# Вступ

В даній роботі реалізована задача виконання символьних операцій з многочленами. Був здійснений математичний опис задачі, розроблені алгоритми її реалізації та здійснена сама реалізація на одній з версій алгоритмічної мови Pascal. Правильність роботи програми перевірена на контрольному прикладі. Здійснювалося також тестування програми на екстремальних вхідних даних (піднесення многочлена до нульового степеня, множення на нуль, ділення многочлена на самого себе і т. д.).

При виконанні роботи використовувався структурний метод програмування, що дало змогу зробити програму короткою, легко модифікованою. Оскільки всі розроблені процедури поміщені в модуль, відкомпільований у tpu‑файл, їх легко використати при написанні програм, які реалізують близькі за темою задачі.

Вибрана алгоритмічна мова TurboPascal повністю забезпечує структурний стиль програмування, оскільки це структурована мова високого рівня, на якій можна написати програму практично необмеженого розміру і довільного призначення.

# 1. Постановка задачі

В представленій роботі реалізується така задача.

Створити тип даних, які б найкраще відповідали означенню многочлена, описати і створити змінні, які б відповідали нулю та одиниці кільця многочленів, передбачити ввід та вивід многочленів згідно із загальноприйнятими стандартами.

Забезпечити реалізацію таких операцій з многочленами:

1. додавання n многочленів
2. множення n многочленів
3. піднесення многочлена до довільного степеня
4. знаходження многочлена від многочлена
5. ділення многочленів
6. знаходження похідної довільного порядку від многочлена
7. знаходження невизначеного інтегралу довільного порядку від многочлена

Операції визначені у розділі 1 представленої роботи.

Інтерфейс користувача створити у вигляді меню.

При створенні програмного забезпечення дотримуватися структурного стилю програмування. Створене програмне забезпечення перевірити на контрольному прикладі. Опис контрольного прикладу приведений у розділі 3 представленої роботи.

# 2. Математичний опис задачі

***Многочленом (поліномом)*** над полем **A** називається вираз виду

**anxn+an-1xn-1+ … +a0 (1)**

де ai – елементи поля **A**, n – натуральне число. n‑максимальне число таке, що коефіцієнт при xn відмінний від 0. n називається степенем многочлена. В даній роботі розглядаються многочлени над полем дійсних чисел.

Загалом многочлени можна трактувати як масиви чисел (елементів поля)

Довільний елемент поля (скаляр) може трактуватися як многочлен нульового степеня.

На множині многочленів введено відношення рівності та операції додавання і множення.

Два многочлени будуть вважатися ***рівними***, якщо рівні їх коефіцієнти ai при однакових степенях невідомого.

Многочлени над довільним полем утворюють ***кільце***. Роль нуля відіграє многочлен (0,0,0,…), роль одиниці – многочлен (1,0,0,…). У представленому у даній роботі програмному забезпеченні вони фігурують під іменами відповідно zero, od.

Операцію множення зручно представити як композицію трьох операцій:

множення на скаляр t

(a0, a1, a2…an,0,0…) ⇒(t\*a0, t\*a1, t\*a2…t\*an,0,0…) (3)

і множення на многочлен xm (підвищення степеня многочлена на m)

(a0, a1, a2…an,0,0…) ⇒(0,0..am, am+1, an+m,0,0.) (4)

та раніше введеною операцією суми.

На кільці многочленів легко ввести ряд операцій, породжених основними: операцію піднесення многочлена до квадрата, куба і довільного ступеня k, де k – натуральне число, операцію знаходження многочлена від многочлена

f(y)=anyn+an-1yn-1+ … +a0

де в ролі y виступає многочлен y(x)= bmym+am-1ym-1+ … +b0.

Крім того, в представленій роботі розглядаються операції знаходження похідної від многочлена

f**’** (x)=(anxn+an-1xn-1+ … +a0)**’**=nanxn-1+(n‑1) an-1 xn-2+ … +a1 (5)

та знаходження невизначеного інтеграла від многочлена

xn+1+ … +a0x+C, (6)

де С – довільна константа. Взяття похідної (диференціювання) і знаходження невизначеного інтеграла (інтегрування) – взаємно обернені операції.

Так визначені операції додавання і множення многочленів легко узагальнити для довільного числа операндів, операції піднесення до степеня, диференціювання та інтегрування – для довільного степеня многочлена, похідної та інтегралу довільного порядку.

В представленій роботі розглядається також операція ***ділення*** многочленів. Многочлени над полем дійсних чисел утворюють кільце, а не поле, тому загалом говорити про ділення многочлена на многочлен нема сенсу. Натомість можна ввести операцію ділення з остачею: для двох многочленів f(x) та g(x) знаходження таких многочленів u(x) та v(x), що f(x)=g(x) u(x)+v(x), причому степінь остачі v(x) менший від ступеня дільника g(x). Загалом степені f(x) i g(x) можуть бути довільні, але задача є нетривіальною тільки тоді, коли степінь дільника менший від степеня діленого. Саме ділення здійснюється згідно з широковідомим алгоритмом, аналогічним до ділення чисел у стовпчик.

# 3.Опис контрольного прикладу

Правильність роботи представленого комплексу програм буде перевірятися на контрольному прикладі. Контрольний приклад виконається коректно, якщо результати, одержані за допомогою програми, співпадуть з результатами, одержаними методами алгебри многочленів, представленим далі.

**Додавання многочленiв**

Доданки

4.00x 2+ 3.00x

2.00x 4-x 3+ 3.00x 2+ 4.50x+ 5.00

4.00x 3- x 2+ 0.40x

**Результат**

2.00x 4+ 3.00x 3+ 6.00x 2+ 7.90x+ 5.00

**Множення многочленiв**

Множники

2.00x 3-1.00x 2+ 2.00x+ 1.00

1.00x 2+ 1.00

-1.00x 2+ 0.50x+ 0.50

**Результат**

-2.00x 7+ 2.00x 6–3.50x 5+ 1.50x 4+ 1.50x+ 0.50

**піднесення многочлена до степеня**

многочлен x 3–2x 2+ 2x‑1

Піднесений до степеня 3

x9-6x 8+18x7-35x 6+48x5-48x4+35x3-18x2+ 6x‑1

**Многочлен від многочлена**

Внутрішній многочлен y=x 2+ x+ 1

Зовнішній многочлен y 2+ 1.00y+ 1.00

**Результат**

x 4+ 2x 3+ 4x 2+ 3x+ 3

**Ділення многочленiв**

Ділене x 5+ 2x 4‑x 3+ 2x 2+ x+ 5.00

Дільник x 2+ x+ 1

Частка x 3+ x 2–3x+ 4

Остача 1.00

**знаходження похідної від многочлена**

многочлен f(x)=x 4‑x 3+ 2x 2+ 3x+ 1

**Похідна 2‑го порядку**

f’’ (x)=12x 2–6x+ 4

**знаходження невизначеного інтеграла від многочлена**

**многочлен**

f(x)=12x 2–6x+ 4

**Інтеграл 2‑го порядку**

x 4‑x 3+ 2x 2+C1x+C0

# 4.Опис програмного забезпечення

Представлене програмне забезпечення розроблене на алгоритмічній мові TurboPascal 7.0. Мова обрана з огляду на її високу структурованість.

Програмне забезпечення записане у двох файлах: файлі kurs.pas, текст якого приведений у додатку 1, та у бібліотеці підпрограм – файлі bibl.pas, який окремо відкомпільований як модуль bibl.tpu. Його текст приведений у додатку 2.

В бібліотеці підпрограм в розділі INTERFACE описані тип poli, що відповідає означенню многочлена (степеня не більше 100), тип mpoli – масиву многочленів, глобальні змінні zero та od, які відіграють роль відповідно нуля та одиниці кільця многочленів.

Там же описані наступні процедури і функції

1. function stepin (a:poli):integer; (знаходження степеня многочлена)
2. procedure riv (a:poli; var b:poli); (присвоєння одному многочлену значення іншого)
3. procedure vvid (n:integer; var a:poli); (ввід многочлена)
4. function poper (a:poli; m:integer):integer; (знаходження коефіцієнта многочлена, попереднього по відношенню до заданого)
5. procedure vyvid (a:poli); (вивід многочлена згідно із загальноприйнятими стандартами)
6. function maxi (n, m:integer):integer; (знаходження числа, більшого з двох)
7. function mini (n, m:integer):integer; (знаходження числа, меншого з двох)
8. procedure suma (a, b:poli; var c:poli); (знаходження суми двох многочленів)
9. procedure nsuma (a:maspoli; n:integer; var c:poli); (знаходження суми n многочленів)
10. procedure dobchy (a:poli; r:real; var c:poli); (добуток многочлена на скаляр)
11. procedure pidvst (a:poli; n:integer; var c:poli); (підвищення степеня многочлена на n одиниць)
12. procedure dobutok (a, b:poli; var c:poli); (знаходження добутку двох многочленів)
13. procedure ndobutok (a:maspoli; n:integer; var c:poli); (знаходження добутку n многочленів)
14. procedure mpoli (a:poli; m:integer; var c:poli); (знаходження m‑го степеня многочлена)
15. procedure polipoli (a, b:poli; var c:poli); (знаходження многочлена від многочлена)
16. procedure dilen (a, b:poli; var c, c1:poli); (знаходження частки і остачі від ділення двох многочленів)
17. procedure dyfer (a:poli; var b:poli); (знаходження похідної від многочлена)
18. procedure integ (a:poli; var b:poli); (знаходження невизначеного інтеграла від многочлена)

Тексти процедур та функцій містяться в розділі IMPLEMENTATION.

Головним файлом пакету є файл kurs.pas. Його текст приведений в додатку 1.

В цьому файлі конструюються вже описані змінні zero i od. Тут же реалізований інтерфейс користувача, який розроблений у вигляді меню. Управління роботою пакету здійснюється через ввід числа, яке відповідає одному із запропонованих режимів роботи. До складу файлу входить директива компілятора $M, яка збільшує розмір стеку до максимально можливого.

# 5. Виконання контрольного прикладу

Додавання многочленiв

Доданки

4.00x^ 2+ 3.00x

2.00x^ 4–1.00x^ 3+ 3.00x^ 2+ 4.50x+ 5.00

4.00x^ 3–1.00x^ 2+ 0.40x

Результат

2.00x^ 4+ 3.00x^ 3+ 6.00x^ 2+ 7.90x+ 5.00

Множення многочленiв

Множники

2.00x^ 3–1.00x^ 2+ 2.00x+ 1.00

1.00x^ 2+ 1.00

-1.00x^ 2+ 0.50x+ 0.50

Результат

-2.00x^ 7+ 2.00x^ 6–3.50x^ 5+ 1.50x^ 4+ 1.50x+ 0.50

пiднесення многочлена до степеня

многочлен 1.00x^ 3–2.00x^ 2+ 2.00x‑1.00

Піднесений до степеня 3

1.00x^9–6.00x^ 8+18.00x^7–35.00x^ 6+48.00x^5–48.00x^4+35.00x^3–18.00x^2+ 6.00x‑1.00

Многочлен від многочлена

Внутрішній многочлен 1.00x^ 2+ 1.00x+ 1.00

Зовнішній многочлен 1.00x^ 2+ 1.00x+ 1.00

Результат 1.00x^ 4+ 2.00x^ 3+ 4.00x^ 2+ 3.00x+ 3.00

Ділення многочленiв

Ділене 1.00x^ 5+ 2.00x^ 4–1.00x^ 3+ 2.00x^ 2+ 1.00x+ 5.00

Дільник 1.00x^ 2+ 1.00x+ 1.00

Частка 1.00x^ 3+ 1.00x^ 2–3.00x+ 4.00

Остача 1.00

знаходження похідної від многочлена

многочлен 1.00x^ 4–1.00x^ 3+ 2.00x^ 2+ 3.00x+ 1.00

Похідна 2‑го порядку

12.00x^ 2–6.00x+ 4.00

знаходження невизначеного інтеграла від многочлена

без врахування констант

многочлен

12.00x^ 2–6.00x+ 4.00

Інтеграл 2‑го порядку

1.00x^ 4–1.00x^ 3+ 2.00x^ 2

# 6. Інструкція користувачу про роботу з програмним забезпеченням

До складу пакету входять файли вихідних текстів kurs.pas i bibl.pas, а також відкомпільовані файли kurs.exe та bibl.tpu.

Робота пакету можлива як у інтегрованому середовищі Turbo-Pascal, так і безпосередньо з операційної системи, запускаючи на виконання файл kurs.exe.

Ввести число для вибору потрібного режиму роботи.

1. Після вибору режиму роботи з’являється вікно вводу, де користувачу надається можливість послідовно вводити потрібні для роботи дані після виводу відповідного повідомлення. Зокрема, коефіцієнти многочлена вводяться, починаючи від коефіцієнта при максимальному степеню невідомого, і після появи відповідного степеня. Коефіцієнтами многочленів є дійсні числа.
2. Після вводу всіх необхідних даних з’являється вікно, у якому приведені умова задачі та результат роботи програми. Для повернення до режиму меню слід натиснути довільну клавішу.
3. Для виходу з пакету в режимі меню ввести число 0.

# Висновки

В даній роботі реалізована задача «Виконання символьних операцій з многочленами». Здійснено математичний опис задачі, постановку задачі та розробку програмного пакету згідно з постановкою. Роботу пакету перевірено на контрольному прикладі, одержано результати, що співпадають з теоретичними.

Дальшим розвитком теми може бути створення «формульного редактора», який спрощував би довільні математичні вирази, до складу яких входять многочлени.

# 

**Додаток**

Вихідні тексти програм

Файл kurs.pas

program kurs;

{$M 65520,0,655360}

Uses Crt, bibl;

label 1;

{початок програми}

var i, m, menu:integer;

n:array [1..20] of integer;

a:maspoli; c, c1:poli;

nb:integer; b:poli;

na:integer; ba:poli;

roz1, roz2:integer;

begin

{створення кiльцевого нуля zero i кiльцевої одиницi od}

for i:=0 to 100 do begin zero[i]:=0.00; od[i]:=0 end;

od[0]:=1;

while true do begin

clrscr;

TextBackground(LightMagenta);

TextColor(Yellow);

roz1:=25; roz2:=10;

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Сума m многочленiв 1');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Добуток m многочленiв 2');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Многочлен в степенi m 3');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Многочлен вiд многочлена 4');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Дiлення многочленiв 5');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Диференцiювання многочлена 6');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Iнтегрування многочлена 7');

gotoxy (roz1, roz2);

roz2:=roz2+1;

writeln ('Вихiд 0');

roz2:=roz2+1;

gotoxy (roz1+10, roz2); readln(menu);

if (menu=0) then goto 1;

TextBackground(LightBlue);

TextColor(Yellow);

{\*\*\*\*\*\*\*\*\*\*\*\*\*додавання\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

if (menu=1) then

begin

Clrscr;

riv (zero, c);

write ('число многочленiв ');

read(m);

for i:=1 to m do

begin

writeln ('степiнь многочлена ');

read (n[i]);

vvid (n[i], a[i]);

end;

nsuma (a, m, c);

Clrscr;

writeln ('Додавання многочленiв');

write('Доданки');

for i:=1 to m do vyvid (a[i]);

writeln;

write('Результат');

vyvid(c);

repeat until keypressed;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*множення\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

if (menu=2) then

begin

Clrscr;

write ('число многочленiв ');

read(m);

for i:=1 to m do

begin

writeln ('степiнь многочлена ');

read (n[i]);

vvid (n[i], a[i]);

end;

ndobutok (a, m, c);

Clrscr;

writeln ('Множення многочленiв');

write('Множники');

for i:=1 to m do vyvid (a[i]);

writeln;

write('Результат');

vyvid(c);

repeat until keypressed;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*пiднесення многочлена до степеня\*\*\*\*\*\*}

if (menu=3) then begin

clrscr;

riv (zero, c);

writeln ('пiднесення многочлена до степеня');

writeln ('степiнь многочлена ');

read(nb);

vvid (nb, b);

writeln;

write ('До якого степеня пiднести ');

read(m);

mpoli (b, m, c);

clrscr;

writeln ('пiднесення многочлена до степеня');

write('многочлен');

vyvid(b);

writeln;

write ('Пiднесений до степеня ', m:2);

vyvid(c);

repeat until keypressed;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*знаходження многочлена вiд многочлена\*\*\*\*\*\*}

if (menu=4) then

begin

riv (zero, c);

clrscr;

writeln ('Знаходження многочлена вiд многочлена ');

writeln ('Внутрiшнiй многочлен');

writeln ('степiнь многочлена ');

read(nb);

vvid (nb, b);

writeln;

writeln ('Зовнiшнiй многочлен');

writeln ('степiнь многочлена ');

read(na);

vvid (na, ba);

writeln;

clrscr;

writeln ('Многочлен вiд многочлена');

write ('Внутрiшнiй многочлен');

vyvid(b);

writeln;

write ('Зовнiшнiй многочлен');

vyvid(ba);

polipoli (b, ba, c);

writeln;

write('Результат');

vyvid(c);

repeat until keypressed;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*дiлення многочленiв\*\*\*\*}

if (menu=5) then

begin

riv (zero, c);

riv (zero, c1);

clrscr;

writeln ('Дiлення многочленiв ');

writeln('Дiлене');

writeln ('степiнь многочлена ');

read(nb);

vvid (nb, b);

writeln;

writeln ('Дiльник ');

writeln ('степiнь многочлена ');

read(na);

vvid (na, ba);

writeln;

clrscr;

dilen (b, ba, c, c1);

clrscr;

writeln ('Дiлення многочленiв');

write('Дiлене');

vyvid(b);

writeln;

write('Дiльник');

vyvid(ba);

writeln;

write('Частка');

vyvid(c);

writeln;

write('Остача');

vyvid(c1);

repeat until keypressed;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*диференцiювання многочленiв\*\*\*\*\*\*\*\*\*\*\*\*}

if (menu=6) then

begin

clrscr;

writeln ('похiдна вiд многочлена');

writeln ('Порядок похiдноi');

read(m);

writeln ('степiнь многочлена ');

read(nb);

vvid (nb, b);

writeln;

riv (b, c);

for i:=1 to m do

dyfer (b, b);

clrscr;

writeln ('знаходження похiдної вiд многочлена');

write('многочлен');

vyvid(c);

writeln;

write ('Похiдна ', m:2,'-го порядку');

vyvid(b);

repeat until keypressed;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*iнтегрування многочленiв\*\*\*\*\*\*\*\*\*\*\*\*\*}

if (menu=7) then

begin

clrscr;

writeln ('невизначений iнтеграл вiд многочлена');

writeln ('Без вказання констант');

writeln ('Порядок iнтеграла');

read(m);

writeln ('степiнь многочлена ');

read(nb);

vvid (nb, b);

writeln;

riv (b, c);

for i:=1 to m do

integ (b, b);

clrscr;

writeln ('знаходження невизначеного iнтеграла вiд многочлена');

writeln ('без врахування констант');

write('многочлен');

vyvid(c);

writeln;

write ('Iнтеграл ', m:2,'-го порядку');

vyvid(b);

repeat until keypressed;

end;

end;

1: end.

Файл bibl.pas

UNIT bibl; {Бiблiотека пiдпрограм}

INTERFACE

uses crt;

TYPE

poli=array [0..100] of real;

type maspoli=array [1..20] of poli;

var zero, od:poli;

{кільцеві нуль і одиниця}

function stepin (a:poli):integer;

procedure riv (a:poli; var b:poli);

procedure vvid (n:integer; var a:poli);

function poper (a:poli; m:integer):integer;

procedure vyvid (a:poli);

function maxi (n, m:integer):integer;

function mini (n, m:integer):integer;

procedure suma (a, b:poli; var c:poli);

procedure nsuma (a:maspoli; n:integer; var c:poli);

procedure dobchy (a:poli; r:real; var c:poli);

procedure pidvst (a:poli; n:integer; var c:poli);

procedure dobutok (a, b:poli; var c:poli);

procedure ndobutok (a:maspoli; n:integer; var c:poli);

procedure mpoli (a:poli; m:integer; var c:poli);

procedure polipoli (a, b:poli; var c:poli);

procedure dilen (a, b:poli; var c, c1:poli);

procedure dyfer (a:poli; var b:poli);

procedure integ (a:poli; var b:poli);

IMPLEMENTATION

function stepin (a:poli):integer;

{визначення степеня многочлена}

var i:integer;

begin

i:=100;

while ((a[i]=0) and (i>=0)) do i:=i‑1;

stepin:=i;

end;

procedure riv (a:poli; var b:poli);

{присвоення одному многочлену значення iншого}

var i:integer;

begin

for i:=0 to 100 do b[i]:=a[i];

end;

procedure vvid (n:integer; var a:poli);

{ввiд многочлена}

var i:integer;

begin

for i:=100 downto n+1 do a[i]:=0;

writeln ('вводьте многочлен');

for i:=n downto 1 do

begin

write ('x^', i:2,'\*');

read (a[i]);

write ('+');

end;

read (a[0]);

end;

function poper (a:poli; m:integer):integer;

{визначення молодшого на 1 коефiцiента многочлена пiсля m}

var i:integer;

begin

i:=m‑1;

while (a[i]=0) and (i>=0) do i:=i‑1;

poper:=i;

end;

procedure vyvid (a:poli);

{вивiд многочлена}

var i, n:integer;

begin

n:=stepin(a);

writeln;

if ((n>0) or (a[0]<>0)) then

begin

i:=n;

while ((i>=1) and (poper(a, i)>-1)) do

begin

if (a[i]<>0) then begin

if (i>1) then

write (a[i]:5:2,'x^', i:2)

else write (a[i]:5:2,'x');

if (a [poper(a, i)]>0) then write ('+');

end;

i:=i‑1;

end;

if (i>1) then write (a[i]:5:2,'x^', i:2)

else

if (i=1) then write (a[i]:5:2,'x')

else

write (a[i]:5:2);

end

else

write('0');

end;

function maxi (n, m:integer):integer;

begin

if (n>=m) then maxi:=n else maxi:=m;

end;

function mini (n, m:integer):integer;

begin

if (n<=m) then mini:=n else mini:=m;

end;

procedure suma (a, b:poli; var c:poli);

{сума 2 многочленiв}

var i, na, nb, nab, nba:integer;

begin

na:=stepin(a);

nb:=stepin(b);

nab:=maxi (na, nb);

riv (zero, c);

for i:=nab downto 0 do c[i]:=a[i]+b[i];

end;

procedure nsuma (a:maspoli; n:integer; var c:poli);

{сума n многочленiв}

var i:integer;

begin

riv (zero, c);

for i:=1 to n do

suma (c, a[i], c);

end;

procedure dobchy (a:poli; r:real; var c:poli);

{добуток скаляра на многочлен}

var i:integer;

begin

riv (zero, c);

for i:=0 to stepin(a) do

c[i]:=r\*a[i];

end;

procedure pidvst (a:poli; n:integer; var c:poli);

(домноження многочлена на x^n)}

var i:integer;

begin

for i:=stepin(a)+n downto n do

c[i]:=a [i-n];

for i:=stepin(a)+n+1 to 100 do c[i]:=0;

for i:=0 to n‑1 do c[i]:=0;

end;

procedure dobutok (a, b:poli; var c:poli);

{добуток 2 многочленiв}

var i:integer;

t, t1, t2:poli;

begin

riv (zero, t);

for i:=0 to stepin(b) do

begin

riv (zero, t1);

riv (zero, t2);

dobchy (a, b[i], t1);

pidvst (t1, i, t2);

suma (t, t2, t);

end;

riv (t, c);

end;

procedure ndobutok (a:maspoli; n:integer; var c:poli);

{добуток n многочленiв}

var i:integer;

begin

riv (od, c);

for i:=1 to n do

dobutok (c, a[i], c);

end;

procedure mpoli (a:poli; m:integer; var c:poli);

{пiднесення многочлена до степеня}

var i:integer;

begin

riv (od, c);

for i:=1 to m do

dobutok (c, a, c);

end;

procedure polipoli (a, b:poli; var c:poli);

{многочлен вiд многочлена}

var i:integer;

t1, t2:poli;

begin

riv (zero, c);

for i:=0 to stepin(b) do

begin

riv (zero, t2);

riv (zero, t1);

mpoli (a, i, t1);

dobchy (t1, b[i], t2);

suma (c, t2, c);

end;

end;

procedure dilen (a, b:poli; var c, c1:poli);

var n, m, i:integer;

t1, t2, t3, t4, t5:poli;

{дiлення многочленiв з остачею}

begin

riv (a, t4);

n:=stepin(a);

m:=stepin(b);

riv (zero, t3);

while n>=m do

begin

riv (zero, t5);

riv (zero, t1);

riv (zero, t2);

t5 [n-m]:=a[n]/b[m];

suma (c, t5, c);

dobutok (t5, b, t1);

dobchy (t1, – 1, t2);

suma (a, t2, a);

n:=stepin(a);

end;

dobutok (c, b, t3);

dobchy (t3, – 1, t3);

suma (t4, t3, c1);

end;

procedure dyfer (a:poli; var b:poli);

{знаходження похiдноi}

var n, i:integer;

begin

n:=stepin(a);

riv (zero, b);

for i:=n downto 1 do

b [i‑1]:=i\*a[i];

end;

procedure integ (a:poli; var b:poli);

{знаходження невизначеного iнтеграла}

var n, i:integer;

begin

n:=stepin(a);

riv (zero, b);

for i:=n downto 0 do

b [i+1]:=a[i]/(i+1);

end;

end.