[ ]:
```
X_vec = [1.41; 2.31; 4.13; 4.89; 5.31; 6.01]
Y_vec = [-1.4156; 2.3901; 3.0567; 0.9812; 4.1245; 2.7569]
x1 = 6.17
x2 = 2.11

tableSize = length(X_vec)
differences = zeros(Float64, tableSize, tableSize)

for i in 1:tableSize
    differences[i, 1] = Y_vec[i]
end

for i in 2:tableSize
    for j in 1:(tableSize-i+1)
        head = j
        tail = j + i - 1

        dif = differences[j+1, i-1] - differences[j, i-1]
        dx = X_vec[tail] - X_vec[head]
        differences[j, i] = dif / dx
    end
end

print("Разделенные разности: \n\n")
for i in 1:tableSize
    for j in 1:tableSize
        print(round(differences[i, j], digits=3), "\t")
    end
    print("\n")
end
```

```
Разделенные разности:

-1.416  4.229   -1.42   0.063   0.826   -0.907
2.39    0.366   -1.2    3.286   -3.344  0.0
3.057   -2.731  8.657   -9.087  0.0     0.0
0.981   7.484   -8.427  0.0     0.0     0.0
4.124   -1.954  0.0     0.0     0.0     0.0
2.757   0.0     0.0     0.0     0.0     0.0
```

[ ]:
```
function NewtonPolyForward(Y_vec, X_vec, x, differences)
    N = length(Y_vec)
    polynom = 1.0
    answer = Y_vec[1]

    for i in 2:N
        row = 1
```

```
        col = i

        polynom *= (x - X_vec[i-1])
        answer += differences[row, col] * polynom
    end

    return answer
end

print("x1 = ", x1, "\n\n")
print("Pn(1)(x1) = ", NewtonPolyForward(Y_vec, X_vec, x1,␣
  ↪differences))
```

```
x1 = 6.17

Pn(1)(x1) = -2.7744682538104186
```

[ ]:
```
function NewtonPolyBackward(Y_vec, X_vec, x, differences)
    N = length(Y_vec)
    polynom = 1.0
    answer = Y_vec[N]

    for i in 2:N
        row = (N - i + 1)
        col = i

        polynom *= (x - X_vec[row+1])
        answer += differences[row, col] * polynom
    end

    return answer
end

print("x2 = ", x2, "\n\n")
print("Pn(2)(x2) = ", NewtonPolyBackward(Y_vec, X_vec, x2,␣
  ↪differences))
```

```
x2 = 2.11

Pn(2)(x2) = -1.1693881919521658
```