CONTENTS

PRIMITIVE POLYNOMIAL – INTRO

A given polynomial whose degree (highest power) is n, is a primitive polynomial if we divide X^T + 1 with the input polynomial and remainder is 0. Where T = 2^n – 1.

Also the division used here uses EXOR operation and hence no negative sign will occur in division.s

Example 1

Let us consider a polynomial X^3 + X^2 + X + 1. We need to check if polynomial is primitive or not.

T = 2^n – 1

T = 2^3 – 1 = 7

So now if X^3 + X^2 + X + 1 divides X^7 + 1, the polynomial is primitive.

In this case polynomial is not primitive as remainder after division is not zero.

Example 2

Let us consider a polynomial X^3 + X^2 + 1. We need to check if polynomial is primitive or not.

T = 2^n – 1

T = 2^3 – 1 = 7

So now if X^3 + X^2 + 1 divides X^7 + 1, the polynomial is primitive.

In this case polynomial is primitive as remainder after division is not zero.

PROJECT DESCRIPTION

We have written a C++ program to find a given polynomial is Primitive or not. The C++ program can perform the operation on any polynomial of degree less than or equal to 10.

Idea Used

In the program we are first letting user to enter the degree of Polynomial. Then the user has to enter 1 or 0 as coefficients of each term. For example if X^2 is present in our polynomial its coefficient will be entered as 1 and if it is not one of the term of polynomial its coefficient will be 0. Hence having complete polynomial input.

These coefficients are placed in an array of denominator [], coefficient of X at index 1, coefficient of X^2 at index 2 and so on.

Now we find X^T + 1 and similarly place 1's and 0's in another array named numerator [].

Now we find a quotient term by subtracting degree of numerator and denominator. The denominators 1's are copied into another array named temp1 [], but the position of 1's are shifted by adding quotient terms. For example if a 1 was present at 2^{nd} index in denominator array, it will be at 2 + quotient term index in temp array.

The temp array and denominator array are EXOR and the result is stored in numerator array. So we now have a new numerator. The above process is repeated till we reach termination condition.

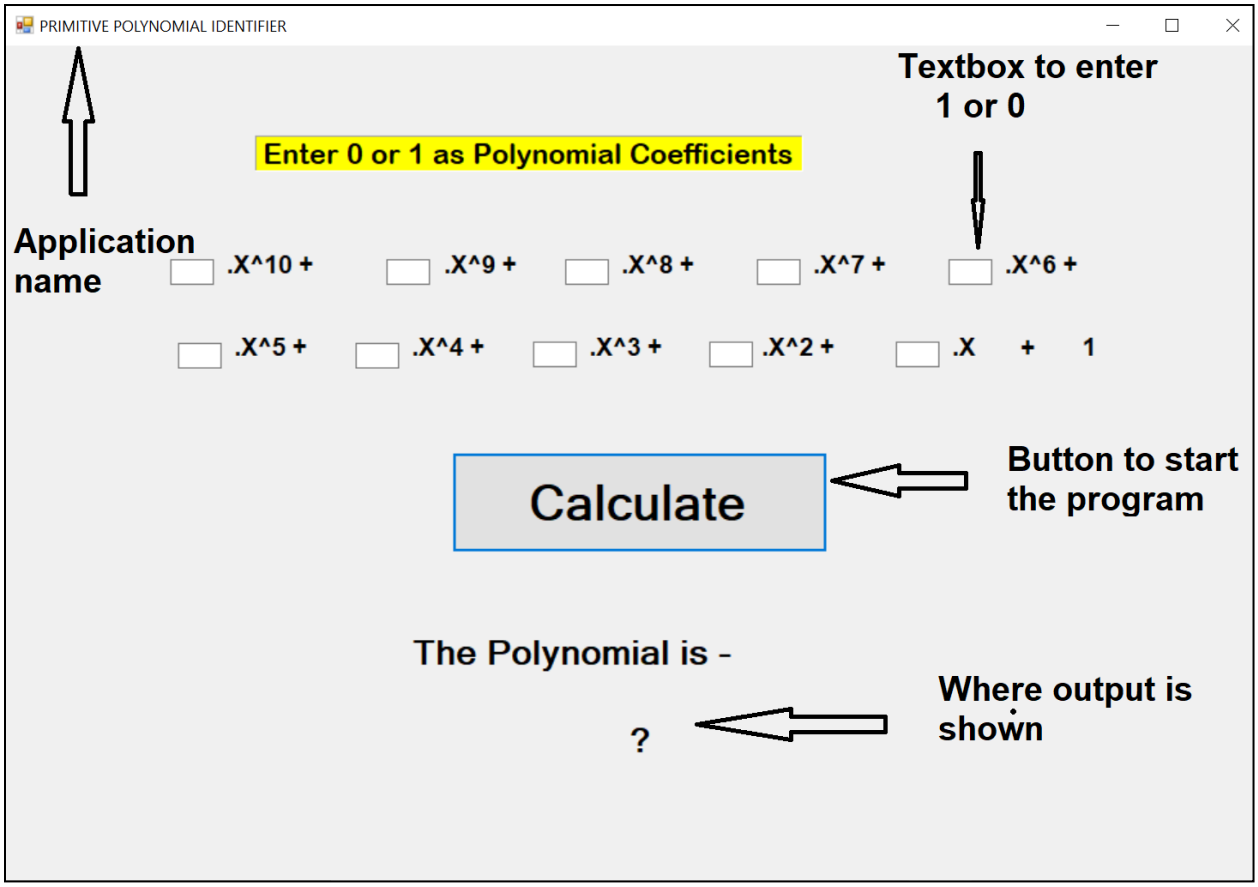
The termination occurs either when numerator index is all 0 (Our polynomial is primitive) or when degree of numerator is less than degree of numerator (Our polynomial is not primitive).

Application development

We used Visual Studio 2019 software to generate executable program of the code. Slight changes in program were made which includes taking inputs from textbox and converting those strings into integer variable. Also output was forwarded to a label.

The design and layout of application was made using features of visual studio. We made changes in predefined template to get required font, layout and size of the application.

APPLICATION LAYOUT



CODE – FOR COMMAND LINE EXECUTION

```
#include<iostream>
#include<stdio.h>
using namespace std;

int main()
{
    int numerator[10000] = {0}, denominator[10000] = {0}, temp1[10000] = {0};
    int degree_of_denominator;
    cout<< "Enter the degree of polynomial  - ";
    cin>>degree_of_denominator;
    cout<< "\n";
    for(int i = 0; i<= degree_of_denominator; i++)
    {
        cout<<"Enter coefficient of X^"<<i<<" - ";
        cin>>denominator[i];
    }
    int X_power_degree=1;
    for(int i=0;i<degree_of_denominator;i++)
    {
        X_power_degree=X_power_degree*2;
    }
    numerator[X_power_degree-1]=1;
    numerator[0]=1;
    int degree_of_numerator;
    while(1)
    {
        degree_of_numerator=-1;
        for(int j=X_power_degree;j>=0;j--)
```

```
        {
            if(numerator[j]==1)
            {
                degree_of_numerator=j;
                break;
            }
        }
        if(degree_of_numerator==-1)
        {
            cout<<"Polynomial is Primitive. \n";
            return 0;
        }
        int quotient_term = degree_of_numerator-degree_of_denominator;
        for(int i =0; i<10000;i++)
        {
            temp1[i] = 0;
        }
        for(int i = degree_of_denominator; i >=0; i--)
        {
            if(denominator[i]==1)
            {
                temp1[i+quotient_term]=1;
            }
        }
        for(int i = degree_of_numerator; i>=0; i--)
        {
            if(((temp1[i]==0) && (numerator[i]==0)) || ((temp1[i]==1) &&
(numerator[i]==1)))
            {
                numerator[i] = 0;
            }
```

```
            else
            {
                numerator[i] = 1;
            }
        }
        for(int j=X_power_degree;j>=0;j--)
        {
            if(numerator[j]==1)
            {
                degree_of_numerator=j;
                break;
            }
        }
        if(degree_of_numerator<degree_of_denominator)
        {
            break;
        }
    }
    cout<<"Polynomial is not Primitive. \n";
    return 0;
}
```

VISUAL STUDIO CODE – FOR APPLICATION DEVELOPMENT

```
#pragma once


namespace Project6 {


	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;


	/// <summary>
	/// Summary for MyForm
	/// </summary>
	public ref class MyForm : public System::Windows::Forms::Form
	{
	public:
		MyForm(void)
		{
			InitializeComponent();
			//
			//TODO: Add the constructor code here
			//
		}


	protected:
		/// <summary>
```

```
        /// Clean up any resources being used.
        /// </summary>
        ~MyForm()
        {
                if (components)
                {
                        delete components;
                }
        }
private: System::Windows::Forms::Button^  button1;
protected:
private: System::Windows::Forms::Label^  label1;
private: System::Windows::Forms::Label^  label2;
private: System::Windows::Forms::Label^  label3;

private: System::Windows::Forms::Label^  label5;

private: System::Windows::Forms::Label^  label7;

private: System::Windows::Forms::Label^  label9;

private: System::Windows::Forms::Label^  label11;
private: System::Windows::Forms::Label^  label12;
private: System::Windows::Forms::Label^  label13;
private: System::Windows::Forms::Label^  label14;

private: System::Windows::Forms::Label^  label16;

private: System::Windows::Forms::Label^  label18;
```

```
	private: System::Windows::Forms::Label^  label20;

	private: System::Windows::Forms::Label^  label22;

	private: System::Windows::Forms::Label^  label24;
	private: System::Windows::Forms::TextBox^  textBox1;
	private: System::Windows::Forms::TextBox^  textBox2;
	private: System::Windows::Forms::TextBox^  textBox3;
	private: System::Windows::Forms::TextBox^  textBox4;
	private: System::Windows::Forms::TextBox^  textBox5;
	private: System::Windows::Forms::TextBox^  textBox6;
	private: System::Windows::Forms::TextBox^  textBox7;
	private: System::Windows::Forms::TextBox^  textBox8;
	private: System::Windows::Forms::TextBox^  textBox9;
	private: System::Windows::Forms::TextBox^  textBox10;



	private:
		/// <summary>
		/// Required designer variable.
		/// </summary>
		System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

		void InitializeComponent(void)
		{
			this->button1 = (gcnew System::Windows::Forms::Button());
			this->label1 = (gcnew System::Windows::Forms::Label());
			this->label2 = (gcnew System::Windows::Forms::Label());
```

```
                this->label3 = (gcnew System::Windows::Forms::Label());
                this->label5 = (gcnew System::Windows::Forms::Label());
                this->label7 = (gcnew System::Windows::Forms::Label());
                this->label9 = (gcnew System::Windows::Forms::Label());
                this->label11 = (gcnew System::Windows::Forms::Label());
                this->label12 = (gcnew System::Windows::Forms::Label());
                this->label13 = (gcnew System::Windows::Forms::Label());
                this->label14 = (gcnew System::Windows::Forms::Label());
                this->label16 = (gcnew System::Windows::Forms::Label());
                this->label18 = (gcnew System::Windows::Forms::Label());
                this->label20 = (gcnew System::Windows::Forms::Label());
                this->label22 = (gcnew System::Windows::Forms::Label());
                this->label24 = (gcnew System::Windows::Forms::Label());
                this->textBox1 = (gcnew System::Windows::Forms::TextBox());
                this->textBox2 = (gcnew System::Windows::Forms::TextBox());
                this->textBox3 = (gcnew System::Windows::Forms::TextBox());
                this->textBox4 = (gcnew System::Windows::Forms::TextBox());
                this->textBox5 = (gcnew System::Windows::Forms::TextBox());
                this->textBox6 = (gcnew System::Windows::Forms::TextBox());
                this->textBox7 = (gcnew System::Windows::Forms::TextBox());
                this->textBox8 = (gcnew System::Windows::Forms::TextBox());
                this->textBox9 = (gcnew System::Windows::Forms::TextBox());
                this->textBox10 = (gcnew
System::Windows::Forms::TextBox());
                this->SuspendLayout();
                //
                // button1
                //
                this->button1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 28,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(0)));
```

```
                this->button1->Location = System::Drawing::Point(348, 316);
                this->button1->Name = L"button1";
                this->button1->Size = System::Drawing::Size(292, 78);
                this->button1->TabIndex = 0;
                this->button1->Text = L"Calculate";
                this->button1->UseVisualStyleBackColor = true;
                this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);
                //
                // label1
                //
                this->label1->AutoSize = true;
                this->label1->Font = (gcnew System::Drawing::Font(L"Arial",
14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(0)));
                this->label1->Location = System::Drawing::Point(834, 223);
                this->label1->Name = L"label1";
                this->label1->Size = System::Drawing::Size(31, 33);
                this->label1->TabIndex = 1;
                this->label1->Text = L"1";
                //
                // label2
                //
                this->label2->AutoSize = true;
                this->label2->Font = (gcnew System::Drawing::Font(L"Arial",
14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(0)));
                this->label2->Location = System::Drawing::Point(781, 224);
                this->label2->Name = L"label2";
                this->label2->Size = System::Drawing::Size(47, 33);
                this->label2->TabIndex = 2;
                this->label2->Text = L" + ";
```

```
                    //
                    // label3
                    //
                    this->label3->AutoSize = true;
                    this->label3->Font = (gcnew System::Drawing::Font(L"Arial",
14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label3->Location = System::Drawing::Point(733, 223);
                    this->label3->Name = L"label3";
                    this->label3->Size = System::Drawing::Size(42, 33);
                    this->label3->TabIndex = 3;
                    this->label3->Text = L".X";
                    //
                    // label5
                    //
                    this->label5->AutoSize = true;
                    this->label5->Font = (gcnew System::Drawing::Font(L"Arial",
14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label5->Location = System::Drawing::Point(585, 223);
                    this->label5->Name = L"label5";
                    this->label5->Size = System::Drawing::Size(106, 33);
                    this->label5->TabIndex = 5;
                    this->label5->Text = L".X^2 + ";
                    //
                    // label7
                    //
                    this->label7->AutoSize = true;
                    this->label7->Font = (gcnew System::Drawing::Font(L"Arial",
14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label7->Location = System::Drawing::Point(451, 223);
```

```
                this->label7->Name = L"label7";
                this->label7->Size = System::Drawing::Size(106, 33);
                this->label7->TabIndex = 7;
                this->label7->Text = L".X^3 + ";
                //
                // label9
                //
                this->label9->AutoSize = true;
                this->label9->Font = (gcnew System::Drawing::Font(L"Arial",
14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(0)));
                this->label9->Location = System::Drawing::Point(313, 223);
                this->label9->Name = L"label9";
                this->label9->Size = System::Drawing::Size(106, 33);
                this->label9->TabIndex = 9;
                this->label9->Text = L".X^4 + ";
                //
                // label11
                //
                this->label11->AutoSize = true;
                this->label11->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 20,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(0)));
                this->label11->Location = System::Drawing::Point(311, 455);
                this->label11->Name = L"label11";
                this->label11->Size = System::Drawing::Size(380, 46);
                this->label11->TabIndex = 11;
                this->label11->Text = L"The Polinomial is - ";
                this->label11->TextAlign =
System::Drawing::ContentAlignment::MiddleCenter;
                //
```

```
                    // label12
                    //
                    this->label12->AutoSize = true;
                    this->label12->BackColor = System::Drawing::Color::Yellow;
                    this->label12->BorderStyle =
System::Windows::Forms::BorderStyle::Fixed3D;
                    this->label12->FlatStyle =
System::Windows::Forms::FlatStyle::Popup;
                    this->label12->Font = (gcnew System::Drawing::Font(L"Arial
Rounded MT Bold", 16, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label12->ForeColor =
System::Drawing::SystemColors::InfoText;
                    this->label12->Location = System::Drawing::Point(195, 70);
                    this->label12->Name = L"label12";
                    this->label12->Size = System::Drawing::Size(614, 39);
                    this->label12->TabIndex = 12;
                    this->label12->Text = L"Enter 0 or 1 as Polynomial
Coefficients";
                    this->label12->TextAlign =
System::Drawing::ContentAlignment::MiddleCenter;
                    //
                    // label13
                    //
                    this->label13->AutoSize = true;
                    this->label13->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 20,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label13->Location = System::Drawing::Point(480, 523);
                    this->label13->Name = L"label13";
                    this->label13->Size = System::Drawing::Size(43, 46);
                    this->label13->TabIndex = 13;
                    this->label13->Text = L"\?";
```

```
                    //
                    // label14
                    //
                    this->label14->AutoSize = true;
                    this->label14->Font = (gcnew
System::Drawing::Font(L"Arial", 14, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label14->Location = System::Drawing::Point(175, 223);
                    this->label14->Name = L"label14";
                    this->label14->Size = System::Drawing::Size(106, 33);
                    this->label14->TabIndex = 14;
                    this->label14->Text = L".X^5 + ";
                    //
                    // label16
                    //
                    this->label16->AutoSize = true;
                    this->label16->Font = (gcnew
System::Drawing::Font(L"Arial", 14, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label16->Location = System::Drawing::Point(774, 159);
                    this->label16->Name = L"label16";
                    this->label16->Size = System::Drawing::Size(106, 33);
                    this->label16->TabIndex = 16;
                    this->label16->Text = L".X^6 + ";
                    //
                    // label18
                    //
                    this->label18->AutoSize = true;
                    this->label18->Font = (gcnew
System::Drawing::Font(L"Arial", 14, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
```

```
                        static_cast<System::Byte>(0)));
                    this->label18->Location = System::Drawing::Point(625, 159);
                    this->label18->Name = L"label18";
                    this->label18->Size = System::Drawing::Size(106, 33);
                    this->label18->TabIndex = 18;
                    this->label18->Text = L".X^7 + ";
                    //
                    // label20
                    //
                    this->label20->AutoSize = true;
                    this->label20->Font = (gcnew
System::Drawing::Font(L"Arial", 14, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                        static_cast<System::Byte>(0)));
                    this->label20->Location = System::Drawing::Point(476, 159);
                    this->label20->Name = L"label20";
                    this->label20->Size = System::Drawing::Size(106, 33);
                    this->label20->TabIndex = 20;
                    this->label20->Text = L".X^8 + ";
                    //
                    // label22
                    //
                    this->label22->AutoSize = true;
                    this->label22->Font = (gcnew
System::Drawing::Font(L"Arial", 14, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                        static_cast<System::Byte>(0)));
                    this->label22->Location = System::Drawing::Point(338, 159);
                    this->label22->Name = L"label22";
                    this->label22->Size = System::Drawing::Size(106, 33);
                    this->label22->TabIndex = 22;
                    this->label22->Text = L".X^9 + ";
```

```
                    //
                    // label24
                    //
                    this->label24->AutoSize = true;
                    this->label24->Font = (gcnew
System::Drawing::Font(L"Arial", 14, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->label24->Location = System::Drawing::Point(169, 159);
                    this->label24->Name = L"label24";
                    this->label24->Size = System::Drawing::Size(122, 33);
                    this->label24->TabIndex = 24;
                    this->label24->Text = L".X^10 + ";
                    //
                    // textBox1
                    //
              this->textBox1->Location = System::Drawing::Point(693, 230);
                    this->textBox1->Name = L"textBox1";
                    this->textBox1->Size = System::Drawing::Size(34, 26);
                    this->textBox1->TabIndex = 26;
                    //
                    // textBox2
                    //
              this->textBox2->Location = System::Drawing::Point(548, 230);
                    this->textBox2->Name = L"textBox2";
                    this->textBox2->Size = System::Drawing::Size(34, 26);
                    this->textBox2->TabIndex = 27;
                    //
                    // textBox3
                    //
              this->textBox3->Location = System::Drawing::Point(411, 230);
```

```
            this->textBox3->Name = L"textBox3";
            this->textBox3->Size = System::Drawing::Size(34, 26);
            this->textBox3->TabIndex = 28;
            //
            // textBox4
            //
        this->textBox4->Location = System::Drawing::Point(273, 231);
            this->textBox4->Name = L"textBox4";
            this->textBox4->Size = System::Drawing::Size(34, 26);
            this->textBox4->TabIndex = 29;
            //
            // textBox5
            //
        this->textBox5->Location = System::Drawing::Point(135, 231);
            this->textBox5->Name = L"textBox5";
            this->textBox5->Size = System::Drawing::Size(34, 26);
            this->textBox5->TabIndex = 30;
            //
            // textBox6
            //
        this->textBox6->Location = System::Drawing::Point(734, 166);
            this->textBox6->Name = L"textBox6";
            this->textBox6->Size = System::Drawing::Size(34, 26);
            this->textBox6->TabIndex = 31;
            //
            // textBox7
            //
        this->textBox7->Location = System::Drawing::Point(585, 166);
            this->textBox7->Name = L"textBox7";
            this->textBox7->Size = System::Drawing::Size(34, 26);
```

```
            this->textBox7->TabIndex = 32;
            //
            // textBox8
            //
        this->textBox8->Location = System::Drawing::Point(436, 166);
            this->textBox8->Name = L"textBox8";
            this->textBox8->Size = System::Drawing::Size(34, 26);
            this->textBox8->TabIndex = 33;
            //
            // textBox9
            //
        this->textBox9->Location = System::Drawing::Point(297, 166);
            this->textBox9->Name = L"textBox9";
            this->textBox9->Size = System::Drawing::Size(34, 26);
            this->textBox9->TabIndex = 34;
            //
            // textBox10
            //
        this->textBox10->Location = System::Drawing::Point(129, 166);
            this->textBox10->Name = L"textBox10";
            this->textBox10->Size = System::Drawing::Size(34, 26);
            this->textBox10->TabIndex = 35;
            //
            // MyForm
            //
            this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::None;
            this->ClientSize = System::Drawing::Size(978, 711);
            this->Controls->Add(this->textBox10);
            this->Controls->Add(this->textBox9);
            this->Controls->Add(this->textBox8);
```

```
            this->Controls->Add(this->textBox7);
            this->Controls->Add(this->textBox6);
            this->Controls->Add(this->textBox5);
            this->Controls->Add(this->textBox4);
            this->Controls->Add(this->textBox3);
            this->Controls->Add(this->textBox2);
            this->Controls->Add(this->textBox1);
            this->Controls->Add(this->label24);
            this->Controls->Add(this->label22);
            this->Controls->Add(this->label20);
            this->Controls->Add(this->label18);
            this->Controls->Add(this->label16);
            this->Controls->Add(this->label14);
            this->Controls->Add(this->label13);
            this->Controls->Add(this->label12);
            this->Controls->Add(this->label11);
            this->Controls->Add(this->label9);
            this->Controls->Add(this->label7);
            this->Controls->Add(this->label5);
            this->Controls->Add(this->label3);
            this->Controls->Add(this->label2);
            this->Controls->Add(this->label1);
            this->Controls->Add(this->button1);
            this->Name = L"MyForm";
            this->Text = L"PRIMITIVE POLYNOMIAL IDENTIFIER";
            this->ResumeLayout(false);
            this->PerformLayout();

        }
#pragma endregion
```

```
private: System::Void button1_Click(System::Object^  sender,
System::EventArgs^  e)

             {

                    int denominator[11];

                    denominator[0]=1;

                    String ^ x1 = textBox1 -> Text;

                    denominator[1]=System::Convert::ToInt16(x1);

                    String ^ x2 = textBox2 -> Text;

                    denominator[2]=System::Convert::ToInt16(x2);

                    String ^ x3 = textBox3 -> Text;

                    denominator[3]=System::Convert::ToInt16(x3);

                    String ^ x4 = textBox4 -> Text;

                    denominator[4]=System::Convert::ToInt16(x4);

                    String ^ x5 = textBox5 -> Text;

                    denominator[5]=System::Convert::ToInt16(x5);

                    String ^ x6 = textBox6 -> Text;

                    denominator[6]=System::Convert::ToInt16(x6);

                    String ^ x7 = textBox7 -> Text;

                    denominator[7]=System::Convert::ToInt16(x7);

                    String ^ x8 = textBox8 -> Text;

                    denominator[8]=System::Convert::ToInt16(x8);

                    String ^ x9 = textBox9 -> Text;

                    denominator[9]=System::Convert::ToInt16(x9);

                    String ^ x10 = textBox10 -> Text;

                    denominator[10]=System::Convert::ToInt16(x10);




                   int numerator[10000] = {0}, temp1[10000] = {0};

                   int degree_of_denominator;
```

```
for(int j=11;j>=0;j--)
{
	if(denominator[j]==1)
	{
		degree_of_denominator=j;
		break;
	}
}
int X_power_degree=1;
for(int i=0;i<degree_of_denominator;i++)
{
	X_power_degree=X_power_degree*2;
}
numerator[X_power_degree-1]=1;
numerator[0]=1;
int degree_of_numerator;
while(1)
{
	degree_of_numerator=-1;
	for(int j=X_power_degree;j>=0;j--)
	{
		if(numerator[j]==1)
		{
			degree_of_numerator=j;
			break;
		}
	}
	if(degree_of_numerator==-1)
	{
		label13 -> Text = "Primitive";
```

```
                        return;
                    }
            int quotient_term = degree_of_numerator-degree_of_denominator;
            for(int i =0; i<10000;i++)
            {
                  temp1[i] = 0;
            }
            for(int i = degree_of_denominator; i >=0; i--)
            {
                  if(denominator[i]==1)
                  {
                        temp1[i+quotient_term]=1;
                  }
            }
            for(int i = degree_of_numerator; i>=0; i--)
            {
                  if(((temp1[i]==0) && (numerator[i]==0)) || ((temp1[i]==1)
&& (numerator[i]==1)))
                  {
                        numerator[i] = 0;
                  }
                  else
                  {
                        numerator[i] = 1;
                  }
            }
            for(int j=X_power_degree;j>=0;j--)
            {
                  if(numerator[j]==1)
                  {
                        degree_of_numerator=j;
```

```
                        break;
                    }
                }

        if(degree_of_numerator<degree_of_denominator)
        {
                break;
        }
}
        label13 -> Text = "Not Primitive.";
        return;
}
};
}
```

SOFTWARES USED

Visual Studio Code

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. We used this software to write C++ code.

Visual Studio 2019

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. We used this software for application development.

REFERENCES

- Class notes and PPT's provided.
- https://docs.microsoft.com//visualstudio to resolve a linker error during application development.