Cairo University

FCAI

CS213

Dr,Muhammed Elramly

Assignment1_Task2,3

Name	ID	Task
Mahmoud Sayed Mahmoud Younis	**20230372**	**Chat-Gpt code, NTI report,free code camp report.**
Saif Omar Emam Abd-Elhalem	**20230183**	**Copilot code, ITI report.**
Hassan Momen Hassan Hosny	**20231047**	**AI code analysis and report.**

Code Report - Task 2

Brief:

Both codes did the assignment ordered from them with the same prompt to ensure fairness. While one of them performed better in almost every scale, both of them fell in somewhat the same tests, except one did one better than the other.

Prompt used: **you will develop a Polynomial class in C++. An object of this class represents a polynomial of arbitrary degree. The public interface of your class will be as in this header file that I will send you after you read this. You will develop 30 test cases and write a program to run the test cases, wait for the header file.**

Codes and Test Cases are provided in the appendix below.

AI assistants used in this report were:
- Microsoft Copilot
- ChatGPT
- Blackbox AI (bonus assistant that fed us with the answers of the test cases in case we got any of them wrong)

ChatGPT:

- When first put the prompt the ai made it so if it answers incorrectly, it would put an exception and stop running ultimately not compiling at all which the first flaw it was then told to just print that the test is incorrect.
- We transformed the first 30 test cases to normal English just so we could feed it to another ai to make the soultions for those test cases just so we could manually check for errors

Explanation:

1. Constructors and Destructor:
   - The class has three constructors: a default constructor that initializes a zero polynomial, one that accepts a vector of coefficients, and a copy constructor for creating a new polynomial from an existing one.

2. Assignment Operator:
   - The assignment operator (operator=) allows one polynomial to be equal to another.

3. Arithmetic Operators:
   - The class implements the addition (operator+), subtraction (operator-), and multiplication (operator*) of polynomials. These operators create and return new Polynomial objects representing the result of the respective arithmetic operation.

4. Equality Operator:
   - The equality operator (operator==) checks if two polynomials are identical by comparing their coefficient vectors.

5. Output Operator:
   - The output operator (operator<<) enables easy printing of polynomial objects in a human-readable format, showing the polynomial in descending order of degree.

6. Utility Functions:
   - The degree() function returns the degree of the polynomial, which is the size of the coefficient vector minus one. The

evaluate(double x) function calculates the polynomial's value at a given point x. The getCoefficient(int degree) function retrieves the coefficient for a specific degree, returning zero if the degree is out of range. The setCoefficients(const vector<double>& coefficients) function updates the polynomial's coefficients.

Testing the Polynomial Class
The runTests() function is designed to test various functionalities of the Polynomial class. It includes a series of tests that cover:
- Creation of default and specific polynomials.
- Basic arithmetic operations like addition, subtraction, and multiplication, with expected outcomes printed to the console.
- Evaluation of polynomials at specific points.
- Checking polynomial degrees and coefficients.
- Handling of polynomials with negative coefficients and large coefficients.
- Equality checks between identical and different polynomials.
- Testing edge cases like evaluating at zero and composing with empty polynomials.
- Mock calculations for definite integrals and polynomial roots, even though the actual root-finding method is not implemented in this code.

Copilot:

Explanations:
1. Constructors and Destructor:
   - The class features three constructors:
     1. Default Constructor: Initializes a zero polynomial with a single coefficient of 0.0.
     2. Coefficient Vector Constructor: Accepts a vector of coefficients to create a polynomial based on user input.
     3. Copy Constructor: Creates a new polynomial by copying the coefficients from another polynomial instance.
   - The destructor is defined but does not perform any specific actions, as the class does not manage dynamic memory directly.

2. Assignment Operator:
   - The assignment operator: (operator=) allows one polynomial object to be assigned to another. It includes a check for self-assignment to prevent unnecessary operations. If the polynomials are different, it copies the coefficients from the source polynomial.

3. Arithmetic Operators:
   - The class implements three arithmetic operators:
   - Addition (operator+): Computes the sum of two polynomials by creating a new polynomial with the resulting coefficients.
   - Subtraction (operator-): Similar to addition, it computes the difference between two polynomials.
   - Multiplication (operator*): Multiplies two polynomials together, resulting in a new polynomial whose coefficients are determined by the product of the individual coefficients.

4. Equality Operator:
   - The equality operator(operator==) checks if two polynomial objects are identical by comparing their coefficient vectors for equality. This allows for straightforward comparison between polynomial instances.

5. Output Operator:
   - The output operator(operator<<) is overloaded to enable easy printing of polynomial objects. It formats the polynomial in a human-readable way, displaying the coefficients in descending order of degree while omitting any terms with a zero coefficient.

6. Utility Functions:
   - The class includes several utility functions:

- degree(): Returns the degree of the polynomial, which is the number of coefficients minus one.
- evaluate(double x): Evaluates the polynomial at a specific value of x using Horner's method for efficiency.
- getCoefficient(int degree): Retrieves the coefficient for a specified degree, returning zero for degrees outside the range of the polynomial.
- setCoefficients(const vector<double>& coefficients): Updates the coefficients of the polynomial.

Testing the Polynomial Class:

The runTests() function is designed to validate the functionalities of the Polynomial class. It covers a variety of test cases, including:
- Creation of both default and specific polynomials.
- Execution of arithmetic operations like addition, subtraction, and multiplication, with expected outcomes printed for verification.
- Evaluation of polynomials at various points to ensure correct computation.
- Checking the degree of polynomials and accessing specific coefficients.
- Handling special cases like polynomials with negative or large coefficients.
- Verifying equality between identical and different polynomial instances.
- Testing edge scenarios, such as evaluating at zero and composing with an empty polynomial.
- Mock calculations for definite integrals and polynomial root finding, though actual root-finding logic is not implemented in this version.

Main Function:

The **main()** function serves as the entry point of the program. It invokes runTests() to execute the defined tests. After all tests are run, a completion message is displayed to indicate that the testing process has finished.

Analysis

1. Readability :

ChatGPT(28/30):

- The code is neatly structured with clear function and variable names. It uses comments to explain complex parts, enhancing understanding.

CoPilot(25/30):

- The code is more concise. It avoids redundancies, The code follows a consistent and minimalist approach, using functions effectively. it lacks detailed comments or explanations in some areas, making it harder to understand.

2. Simplicity:

ChatGPT(25/30):

- Follows a step-by-step approach with an explicit handling of terms in polynomial operations. The use of for-loops is straightforward.

- The redundancy adds unnecessary complexity, especially in operations like add() and multiply(). Each method involves multiple steps that are not needed and could be condensed.

CoPilot(28/30):

- The code is much simpler and more efficient. By reducing repeated logic and combining steps, it achieves the same outcome with fewer lines of code.

3. Performance (Speed):

ChatGPT(25/30):

- Due to the redundancy in the add() and multiply() functions, the first code performs more computations than necessary. The multiple iterations and explicit handling of each term in loops make it slower when dealing with larger polynomials.

CoPilot(28/30):

- The code reduces the number of unnecessary computations by using efficient looping constructs and reusing logic. This results in faster execution.

4.Correctness:

ChatGPT(14/20):

- Failing 6 out of 30 test cases (80% pass rate) indicates issues in edge cases or specific logic, reducing the reliability
- failed cases (12 , 19 , 21 , 23 , 25 , 28).

CoPilot(16/20):

- With only 5 out of 30 test case failures (83% pass rate), the second code is slightly more reliable, though still needing improvement.
- failed cases (12 , 19 , 21 , 23 , 24).

Conclusion:

- With ChatGPT getting an 83% in general performance and CoPilot getting an 88% it's clear that copilot proved to be superior in certain parts.
- Copilot also added a couple of functions that GPT lacked like integral , definite integral,derivative and better exception handleing for get root function .
- Copilot was different in handling the output as if the output was only 0 it would print out an empty line.
- Both of them lacked handeling bigger roots and imaginary roots although it was ordered to do so multiple times.
- Both of them used mostly the same logic at the beginning but each went with their own way of implementing it.
- Generally AI can be a useful tool in coding but it shouldn't be a primary source for code unless it's understood and revised with caution.

Training research-Task 3

DEY - Egypt Makes Electronics

1. Who Offers It and Where It Is Located

This training is part of the capacity building programs offered by the NTI under the name of Digital Egypt Youth (DEY).

About the NTI :

The National Telecommunication Institute (NTI), established in 1983, excels in Egypt's education, research, and technical consultation. Aligned with the Ministry of Telecommunication and Information Technology (MCIT), NTI recognizes ICT's crucial role in the Egyptian societal advancement. It plays a key role in Egypt's Vision 2030 for digital transformation, supported by the government and industry collaboration. NTI prioritizes high-quality education, training, and innovative research, offering consultancy and technical services.

NTI Sites in Egypt :

1. Nasr City Campus .

2. Smart Village Campus.

3. Knowledge City Campus (New Capital, Egypt).

2. What It Offers

The Digital Egypt Youth (DEY) initiative plays a vital role in achieving one of the strategic objectives of the ICT sector by empowering young Egyptians to thrive in the digital economy, contributing to the country's socioeconomic development and digital transformation initiatives.

The Digital Egypt Youth (DEY) initiative aims to provide young Egyptians with the necessary skills and knowledge to thrive in the digital age. The initiative focuses on training youth in various digital fields to increase their employability and entrepreneurship opportunities. The initiative aims to bridge the digital skills gap in Egypt and create a digitally skilled workforce to support the country's digital transformation. It specifically targets young Egyptians, equipping them with in-demand skills needed for jobs in the digital economy.

The initiative offers a full-fledged scholarship and targets 2500 fresh graduates from Egyptian universities nationwide. The initiative adopts the upskilling approach: technical up-skilling and personal and soft skills up-skilling aspects. The program seeks to cover various digital skills to cater to diverse interests and career paths.

The Digital Egypt Youth (DEY) initiative collaborates with industry partners and experts to develop and deliver relevant training content. This ensures that the program aligns with industry requirements and equips participants with practical skills and knowledge that are in demand in the job market. The program aims to facilitate job placement for trained participants by partnering with industry stakeholders and creating linkages between trained individuals and potential employers through one month of job training at the industry partner premises. This helps connect young talent with job opportunities in the digital sector. NTI collaborated with more than 60 national and multinational industry partners including but not limited to, Amazon, Etisalat Egypt, Vodafone, Huawei, WE, Cisco, VMware, Orange, Valeo, Avelabs, and SEITech, Brightsky, VOIS to design more than 25 professional tracks to meet job profiles and ICT

market need. At the end of the program, participants receive certification that validate their skills and competencies. Theses certificates help enhance their employability and serve as credentials for future job opportunities.

A key target field of the Digital Egypt Initiative is the electronics & embedded systems to match the presidential initiative Egypt Makes Electronics (EME)

3. When It Opens, Duration, and Mode (Online/Offline)

Registration is currently closed, and expected to be opened during summer, the Duration of the training is

4 months in an offline study mode.

4. Conditions for Application, Acceptance, Fees, and Other Conditions

1. Commitment to submit all required documents on the initiative's electronic platform upon approval of these terms and regulations is a fundamental requirement for scholarship acceptance:

2. Copy of the graduation certificate.

3. Scanned copy of the national ID card.

4. Signed declaration by the trainee.

5. Compliance with the fundamental requirements of each training program is a prerequisite for acceptance into the scholarship.

6. Commitment to the scheduled training hours (6 training hours daily).

7. Adherence to the prescribed attendance percentage of not less than 75%.

8. Commitment to appropriate behavior and professional etiquette with everyone during the training period within the initiative.

9. Commitment to preserving academic materials and refraining from posting them on electronic platforms (intellectual property rights).

10. Commitment to the accuracy of all information stated in the declaration.

11. Commitment to submitting a new signed declaration for each training course the trainee enrolls in.

12. Commitment to providing a notification or sending an apology via email for non-completion of the scholarship, at least 48 hours before the commencement of the training.

13. Only communications through the official email of the initiative (D4m@nti.sci.eg) will be considered.

14. There are no fees once accepted in the initiative.

5. Different Learning Tracks

training tracks that the initiative offers are :

        1. Electronics and Embedded Systems

2. Network Infrastructure and Enablers

3. Cybersecurity.

4. Devops Engineering.

5. Telecom Engineering.

6. Software Engineering.

6. Prefferd track and why:

Electronics and Embedded Systems:

I chose this track specificly because I have always been fascinated by the integration process

between software and hardware and the creativity that it offers the learner, also learning

Embedded Systems gives the learner a deeper understanding of how computers function.

check appendix (D) for resources

Free Code Camp

1. Who Offers It and Where It Is Located

 training is offerd by freeCodeCamp.org

About freecodecamp.org :

freeCodeCamp is a donor-supported charitable organization their message is to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. Donations to freeCodeCamp go toward their education initiatives, and help pay for servers, services, and staff.

2. What It Offers

Free Code Camp trainings offer courses in various feilds like web design, data analysis and machine learning.training is completly free and online. The trainings offered by free code camp make use of avtive learning by explaining concepts using projects suitable for beginners and at the end of each training a certification is given for free as and indication that the trainee completed the training to the end.

3. When It Opens, Duration, and Mode (Online/Offline)

training is available on freecodecamp.org completely free and online only which makes learning at your preffered pace.

4. Conditions for Application, Acceptance, Fees, and Other Conditions

There are no conditions nor fees for applying for the training, however there are some conditions that govern the use of the website :

1. You must be at least thirteen years old.

2. You may no longer use the website if the company contacts you directly to say that you may not.

3. You must use the website in accordance with Acceptable Use and Content Standards.

5. Different Learning Tracks

 training tracks that the website offers are :

1. Responsive Web Design Certification.

2. Front End Development Libraries Certification

3. Information Security Certification.

4. Relational Database Certification.

5. Back End Development and APIs Certification.

6. Machine Learning with Python Certification.

6. Preffered track and why

Datat Analysis with python :

Every industry uses data to drive the decisions it makes, and that means every company needs people who can organize and interpret data. Industries as varied as finance, healthcare, marketing, and technology all rely on data analysts to shape their strategies, optimize their processes, and drive their strategic initiatives. Learning data analysis skills can make you a highly sought-after candidate, even if you haven't worked in the field before.

check appendix (D) for resources

Information technology Institute (ITI)

1. Who Offers It and Where It Is Located

ITI is offered by the Egyptian Ministry of Communications and Information Technology. It has multiple locations across Egypt, including Smart Village, Alexandria, Mansoura, Assiut, and other cities. Information technology Institute (ITI) : It is a national training institute established by the Egyptian government in 1993, under the Ministry of Communications and Information Technology. ITI specializes in providing professional training in the field of information technology to equip graduates with the skills needed for the job market. It collaborates with various industry partners and offers programs in tech-related fields like software development, data science, AI, and cybersecurity

2. What It Offers

ITI provides two main programs: a 1-month and a 9-month professional training program. The 9-month program includes over 30 specialized technology tracks like Artificial Intelligence (AI), data analysis, cybersecurity, and software development.

there are some information about the training programs :

1-Month Program:

Who can apply: Generally, anyone can apply to ITI's 1-month program, as it is designed to offer short-term, specialized training in various IT tracks. It is suitable for:

Fresh graduates,

Mid-career professionals seeking to upskill or reskill,

Individuals with or without a strong technical background.

The application process is relatively open, and while some technical tracks may require basic prerequisites, the program is designed to be accessible to a broader audience.

9-Month Program (Scholarship):

Who can apply: The 9-month program is more exclusive. The primary eligibility criteria are:

Egyptian university graduates who have graduated within the past 5 years.

Applicants must have a minimum grade of "Good".

The program is highly competitive, accepting only around 10% of applicants, as it aims to prepare participants for specific job roles in the tech industry.

The selection process includes:

Submitting academic and personal documents,

Passing an aptitude exam,

Completing interviews and additional tests based on the chosen track

This means anyone can apply for the 1-month program if they meet the basic requirements, but the 9-month program is limited to recent graduates and has a more rigorous selection process.

3. When It Opens, Duration, and Mode (Online/Offline)

The 9-month program starts every October and runs until June the following year. It is mainly conducted offline at ITI's premises.

4. Conditions for Application, Acceptance, Fees, and Other Conditions

Applicants must be Egyptian university graduates within the last five years. The program is fully funded, meaning no fees are required, but acceptance is competitive with a 10% selection rate

5. Different Learning Tracks

1-Artificial Intelligence (AI): This track covers machine learning, natural language processing, and computer vision.

2-Cybersecurity: Focuses on network security, ethical hacking, and cyber defense strategies.

3-Data Science and Analytics: Teaches data mining, machine learning, and business analytics

4-Embedded Systems: For those interested in microcontroller programming, real-time systems, and IoT devices

5-Mobile Application Development: Covers both Android and iOS app development.

6-Cloud Computing: Focuses on deploying and managing applications in cloud environments.

7-Software Development: This broad track includes web development, mobile app development, and DevOps

8-Geographic Information Systems (GIS): Covers spatial data analysis and mapping technologies.

9-Digital Arts and Cinematic Effects: Aimed at individuals interested in visual effects, animation, and 3D modeling

10-Building Information Modeling (BIM): For professionals in construction and architecture looking to implement digital models in their workflows.

11-Electronic Circuit Design: For those interested in the design and implementation of electronic components and system

6. Preffered track and why

Artificial Intelligence and Machine Learning

Why: AI and Machine Learning are rapidly growing fields with high demand across industries. ITI's 9-month program offers in-depth training in AI, providing a strong theoretical foundation and practical experience, which aligns with your preferred focus. Additionally, the fully funded nature of ITI's program makes it a valuable option compared to other programs that may charge fees or provide less structured guidance.

check appendix (D) for resources.

Appendix (A)

chat-gpt

Chat gpt code(implementation and menu)

Menu:

#include "Polynomial.h"

```cpp
#include "Polynomial.hpp"


void displayMenu() {
    cout << "Polynomial Operations Menu:\n";
    cout << "1. Add two polynomials\n";
    cout << "2. Subtract two polynomials\n";
    cout << "3. Multiply two polynomials\n";
    cout << "4. Evaluate a polynomial at a point\n";
    cout << "5. Display a polynomial\n";
    cout << "6. Exit\n";
    cout << "Please enter your choice: ";
}


int main() {
    int choice;
    Polynomial p1, p2, result;
    vector<double> coefficients;
    int degree;
    double value;

    // Menu loop
    do {
        displayMenu();
        cin >> choice;

        switch (choice) {
            case 1: // Add two polynomials
                cout << "Enter the degree of the first polynomial: ";
                cin >> degree;
                coefficients.resize(degree + 1);
                cout << "Enter the coefficients of the first polynomial: ";
                for (int i = 0; i <= degree; ++i) {
```

```cpp
        cin >> coefficients[i];
    }
    p1 = Polynomial(coefficients);

    cout << "Enter the degree of the second polynomial: ";
    cin >> degree;
    coefficients.resize(degree + 1);
    cout << "Enter the coefficients of the second polynomial: ";
    for (int i = 0; i <= degree; ++i) {
        cin >> coefficients[i];
    }
    p2 = Polynomial(coefficients);

    result = p1 + p2;
    cout << "The result of addition is: " << result << endl;
    break;

case 2: // Subtract two polynomials
    cout << "Enter the degree of the first polynomial: ";
    cin >> degree;
    coefficients.resize(degree + 1);
    cout << "Enter the coefficients of the first polynomial: ";
    for (int i = 0; i <= degree; ++i) {
        cin >> coefficients[i];
    }
    p1 = Polynomial(coefficients);

    cout << "Enter the degree of the second polynomial: ";
    cin >> degree;
    coefficients.resize(degree + 1);
    cout << "Enter the coefficients of the second polynomial: ";
    for (int i = 0; i <= degree; ++i) {
        cin >> coefficients[i];
```

```cpp
        }
        p2 = Polynomial(coefficients);

        result = p1 - p2;
        cout << "The result of subtraction is: " << result << endl;
        break;

    case 3: // Multiply two polynomials
        cout << "Enter the degree of the first polynomial: ";
        cin >> degree;
        coefficients.resize(degree + 1);
        cout << "Enter the coefficients of the first polynomial: ";
        for (int i = 0; i <= degree; ++i) {
            cin >> coefficients[i];
        }
        p1 = Polynomial(coefficients);

        cout << "Enter the degree of the second polynomial: ";
        cin >> degree;
        coefficients.resize(degree + 1);
        cout << "Enter the coefficients of the second polynomial: ";
        for (int i = 0; i <= degree; ++i) {
            cin >> coefficients[i];
        }
        p2 = Polynomial(coefficients);

        result = p1 * p2;
        cout << "The result of multiplication is: " << result << endl;
        break;

    case 4: // Evaluate polynomial at a point
        cout << "Enter the degree of the polynomial: ";
        cin >> degree;
```

```cpp
            coefficients.resize(degree + 1);
            cout << "Enter the coefficients of the polynomial: ";
            for (int i = 0; i <= degree; ++i) {
                cin >> coefficients[i];
            }
            p1 = Polynomial(coefficients);

            cout << "Enter the value of x to evaluate the polynomial: ";
            cin >> value;

            cout << "The value of the polynomial at x = " << value << " is: " << p1.evaluate(value) << endl;
            break;

        case 5: // Display polynomial
            cout << "Enter the degree of the polynomial: ";
            cin >> degree;
            coefficients.resize(degree + 1);
            cout << "Enter the coefficients of the polynomial: ";
            for (int i = 0; i <= degree; ++i) {
                cin >> coefficients[i];
            }
            p1 = Polynomial(coefficients);

            cout << "The polynomial is: " << p1 << endl;
            break;

        case 6: // Exit
            cout << "Exiting the program." << endl;
            break;

        default:
            cout << "Invalid choice. Please try again." << endl;
            break;
```

```
        }

    } while (choice != 6);


    return 0;
}
```

Implementaion:

```cpp
#include "Polynomial.h"


// Constructors
Polynomial::Polynomial() : coeffs(1, 0.0) {}


Polynomial::Polynomial(const vector<double>& coefficients) : coeffs(coefficients) {}


Polynomial::Polynomial(const Polynomial& other) : coeffs(other.coeffs) {}


// Destructor
Polynomial::~Polynomial() {}


// Assignment operator
Polynomial& Polynomial::operator=(const Polynomial& other) {
    if (this != &other) {
        coeffs = other.coeffs;
    }
    return *this;
}


// Arithmetic operators
Polynomial Polynomial::operator+(const Polynomial& other) const {
    size_t maxDegree = max(coeffs.size(), other.coeffs.size());
    vector<double> result(maxDegree, 0.0);
```

```cpp
    for (size_t i = 0; i < maxDegree; ++i) {
        if (i < coeffs.size()) result[i] += coeffs[i];
        if (i < other.coeffs.size()) result[i] += other.coeffs[i];
    }
    return Polynomial(result);
}


Polynomial Polynomial::operator-(const Polynomial& other) const {
    size_t maxDegree = max(coeffs.size(), other.coeffs.size());
    vector<double> result(maxDegree, 0.0);


    for (size_t i = 0; i < maxDegree; ++i) {
        if (i < coeffs.size()) result[i] += coeffs[i];
        if (i < other.coeffs.size()) result[i] -= other.coeffs[i];
    }
    return Polynomial(result);
}


Polynomial Polynomial::operator*(const Polynomial& other) const {
    vector<double> result(coeffs.size() + other.coeffs.size() - 1, 0.0);
    for (size_t i = 0; i < coeffs.size(); ++i) {
        for (size_t j = 0; j < other.coeffs.size(); ++j) {
            result[i + j] += coeffs[i] * other.coeffs[j];
        }
    }
    return Polynomial(result);
}


// Equality operator
bool Polynomial::operator==(const Polynomial& other) const {
    return coeffs == other.coeffs;
}
```

```cpp
// Output operator
ostream& operator<<(ostream& out, const Polynomial& poly) {
    for (int i = poly.coeffs.size() - 1; i >= 0; --i) {
        if (i != poly.coeffs.size() - 1) out << (poly.coeffs[i] >= 0 ? "+" : "");
        out << poly.coeffs[i];
        if (i > 0) out << "x^" << i;
    }
    return out;
}


// Utility functions
int Polynomial::degree() const {
    return coeffs.size() - 1;
}


double Polynomial::evaluate(double x) const {
    double result = 0.0;
    for (int i = degree(); i >= 0; --i) {
        result = result * x + coeffs[i];
    }
    return result;
}


double Polynomial::getCoefficient(int degree) const {
    if (degree < 0 || degree >= coeffs.size()) return 0.0;
    return coeffs[degree];
}


void Polynomial::setCoefficients(const vector<double>& coefficients) {
    coeffs = coefficients;
}
```

Appendix (B)

copilot

Copilot (Code and Implementaion)

Menu

#include "Polynomial.h"

#include "Polynomial.hpp"

```cpp
void displayMenu() {
    cout << "\n===== Polynomial Operations Menu =====\n";
    cout << "1. Enter Polynomial\n";
    cout << "2. Display Polynomial\n";
    cout << "3. Add Two Polynomials\n";
    cout << "4. Subtract Two Polynomials\n";
    cout << "5. Multiply Two Polynomials\n";
    cout << "6. Evaluate Polynomial at a Given Point\n";
    cout << "7. Find Derivative of Polynomial\n";
    cout << "8. Find Integral of Polynomial\n";
    cout << "9. Calculate Definite Integral\n";
    cout << "10. Find Root using Newton's Method\n";
    cout << "0. Exit\n";
    cout << "Choose an option: ";
}


Polynomial inputPolynomial() {
    int degree;
    cout << "Enter the degree of the polynomial: ";
    cin >> degree;

    vector<double> coefficients(degree + 1);
    cout << "Enter the coefficients (from constant to highest degree term): ";
    for (int i = 0; i <= degree; ++i) {
        cin >> coefficients[i];
    }

    return Polynomial(coefficients);
}


int main() {
    Polynomial p1, p2, result;
```

```cpp
bool running = true;

while (running) {
    displayMenu();
    int choice;
    cin >> choice;

    switch (choice) {
        case 1: {
            p1 = inputPolynomial();
            cout << "Polynomial set!\n";
            break;
        }
        case 2: {
            cout << "Polynomial: " << p1 << endl;
            break;
        }
        case 3: {
            cout << "Enter the second polynomial to add:\n";
            p2 = inputPolynomial();
            result = p1 + p2;
            cout << "Result: " << result << endl;
            break;
        }
        case 4: {
            cout << "Enter the second polynomial to subtract:\n";
            p2 = inputPolynomial();
            result = p1 - p2;
            cout << "Result: " << result << endl;
            break;
        }
        case 5: {
            cout << "Enter the second polynomial to multiply:\n";
```

```cpp
            p2 = inputPolynomial();
            result = p1 * p2;
            cout << "Result: " << result << endl;
            break;
        }
        case 6: {
            double x;
            cout << "Enter the value of x: ";
            cin >> x;
            cout << "Value of the polynomial at x = " << x << ": " << p1.evaluate(x) << endl;
            break;
        }
        case 7: {
            result = p1.derivative();
            cout << "Derivative: " << result << endl;
            break;
        }
        case 8: {
            result = p1.integral();
            cout << "Integral: " << result << endl;
            break;
        }
        case 9: {
            double x1, x2;
            cout << "Enter the limits (x1 and x2) for the definite integral: ";
            cin >> x1 >> x2;
            cout << "Definite integral: " << p1.integral(x1, x2) << endl;
            break;
        }
        case 10: {
            double guess, tolerance;
            int maxIter;
            cout << "Enter the initial guess: ";
```

```cpp
                cin >> guess;

                cout << "Enter the tolerance: ";

                cin >> tolerance;

                cout << "Enter the maximum number of iterations: ";

                cin >> maxIter;

                try {

                    double root = p1.getRoot(guess, tolerance, maxIter);

                    cout << "Root found at: " << root << endl;

                } catch (const runtime_error& e) {

                    cout << e.what() << endl;

                }

                break;

            }

            case 0:

                running = false;

                cout << "Exiting program. Goodbye!\n";

                break;

            default:

                cout << "Invalid choice. Please try again.\n";

                break;

        }

    }


    return 0;

}
```

Implementaion

```cpp
#include "Polynomial.h"


// Constructors

Polynomial::Polynomial() : coeffs(1, 0.0) {}
```

```cpp
Polynomial::Polynomial(const vector<double>& coefficients) : coeffs(coefficients) {}


Polynomial::Polynomial(const Polynomial& other) : coeffs(other.coeffs) {}


// Destructor
Polynomial::~Polynomial() {}


// Assignment operator
Polynomial& Polynomial::operator=(const Polynomial& other) {
    if (this != &other) {
        coeffs = other.coeffs;
    }
    return *this;
}


// Arithmetic operators
Polynomial Polynomial::operator+(const Polynomial& other) const {
    int max_deg = max(coeffs.size(), other.coeffs.size());
    vector<double> result_coeffs(max_deg, 0.0);
    for (int i = 0; i < max_deg; ++i) {
        if (i < coeffs.size()) result_coeffs[i] += coeffs[i];
        if (i < other.coeffs.size()) result_coeffs[i] += other.coeffs[i];
    }
    return Polynomial(result_coeffs);
}


Polynomial Polynomial::operator-(const Polynomial& other) const {
    int max_deg = max(coeffs.size(), other.coeffs.size());
    vector<double> result_coeffs(max_deg, 0.0);
    for (int i = 0; i < max_deg; ++i) {
        if (i < coeffs.size()) result_coeffs[i] += coeffs[i];
        if (i < other.coeffs.size()) result_coeffs[i] -= other.coeffs[i];
    }
```

```cpp
    return Polynomial(result_coeffs);

}


Polynomial Polynomial::operator*(const Polynomial& other) const {

  vector<double> result_coeffs(coeffs.size() + other.coeffs.size() - 1, 0.0);

  for (int i = 0; i < coeffs.size(); ++i) {

    for (int j = 0; j < other.coeffs.size(); ++j) {

      result_coeffs[i + j] += coeffs[i] * other.coeffs[j];

    }

  }

  return Polynomial(result_coeffs);

}


// Equality operator

bool Polynomial::operator==(const Polynomial& other) const {

  return coeffs == other.coeffs;

}


// Output operator

ostream& operator<<(ostream& out, const Polynomial& poly) {

  for (int i = poly.coeffs.size() - 1; i >= 0; --i) {

    if (poly.coeffs[i] != 0) {

      if (i != poly.coeffs.size() - 1 && poly.coeffs[i] > 0)

        out << "+";

      out << poly.coeffs[i];

      if (i > 0)

        out << "x^" << i << " ";

    }

  }

  return out;

}


// Utility functions
```

```cpp
int Polynomial::degree() const {

    return coeffs.size() - 1;

}


double Polynomial::evaluate(double x) const {

    double result = 0.0;

    for (int i = 0; i < coeffs.size(); ++i) {

        result += coeffs[i] * pow(x, i);

    }

    return result;

}


Polynomial Polynomial::compose(const Polynomial& q) const {

    Polynomial result;

    for (int i = coeffs.size() - 1; i >= 0; --i) {

        result = result * q + Polynomial(vector<double>{coeffs[i]});

    }

    return result;

}


Polynomial Polynomial::derivative() const {

    vector<double> result_coeffs(coeffs.size() - 1, 0.0);

    for (int i = 1; i < coeffs.size(); ++i) {

        result_coeffs[i - 1] = coeffs[i] * i;

    }

    return Polynomial(result_coeffs);

}


Polynomial Polynomial::integral() const {

    vector<double> result_coeffs(coeffs.size() + 1, 0.0);

    for (int i = 0; i < coeffs.size(); ++i) {

        result_coeffs[i + 1] = coeffs[i] / (i + 1);

    }
```

```cpp
        return Polynomial(result_coeffs);
}


double Polynomial::integral(double x1, double x2) const {
    Polynomial integrated = integral();
    return integrated.evaluate(x2) - integrated.evaluate(x1);
}


double Polynomial::getRoot(double guess, double tolerance, int maxIter) {
    Polynomial deriv = derivative();
    for (int i = 0; i < maxIter; ++i) {
        double value = evaluate(guess);
        if (abs(value) < tolerance) return guess;
        double slope = deriv.evaluate(guess);
        if (slope == 0) break;
        guess -= value / slope;
    }
    throw runtime_error("Root not found within the specified tolerance and iterations.");
}


void Polynomial::setCoefficients(const vector<double>& coefficients) {
    coeffs = coefficients;
}


double Polynomial::getCoefficient(int degree) const {
    if (degree < 0 || degree >= coeffs.size())
        throw out_of_range("Degree is out of range.");
    return coeffs[degree];
}
```

Appendix (C)

Test Cases

Test Case (1):

9

4 13 3 2 6 10 10 -8 -12

4

7 -10 10 -2

Expected output:

Polynomial 1: -12*x^8 - 8*x^7 + 10*x^6 + 10*x^5 + 6*x^4 + 2*x^3 + 3*x^2 + 13*x + 4

Polynomial 2: -2*x^3 + 10*x^2 - 10*x + 7

Sum: -12*x^8 - 8*x^7 + 10*x^6 + 10*x^5 + 6*x^4 + 13*x^2 + 3*x + 11

Subtraction: -12*x^8 - 8*x^7 + 10*x^6 + 10*x^5 + 6*x^4 + 4*x^3 - 7*x^2 + 23*x - 3

Product: 24*x^11 - 104*x^10 + 20*x^9 + 76*x^8 - 68*x^7 + 26*x^6 + 24*x^5 + 26*x^4 + 106*x^3 - 69*x^2 + 51*x + 28

Degree of Polynomial: 9

Evaluation of polynomial at x=2: -2982

Derivative: -96*x^7 - 56*x^6 + 60*x^5 + 50*x^4 + 24*x^3 + 6*x^2 + 6*x + 13

Integral: -4*x^9/3 - x^8 + 10*x^7/7 + 5*x^6/3 + 6*x^5/5 + x^4/2 + x^3 + 13*x^2/2 + 4*x

Definite integration from x=0 to x=1: 1466/105

Real roots of Polynomial 1: [-0.331031267379015, 1.25830027776256]

Test Case (2):

10

-11 13 -6 14 -5 -4 3 6 -8 -4

9

5 15 -4 -10 9 4 -13 5 11

Expected output:

Polynomial 1: -4*x^9 - 8*x^8 + 6*x^7 + 3*x^6 - 4*x^5 - 5*x^4 + 14*x^3 - 6*x^2 + 13*x - 11

Polynomial 2: 11*x^8 + 5*x^7 - 13*x^6 + 4*x^5 + 9*x^4 - 10*x^3 - 4*x^2 + 15*x + 5

Sum: -4*x^9 + 3*x^8 + 11*x^7 - 10*x^6 + 4*x^4 + 4*x^3 - 10*x^2 + 28*x - 6

Subtraction: -4*x^9 - 19*x^8 + x^7 + 16*x^6 - 8*x^5 - 14*x^4 + 24*x^3 - 2*x^2 - 2*x - 16

Product: -44*x^17 - 108*x^16 + 78*x^15 + 151*x^14 - 175*x^13 - 122*x^12 + 343*x^11 - 8*x^10 - 319*x^9 + 111*x^8 + 19*x^7 - 24*x^6 - 18*x^5 - 20*x^4 + 38*x^3 + 209*x^2 - 100*x - 55

Degree of Polynomial: 10

Evaluation of polynomial at x=2: -3241

Derivative: -36*x^8 - 64*x^7 + 42*x^6 + 18*x^5 - 20*x^4 - 20*x^3 + 42*x^2 - 12*x + 13

Integral: -2*x^10/5 - 8*x^9/9 + 3*x^8/4 + 3*x^7/7 - 2*x^6/3 - x^5 + 7*x^4/2 - 2*x^3 + 13*x^2/2 - 11*x

Definite integration from x=0 to x=1: -6019/1260

Real roots of Polynomial 1: [-2.49764339948220]

Test Case (3):

11

8 4 -8 -9 8 -8 -13 4 -3 7 -4

12

-2 -6 2 12 -11 -3 13 15 3 10 -9 14

Expected output:

Polynomial 1: -4*x^10 + 7*x^9 - 3*x^8 + 4*x^7 - 13*x^6 - 8*x^5 + 8*x^4 - 9*x^3 - 8*x^2 + 4*x + 8

Polynomial 2: 14*x^11 - 9*x^10 + 10*x^9 + 3*x^8 + 15*x^7 + 13*x^6 - 3*x^5 - 11*x^4 + 12*x^3 + 2*x^2 - 6*x - 2

Sum: 14*x^11 - 13*x^10 + 17*x^9 + 19*x^7 - 11*x^5 - 3*x^4 + 3*x^3 - 6*x^2 - 2*x + 6

Subtraction: -14*x^11 + 5*x^10 - 3*x^9 - 6*x^8 - 11*x^7 - 26*x^6 - 5*x^5 + 19*x^4 - 21*x^3 - 10*x^2 + 10*x + 10

Product: -56*x^21 + 134*x^20 - 145*x^19 + 141*x^18 - 287*x^17 + 89*x^16 + 124*x^15 - 273*x^14 - 234*x^13 - 130*x^12 - 18*x^11 + 88*x^10 - 225*x^9 - 221*x^8 + 445*x^7 + 162*x^6 - 214*x^5 - 18*x^4 + 170*x^3 + 8*x^2 - 56*x - 16

Degree of Polynomial: 11

Evaluation of polynomial at x=2: -1816

Derivative: -40*x^9 + 63*x^8 - 24*x^7 + 28*x^6 - 78*x^5 - 40*x^4 + 32*x^3 - 27*x^2 - 16*x + 4

Integral: -4*x^11/11 + 7*x^10/10 - x^9/3 + x^8/2 - 13*x^7/7 - 4*x^6/3 + 8*x^5/5 - 9*x^4/4 - 8*x^3/3 + 2*x^2 + 8*x

Definite integration from x=0 to x=1: 18461/4620

Real roots of Polynomial 1: [-0.903702020607161, 0.789936856605057]

Test Case (4):

3

12 -6 -5

6

-2 1 -2 -2 6 15

Expected output:

Polynomial 1: -5*x^2 - 6*x + 12

Polynomial 2: 15*x^5 + 6*x^4 - 2*x^3 - 2*x^2 + x - 2

Sum: 15*x^5 + 6*x^4 - 2*x^3 - 7*x^2 - 5*x + 10

Subtraction: -15*x^5 - 6*x^4 + 2*x^3 - 3*x^2 - 7*x + 14

Product: -75*x^7 - 120*x^6 + 154*x^5 + 94*x^4 - 17*x^3 - 20*x^2 + 24*x - 24

Degree of Polynomial: 3

Evaluation of polynomial at x=2: -20

Derivative: -10*x - 6

Integral: -5*x^3/3 - 3*x^2 + 12*x

Definite integration from x=0 to x=1: 22/3

Real roots of Polynomial 1: [-2.26132477258361, 1.06132477258362]

Test Case (5):

10

0 15 -1 14 5 -8 9 -12 -10 6

11

-1 10 -13 -10 2 -8 -5 -9 -4 2 12

Expected output:

Polynomial 1: 6*x^9 - 10*x^8 - 12*x^7 + 9*x^6 - 8*x^5 + 5*x^4 + 14*x^3 - x^2 + 15*x

Polynomial 2: 12*x^10 + 2*x^9 - 4*x^8 - 9*x^7 - 5*x^6 - 8*x^5 + 2*x^4 - 10*x^3 - 13*x^2 + 10*x - 1

Sum: 12*x^10 + 8*x^9 - 14*x^8 - 21*x^7 + 4*x^6 - 16*x^5 + 7*x^4 + 4*x^3 - 14*x^2 + 25*x - 1

Subtraction: -12*x^10 + 4*x^9 - 6*x^8 - 3*x^7 + 14*x^6 + 3*x^4 + 24*x^3 + 12*x^2 + 5*x + 1

Product: 72*x^19 - 108*x^18 - 188*x^17 + 70*x^16 + 30*x^15 + 118*x^14 + 281*x^13 + 39*x^12 + 43*x^11 + 275*x^10 - 217*x^9 - 379*x^8 + 117*x^7 - 416*x^6 - 84*x^5 - 2*x^4 - 219*x^3 + 151*x^2 - 15*x

Degree of Polynomial: 10

Evaluation of polynomial at x=2: -486

Derivative: 54*x^8 - 80*x^7 - 84*x^6 + 54*x^5 - 40*x^4 + 20*x^3 + 42*x^2 - 2*x + 15

Integral: 3*x^10/5 - 10*x^9/9 - 3*x^8/2 + 9*x^7/7 - 4*x^6/3 + x^5 + 7*x^4/2 - x^3/3 + 15*x^2/2

Definite integration from x=0 to x=1: 6053/630

Real roots of Polynomial 1: [0, 1.21963019623181, 2.29111625392896]

Test Case (6):

11

7 -6 -2 -3 2 -6 14 -3 -4 6 10

5

0 -13 13 1 -8

Expected output:

Polynomial 1: 10*x^10 + 6*x^9 - 4*x^8 - 3*x^7 + 14*x^6 - 6*x^5 + 2*x^4 - 3*x^3 - 2*x^2 - 6*x + 7

Polynomial 2: -8*x^4 + x^3 + 13*x^2 - 13*x

Sum: 10*x^10 + 6*x^9 - 4*x^8 - 3*x^7 + 14*x^6 - 6*x^5 - 6*x^4 - 2*x^3 + 11*x^2 - 19*x + 7

Subtraction: 10*x^10 + 6*x^9 - 4*x^8 - 3*x^7 + 14*x^6 - 6*x^5 + 10*x^4 - 4*x^3 - 15*x^2 + 7*x + 7

Product: -80*x^14 - 38*x^13 + 168*x^12 - 32*x^11 - 245*x^10 + 75*x^9 + 199*x^8 - 234*x^7 + 117*x^6 - 19*x^5 - 49*x^4 - 45*x^3 + 169*x^2 - 91*x

Degree of Polynomial: 11

Evaluation of polynomial at x=2: 12603

Derivative: 100*x^9 + 54*x^8 - 32*x^7 - 21*x^6 + 84*x^5 - 30*x^4 + 8*x^3 - 9*x^2 - 4*x - 6

Integral: 10*x^11/11 + 3*x^10/5 - 4*x^9/9 - 3*x^8/8 + 2*x^7 - x^6 + 2*x^5/5 - 3*x^4/4 - 2*x^3/3 - 3*x^2 + 7*x

Definite integration from x=0 to x=1: 3701/792

Real roots of Polynomial 1: []

Test Case (7):

9

-13 9 -4 -1 -11 12 14 -4 -13

10

15 3 0 14 12 12 -14 -13 2 6

Expected output:

Polynomial 1: -13*x^8 - 4*x^7 + 14*x^6 + 12*x^5 - 11*x^4 - x^3 - 4*x^2 + 9*x - 13

Polynomial 2: 6*x^9 + 2*x^8 - 13*x^7 - 14*x^6 + 12*x^5 + 12*x^4 + 14*x^3 + 3*x + 15

Sum: 6*x^9 - 11*x^8 - 17*x^7 + 24*x^5 + x^4 + 13*x^3 - 4*x^2 + 12*x + 2

Subtraction: -6*x^9 - 15*x^8 + 9*x^7 + 28*x^6 - 23*x^4 - 15*x^3 - 4*x^2 + 6*x - 28

Product: -78*x^17 - 50*x^16 + 245*x^15 + 334*x^14 - 324*x^13 - 584*x^12 - 113*x^11 + 469*x^10 + 175*x^9 - 270*x^8 - 189*x^7 + 474*x^6 + 43*x^5 - 198*x^4 - 209*x^3 - 33*x^2 + 96*x - 195

Degree of Polynomial: 9

Evaluation of polynomial at x=2: -2755

Derivative: -104*x^7 - 28*x^6 + 84*x^5 + 60*x^4 - 44*x^3 - 3*x^2 - 8*x + 9

Integral: -13*x^9/9 - x^8/2 + 2*x^7 + 2*x^6 - 11*x^5/5 - x^4/4 - 4*x^3/3 + 9*x^2/2 - 13*x

Definite integration from x=0 to x=1: -1841/180

Real roots of Polynomial 1: []

Test Case (8):

11

1 6 13 5 -5 12 10 11 -14 -4 0

12

-8 0 2 -7 -13 -7 5 -13 -2 6 15 -5

Expected output:

Polynomial 1: -4*x^9 - 14*x^8 + 11*x^7 + 10*x^6 + 12*x^5 - 5*x^4 + 5*x^3 + 13*x^2 + 6*x + 1

Polynomial 2: -5*x^11 + 15*x^10 + 6*x^9 - 2*x^8 - 13*x^7 + 5*x^6 - 7*x^5 - 13*x^4 - 7*x^3 + 2*x^2 - 8

Sum: -5*x^11 + 15*x^10 + 2*x^9 - 16*x^8 - 2*x^7 + 15*x^6 + 5*x^5 - 18*x^4 - 2*x^3 + 15*x^2 + 6*x - 7

Subtraction: 5*x^11 - 15*x^10 - 10*x^9 - 12*x^8 + 24*x^7 + 5*x^6 + 19*x^5 + 8*x^4 + 12*x^3 + 11*x^2 + 6*x + 9

Product: 20*x^20 + 10*x^19 - 289*x^18 + 39*x^17 + 236*x^16 + 405*x^15 - 233*x^14 + 31*x^13 + 232*x^12 + 155*x^11 - 384*x^10 - 287*x^9 + 63*x^8 - 168*x^7 - 331*x^6 - 262*x^5 + 11*x^4 - 35*x^3 - 102*x^2 - 48*x - 8

Degree of Polynomial: 11

Evaluation of polynomial at x=2: -3175

Derivative: -36*x^8 - 112*x^7 + 77*x^6 + 60*x^5 + 60*x^4 - 20*x^3 + 15*x^2 + 26*x + 6

Integral: -2*x^10/5 - 14*x^9/9 + 11*x^8/8 + 10*x^7/7 + 2*x^6 - x^5 + 5*x^4/4 + 13*x^3/3 + 3*x^2 + x

Definite integration from x=0 to x=1: 28807/2520

Real roots of Polynomial 1: [-4.07379773511831, -0.486036721404166, 1.39252192280508]

Test Case (9):

2

-4 10

6

-11 2 12 -7 -2 10

Expected output:

Polynomial 1: 10*x - 4

Polynomial 2: 10*x^5 - 2*x^4 - 7*x^3 + 12*x^2 + 2*x - 11

Sum: 10*x^5 - 2*x^4 - 7*x^3 + 12*x^2 + 12*x - 15

Subtraction: -10*x^5 + 2*x^4 + 7*x^3 - 12*x^2 + 8*x + 7

Product: 100*x^6 - 60*x^5 - 62*x^4 + 148*x^3 - 28*x^2 - 118*x + 44

Degree of Polynomial: 2

Evaluation of polynomial at x=2: 16

Derivative: 10

Integral: 5*x^2 - 4*x

Definite integration from x=0 to x=1: 1

Real roots of Polynomial 1: [0.400000000000000]

Test Case (10):

12

-9 -14 3 13 4 15 6 -2 -3 -11 -13 10

2

-12 -2

Expected output:

Polynomial 1: 10*x^11 - 13*x^10 - 11*x^9 - 3*x^8 - 2*x^7 + 6*x^6 + 15*x^5 + 4*x^4 + 13*x^3 + 3*x^2 - 14*x - 9

Polynomial 2: -2*x - 12

Sum: 10*x^11 - 13*x^10 - 11*x^9 - 3*x^8 - 2*x^7 + 6*x^6 + 15*x^5 + 4*x^4 + 13*x^3 + 3*x^2 - 16*x - 21

Subtraction: 10*x^11 - 13*x^10 - 11*x^9 - 3*x^8 - 2*x^7 + 6*x^6 + 15*x^5 + 4*x^4 + 13*x^3 + 3*x^2 - 12*x + 3

Product: -20*x^12 - 94*x^11 + 178*x^10 + 138*x^9 + 40*x^8 + 12*x^7 - 102*x^6 - 188*x^5 - 74*x^4 - 162*x^3 - 8*x^2 + 186*x + 108

Degree of Polynomial: 12

Evaluation of polynomial at x=2: 1519

Derivative: 110*x^10 - 130*x^9 - 99*x^8 - 24*x^7 - 14*x^6 + 36*x^5 + 75*x^4 + 16*x^3 + 39*x^2 + 6*x - 14

Integral: 5*x^12/6 - 13*x^11/11 - 11*x^10/10 - x^9/3 - x^8/4 + 6*x^7/7 + 5*x^6/2 + 4*x^5/5 + 13*x^4/4 + x^3 - 7*x^2 - 9*x

Definite integration from x=0 to x=1: -7411/770

Real roots of Polynomial 1: [1.86914934288554]

Test Case (11):

10

4 -6 8 14 6 4 6 1 5 -7

5

2 -10 -13 10 7

Expected output:

Polynomial 1: -7*x^9 + 5*x^8 + x^7 + 6*x^6 + 4*x^5 + 6*x^4 + 14*x^3 + 8*x^2 - 6*x + 4

Polynomial 2: 7*x^4 + 10*x^3 - 13*x^2 - 10*x + 2

Sum: -7*x^9 + 5*x^8 + x^7 + 6*x^6 + 4*x^5 + 13*x^4 + 24*x^3 - 5*x^2 - 16*x + 6

Subtraction: -7*x^9 + 5*x^8 + x^7 + 6*x^6 + 4*x^5 - x^4 + 4*x^3 + 21*x^2 + 4*x + 2

Product: -49*x^13 - 35*x^12 + 148*x^11 + 57*x^10 + 11*x^9 + 4*x^8 + 48*x^7 + 90*x^6 - 196*x^5 - 264*x^4 + 66*x^3 + 24*x^2 - 52*x + 8

Degree of Polynomial: 10

Evaluation of polynomial at x=2: -1432

Derivative: -63*x^8 + 40*x^7 + 7*x^6 + 36*x^5 + 20*x^4 + 24*x^3 + 42*x^2 + 16*x - 6

Integral: -7*x^10/10 + 5*x^9/9 + x^8/8 + 6*x^7/7 + 2*x^6/3 + 6*x^5/5 + 7*x^4/2 + 8*x^3/3 - 3*x^2 + 4*x

Definite integration from x=0 to x=1: 4975/504

Real roots of Polynomial 1: [1.61751110316682]

Test Case (12):

4

-12 3 -8 -14

3

0 14 3

Expected output:

Polynomial 1: -14*x^3 - 8*x^2 + 3*x - 12

Polynomial 2: 3*x^2 + 14*x

Sum: -14*x^3 - 5*x^2 + 17*x - 12

Subtraction: -14*x^3 - 11*x^2 - 11*x - 12

Product: -42*x^5 - 220*x^4 - 103*x^3 + 6*x^2 - 168*x

Degree of Polynomial: 4

Evaluation of polynomial at x=2: -150

Derivative: -42*x^2 - 16*x + 3

Integral: -7*x^4/2 - 8*x^3/3 + 3*x^2/2 - 12*x

Definite integration from x=0 to x=1: -50/3

Real roots of Polynomial 1: [-1.27080610264932]

Test Case (13):

1

-7

10

6 -3 -1 -9 5 14 -11 10 -12 -7

Expected output:

Polynomial 1: -7

Polynomial 2: -7*x^9 - 12*x^8 + 10*x^7 - 11*x^6 + 14*x^5 + 5*x^4 - 9*x^3 - x^2 - 3*x + 6

Sum: -7*x^9 - 12*x^8 + 10*x^7 - 11*x^6 + 14*x^5 + 5*x^4 - 9*x^3 - x^2 - 3*x - 1

Subtraction: 7*x^9 + 12*x^8 - 10*x^7 + 11*x^6 - 14*x^5 - 5*x^4 + 9*x^3 + x^2 + 3*x - 13

Product: 49*x^9 + 84*x^8 - 70*x^7 + 77*x^6 - 98*x^5 - 35*x^4 + 63*x^3 + 7*x^2 + 21*x - 42

Degree of Polynomial: 1

Evaluation of polynomial at x=2: -7

Derivative: 0

Integral: -7*x

Definite integration from x=0 to x=1: -7

Real roots of Polynomial 1: []

Test Case (14):

9

3 10 3 6 0 -8 7 15 -3

10

8 3 1 -3 -5 3 -8 3 -11 3

Expected output:

Polynomial 1: -3*x^8 + 15*x^7 + 7*x^6 - 8*x^5 + 6*x^3 + 3*x^2 + 10*x + 3

Polynomial 2: 3*x^9 - 11*x^8 + 3*x^7 - 8*x^6 + 3*x^5 - 5*x^4 - 3*x^3 + x^2 + 3*x + 8

Sum: 3*x^9 - 14*x^8 + 18*x^7 - x^6 - 5*x^5 - 5*x^4 + 3*x^3 + 4*x^2 + 13*x + 11

Subtraction: -3*x^9 + 8*x^8 + 12*x^7 + 15*x^6 - 11*x^5 + 5*x^4 + 9*x^3 + 2*x^2 + 7*x - 5

Product: -9*x^17 + 78*x^16 - 153*x^15 - 32*x^14 - 20*x^13 - 2*x^12 - 38*x^11 - 92*x^10 - 115*x^9 + 43*x^8 + 41*x^7 + 5*x^6 - 108*x^5 - 24*x^4 + 58*x^3 + 57*x^2 + 89*x + 24

Degree of Polynomial: 9

Evaluation of polynomial at x=2: 1427

Derivative: -24*x^7 + 105*x^6 + 42*x^5 - 40*x^4 + 18*x^2 + 6*x + 10

Integral: -x^9/3 + 15*x^8/8 + x^7 - 4*x^6/3 + 3*x^4/2 + x^3 + 5*x^2 + 3*x

Definite integration from x=0 to x=1: 281/24

Real roots of Polynomial 1: [-0.313616526318418, 5.34598620115800]

Test Case (15):

9

10 15 7 10 10 4 8 -1 3

6

-4 -2 3 14 8 12

Expected output:

Polynomial 1: 3*x^8 - x^7 + 8*x^6 + 4*x^5 + 10*x^4 + 10*x^3 + 7*x^2 + 15*x + 10

Polynomial 2: 12*x^5 + 8*x^4 + 14*x^3 + 3*x^2 - 2*x - 4

Sum: 3*x^8 - x^7 + 8*x^6 + 16*x^5 + 18*x^4 + 24*x^3 + 10*x^2 + 13*x + 6

Subtraction: 3*x^8 - x^7 + 8*x^6 - 8*x^5 + 2*x^4 - 4*x^3 + 4*x^2 + 17*x + 14

Product: 36*x^13 + 12*x^12 + 130*x^11 + 107*x^10 + 255*x^9 + 270*x^8 + 304*x^7 + 366*x^6 + 332*x^5 + 251*x^4 + 131*x^3 - 28*x^2 - 80*x - 40

Degree of Polynomial: 9

Evaluation of polynomial at x=2: 1588

Derivative: 24*x^7 - 7*x^6 + 48*x^5 + 20*x^4 + 40*x^3 + 30*x^2 + 14*x + 15

Integral: x^9/3 - x^8/8 + 8*x^7/7 + 2*x^6/3 + 2*x^5 + 5*x^4/2 + 7*x^3/3 + 15*x^2/2 + 10*x

Definite integration from x=0 to x=1: 4427/168

Real roots of Polynomial 1: []

Test Case (16):

12

-1 -13 -3 10 -11 10 -10 10 -10 0 6 -8

12

13 -7 -8 12 7 5 12 14 -5 -6 8 12

Expected output:

Polynomial 1: -8*x^11 + 6*x^10 - 10*x^8 + 10*x^7 - 10*x^6 + 10*x^5 - 11*x^4 + 10*x^3 - 3*x^2 - 13*x - 1

Polynomial 2: 12*x^11 + 8*x^10 - 6*x^9 - 5*x^8 + 14*x^7 + 12*x^6 + 5*x^5 + 7*x^4 + 12*x^3 - 8*x^2 - 7*x + 13

Sum: 4*x^11 + 14*x^10 - 6*x^9 - 15*x^8 + 24*x^7 + 2*x^6 + 15*x^5 - 4*x^4 + 22*x^3 - 11*x^2 - 20*x + 12

Subtraction: -20*x^11 - 2*x^10 + 6*x^9 - 5*x^8 - 4*x^7 - 22*x^6 + 5*x^5 - 18*x^4 - 2*x^3 + 5*x^2 - 6*x - 14

Product: -96*x^22 + 8*x^21 + 96*x^20 - 116*x^19 - 102*x^18 + 8*x^17 + 62*x^16 - 208*x^15 - 12*x^14 + 126*x^13 - 177*x^12 - 428*x^11 + 351*x^10 + 34*x^9 - 240*x^8 - 127*x^7 - 90*x^6 - 5*x^5 - 352*x^4 + 243*x^3 + 60*x^2 - 162*x - 13

Degree of Polynomial: 12

Evaluation of polynomial at x=2: -11975

Derivative: -88*x^10 + 60*x^9 - 80*x^7 + 70*x^6 - 60*x^5 + 50*x^4 - 44*x^3 + 30*x^2 - 6*x - 13

Integral: -2*x^12/3 + 6*x^11/11 - 10*x^9/9 + 5*x^8/4 - 10*x^7/7 + 5*x^6/3 - 11*x^5/5 + 5*x^4/2 - x^3 - 13*x^2/2 - x

Definite integration from x=0 to x=1: -110107/13860

Real roots of Polynomial 1: [-1.35923528211261, -0.617892783493185, -0.0787657776801827]

Test Case (17):

10

-3 6 -8 5 9 -13 2 -10 -10 15

2

-12 -4

Expected output:

Polynomial 1: 15*x^9 - 10*x^8 - 10*x^7 + 2*x^6 - 13*x^5 + 9*x^4 + 5*x^3 - 8*x^2 + 6*x - 3

Polynomial 2: -4*x - 12

Sum: 15*x^9 - 10*x^8 - 10*x^7 + 2*x^6 - 13*x^5 + 9*x^4 + 5*x^3 - 8*x^2 + 2*x - 15

Subtraction: 15*x^9 - 10*x^8 - 10*x^7 + 2*x^6 - 13*x^5 + 9*x^4 + 5*x^3 - 8*x^2 + 10*x + 9

Product: -60*x^10 - 140*x^9 + 160*x^8 + 112*x^7 + 28*x^6 + 120*x^5 - 128*x^4 - 28*x^3 + 72*x^2 - 60*x + 36

Degree of Polynomial: 10

Evaluation of polynomial at x=2: 3713

Derivative: 135*x^8 - 80*x^7 - 70*x^6 + 12*x^5 - 65*x^4 + 36*x^3 + 15*x^2 - 16*x + 6

Integral: 3*x^10/2 - 10*x^9/9 - 5*x^8/4 + 2*x^7/7 - 13*x^6/6 + 9*x^5/5 + 5*x^4/4 - 8*x^3/3 + 3*x^2 - 3*x

Definite integration from x=0 to x=1: -743/315

Real roots of Polynomial 1: [1.27791195205482]

Test Case (18):

10

10 -5 -6 10 -5 9 -2 13 -9 -3

6

-7 -14 11 -8 -2 -10

Expected output:

Polynomial 1: -3*x^9 - 9*x^8 + 13*x^7 - 2*x^6 + 9*x^5 - 5*x^4 + 10*x^3 - 6*x^2 - 5*x + 10

Polynomial 2: -10*x^5 - 2*x^4 - 8*x^3 + 11*x^2 - 14*x - 7

Sum: -3*x^9 - 9*x^8 + 13*x^7 - 2*x^6 - x^5 - 7*x^4 + 2*x^3 + 5*x^2 - 19*x + 3

Subtraction: -3*x^9 - 9*x^8 + 13*x^7 - 2*x^6 + 19*x^5 - 3*x^4 + 18*x^3 - 17*x^2 + 9*x + 17

Product: 30*x^14 + 96*x^13 - 88*x^12 + 33*x^11 - 247*x^10 + 338*x^9 - 303*x^8 + 116*x^7 - 185*x^6 + 75*x^5 - 151*x^4 - 121*x^3 + 222*x^2 - 105*x - 70

Degree of Polynomial: 10

Evaluation of polynomial at x=2: -2040

Derivative: -27*x^8 - 72*x^7 + 91*x^6 - 12*x^5 + 45*x^4 - 20*x^3 + 30*x^2 - 12*x - 5

Integral: -3*x^10/10 - x^9 + 13*x^8/8 - 2*x^7/7 + 3*x^6/2 - x^5 + 5*x^4/2 - 2*x^3 - 5*x^2/2 + 10*x

Definite integration from x=0 to x=1: 2391/280

Real roots of Polynomial 1: [-4.13738236859391, -0.745211506059180, 1.30687498187249]

Test Case (19):

10

12 0 3 3 3 1 0 -6 12 -12

4

7 13 -10 -13

Expected output:

Polynomial 1: -12*x^9 + 12*x^8 - 6*x^7 + x^5 + 3*x^4 + 3*x^3 + 3*x^2 + 12

Polynomial 2: -13*x^3 - 10*x^2 + 13*x + 7

Sum: -12*x^9 + 12*x^8 - 6*x^7 + x^5 + 3*x^4 - 10*x^3 - 7*x^2 + 13*x + 19

Subtraction: -12*x^9 + 12*x^8 - 6*x^7 + x^5 + 3*x^4 + 16*x^3 + 13*x^2 - 13*x + 5

Product: 156*x^12 - 36*x^11 - 198*x^10 + 132*x^9 - 7*x^8 - 91*x^7 - 56*x^6 - 23*x^5 + 30*x^4 - 96*x^3 - 99*x^2 + 156*x + 84

Degree of Polynomial: 10

Evaluation of polynomial at x=2: -3712

Derivative: -108*x^8 + 96*x^7 - 42*x^6 + 5*x^4 + 12*x^3 + 9*x^2 + 6*x

Integral: -6*x^10/5 + 4*x^9/3 - 3*x^8/4 + x^6/6 + 3*x^5/5 + 3*x^4/4 + x^3 + 12*x

Definite integration from x=0 to x=1: 139/10

Real roots of Polynomial 1: [1.19145175745237]

Test Case (20):

2

-7 10

5

-6 15 9 -2 -7

Expected output:

Polynomial 1: 10*x - 7

Polynomial 2: -7*x^4 - 2*x^3 + 9*x^2 + 15*x - 6

Sum: -7*x^4 - 2*x^3 + 9*x^2 + 25*x - 13

Subtraction: 7*x^4 + 2*x^3 - 9*x^2 - 5*x - 1

Product: -70*x^5 + 29*x^4 + 104*x^3 + 87*x^2 - 165*x + 42

Degree of Polynomial: 2

Evaluation of polynomial at x=2: 13

Derivative: 10

Integral: 5*x^2 - 7*x

Definite integration from x=0 to x=1: -2

Real roots of Polynomial 1: [0.700000000000000]

Test Case (21):

7

5 -6 -12 -5 -10 -12 5

6

-7 -7 -3 2 2 13

Expected output:

Polynomial 1: 5*x^6 - 12*x^5 - 10*x^4 - 5*x^3 - 12*x^2 - 6*x + 5

Polynomial 2: 13*x^5 + 2*x^4 + 2*x^3 - 3*x^2 - 7*x - 7

Sum: 5*x^6 + x^5 - 8*x^4 - 3*x^3 - 15*x^2 - 13*x - 2

Subtraction: 5*x^6 - 25*x^5 - 12*x^4 - 7*x^3 - 9*x^2 + x + 12

Product: 65*x^11 - 146*x^10 - 144*x^9 - 124*x^8 - 185*x^7 - 33*x^6 + 198*x^5 + 139*x^4 + 147*x^3 + 111*x^2 + 7*x - 35

Degree of Polynomial: 7

Evaluation of polynomial at x=2: -319

Derivative: 30*x^5 - 60*x^4 - 40*x^3 - 15*x^2 - 24*x - 6

Integral: 5*x^7/7 - 2*x^6 - 2*x^5 - 5*x^4/4 - 4*x^3 - 3*x^2 + 5*x

Definite integration from x=0 to x=1: -183/28

Real roots of Polynomial 1: [0.400073423967370, 3.20326958374905]

Test Case (22):

4

15 -4 10 -9

1

2

Expected output:

Polynomial 1: -9*x^3 + 10*x^2 - 4*x + 15

Polynomial 2: 2

Sum: -9*x^3 + 10*x^2 - 4*x + 17

Subtraction: -9*x^3 + 10*x^2 - 4*x + 13

Product: -18*x^3 + 20*x^2 - 8*x + 30

Degree of Polynomial: 4

Evaluation of polynomial at x=2: -25

Derivative: -27*x^2 + 20*x - 4

Integral: -9*x^4/4 + 10*x^3/3 - 2*x^2 + 15*x

Definite integration from x=0 to x=1: 169/12

Real roots of Polynomial 1: [1.53149536957870]

Test Case (23):

2

0 -8

1

7

Expected output:

Polynomial 1: -8*x

Polynomial 2: 7

Sum: 7 - 8*x

Subtraction: -8*x - 7

Product: -56*x

Degree of Polynomial: 2

Evaluation of polynomial at x=2: -16

Derivative: -8

Integral: -4*x^2

Definite integration from x=0 to x=1: -4

Real roots of Polynomial 1: [0]

Test Case (24):

11

0 14 14 5 -3 -14 2 0 -5 -5 11

10

3 13 1 10 9 -8 -10 -1 -5 10

Expected output:

Polynomial 1: 11*x^10 - 5*x^9 - 5*x^8 + 2*x^6 - 14*x^5 - 3*x^4 + 5*x^3 + 14*x^2 + 14*x

Polynomial 2: 10*x^9 - 5*x^8 - x^7 - 10*x^6 - 8*x^5 + 9*x^4 + 10*x^3 + x^2 + 13*x + 3

Sum: 11*x^10 + 5*x^9 - 10*x^8 - x^7 - 8*x^6 - 22*x^5 + 6*x^4 + 15*x^3 + 15*x^2 + 27*x + 3

Subtraction: 11*x^10 - 15*x^9 + x^7 + 12*x^6 - 6*x^5 - 12*x^4 - 5*x^3 + 13*x^2 + x - 3

Product: 110*x^19 - 105*x^18 - 36*x^17 - 80*x^16 - 13*x^15 + 39*x^14 + 143*x^13 - 25*x^12 + 330*x^11 + 188*x^10 - 296*x^9 - 374*x^8 - 225*x^7 - 115*x^6 + 190*x^5 + 210*x^4 + 211*x^3 + 224*x^2 + 42*x

Degree of Polynomial: 11

Evaluation of polynomial at x=2: 7180

Derivative: 110*x^9 - 45*x^8 - 40*x^7 + 12*x^5 - 70*x^4 - 12*x^3 + 15*x^2 + 28*x + 14

Integral: x^11 - x^10/2 - 5*x^9/9 + 2*x^7/7 - 7*x^6/3 - 3*x^5/5 + 5*x^4/4 + 14*x^3/3 + 7*x^2

Definite integration from x=0 to x=1: 12869/1260

Real roots of Polynomial 1: [-0.797516312888307, 0]

Test Case (25):

9

15 1 -1 -8 9 12 -11 -14 1

4

-10 11 -4 -9

Expected output:

Polynomial 1: x^8 - 14*x^7 - 11*x^6 + 12*x^5 + 9*x^4 - 8*x^3 - x^2 + x + 15

Polynomial 2: -9*x^3 - 4*x^2 + 11*x - 10

Sum: x^8 - 14*x^7 - 11*x^6 + 12*x^5 + 9*x^4 - 17*x^3 - 5*x^2 + 12*x + 5

Subtraction: x^8 - 14*x^7 - 11*x^6 + 12*x^5 + 9*x^4 + x^3 + 3*x^2 - 10*x + 25

Product: -9*x^11 + 122*x^10 + 166*x^9 - 228*x^8 - 110*x^7 + 278*x^6 + 20*x^5 - 183*x^4 - 70*x^3 - 39*x^2 + 155*x - 150

Degree of Polynomial: 9

Evaluation of polynomial at x=2: -1763

Derivative: 8*x^7 - 98*x^6 - 66*x^5 + 60*x^4 + 36*x^3 - 24*x^2 - 2*x + 1

Integral: x^9/9 - 7*x^8/4 - 11*x^7/7 + 2*x^6 + 9*x^5/5 - 2*x^4 - x^3/3 + x^2/2 + 15*x

Definite integration from x=0 to x=1: 17333/1260

Real roots of Polynomial 1: [1.04104142730940, 14.6905126633698]

Test Case (26):

3

6 12 -3

6

14 -9 4 0 0 13

Expected output:

Polynomial 1: -3*x^2 + 12*x + 6

Polynomial 2: 13*x^5 + 4*x^2 - 9*x + 14

Sum: 13*x^5 + x^2 + 3*x + 20

Subtraction: -13*x^5 - 7*x^2 + 21*x - 8

Product: -39*x^7 + 156*x^6 + 78*x^5 - 12*x^4 + 75*x^3 - 126*x^2 + 114*x + 84

Degree of Polynomial: 3

Evaluation of polynomial at x=2: 18

Derivative: 12 - 6*x

Integral: -x^3 + 6*x^2 + 6*x

Definite integration from x=0 to x=1: 11

Real roots of Polynomial 1: [-0.449489742783178, 4.44948974278318]

Test Case (27):

3

-8 8 0

1

7

Expected output:

Polynomial 1: 8*x - 8

Polynomial 2: 7

Sum: 8*x - 1

Subtraction: 8*x - 15

Product: 56*x - 56

Degree of Polynomial: 3

Evaluation of polynomial at x=2: 8

Derivative: 8

Integral: 4*x^2 - 8*x

Definite integration from x=0 to x=1: -4

Real roots of Polynomial 1: [1.00000000000000]

Test Case (28):

8

13 -5 -5 -8 -10 0 8 -8

9

0 -6 -11 -2 13 11 -2 -5 -11

Expected output:

Polynomial 1: -8*x^7 + 8*x^6 - 10*x^4 - 8*x^3 - 5*x^2 - 5*x + 13

Polynomial 2: -11*x^8 - 5*x^7 - 2*x^6 + 11*x^5 + 13*x^4 - 2*x^3 - 11*x^2 - 6*x

Sum: -11*x^8 - 13*x^7 + 6*x^6 + 11*x^5 + 3*x^4 - 10*x^3 - 16*x^2 - 11*x + 13

Subtraction: 11*x^8 - 3*x^7 + 10*x^6 - 11*x^5 - 23*x^4 - 6*x^3 + 6*x^2 + x + 13

Product: 88*x^15 - 48*x^14 - 24*x^13 + 6*x^12 + 122*x^11 + 235*x^10 + 58*x^9 - 366*x^8 - 242*x^7 - 20*x^6 + 236*x^5 + 282*x^4 + 59*x^3 - 113*x^2 - 78*x

Degree of Polynomial: 8

Evaluation of polynomial at x=2: -753

Derivative: -56*x^6 + 48*x^5 - 40*x^3 - 24*x^2 - 10*x - 5

Integral: -x^8 + 8*x^7/7 - 2*x^5 - 2*x^4 - 5*x^3/3 - 5*x^2/2 + 13*x

Definite integration from x=0 to x=1: 209/42

Real roots of Polynomial 1: [0.756084505909115]

Test Case (29):

9

13 6 -4 -10 8 -4 -13 14 2

12

-13 6 8 1 9 -3 -7 -1 -4 1 -10 8

Expected output:

Polynomial 1: 2*x^8 + 14*x^7 - 13*x^6 - 4*x^5 + 8*x^4 - 10*x^3 - 4*x^2 + 6*x + 13

Polynomial 2: 8*x^11 - 10*x^10 + x^9 - 4*x^8 - x^7 - 7*x^6 - 3*x^5 + 9*x^4 + x^3 + 8*x^2 + 6*x - 13

Sum: 8*x^11 - 10*x^10 + x^9 - 2*x^8 + 13*x^7 - 20*x^6 - 7*x^5 + 17*x^4 - 9*x^3 + 4*x^2 + 12*x

Subtraction: -8*x^11 + 10*x^10 - x^9 + 6*x^8 + 15*x^7 - 6*x^6 - x^5 - x^4 - 11*x^3 - 12*x^2 + 26

Product: 16*x^19 + 92*x^18 - 242*x^17 + 104*x^16 + 33*x^15 - 140*x^14 + x^13 + 117*x^12 + 267*x^11 - 229*x^10 + 114*x^9 + 22*x^8 - 417*x^7 + 54*x^6 + 31*x^5 - 73*x^4 + 167*x^3 + 192*x^2 - 169

Degree of Polynomial: 9

Evaluation of polynomial at x=2: 1401

Derivative: 16*x^7 + 98*x^6 - 78*x^5 - 20*x^4 + 32*x^3 - 30*x^2 - 8*x + 6

Integral: 2*x^9/9 + 7*x^8/4 - 13*x^7/7 - 2*x^6/3 + 8*x^5/5 - 5*x^4/2 - 4*x^3/3 + 3*x^2 + 13*x

Definite integration from x=0 to x=1: 16651/1260

Real roots of Polynomial 1: [-7.79155807544242, -1.00000000000000]

Test Case (30):

2

-14 -14

11

10 9 -8 1 -9 13 12 -5 -2 11 -6

Expected output:

Polynomial 1: -14*x - 14

Polynomial 2: -6*x^10 + 11*x^9 - 2*x^8 - 5*x^7 + 12*x^6 + 13*x^5 - 9*x^4 + x^3 - 8*x^2 + 9*x + 10

Sum: -6*x^10 + 11*x^9 - 2*x^8 - 5*x^7 + 12*x^6 + 13*x^5 - 9*x^4 + x^3 - 8*x^2 - 5*x - 4

Subtraction: 6*x^10 - 11*x^9 + 2*x^8 + 5*x^7 - 12*x^6 - 13*x^5 + 9*x^4 - x^3 + 8*x^2 - 23*x - 24

Product: 84*x^11 - 70*x^10 - 126*x^9 + 98*x^8 - 98*x^7 - 350*x^6 - 56*x^5 + 112*x^4 + 98*x^3 - 14*x^2 - 266*x - 140

Degree of Polynomial: 2

Evaluation of polynomial at x=2: -42

Derivative: -14

Integral: -7*x^2 - 14*x

Definite integration from x=0 to x=1: -21

Real roots of Polynomial 1: [-1.00000000000000]

Test Case (31):

6

3 -11 -5 4 -7 14

2

15 -3

Expected output:

Polynomial 1: 14*x^5 - 7*x^4 + 4*x^3 - 5*x^2 - 11*x + 3

Polynomial 2: 15 - 3*x

Sum: 14*x^5 - 7*x^4 + 4*x^3 - 5*x^2 - 14*x + 18

Subtraction: 14*x^5 - 7*x^4 + 4*x^3 - 5*x^2 - 8*x - 12

Product: -42*x^6 + 231*x^5 - 117*x^4 + 75*x^3 - 42*x^2 - 174*x + 45

Degree of Polynomial: 6

Evaluation of polynomial at x=2: 329

Derivative: 70*x^4 - 28*x^3 + 12*x^2 - 10*x - 11

Integral: 7*x^6/3 - 7*x^5/5 + x^4 - 5*x^3/3 - 11*x^2/2 + 3*x

Definite integration from x=0 to x=1: -67/30

Real roots of Polynomial 1: [-0.780154540661465, 0.248941106652295, 1.05162773230181]

Test Case (32):

3

2 -3 -11

4

-13 1 -12 5

Expected output:

Polynomial 1: -11*x^2 - 3*x + 2

Polynomial 2: 5*x^3 - 12*x^2 + x - 13

Sum: 5*x^3 - 23*x^2 - 2*x - 11

Subtraction: -5*x^3 + x^2 - 4*x + 15

Product: -55*x^5 + 117*x^4 + 35*x^3 + 116*x^2 + 41*x - 26

Degree of Polynomial: 3

Evaluation of polynomial at x=2: -48

Derivative: -22*x - 3

Integral: -11*x^3/3 - 3*x^2/2 + 2*x

Definite integration from x=0 to x=1: -19/6

Real roots of Polynomial 1: [-0.584038990990732, 0.311311718263459]

Test Case (33):

10

11 9 -3 -12 -8 -14 3 7 6 7

2

-8 -2

Expected output:

Polynomial 1: 7*x^9 + 6*x^8 + 7*x^7 + 3*x^6 - 14*x^5 - 8*x^4 - 12*x^3 - 3*x^2 + 9*x + 11

Polynomial 2: -2*x - 8

Sum: 7*x^9 + 6*x^8 + 7*x^7 + 3*x^6 - 14*x^5 - 8*x^4 - 12*x^3 - 3*x^2 + 7*x + 3

Subtraction: 7*x^9 + 6*x^8 + 7*x^7 + 3*x^6 - 14*x^5 - 8*x^4 - 12*x^3 - 3*x^2 + 11*x + 19

Product: -14*x^10 - 68*x^9 - 62*x^8 - 62*x^7 + 4*x^6 + 128*x^5 + 88*x^4 + 102*x^3 + 6*x^2 - 94*x - 88

Degree of Polynomial: 10

Evaluation of polynomial at x=2: 5553

Derivative: 63*x^8 + 48*x^7 + 49*x^6 + 18*x^5 - 70*x^4 - 32*x^3 - 36*x^2 - 6*x + 9

Integral: 7*x^10/10 + 2*x^9/3 + 7*x^8/8 + 3*x^7/7 - 7*x^6/3 - 8*x^5/5 - 3*x^4 - x^3 + 9*x^2/2 + 11*x

Definite integration from x=0 to x=1: 8599/840

Real roots of Polynomial 1: [-1.27299545255986]

Test Case (34):

6

4 -4 2 -4 -1 -2

7

-12 -7 0 -12 -2 3 -2

Expected output:

Polynomial 1: -2*x^5 - x^4 - 4*x^3 + 2*x^2 - 4*x + 4

Polynomial 2: -2*x^6 + 3*x^5 - 2*x^4 - 12*x^3 - 7*x - 12

Sum: -2*x^6 + x^5 - 3*x^4 - 16*x^3 + 2*x^2 - 11*x - 8

Subtraction: 2*x^6 - 5*x^5 + x^4 + 8*x^3 + 2*x^2 + 3*x + 16

Product: 4*x^11 - 4*x^10 + 9*x^9 + 10*x^8 + 34*x^7 + 38*x^6 + 27*x^5 + 80*x^4 - 14*x^3 + 4*x^2 + 20*x - 48

Degree of Polynomial: 6

Evaluation of polynomial at x=2: -108

Derivative: -10*x^4 - 4*x^3 - 12*x^2 + 4*x - 4

Integral: -x^6/3 - x^5/5 - x^4 + 2*x^3/3 - 2*x^2 + 4*x

Definite integration from x=0 to x=1: 17/15

Real roots of Polynomial 1: [0.720701095494268]

Test Case (35):

5

-10 -8 7 -12 -11

11

-11 -9 -12 3 -14 -13 1 11 -13 8 0

Expected output:

Polynomial 1: -11*x^4 - 12*x^3 + 7*x^2 - 8*x - 10

Polynomial 2: 8*x^9 - 13*x^8 + 11*x^7 + x^6 - 13*x^5 - 14*x^4 + 3*x^3 - 12*x^2 - 9*x - 11

Sum: 8*x^9 - 13*x^8 + 11*x^7 + x^6 - 13*x^5 - 25*x^4 - 9*x^3 - 5*x^2 - 17*x - 21

Subtraction: -8*x^9 + 13*x^8 - 11*x^7 - x^6 + 13*x^5 + 3*x^4 - 15*x^3 + 19*x^2 + x + 1

Product: -88*x^13 + 47*x^12 + 91*x^11 - 298*x^10 + 232*x^9 + 359*x^8 - 74*x^7 + 92*x^6 + 506*x^5 + 261*x^4 + 135*x^3 + 115*x^2 + 178*x + 110

Degree of Polynomial: 5

Evaluation of polynomial at x=2: -270

Derivative: -44*x^3 - 36*x^2 + 14*x - 8

Integral: -11*x^5/5 - 3*x^4 + 7*x^3/3 - 4*x^2 - 10*x

Definite integration from x=0 to x=1: -253/15

Real roots of Polynomial 1: [-1.55849722938610, -0.675177063532718]

Test Case (36):

4

8 9 13 4

8

1 11 -1 14 1 14 8 -10

Expected output:

Polynomial 1: 4*x^3 + 13*x^2 + 9*x + 8

Polynomial 2: -10*x^7 + 8*x^6 + 14*x^5 + x^4 + 14*x^3 - x^2 + 11*x + 1

Sum: -10*x^7 + 8*x^6 + 14*x^5 + x^4 + 18*x^3 + 12*x^2 + 20*x + 9

Subtraction: 10*x^7 - 8*x^6 - 14*x^5 - x^4 - 10*x^3 + 14*x^2 - 2*x + 7

Product: -40*x^10 - 98*x^9 + 70*x^8 + 178*x^7 + 259*x^6 + 299*x^5 + 165*x^4 + 250*x^3 + 104*x^2 + 97*x + 8

Degree of Polynomial: 4

Evaluation of polynomial at x=2: 110

Derivative: 12*x^2 + 26*x + 9

Integral: x^4 + 13*x^3/3 + 9*x^2/2 + 8*x

Definite integration from x=0 to x=1: 107/6

Real roots of Polynomial 1: [-2.68995612604420]

Test Case (37):

9

-9 10 -11 -12 -7 -3 -5 -8 2

4

12 -8 13 3

Expected output:

Polynomial 1: 2*x^8 - 8*x^7 - 5*x^6 - 3*x^5 - 7*x^4 - 12*x^3 - 11*x^2 + 10*x - 9

Polynomial 2: 3*x^3 + 13*x^2 - 8*x + 12

Sum: 2*x^8 - 8*x^7 - 5*x^6 - 3*x^5 - 7*x^4 - 9*x^3 + 2*x^2 + 2*x + 3

Subtraction: 2*x^8 - 8*x^7 - 5*x^6 - 3*x^5 - 7*x^4 - 15*x^3 - 24*x^2 + 18*x - 21

Product: 6*x^11 + 2*x^10 - 135*x^9 + 14*x^8 - 116*x^7 - 163*x^6 - 169*x^5 - 101*x^4 + 47*x^3 - 329*x^2 + 192*x - 108

Degree of Polynomial: 9

Evaluation of polynomial at x=2: -1169

Derivative: 16*x^7 - 56*x^6 - 30*x^5 - 15*x^4 - 28*x^3 - 36*x^2 - 22*x + 10

Integral: 2*x^9/9 - x^8 - 5*x^7/7 - x^6/2 - 7*x^5/5 - 3*x^4 - 11*x^3/3 + 5*x^2 - 9*x

Definite integration from x=0 to x=1: -8857/630

Real roots of Polynomial 1: [-1.20544080836853, 4.65574257973263]

Test Case (38):

8

12 -7 -7 8 14 -14 9 6

6

12 1 10 -2 -6 12

Expected output:

Polynomial 1: 6*x^7 + 9*x^6 - 14*x^5 + 14*x^4 + 8*x^3 - 7*x^2 - 7*x + 12

Polynomial 2: 12*x^5 - 6*x^4 - 2*x^3 + 10*x^2 + x + 12

Sum: 6*x^7 + 9*x^6 - 2*x^5 + 8*x^4 + 6*x^3 + 3*x^2 - 6*x + 24

Subtraction: 6*x^7 + 9*x^6 - 26*x^5 + 20*x^4 + 10*x^3 - 17*x^2 - 8*x

Product: 72*x^12 + 72*x^11 - 234*x^10 + 294*x^9 + 136*x^8 - 219*x^7 + 176*x^6 + 126*x^5 + 48*x^4 - 5*x^3 + 29*x^2 - 72*x + 144

Degree of Polynomial: 8

Evaluation of polynomial at x=2: 1154

Derivative: 42*x^6 + 54*x^5 - 70*x^4 + 56*x^3 + 24*x^2 - 14*x - 7

Integral: 3*x^8/4 + 9*x^7/7 - 7*x^6/3 + 14*x^5/5 + 2*x^4 - 7*x^3/3 - 7*x^2/2 + 12*x

Definite integration from x=0 to x=1: 4481/420

Real roots of Polynomial 1: [-2.63809982287904]

Test Case (39):

7

12 -4 -6 2 6 10 3

8

7 5 -2 8 -3 -6 14 -6

Expected output:

Polynomial 1: 3*x^6 + 10*x^5 + 6*x^4 + 2*x^3 - 6*x^2 - 4*x + 12

Polynomial 2: -6*x^7 + 14*x^6 - 6*x^5 - 3*x^4 + 8*x^3 - 2*x^2 + 5*x + 7

Sum: -6*x^7 + 17*x^6 + 4*x^5 + 3*x^4 + 10*x^3 - 8*x^2 + x + 19

Subtraction: 6*x^7 - 11*x^6 + 16*x^5 + 9*x^4 - 6*x^3 - 4*x^2 - 9*x + 5

Product: -18*x^13 - 18*x^12 + 86*x^11 + 3*x^10 + 22*x^9 - 16*x^8 - 55*x^7 + 285*x^6 - 12*x^5 - 4*x^4 + 88*x^3 - 86*x^2 + 32*x + 84

Degree of Polynomial: 7

Evaluation of polynomial at x=2: 604

Derivative: 18*x^5 + 50*x^4 + 24*x^3 + 6*x^2 - 12*x - 4

Integral: 3*x^7/7 + 5*x^6/3 + 6*x^5/5 + x^4/2 - 2*x^3 - 2*x^2 + 12*x

Definite integration from x=0 to x=1: 2477/210

Real roots of Polynomial 1: [-2.73984707412938, -1.23229210741421]

Test Case (40):

8

9 -3 9 -9 -10 -5 9 -12

9

-5 5 7 11 -6 -3 -4 3 0

Expected output:

Polynomial 1: -12*x^7 + 9*x^6 - 5*x^5 - 10*x^4 - 9*x^3 + 9*x^2 - 3*x + 9

Polynomial 2: 3*x^7 - 4*x^6 - 3*x^5 - 6*x^4 + 11*x^3 + 7*x^2 + 5*x - 5

Sum: -9*x^7 + 5*x^6 - 8*x^5 - 16*x^4 + 2*x^3 + 16*x^2 + 2*x + 4

Subtraction: -15*x^7 + 13*x^6 - 2*x^5 - 4*x^4 - 20*x^3 + 2*x^2 - 8*x + 14

Product: -36*x^14 + 75*x^13 - 15*x^12 + 35*x^11 - 158*x^10 + 138*x^9 - 10*x^8 + 26*x^7 - 320*x^6 + 2*x^5 - 19*x^4 + 168*x^3 + 3*x^2 + 60*x - 45

Degree of Polynomial: 8

Evaluation of polynomial at x=2: -1313

Derivative: -84*x^6 + 54*x^5 - 25*x^4 - 40*x^3 - 27*x^2 + 18*x - 3

Integral: -3*x^8/2 + 9*x^7/7 - 5*x^6/6 - 2*x^5 - 9*x^4/4 + 3*x^3 - 3*x^2/2 + 9*x

Definite integration from x=0 to x=1: 437/84

Real roots of Polynomial 1: [0.840935386948200]

Test Case (41):

2

-7 -7

10

-6 -5 -7 -7 13 1 13 2 0 -4

Expected output:

Polynomial 1: -7*x - 7

Polynomial 2: -4*x^9 + 2*x^7 + 13*x^6 + x^5 + 13*x^4 - 7*x^3 - 7*x^2 - 5*x - 6

Sum: -4*x^9 + 2*x^7 + 13*x^6 + x^5 + 13*x^4 - 7*x^3 - 7*x^2 - 12*x - 13

Subtraction: 4*x^9 - 2*x^7 - 13*x^6 - x^5 - 13*x^4 + 7*x^3 + 7*x^2 - 2*x - 1

Product: 28*x^10 + 28*x^9 - 14*x^8 - 105*x^7 - 98*x^6 - 98*x^5 - 42*x^4 + 98*x^3 + 84*x^2 + 77*x + 42

Degree of Polynomial: 2

Evaluation of polynomial at x=2: -21

Derivative: -7

Integral: -7*x^2/2 - 7*x

Definite integration from x=0 to x=1: -21/2

Real roots of Polynomial 1: [-1.00000000000000]

Test Case (42):

6

-7 11 -3 13 -13 3

4

-5 2 5 14

Expected output:

Polynomial 1: 3*x^5 - 13*x^4 + 13*x^3 - 3*x^2 + 11*x - 7

Polynomial 2: 14*x^3 + 5*x^2 + 2*x - 5

Sum: 3*x^5 - 13*x^4 + 27*x^3 + 2*x^2 + 13*x - 12

Subtraction: 3*x^5 - 13*x^4 - x^3 - 8*x^2 + 9*x - 2

Product: 42*x^8 - 167*x^7 + 123*x^6 - 18*x^5 + 230*x^4 - 114*x^3 + 2*x^2 - 69*x + 35

Degree of Polynomial: 6

Evaluation of polynomial at x=2: -5

Derivative: 15*x^4 - 52*x^3 + 39*x^2 - 6*x + 11

Integral: x^6/2 - 13*x^5/5 + 13*x^4/4 - x^3 + 11*x^2/2 - 7*x

Definite integration from x=0 to x=1: -27/20

Real roots of Polynomial 1: [0.610175451506496, 1.75264369417805, 2.76093996629644]

Test Case (43):

10

8 -7 -2 -4 4 -6 9 -9 -6 -9

7

-3 3 -14 0 5 -5 5

Expected output:

Polynomial 1: -9*x^9 - 6*x^8 - 9*x^7 + 9*x^6 - 6*x^5 + 4*x^4 - 4*x^3 - 2*x^2 - 7*x + 8

Polynomial 2: 5*x^6 - 5*x^5 + 5*x^4 - 14*x^2 + 3*x - 3

Sum: -9*x^9 - 6*x^8 - 9*x^7 + 14*x^6 - 11*x^5 + 9*x^4 - 4*x^3 - 16*x^2 - 4*x + 5

Subtraction: -9*x^9 - 6*x^8 - 9*x^7 + 4*x^6 - x^5 - x^4 - 4*x^3 + 12*x^2 - 10*x + 11

Product: -45*x^15 + 15*x^14 - 60*x^13 + 60*x^12 + 6*x^11 + 152*x^10 + 65*x^9 - 105*x^8 + 93*x^7 - 36*x^6 + 11*x^5 + 44*x^4 + 104*x^3 - 127*x^2 + 45*x - 24

Degree of Polynomial: 10

Evaluation of polynomial at x=2: -6894

Derivative: -81*x^8 - 48*x^7 - 63*x^6 + 54*x^5 - 30*x^4 + 16*x^3 - 12*x^2 - 4*x - 7

Integral: -9*x^10/10 - 2*x^9/3 - 9*x^8/8 + 9*x^7/7 - x^6 + 4*x^5/5 - x^4 - 2*x^3/3 - 7*x^2/2 + 8*x

Definite integration from x=0 to x=1: 1031/840

Real roots of Polynomial 1: [0.712255050975951]

Test Case (44):

5

-11 -1 -9 -5 10

5

-2 -8 -11 4 12

Expected output:

Polynomial 1: 10*x^4 - 5*x^3 - 9*x^2 - x - 11

Polynomial 2: 12*x^4 + 4*x^3 - 11*x^2 - 8*x - 2

Sum: 22*x^4 - x^3 - 20*x^2 - 9*x - 13

Subtraction: -2*x^4 - 9*x^3 + 2*x^2 + 7*x - 9

Product: 120*x^8 - 20*x^7 - 238*x^6 - 73*x^5 - 17*x^4 + 49*x^3 + 147*x^2 + 90*x + 22

Degree of Polynomial: 5

Evaluation of polynomial at x=2: 71

Derivative: 40*x^3 - 15*x^2 - 18*x - 1

Integral: 2*x^5 - 5*x^4/4 - 3*x^3 - x^2/2 - 11*x

Definite integration from x=0 to x=1: -55/4

Real roots of Polynomial 1: [-1.09023399943464, 1.48607620951496]

Test Case (45):

2

1 -12

12

-12 -14 -14 -10 -12 8 7 12 -7 -5 11 12

Expected output:

Polynomial 1: 1 - 12*x

Polynomial 2: 12*x^11 + 11*x^10 - 5*x^9 - 7*x^8 + 12*x^7 + 7*x^6 + 8*x^5 - 12*x^4 - 10*x^3 - 14*x^2 - 14*x - 12

Sum: 12*x^11 + 11*x^10 - 5*x^9 - 7*x^8 + 12*x^7 + 7*x^6 + 8*x^5 - 12*x^4 - 10*x^3 - 14*x^2 - 26*x - 11

Subtraction: -12*x^11 - 11*x^10 + 5*x^9 + 7*x^8 - 12*x^7 - 7*x^6 - 8*x^5 + 12*x^4 + 10*x^3 + 14*x^2 + 2*x + 13

Product: -144*x^12 - 120*x^11 + 71*x^10 + 79*x^9 - 151*x^8 - 72*x^7 - 89*x^6 + 152*x^5 + 108*x^4 + 158*x^3 + 154*x^2 + 130*x - 12

Degree of Polynomial: 2

Evaluation of polynomial at x=2: -23

Derivative: -12

Integral: -6*x^2 + x

Definite integration from x=0 to x=1: -5

Real roots of Polynomial 1: [0.0833333333333333]

Test Case (46):

3

-4 2 1

4

-1 -2 -11 9

Expected output:

Polynomial 1: x^2 + 2*x - 4

Polynomial 2: 9*x^3 - 11*x^2 - 2*x - 1

Sum: 9*x^3 - 10*x^2 - 5

Subtraction: -9*x^3 + 12*x^2 + 4*x - 3

Product: 9*x^5 + 7*x^4 - 60*x^3 + 39*x^2 + 6*x + 4

Degree of Polynomial: 3

Evaluation of polynomial at x=2: 4

Derivative: 2*x + 2

Integral: x^3/3 + x^2 - 4*x

Definite integration from x=0 to x=1: -8/3

Real roots of Polynomial 1: [-3.23606797749979, 1.23606797749979]

Test Case (47):

4

-12 13 11 -13

3

0 -8 -12

Expected output:

Polynomial 1: -13*x^3 + 11*x^2 + 13*x - 12

Polynomial 2: -12*x^2 - 8*x

Sum: -13*x^3 - x^2 + 5*x - 12

Subtraction: -13*x^3 + 23*x^2 + 21*x - 12

Product: 156*x^5 - 28*x^4 - 244*x^3 + 40*x^2 + 96*x

Degree of Polynomial: 4

Evaluation of polynomial at x=2: -46

Derivative: -39*x^2 + 22*x + 13

Integral: -13*x^4/4 + 11*x^3/3 + 13*x^2/2 - 12*x

Definite integration from x=0 to x=1: -61/12

Real roots of Polynomial 1: [-1.02039763534099]

Test Case (48):

4

9 14 -1 -4

10

-13 -11 -11 3 -7 6 10 5 -9 -8

Expected output:

Polynomial 1: -4*x^3 - x^2 + 14*x + 9

Polynomial 2: -8*x^9 - 9*x^8 + 5*x^7 + 10*x^6 + 6*x^5 - 7*x^4 + 3*x^3 - 11*x^2 - 11*x - 13

Sum: -8*x^9 - 9*x^8 + 5*x^7 + 10*x^6 + 6*x^5 - 7*x^4 - x^3 - 12*x^2 + 3*x - 4

Subtraction: 8*x^9 + 9*x^8 - 5*x^7 - 10*x^6 - 6*x^5 + 7*x^4 - 7*x^3 + 10*x^2 + 25*x + 22

Product: 32*x^12 + 44*x^11 - 123*x^10 - 243*x^9 - 45*x^8 + 207*x^7 + 169*x^6 - 3*x^5 + 34*x^4 - 64*x^3 - 240*x^2 - 281*x - 117

Degree of Polynomial: 4

Evaluation of polynomial at x=2: 1

Derivative: -12*x^2 - 2*x + 14

Integral: -x^4 - x^3/3 + 7*x^2 + 9*x

Definite integration from x=0 to x=1: 44/3

Real roots of Polynomial 1: [-1.56720136657657, -0.708672178386541, 2.02587354496311]

Test Case (49):

11

-1 -3 2 -7 4 -6 -3 6 15 -13 -9

3

10 9 -7

Expected output:

Polynomial 1: -9*x^10 - 13*x^9 + 15*x^8 + 6*x^7 - 3*x^6 - 6*x^5 + 4*x^4 - 7*x^3 + 2*x^2 - 3*x - 1

Polynomial 2: -7*x^2 + 9*x + 10

Sum: -9*x^10 - 13*x^9 + 15*x^8 + 6*x^7 - 3*x^6 - 6*x^5 + 4*x^4 - 7*x^3 - 5*x^2 + 6*x + 9

Subtraction: -9*x^10 - 13*x^9 + 15*x^8 + 6*x^7 - 3*x^6 - 6*x^5 + 4*x^4 - 7*x^3 + 9*x^2 - 12*x - 11

Product: 63*x^12 + 10*x^11 - 312*x^10 - 37*x^9 + 225*x^8 + 75*x^7 - 112*x^6 + 25*x^5 - 37*x^4 - 31*x^3 - 39*x - 10

Degree of Polynomial: 11

Evaluation of polynomial at x=2: -11639

Derivative: -90*x^9 - 117*x^8 + 120*x^7 + 42*x^6 - 18*x^5 - 30*x^4 + 16*x^3 - 21*x^2 + 4*x - 3

Integral: -9*x^11/11 - 13*x^10/10 + 5*x^9/3 + 3*x^8/4 - 3*x^7/7 - x^6 + 4*x^5/5 - 7*x^4/4 + 2*x^3/3 - 3*x^2/2 - x

Definite integration from x=0 to x=1: -904/231

Real roots of Polynomial 1: [-2.10546624177659, -0.249109963746682]

Test Case (50):

8

2 -5 -6 10 11 -11 -7 -9

8

-3 1 -12 11 -4 -3 -12 -10

Expected output:

Polynomial 1: -9*x^7 - 7*x^6 - 11*x^5 + 11*x^4 + 10*x^3 - 6*x^2 - 5*x + 2

Polynomial 2: -10*x^7 - 12*x^6 - 3*x^5 - 4*x^4 + 11*x^3 - 12*x^2 + x - 3

Sum: -19*x^7 - 19*x^6 - 14*x^5 + 7*x^4 + 21*x^3 - 18*x^2 - 4*x - 1

Subtraction: x^7 + 5*x^6 - 8*x^5 + 15*x^4 - x^3 + 6*x^2 - 6*x + 5

Product: 90*x^14 + 178*x^13 + 221*x^12 + 79*x^11 - 270*x^10 - 18*x^9 + 2*x^8 + 291*x^7 + 3*x^6 - 128*x^5 - 14*x^4 + 46*x^3 - 11*x^2 + 17*x - 6

Degree of Polynomial: 8

Evaluation of polynomial at x=2: -1728

Derivative: -63*x^6 - 42*x^5 - 55*x^4 + 44*x^3 + 30*x^2 - 12*x - 5

Integral: -9*x^8/8 - x^7 - 11*x^6/6 + 11*x^5/5 + 5*x^4/2 - 2*x^3 - 5*x^2/2 + 2*x

Definite integration from x=0 to x=1: -211/120

Real roots of Polynomial 1: [0.356797886917835]

Appendix (D)

Resources

[NTI offical web site](#)

[free code camp](#)

https://en.wikipedia.org/wiki/Information_Technology_Institute

https://ititic.zohodesk.com/portal/en/kb/articles/9monthtraining

https://www.bibalex.org/he_funding/Donors/Details.aspx?ID=59

https://eduhub21.com/en/iti-scholarship-for-9-months-2024/

https://iti.gov.eg/home