**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

# «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО» Інститут прикладного системного аналізу

# Звіт

## про виконання лабораторної роботи №1

## з дисципліни «Проектування та аналіз обчислювальних алгоритмів»

Виконала: студентка II курсу, групи ДА-02

Лесечко Олеся Романівна

Київ – 2021

Завдання 1

Вычислить результат умножения многочлена А(х)=4х4+3х2+2х+1х+0,5 на многочлен В(х)=2х2 +1х+0,4.

#include "polynomials.h"

std::vector<double> input(int n) {

std::cout << "\nPolynomial " << n << "\n\n";

std::cout << "Please enter the highest degree: ";

int degree;

std::cin >> degree;

std::vector <double> vectorCoeff;

while (degree >= 0) {

std::cout << "x^" << degree << " coefficient = ";

double coeff;

std::cin >> coeff;

vectorCoeff.push\_back(coeff);

degree--;

}

return vectorCoeff;

}

void myTask(std::vector <double> vectorCoeff) {

int degree = vectorCoeff.size();

std::cout << "(";

for (int i = 0; i < vectorCoeff.size(); i++) {

degree--;

if (vectorCoeff[i]) {

if (i != 0) std::cout << " + ";

std::cout << vectorCoeff[i];

if (degree > 1) std::cout << "x^" << degree;

if (degree == 1) std::cout << "x";

}

}

std::cout << ")";

}

std::vector <double> calculate(std::vector <double> pol1Coeff, std::vector<double> pol2Coeff) {

int MAXdegree1 = pol1Coeff.size() - 1;

int MAXdegree2 = pol2Coeff.size() - 1;

std::vector<double> result(MAXdegree1 + MAXdegree2 + 1);

int degree1 = pol1Coeff.size() - 1;

for (int i = 0; i < pol1Coeff.size(); i++) {

int degree2 = pol2Coeff.size() - 1;

for (int j = 0; j < pol2Coeff.size(); j++) {

result[degree1 + degree2] += pol1Coeff[i] \* pol2Coeff[j];

degree2--;

}

degree1--;

}

return result;

}

void printResult(std::vector <double> r) {

for (int i = r.size() - 1; i >= 0; i--) {

if (r[i] != 0) {

if(i != r.size() - 1) std::cout << " + ";

std::cout << r[i];

if (i > 1) std::cout << "x^" << i;

if (i == 1) std::cout << "x";

}

}

std::cout << std::endl;

}

void MultiplicationPolynomials() {

std::cout << "TASK 1:\n";

std::cout << "The calculator calculates the product of two polynomials\n";

std::vector <double> pol1Coeff;

std::vector <double> pol2Coeff;

pol1Coeff = input(1);

pol2Coeff = input(2);

std::cout << "\nEntered task\n";

myTask(pol1Coeff);

myTask(pol2Coeff);

std::cout << "\n\nResult\n";

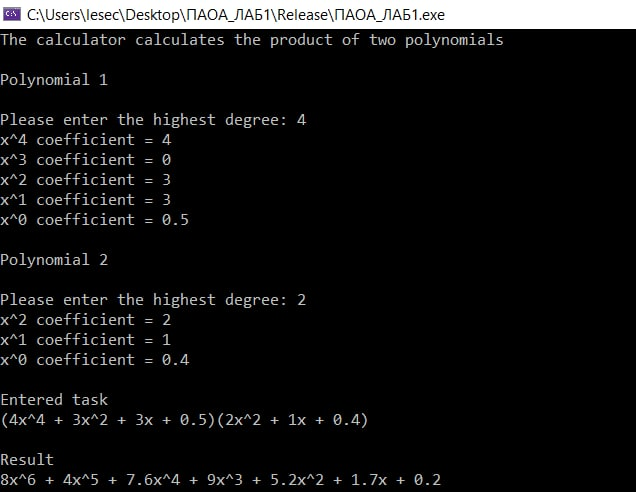
std::vector <double> result;

result = calculate(pol1Coeff, pol2Coeff);

printResult(result);

std::cout << std::endl << std::endl;

}



Завдання 2

Метод «розділяй і володарюй» використовує два рекурсивних виклики, кожен з яких працює приблизно з половиною вхідних даних. За допомогою методу «розділяй і володарюй» знайти кількість додатних елементів цілочисельного масиву А.

#include "divideANDconquer.h"

void personally(std::vector<int>& A, int size) {

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

int a;

std::cin >> a;

A.push\_back(a);

}

}

void randomly(std::vector<int>& A,int size){

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

A.push\_back(rand() % 100 - 50);

}

}

void print(std::vector <int>& A) {

std::cout << "Print array:\n";

for (int i = 0; i < A.size(); i++) {

std::cout << A[i] << " ";

}

}

int Count(std::vector <int>& A, int l, int r) { // Divide-and-conquer algorithm

if (l > r) {

return 0;

}

else if (l == r) {

return A[l] > 0;

}

else {

return Count(A, l, (l + r) / 2) + Count(A, (l + r) / 2 + 1, r);

}

}

int Count2(std::vector <int>& A) { // Linear algorithm

int count = 0;

for (int i = 0; i < A.size(); i++) {

if(A[i] > 0){

count++;

}

}

return count;

}

void counter() {

std::cout << "TASK 2:\n";

std::cout << "Find the number of positive elements integer array A\n\n";

std::cout << "Number of elements: ";

int num;

std::cin >> num;

std::vector <int> A;

std::cout << "Enter manually or randomly? (1 or 2): ";

int opt;

std::cin >> opt;

opt == 1 ? personally(A, num) : randomly(A, num);

//print(A);

clock\_t Start1 = clock();

std::cout << "\nCount 1 = " << Count(A, 0, num - 1) << std::endl;

clock\_t End1 = clock();

clock\_t Start2 = clock();

std::cout << "Count 2 = " << Count2(A) << std::endl << std::endl;

clock\_t End2 = clock();

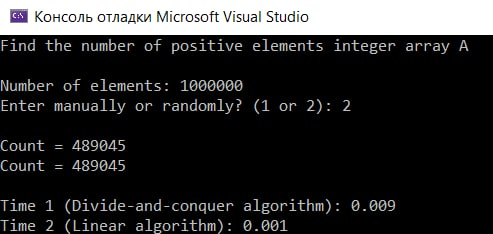
float Time1 = (float(End1 - Start1)) / CLOCKS\_PER\_SEC;

float Time2 = (float(End2 - Start2)) / CLOCKS\_PER\_SEC;

std::cout << "Time 1 (Divide-and-conquer algorithm): " << Time1 << std::endl;

std::cout << "Time 2 (Linear algorithm): " << Time2 << std::endl;

}



Висновки

В ході виконання лабораторної роботи №1 ознайомилась та реалізувала найпростіші алгоритми обробки числових даних, пошуку та сортування