1] - table[j][i - 1])) / (table[j - i + 1][0] - table[j][0])
    return table

def get_coefficient(table):
    coefficient = []
    for i in range(len(table[0])-1):
        coefficient.append(table[i][i + 1])
    return coefficient

def get_polynomio(coefficient, y):
    polinomio = ['']
    n = len(y) - 1
    for i in range(len(y)):
        polinomio += "("+str(coefficient[i])+")x^"+ str(n)+ " + "
        n -= 1
    return "".join(polinomio)[:-2]

def main():
    x = [-4,-2, -1]
    f = [-2.5962, -0.6969, 0.9081]
    table = NewtonPolynomial(x, f)
    coefficient = get_coefficient(table)
    print(get_polynomio(coefficient, f))

main()
```