Федеральное государственное автономное образовательное учреждение

высшего образования

«Нижегородский государственный университет

им. Н.И. Лобачевского»

Институт информационных технологий математики и механики

Специальность прикладная математика и информатика

Отчёт по лабораторной работе

Арифметические операции с полиномами

Выполнил:

Студент ПМИгр. 381703-1

Арефьев А. М.

Проверил:

Ассистент кафедры МОСТ ИИТММ

Волокитин В. Д.

Нижний Новгород

2019 г.

Оглавление

[Введение 3](#_Toc499325868)

[Постановка задачи 4](#_Toc499325869)

[Руководство пользователя 5](#_Toc499325870)

[Руководство программиста 6](#_Toc499325871)

[Описание структур данных 6](#_Toc499325872)

[Описание алгоритмов 6](#_Toc499325873)

[Описание структуры программы 7](#_Toc499325874)

[Заключение 8](#_Toc499325875)

[Литература 9](#_Toc499325876)

[Приложения 10](#_Toc499325877)

# Введение

Изучение полиномиальных уравнений и их решений составляло едва ли не главный объект «классической алгебры».

С изучением полиномов связан целый ряд преобразований в математике: введение в рассмотрение нуля, отрицательных, а затем и комплексных чисел, а также появление теории групп как раздела математики и выделение классов специальных функций в анализе.

Техническая простота вычислений, связанных с полиномами способствовали развитию методов разложения в ряды и полиномиальной интерполяции в математическом анализе.

Полиномы также играют ключевую роль в алгебраической геометрии, объектом которой являются множества, определённые как решения систем многочленов.

Особые свойства преобразования коэффициентов при умножении многочленов используются в алгебраической геометрии, алгебре, теории узлов и других разделах математики для кодирования или выражения полиномами свойств различных объектов.

Поэтому такая задача, как создание программы для работы с полиномами имеет важное практическое значение и всегда будет актуальна.

# Постановка задачи

# Разработать программу, выполняющую арифметические операции с полиномами трех переменных (x, y и z): сложение, вычитание, умножение на константу, умножение двух полиномов. Считается, что полином составлен из мономов от трех переменных с ограничением на степень каждой переменой от 0 до 9 (Опционально можно расширить данное ограничение). Коэффициенты полинома - вещественные числа. Работоспособность программы необходимо проверить с помощью Google Test-ов. Кроме того, необходимо разработать пользовательское консольное приложение.

# Особенности реализации:

# В качестве структуры хранения полинома использовать список мономов с ненулевыми коэффициентами (односвязный или двухсвязный). Односвязный список удобнее реализовать с фиктивной головой. Элементы списка хранить упорядоченными.

# Степень полинома хранить в "свернутом" виде, т. е. степень должна быть представлена как трехзначное число, где число сотен – это степень при переменной “x”, число десятков - степень при переменной “y”, число единиц - степень при переменной “z”.

# Сложение полиномов осуществлять алгоритмом слияния упорядоченных массивов.

# Вычитание полиномов допускается выполнять через сложение с умножением на константу (C = A - B = A + (-1)\*B)

# При умножении и сложение (вычитание) необходимо следить, чтобы в итоговом полиноме были приведены подобные слагаемые и не хранилось мономов с нулевым коэффициентом.

# Если при умножении полиномов полученные степени переменных больше 9, выводить сообщение об ошибке.

# Считывание полинома у пользователя допускается в любом удобном для вас виде. Необходимо предоставить пользователю правила ввода данных

1. Следует учесть, что пользователь может вводить полином, не упорядочив в нем мономы.

# Руководство пользователя

1. Запустите программу.
2. Введите первый полином.
3. Введите второй полином.
4. Выберите операцию.

## Правила ввода математического выражения

# Степень должна быть не отрицательна и меньше 10

# Коэффициент монома не должен быть равен 0

# Сначала вводится количество мономов

# Затем для каждого монома требуется ввести его коэффициент и степени переменных

# Руководство программиста

## Описание структур данных

Мономы представляют собой структуру, в которой содержатся коэффициент монома и свернутая степень переменных, а также указатель на следующий моном.

Класс полиномов представляет из себя односвязный список мономов с фиктивной головой. Элементы списка хранятся упорядоченными.

## Описание алгоритмов

1.Добавление монома.

1. Вход: моном
   1. Если это первый моном, то добавить его
   2. Если не первый, то
      1. В цикле, пока указатель на следующий моном не будет равен 0 и его степень будет меньше степени данного монома, двигаться к следующему моному
      2. Если цикл прервался из-за первого условия, то добавить моном в конец списка
      3. Если цикл прервался из-за второго условия и степени равны, то поменять значение коэффициент последнего монома на сумму коэффициентов последнего и вставляемого
      4. Если цикл прервался из-за второго условия и степени не равны, то добавить моном перед мономом с меньшей степенью
2. Полином с добавленным мономом

2.Сложение полиномов

1. Вход: Полином А, полином Б
2. В цикле пока не обошли один из полиномов
   1. Если моном полинома А больше монома полинома Б добавляем его в полином С, двигаемся к следующему моному полинома А
   2. Если моном полинома Б больше монома полинома А добавляем его в полином С, двигаемся к следующему моному полинома Б
3. После цикла добавляем в полином С все мономы не закончившегося полинома
4. Выход: полином С

3. Умножение полиномов

1. Вход: Полином А, полином Б,N- размер полинома А, M - размер полинома Б
2. В цикле от 0 до N
   1. Берем моном полинома А (К)
   2. В цикле от 0 до М
      1. Берем моном полинома Б (Л)
      2. Умножаем К и Л
      3. Добавляем получившийся моном в полином С
      4. Двигаемся в полиноме Б к следующему моному
   3. После цикла двигаемся к следующему моному полинома А
3. Выход: полином С

## Описание структуры программы

Программа состоит из 4 проектов:

1. main(Запускаемый проект, в котором реализован пользовательский интерфейс)
2. pol(Библиотека, содержащая реализацию класса полиномов)
3. gtest(Библиотека, необходимая для запуска тестов )
4. test(Проект, в котором содержатся тесты)

# Заключение

В итоге удалось реализовать программу,которая производит арифметические операции над полиномами.

Реализовано следующее:

1. В качестве структуры хранения полинома использован с фиктивной головой односвязный список мономов с ненулевыми коэффициентами и упорядоченными элементами.
2. Степень полинома хранится в "свернутом" виде.
3. Вычитание полиномов выполняется через сложение с умножением на константу (C = A - B = A + (-1)\*B)
4. При умножении и сложении в итоговом полиноме приводятся подобные слагаемые и не хранятся мономы с нулевым коэффициентом.
5. Если при умножении полиномов полученные степени переменных больше 9, выводится сообщение об ошибке.
6. Правила ввода данных выводятся в приложении
7. Пользователь может вводить полином, не упорядочив в нем мономы.

# Литература

1. Левитин А. В. Глава 4. Метод декомпозиции: Сортировка слиянием // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 169–172. — 576 с.
2. Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Алгоритмы: построение и анализ = IntroductiontoAlgorithms / Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — 1296 с.

# Приложения

Вставка

voidpol::insert(monomm) {

if(m.a!=0 &&m.s<1000 &&m.s>=0){

if (size == 0) {

monom \*nw = newmonom;

head->next = nw;

nw->next = NULL;

nw->a = m.a;

nw->s = m.s;

size++;

}

else {

monom \*tmp = head, \*tmp2 = head;

inti = 0;

monom \*nw = newmonom;

while ((tmp->next != NULL) && ((tmp->next)->s >m.s)) {

tmp = tmp->next;

i++;

}

for (int j = 0; j <i; j++) {

tmp2 = tmp2->next;

}

if (tmp->next == NULL) {

tmp->next = nw;

nw->next = NULL;

size++;

nw->a = m.a;

nw->s = m.s;

}

elseif (tmp->next->s == m.s) {

tmp->next->a = m.a + tmp->next->a;

if (tmp->next->a < 0.00000000001 &&tmp->next->a > -0.00000000001) {

tmp2->next = tmp->next;

size--;

deletenw;

}

}

else {

nw->next = tmp->next;

tmp2->next = nw;

size++;

nw->a = m.a;

nw->s = m.s;

}

}

}

else {

cout<<"неверновведенмоном"<<endl;

throw ("wrong monom");

}

}

Сложение

polpol::operator+ (polsec) {

pol \*np = newpol;

monom \*tmp = head, \*tmp2 = sec.head;

while ((tmp->next != NULL) && (tmp2->next != NULL)) {

if (((tmp->next)->s) >= ((tmp2->next)->s)) {

np->insert(\*(tmp->next));

tmp = tmp->next;

}

else {

np->insert(\*(tmp2->next));

tmp2 = tmp2->next;

}

}

while (tmp->next != NULL)

{

np->insert(\*(tmp->next));

tmp = tmp->next;

}

while (tmp2->next != NULL)

{

np->insert(\*(tmp2->next));

tmp2 = tmp2->next;

}

return \*np;

}

Умножение

polpol::operator\* (polsec) {

setlocale(LC\_ALL, "Russian");

pol \*np = newpol;

monom \*tmp = head, \*tmp2;

for (inti = 0; i< size; i++) {

tmp2 = sec.head;

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

if (((tmp->next->s) / 100 + (tmp2->next->s) / 100)<10 && (((tmp->next->s) % 100)/10 + ((tmp2->next->s) % 100)/10 )<10 && ((tmp->next->s) % 10 + (tmp2->next->s) % 10)<10) {

monom \*nw = newmonom;

nw->s = (tmp->next->s) + (tmp2->next->s);

nw->a = (tmp->next->a)\*(tmp2->next->a);

nw->next;

tmp2 = tmp2->next;

np->insert(\*nw);

}

else {

cout<<"превышенастепень"<<endl;

throw ("limit power is exceeded");

}

}

tmp = tmp->next;

}

return \*np;

}