|  |  |
| --- | --- |
| Assignment No.: | 6 |
| Title: | Write a C++ program to realize polynomial equation and perform operations |
| Subject: | Data Structures Laboratory |
| Class: | S.Y. (C.s.E.) |
| Roll No.: |  |
| Assessment (Marks): |  |
| Signature and Date of Assessment: |  |

Experiment No.

Experiment Title: Write a C++ program to realize polynomial equation and perform operations

Objectives:

1. To understand how to realize polynomial using structure.
2. To understand polynomial operations using structures.

Problem Statement: Write a C++ program to realize polynomial equation and perform operations. Write the following functions:

l. To input and output polynomials

* 1. To add two polynomials
  2. To multiply two polynomials

Outcomes:

Understanding the polynomial representation using structure.

g++ compiler on Ubuntu 14.04 (64bit)

# Structure

Structure is a collection of variables of different data types under a single name. It is similar to a class in that, bothholds a collection of data of different data types.

For example: You want to store some information about a person: his,'her name, citizenship number and salary. You can easily create different variables name, age, salary to store this information separately. However, in the future, you would want to store information about multiple persons. Now, you'd need to createdifferent variables for each information per person: namel , citNol, salaryl , name2, citN02, salary2 You can easily visualize how big and messy the code would look. Also, since no relation between the variables(information) would exist, it's going to be a daunting task.

A better approach will be to have a collection of all related information under a single name Person, and use it forevery person. Now, the code looks much cleaner, readable and efficient as well. This collection of all related information under a single name Person is a structure.

How to declare a structure in C++ programming?

The struct keyword defines a structure type followed by an identifier (name of the structure).

Then inside the curly braces, you can declare one or more members (declare variables inside curly braces) of thatstructure. For example:

struct Person

Char name [501; int age; float salary;

Here a structure person is defined which has three members: name, age and salary.

When a structure is created, no memory is allocated.

The structure definition is only the blueprint for the creating of variables. You can imagine it as a datatype. Whenyou define an integer as below: int foo;

The int specifies that, variable foo can hold integer element only. Similarly, structure definition only specifies that,what property a structure variable holds when it is defined.

How to define a structure variable?

Once you declare a structure person as above. You can define a structure variable as:Person bill;

Here, a structure variable bill is defined which is of type structure Person.

When structure variable is defined, only then the required memory is allocated by the compiler.

Considering you have either 32-bit or 64-bit system, the memory of float is 4 bytes, memory of int is 4 bytes andmemory ofchar is I byte.

Hence, 58 bytes of memory is allocated for structure variable bill.How to access members of a structure?

The members of structure variable is accessed using a dot (.) operator.

Suppose, you want to access age of structure variable bill and assign it 50 to it. You can perform this task by usingfollowing code below:

bill.age = 50;

A polynomial may be represented using array or structure. A structure may be defined such that it contains two parts — one is the coefficient and second is the corresponding exponent. structure definition may be given asshown below: Struct polynomial int coefficient;int exponent;

## Addition of polynomials

While adding two polynomials, following cases need to be considered. When the degrees of corresponding terms of the two polynomials are same:

This is the normal case when corresponding coefficients of each term can be added directly. For example, thesum of the polynomials



### 7x3+9x2+12

When a structure is created, no memory is allocated.

The structure definition is only the blueprint for the creating of variables. You can imagine it as a datatype. Whenyou define an integer as below: int foo;

The int specifies that, variable foo can hold integer element only. Similarly, structure definition only specifies that,what property a structure variable holds when it is defined.

How to define a structure variable?

Once you declare a structure person as above. You can define a structure variable as:Person bill;

Here, a structure variable bill is defined which is of type structure Person.

When structure variable is defined, only then the required memory is allocated by the compiler.

Considering you have either 32-bit or 64-bit system, the memory of float is 4 bytes, memory of int is 4 bytes andmemory ofchar is I byte.

Hence, 58 bytes of memory is allocated for structure variable bill.How to access members of a structure?

The members of structure variable is accessed using a dot (.) operator.

Suppose, you want to access age of structure variable bill and assign it 50 to it. You can perform this task by usingfollowing code below:

bill.age = 50;

A polynomial may be represented using array or structure. A structure may be defined such that it contains two parts — one is the coefficient and second is the corresponding exponent. structure definition may be given asshown below: Struct polynomial int coefficient;int exponent;

## Addition of polynomials

While adding two polynomials, following cases need to be considered. When the degrees of corresponding terms of the two polynomials are same:

This is the normal case when corresponding coefficients of each term can be added directly. For example, thesum of the polynomials



### 7x3+9x2+12

12x3+11x2+19 is a simple addition where all the degrees of the corresponding terms are same.

2. When the degrees of corresponding terms of the polynomials are different: Here, the term with the larger degree pre-dominates.

9x4+5x3+ +2x

### 3x4+ +4x2+7x



### Multiplication of polynomials

Suppose the given linked lists are:

Polyl: 3X3 + 6xl - 9

Poly2: 9X3 - 8x2+ 7xi +2

* Now, according to the problem statement, we need to multiply these polynomials Polyl and Poly2.
* So, we will multiply each term in Polyl with every term in Poly2, and then we will add up all the term with the same power of x such that each term in the final resultant polynomial list has a different power ofx.

Output

Resultant Polynomial: 27x6 - 24x5 + 75x4- 123x3 + 114x2 - 51xl - 18

Conclusion: Thus, we implemented polynomial operations using structure.

Questions

What is structure?

How can we access structure variables? iii. What is structure for polynomial?



Code:-

#include <iostream>

using namespace std;

struct polynomial

{

    int coeff;

    int expo;

};

int ipoly(struct polynomial p[])

{

    int t1, i, k;

    cout << "\n Enter the polynomial details:";

    cout << "\n Enter the Total number of terms in the polynomial:";

    cin >> t1;

    cout << "\n Enter the COEFFICIENT and EXPONENT" << endl;

    for (i = 0; i < t1; i++)

    {

        cout << " Enter the Coefficient(" << i + 1 << "):";

        cin >> p[i].coeff;

        cout << " Enter the Exponent(" << i + 1 << "):";

        cin >> p[i].expo;

    }

    cout << "\n The polynomial is :";

    for (k = 0; k < t1 - 1; k++)

    {

        cout << p[k].coeff << "(x^" << p[k].expo << ")+";

    }

    cout << p[k].coeff << "(x^" << p[k].expo << ")";

    return t1;

}

void add\_polynomial()

{

    int t1, t2;

    struct polynomial p2[20];

    struct polynomial p3[20];

    struct polynomial p4[20];

    cout << "\n First polynomial :";

    t1 = ipoly(p2);

    cout << "\n Second polynomial:";

    t2 = ipoly(p3);

    int i, j, k;

    i = 0;

    j = 0;

    k = 0;

    while (i < t1 && j < t2)

    {

        if (p2[i].expo == p3[j].expo)

        {

            p4[k].coeff = p2[i].coeff;

            p4[k].expo = p2[i].expo;

            i++;

            k++;

        }

        else

        {

            p4[k].coeff = p3[j].coeff;

            p4[k].expo = p3[j].expo;

            j++;

            k++;

        }

    }

    /\*for rest over terms of polynomial 1 \*/

    while (i < t1)

    {

        p4[k].coeff = p2[i].coeff;

        p4[k].expo = p2[i].expo;

        i++;

        k++;

    }

    /\* for rest over terms of polynomial 2 \*/

    while (j < t2)

    {

        p4[k].coeff = p3[j].coeff;

        p4[k].expo = p3[j].expo;

        j++;

        k++;

    }

    cout << "\n Addition of the given polynomial is:";

    for (i = 0; k - 1; i++)

        cout << p4[i].coeff << "(x^" << p4[i].expo << ")+";

    cout << p4[i].coeff << "(x^" << p4[i].expo << ")";

}

void mul\_polynomial()

{

    int t1, t2, i, j, c;

    struct polynomial p2[20];

    struct polynomial p3[20];

    struct polynomial p4[50];

    cout << "\n First polynomial";

    t1 = ipoly(p2);

    cout << "\n Second polynomial";

    t2 = ipoly(p3);

    c = -1;

    for (i = 0; i < t1; i++)

    {

        for (j = 0; j < t2; j++)

        {

            p4[++c].expo = p2[i].expo + p3[j].expo;

            p4[c].coeff = p2[i].coeff \* p3[j].coeff;

        } /\*end of j loop\*/

    } /\* end of i loop\*/

    cout << "\n Product of two polynomial is:\n";

    cout << endl;

    for (i = 0; i < c; i++)

        cout << p4[i].coeff << "(x^" << p4[i].expo << ")+";

    cout << p4[i].coeff << "(x^" << p4[i].expo << ")";

}

int main()

{

    int ch;

    char ans;

    struct polynomial p1[20];

    do

    {

        cout << "\n 1.Taking input and printing polynomial\n 2.Addition \n 3.Multiplication \n Enter choice";

        cin >> ch;

        switch (ch)

        {

        case 1:

            ipoly(p1);

            break;

        case 2:

            add\_polynomial();

            break;

        case 3:

            mul\_polynomial();

            break;

        }

        cout << "\n Do you want to continue(Y/y)=>";

        cin >> ans;

    } while (ans == 'Y' || ans == 'y');

    return 0;

}

Output:-

1.Taking input and printing polynomial

2.Addition

3.Multiplication

Enter choice1

Enter the polynomial details:

Enter the Total number of terms in the polynomial:3

Enter the COEFFICIENT and EXPONENT

Enter the Coefficient(1):x y