http://www.pm298.ru/Math/f1333.JPG

http://www.pm298.ru/Math/f1334.JPG

**void** polynomial::multiplication(polynomial P1, polynomial P2) {

**int** i, j, max;

max = P1.degree + P2.degree;

**int** \*mul = **new** **int**[max + 1];

**for** (i = P1.degree; i >= 0; i--)

**for** (j = P2.degree; j >= 0; j--)

mul[i + j] += P1.coeff[i] \* P2.coeff[j];

cout << **"\nMultiplication:"**;

display(mul, max);

}

Деление

Polynom operator / (const Polynom &p1, const Polynom &p2) {

Polynom temp = p1;

int rdeg = temp.degree - p2.degree + 1;

Polynom res(rdeg);

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

res.coef[rdeg - i - 1] = temp.coef[temp.degree - i - 1] / p2.coef[p2.degree - 1];

for (int j = 0; j < p2.degree; j++) {

temp.coef[temp.degree - j - i - 1] -= p2.coef[p2.degree - j - 1] \* res.coef[rdeg - i - 1];

}

}

temp.reduce();

if (temp.degree != 0) {

cout << "!!! имеется остаток от деления " << temp << endl;

}

return res;

}

bool DividePolynomials(double p[], int n, double q[], int m, double \*\*h, int &l, double \*\*r, int &k)

{

if(n < 0 || m < 0)

return false;

const double epsilon = 0.0000001;

int i, j;

double \*\_p = new double[n + 1];

for(i = 0; i <= n; i ++)

\_p[i] = p[i];

l = n - m;

\*h = new double[l + 1];

if(n >= m){

for(j = n - m; j >= 0; j --){

(\*h)[j] = \_p[j + m] / q[m];

for(i = m; i >= 0; i --)

\_p[i + j] = \_p[i + j] - (\*h)[j] \* q[i];

}

k = m - 1;

while(true){

if(k < 0)

break;

if(fabs(\_p[k]) >= epsilon)

break;

k --;

}

}

else

k = n;

if(k >= 0){

\*r = new double[k + 1];

for(i = k; i >= 0; i --)

(\*r)[i] = \_p[i];

}

delete[] \_p;

return true;

}