**1-** **Matematikte bir polinom, değişkenlerden (belirsiz olarak da bilinir) ve sabitlerden oluşturulan bir ifadedir. Toplama, çıkarma, çarpma ve sabit negatif olmayan tam sayı üsleri işlemlerini kullanılır. Örneğin, x^2 − 4x + 7 bir polinomdur, ancak**

**x^2 −4/x + 7x^3/2 değil, çünkü ikinci terimi değişkene göre bölmeyi içeriyor x ve ayrıca üçüncü terimi tam sayı olmayan bir üs içerdiği için.**

**2- Polinomlar cebirde, matematik ve fen bilimlerinde en önemli kavramlardan biridir. Temel kelime problemlerinden çok çeşitli problemleri kodlayan polinom denklemleri oluşturmak için kullanılırlar. Bilimlerde karmaşık problemler; arasında değişen ayarlarda görünen polinom işlevlerini tanımlamak için kullanılırlar.**

**Temel kimya ve fizikten ekonomiye kadar uzanır ve diğer fonksiyonlara yaklaşmak için matematik ve sayısal analizde kullanılır. Polinomlar, cebir ve cebirsel geometrideki en güçlü kavramlardan biri olan polinom halkalarını oluşturmak için kullanılır.**

**3- Bu laboratuvarda polinomları oluşturmak ve işlemek için çeşitli işlevler uygulayacaksınız. Polinomun terimleri**

**üs değerlerinin azalan sırasına göre tek bir linked list’de saklanacaktır.**

**4- Şekil 1 nasıl yapıldığını gösterir.**

**P(x)= 2.34x^4 − 4x + 5 ve −x^5 + 6x^3 + 5.1x^2 − 8x linked listelerle temsil edilir. Görüldüğü gibi her polinomun terimi, bir katsayı ve bir üsten oluşan bir PolyNode yapısı (PolyNode.h'de tanımlanmıştır) ile temsil edilir. Bir katsayı gerçek bir sayı olabilirken, üs bir doğal sayı, yani bir tam sayı olmalıdır. Polinomun terimleri, linked list’de üslerine göre azalan sırada saklanır.**

**Poly Creation and Operations**

**1- PolyNode.h'de bir polinom struct’ı tanımlanmıştır ve aşağıdaki yapıya sahiptir:**

**2- Her node’ın 0 olamayacak bir katsayısı ve negatif olmaması gereken bir üssü vardır. Polinom işlevleri Poly.h'de tanımlanır ve Poly.cpp'de uygulanır ve aşağıdaki prototiplere sahiptir:**

**3- Bir dize ifadesi alan CreatePoly tarafından bir polinom oluşturulabilir. Bu işlevi uygulamak için, polinom ifadesini içeren verilen dizeyi polinom terimlerine ayrıştırmanız ve oluşturulan polinom için her terimi eklemeniz lazım. Aşağıdakiler, string dizeleriyle polinom oluşturma örnekleridir:**

**4- Her terim (birinci terim hariç) bir karakter işareti (+ veya -) ile başlar. İşaretin ardından katsayı belirtilir ardından 'x' ve 'ˆ' karakterleri gelir. Sonunda üs geliyor. Katsayı 1 ise, ihmal edilebileceğine dikkat edin. “xˆ3” ifadesindeki gibi. Benzer şekilde, üs 1 ise, “2x” ifadesindeki gibi atlanabilir. Son olarak, üs 0 ise, "x" ve "ˆ" karakterleri tamamen atlanır, örneğin "44".**

**5- Polinom ifadelerinin iyi biçimli olacağını, yani herhangi bir hata içermeyeceğini varsayabilirsiniz. Ancak terimler arasında bir veya daha fazla boşluk olabilir, yani '+' ve '-' işaret sembollerinden önce veya sonra. Verilen bir ayrıştırmak için string dizesi, tüm boşluk karakterlerini kaldırarak yeni bir string dizesi oluşturmanızı öneririz. Ardından yeni dizeyi her seferinde bir terim ayrıştırın. Polinomun her teriminin bir işaret sembolü ile ayrıldığına dikkat edin.**

**6- Yukarıda gösterildiği gibi polinomlar üzerinde çalışmak için kullanılan birkaç fonksiyon daha vardır. AddNode işlevi, yeni bir terim eklemek veya bir polinomdan/bir polinomdan mevcut bir terimi silmek için kullanılır. Örneğin, aşağıdaki polinom verildiğinde**

**x^3 + x + 2, AddNode(poly, 3, 1) polinomuna “3x” terimini ekleyerek polinomu x^3 + 4x + 2'ye dönüştürür. Benzer şekilde, AddNode(poly, -1, 3), ilk terimi kaldırarak x^3 + x + 2'yi x + 2'ye dönüştürür. "AddNode" yeni polinomun olası yeni başlığına bir pointer döndürür.**

**7- Evalute(x), belirli bir x noktasında polinomun değerini almak için kullanılır. Print() polinomun terimlerini yazdırırken, Plot(x1, x2) için polinomu çizer. x1 <= x <= x2 ve 12 >= y >= -12.**

**8- Ayrıca bir veya iki polinom üzerinde çalışan 4 fonksiyon tanımladık, ve sonucu içeren yeni bir polinom üretin. Bu işlevler şunlardır:**

**9- İlk işlev olan Derivative, bir polinomun türevini almak ve elde edilen polinomu döndürmek için kullanılır. İkinci işlev Add, iki polinomu alan, onları toplayan ve elde edilen polinomu döndüren polinom toplamadır. Üçüncü işlev Subtract, polinom çıkarma işlemidir, iki polinom alan, ikinci polinomu birinciden çıkaran ve elde edilen polinomu döndürür. Son olarak Multiply, iki polinom alır, birinciyi ikinci ile çarpar ve elde edilen polinomu döndürür. Bu fonksiyonların hiçbirinin bu fonksiyonlara parametre olarak aktarılan polinomları değiştirmediğine dikkat edilmelidir. Sadece geçen polinomları kullanırlar ve tamamen yeni polinomlar üretirler.**

**TESTİNG**

**1- Projenin temel kodu verilir. PolyNode.h, Poly.h ve Poly.cpp, polinom node ve polinom işlevleri için bildirimleri içerir. Sizin işiniz Poly.cpp'deki yöntemleri yukarıda anlatıldığı gibi doldurmak.**

**2- main.cpp, polinom sınıfınızı test etmek için çeşitli testler içerir. Polinom fonksiyonlarınızı kapsamlı bir şekilde test etmek için kendi test programınızı uygulamanızı öneririz. Not verme sırasında diğer test programlarını kullanacağız.**