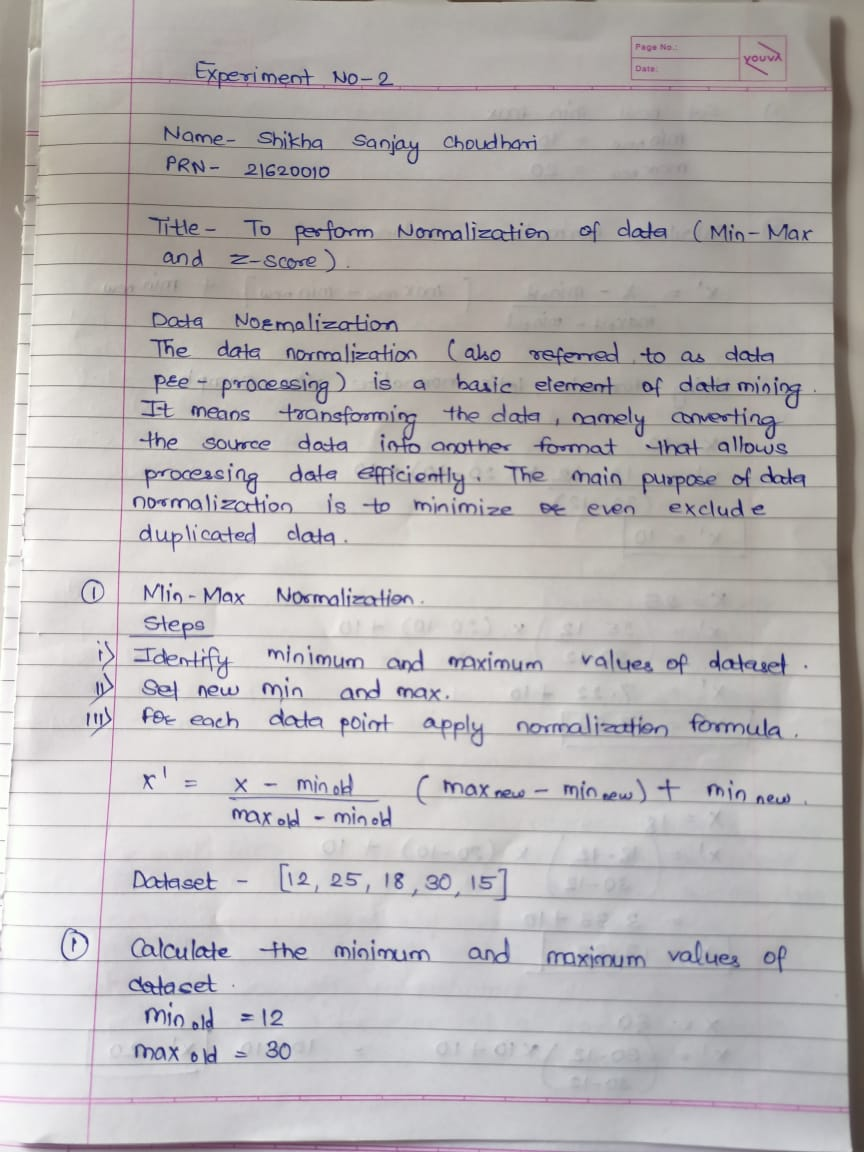
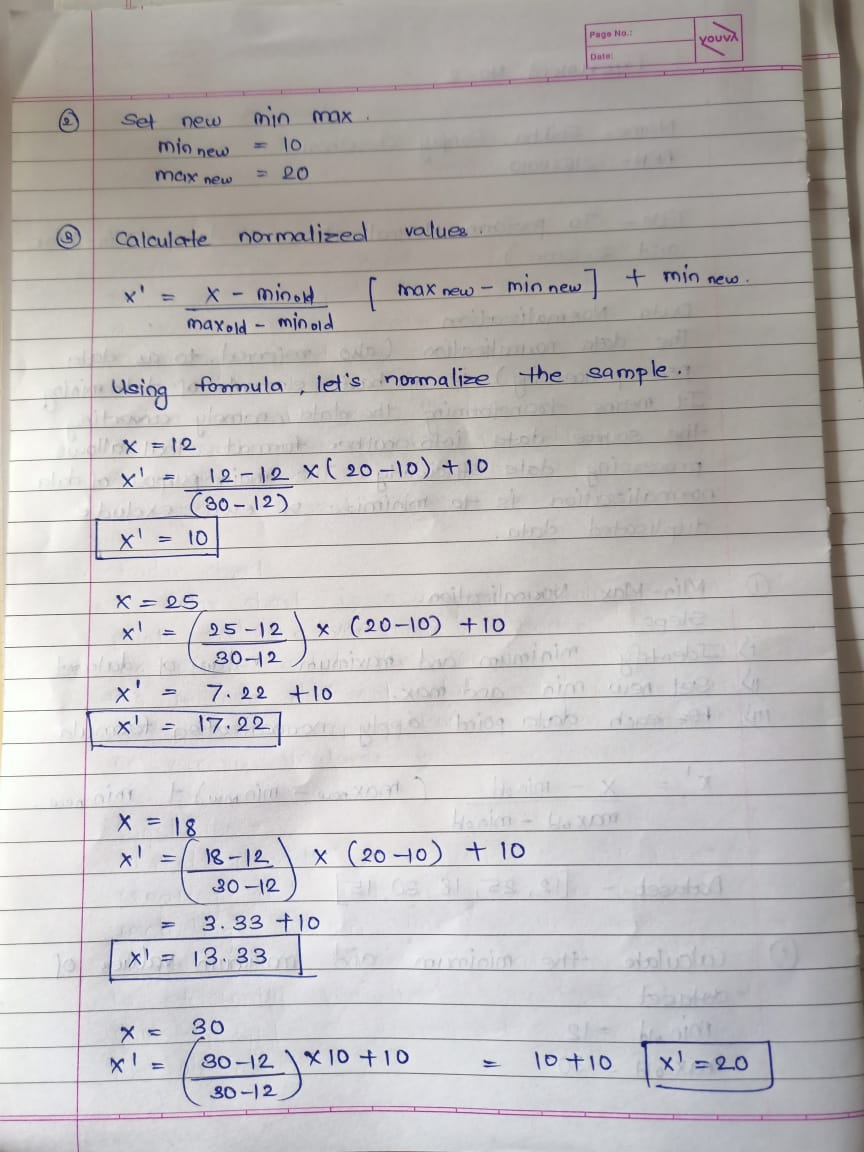
**Experiment 2**

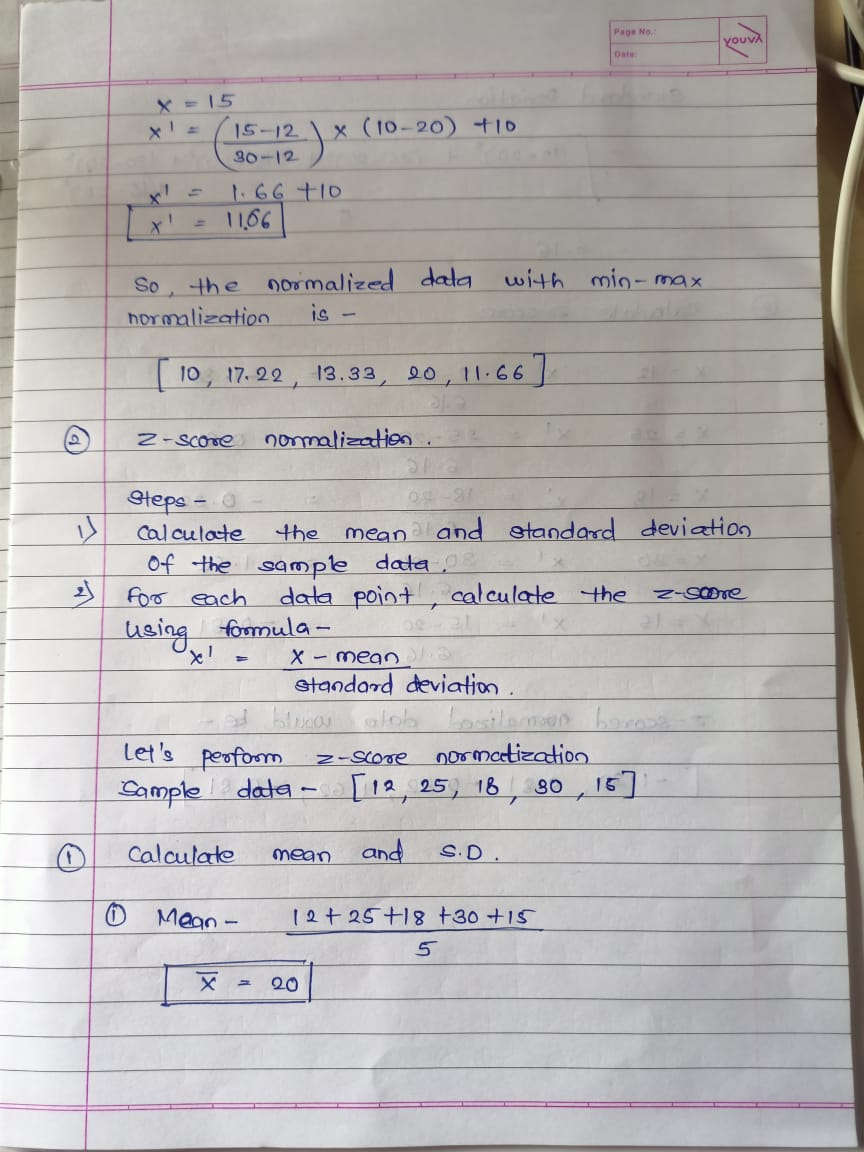
Name - Shikha Sanjay Choudhari

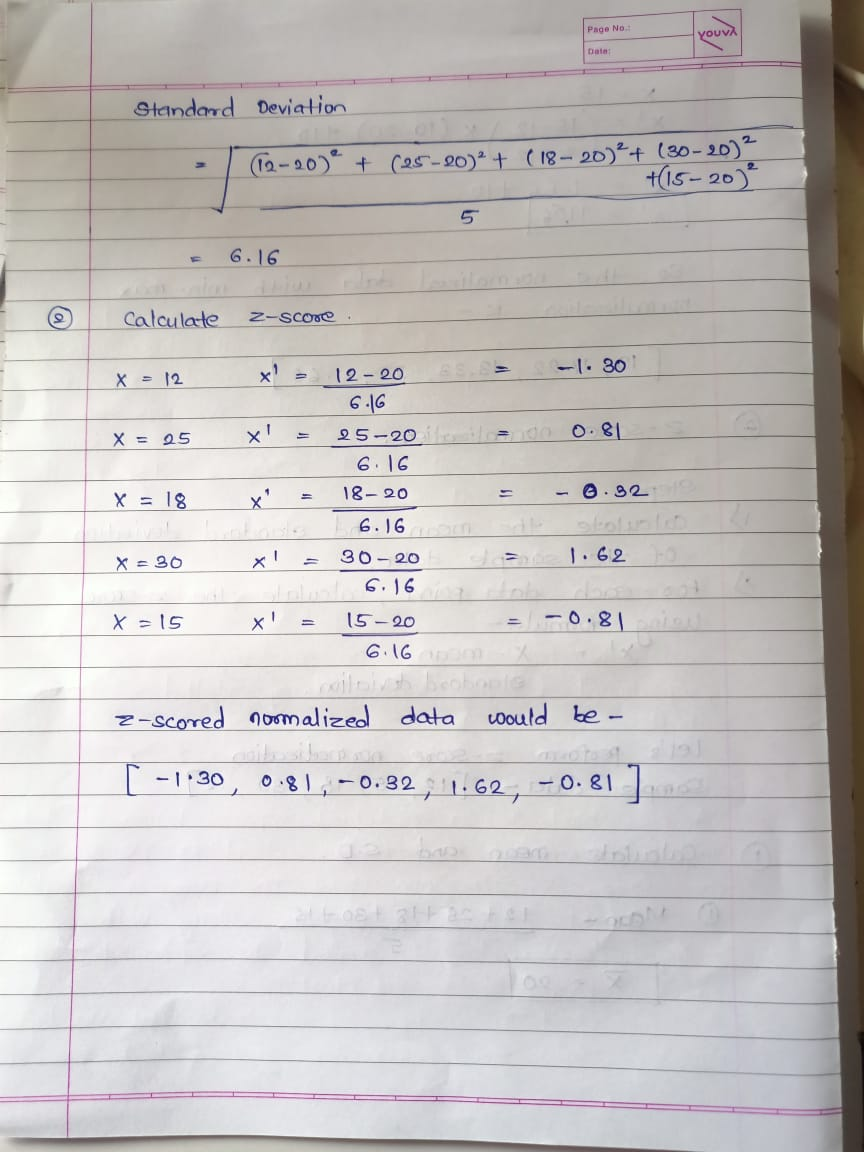
PRN- 21620010

**Title- To perform normalization of data (Min-Max and Z-score)**

****

****

****

****

* **Code**

#include <bits/stdc++.h>

#include <fstream>

#include <vector>

#include <cmath>

#include <string>

using namespace std;

// Function to perform min-max normalization using the specified formula

vector<double> minMaxNormalization(const vector<double>& data, double min\_new, double max\_new) {

    double min\_old = \*min\_element(data.begin(), data.end());

    double max\_old = \*max\_element(data.begin(), data.end());

    vector<double> normalized\_data;

    for (double val : data) {

        double normalized\_val = (val - min\_old) / (max\_old - min\_old) \* (max\_new - min\_new) + min\_new;

        normalized\_data.push\_back(normalized\_val);

    }

    return normalized\_data;

}

// Function to perform z-score normalization

vector<double> zScoreNormalization(const vector<double>& data) {

    double mean = 0;

    for (double val : data) {

        mean += val;

    }

    mean /= data.size();

    double variance = 0;

    for (double val : data) {

        variance += (val - mean) \* (val - mean);

    }

    variance /= data.size();

    double std\_deviation = sqrt(variance);

    vector<double> normalized\_data;

    for (double val : data) {

        double z\_score = (val - mean) / std\_deviation;

        normalized\_data.push\_back(z\_score);

    }

    return normalized\_data;

}

int main() {

    ifstream input\_file("input.txt");

    ofstream output\_file("output.txt");

    vector<double> data;

    double value;

    while (input\_file >> value) {

        data.push\_back(value);

    }

    double min\_new, max\_new;

    cout << "Enter the new range for min-max normalization (min new max): ";

    cin >> min\_new >> max\_new;

    vector<double> min\_max\_normalized = minMaxNormalization(data, min\_new, max\_new);

    vector<double> z\_score\_normalized = zScoreNormalization(data);

    output\_file << "Min-Max Normalized Data:\n";

    for (double val : min\_max\_normalized) {

        output\_file << val << "\n";

    }

    output\_file << "\nZ-Score Normalized Data:\n";

    for (double val : z\_score\_normalized) {

        output\_file << val << "\n";

    }

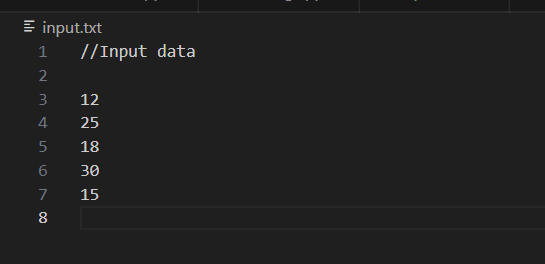
    input\_file.close();

    output\_file.close();

    return 0;

}

* **Input data**

****

* **Output**

