**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО”**

**ЗВІТ**

**з практичної роботи**

**з навчальної дисципліни “Програмування складних алгоритмів”**

**Варіант 1**

**Тема: РОЗРОБКА ПРОГРАМ З ПОШУКУ МІНІМАЛЬНИХ (МАКСИМАЛЬНИХ) ЕЛЕМЕНТІВ В СТАТИЧНИХ ОДНОВИМІРНИХ І ДВОВИМІРНИХ МАСИВАХ**

**Виконав студент групи ТР-15**

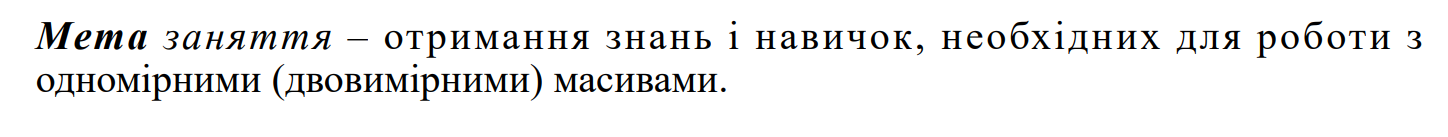
Руденко Владислав

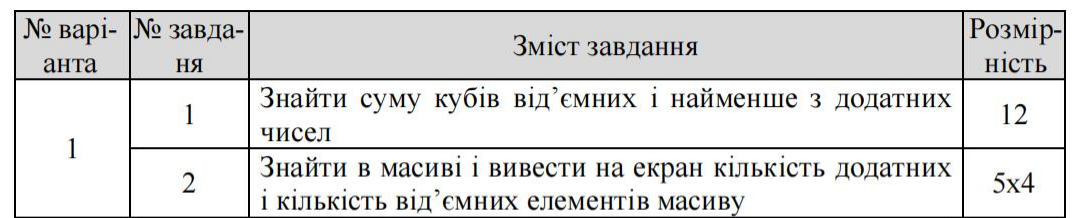
**Перевірив доцент кафедри**

Андрій ОНИСЬКО

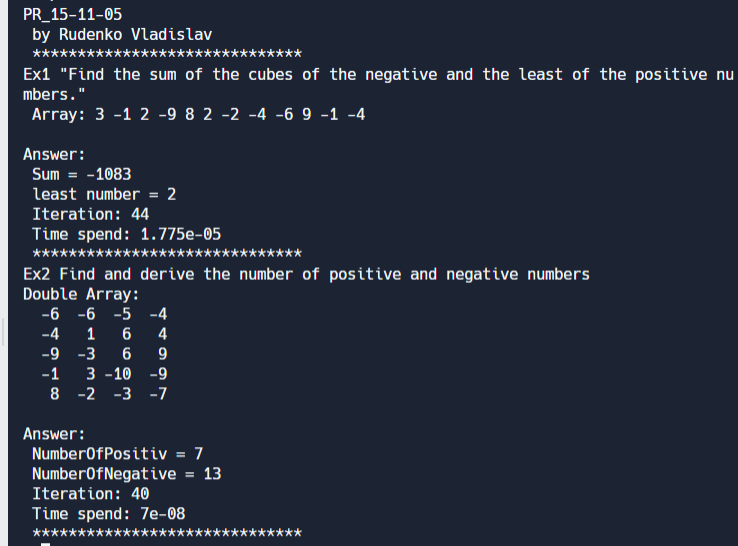
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ року

**Київ 2022**

**Мета роботи + Завдання:**   




**Результат роботи програми**



**Посилання на код**

**<https://replit.com/join/gpdigykafo-hetik>**

**Код:**

**#include <iostream>**

**#include <chrono>**

**#include <cmath>**

**using namespace std;**

**using milliseconds = chrono::duration<long long, micro>;**

**#define ArraySize 12**

**#define DoubleArraySize\_Column 5**

**#define DoubleArraySize\_raw 4**

**int Array[12];**

**int DoubleArray[5][4];**

**void FillIt();**

**void LowerCub();**

**void HowMany();**

**int main() {**

**srand(time(0));**

**FillIt();**

**cout << "PR\_15-11-05\n by Rudenko Vladislav\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nEx1 \"Find the sum of the cubes of the negative and the least of the positive numbers.\"";**

**LowerCub();**

**cout << "Ex2 Find and derive the number of positive and negative numbers\n";**

**HowMany();**

**}**

**void FillIt()**

**{**

**int i,j;**

**for(i=0;i<ArraySize;i++)**

**Array[i] = rand()%20-10;**

**for(i =0;i<DoubleArraySize\_Column;i++)**

**for(j=0;j<DoubleArraySize\_raw;j++)**

**DoubleArray[i][j] = rand()%20-10;**

**}**

**void LowerCub()**

**{**

**int i,j, lowest=Array[0], sum=0, INTER=0;**

**cout << "\n Array: ";**

**for(i=0;i<ArraySize;i++)**

**{**

**cout << Array[i] << " ";**

**}**

**const auto TimeStart = std::chrono::high\_resolution\_clock::now();**

**for(i=0;i<ArraySize;i++)**

**{**

**if(Array[i]<0)**

**{sum += pow(Array[i],3); INTER++;}**

**else if (Array[i]>0 && lowest<0)**

**{lowest = Array[i]; INTER++;}**

**else if(Array[i]>0 && Array[i]<lowest)**

**{lowest = Array[i]; INTER++;}**

**INTER+=3;**

**}**

**const auto TimeEnd = std::chrono::high\_resolution\_clock::now();**

**cout << "\n\nAnswer:\n Sum = " << sum << "\n least number = " << lowest << "\n Iteration: " << INTER << "\n Time spend: " << chrono::duration<float>(TimeEnd - TimeStart).count() << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void HowMany()**

**{**

**int i,j, NofNeganive=0, NofPositiv=0, INTER=0;**

**cout << "Double Array: \n";**

**for(i=0;i<DoubleArraySize\_Column;i++)**

**{**

**for(j=0;j<DoubleArraySize\_raw;j++)**

**printf("%4.i", DoubleArray[i][j]);**

**cout << endl;**

**}**

**const auto TimeStart = std::chrono::high\_resolution\_clock::now();**

**for(i=0;i<DoubleArraySize\_Column;i++)**

**for(j=0;j<DoubleArraySize\_raw;j++)**

**{**

**if(DoubleArray[i][j] >= 0)**

**{**

**INTER++;**

**NofPositiv++;**

**}**

**else**

**{**

**INTER++;**

**NofNeganive++;**

**}**

**INTER++;**

**}**

**const auto TimeEnd = std::chrono::high\_resolution\_clock::now();**

**cout << "\nAnswer:\n NumberOfPositiv = "<< NofPositiv << "\n NumberOfNegative = " << NofNeganive << "\n Iteration: " << INTER << "\n Time spend: " << chrono::duration<float>(TimeEnd - TimeStart).count() << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**