МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСАФАКУЛЬТЕТ МАТЕМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙКАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ І ТЕОРІЇ СИСТЕМ УПРАВЛІННЯЗВІТЗ НАВЧАЛЬНОЇ (КОМП’ЮТЕРНОЇ) ПРАКТИКИ

ЗВІТ

З НАВЧАЛЬНОЇ (КОМП’ЮТЕРНОЇ) ПРАКТИКИ

**Виконав:**

студент 1 курсу

денної форми навчання

спеціальності 122 Комп’ютерні науки

**ОП**: «Сучасні інформаційні технології

та програмування»

Пустовойтенко Владислав Вікторович

**Керівник практики**:

Смоктій К. В., старший викладач

**Оцінка**:\_\_\_\_\_\_\_\_\_\_ (балів)

\_\_\_\_\_\_\_\_\_\_ (ECTS)

\_\_\_\_\_\_\_\_\_\_ (національна)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_р.

Вінниця –2019

**ОФОРМЛЕННЯ ЗВІТУ З ПРАКТИКИ**

**Завдання №1**: Дано координати (як цілі від 1 до 8) двох полів шахової дошки. Визначити, чи може тура за один хід перейти з одного із цих полів на інше.

**Алгоритм:** Тура ходить лише по горизонталі та вертикалі, отже потрібно щоб 1 координата цілі співпадала з відповідною початковою

**Лістинг програми.**

**#include <stdio.h>**

**#include <conio.h>**

**#include <stdlib.h>**

**#include <windows.h>**

**int main()**

**{**

**SetConsoleCP(1251);**

**SetConsoleOutputCP(1251);**

**int x1, x2, y1, y2;**

**while(666)**

**{**

**printf("Введiть початкову позицію тури:\n");**

**scanf("%i%i",&x1,&y1);**

**if(x1 > 8 || x1 < 1 || y1>8 || y1 <1 )**

**printf("Введiть координати y вiд 1 до 8.Спробуйте ще раз!\n\n");**

**else**

**break;**

**}**

**while(666)**

**{**

**printf("Введiть точку в яку хочете перемістити туру:\n");**

**scanf("%i%i",&x2,&y2);**

**if(x2 > 8 || x2 < 1 || y2>8 || y2 < 1 )**

**printf("Введiть координати y вiд 1 до 8.Спробуйте ще раз!\n\n");**

**else**

**break;**

**}**

**if(x1 == x2 && y1 == y2)**

**printf("Ви ввели стартову позицію.");**

**else if(x1 == x2 || y1 == y2)**

**{**

**printf("Перехід можливий.");**

**return 0;**

**}**

**else**

**{**

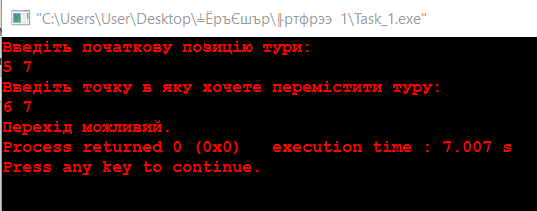
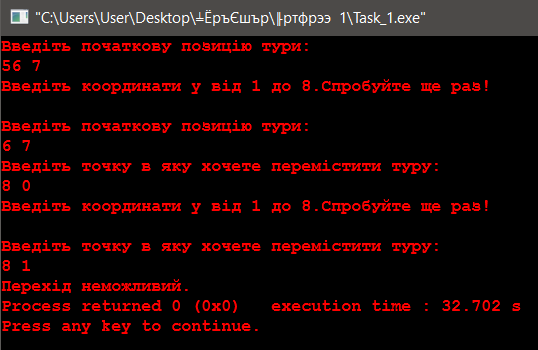
**printf("Перехід неможливий.");**

**return 0;**

**}**

**getch();**

**}**



**Рисунок 1 -Результат роботи програми**

**Завдання №2**: Дане натуральне число N. Отримати всі Піфагорові трійки натуральних чисел, кожне з яких не перевищує N, тобто всі такі трійки натуральних чисел a,b,c, що a2+b=c2(a≤b ≤ c≤N).

**Алгоритм**: Створюємо три вкладені цикли , де в останьому лежить умова a2+b=c2

**Лістинг програми.**

**#include <stdio.h>**

**#include <conio.h>**

**#include <stdlib.h>**

**#include <windows.h>**

**int main()**

**{**

**SetConsoleCP(1251);**

**SetConsoleOutputCP(1251);**

**int N, i, a, b, c;**

**printf("Введіть натуральне число N:\n");**

**while(666)**

**{**

**scanf("%i",&N);**

**if(N<1)**

**printf("Число N має бути > 0.Спробуйте ще раз!\n");**

**else**

**break;**

**}**

**for(c=1; c<=N; c++)**

**{**

**for(b=1; b<=c; b++)**

**{**

**for(a=1; a<=b; a++)**

**{**

**if(c\*c==a\*a+b\*b)**

**{**

**printf("%3i^2 + %3i^2 = %3i^2; Перевірка: %5i + %5i = %5i; \n",a, b, c,a\*a,b\*b,c\*c);**

**}**

**}**

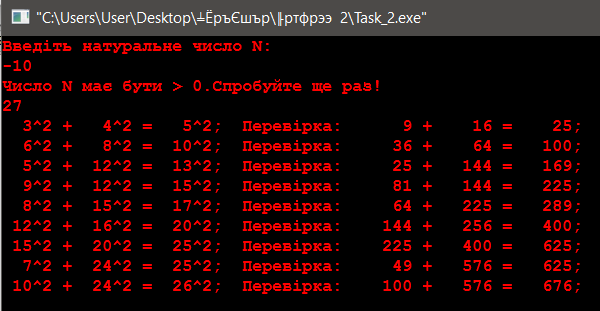
**}**

**}**

**getch();**

**return 0;**

**}**



**Рисунок 2 -Результат роботи програми**

**Завдання №2**: Для роботи з грошима у форматі: гривні, копійкистворити структуру та функції, що дозволяють: a)вводити грошову суму з клавіатури (файлу); b)виводити грошову суму на екран (файл); c)знаходити суму/різницю двох грошових сум; d)переведення грошової суми у копійки та навпаки. Створити приклад для демонстрації усіх функціональних можливостей.

**Алгоритм**: Створюємо клас Groshі, де оголошуєму 2 змінні grn i cop. Написати 4 окремі функції, аргументами яких є суми типу Groshi, для вводу\виводу з клавіатури\файлу певних сум грошей в форматі гривні копійки. Далі створюємо 2 функції-оператори для додавання і віднімання введених 2х сум, для коректного виводу на екран застосовуємо оператор умови, коли cop>99, cop -100, grn+1, якщо ж cop<0,то cop+100, grn -1. Далі пишемо 2 функції для переводу всіх грошових сум в копійки і назад.В копійки: grn\*100+cop, якщо grn<0, grn\*100-cop , результат записуємо в змінну sum. З копійок: grn=sum/100,cop=sum%100.Оголошуємо всі функції в мейні івикликаємо за допомогою swich(),case.Всі некоректні введення передбачені.

**Лістинг програми.**

(файл з класом Groshi)

#ifndef GROSHI\_H\_INCLUDED

#define GROSHI\_H\_INCLUDED

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

using namespace std;

class Groshi

{

private:

int grn,cop;

public:

void InputFromKB()

{

while(666)

{

cin>>grn>>cop;

if(cop < 0 || cop >99 || grn <0)

{

cout<<"Cop must be from 0 to 99, Grn only positive!!!\nInput grn and cop again!\n";

}

else

break;

}

}

void InputFromFile()

{

ifstream fin("Groshi.txt");

fin>>grn>>cop;

InputFromKB();

fin.close();

}

void OutputToScreen()

{

cout<<grn<<","<<cop<<endl;

}

void OutputToFile()

{

ofstream fout("Groshi.txt");

fout<<grn<<" "<<cop<< endl;

cout<<grn<<","<<cop<<endl;

fout.close();

}

Groshi operator + (Groshi sum)

{

Groshi result;

result.grn=grn+sum.grn;

result.cop=cop+sum.cop;

if(result.cop >= 100)

{

result.cop-=100;

result.grn++;

}

return result;

}

Groshi operator - (Groshi sum)

{

Groshi result;

result.grn=grn-sum.grn;

result.cop=cop-sum.cop;

if(result.cop < 0)

{

result.cop+=100;

result.grn--;

}

return result;

}

void ToCop()

{

if(grn>= 0)

cout<<grn\*100+cop<<endl;

else

cout<<grn\*100-cop<<endl;

}

void FromCop()

{

int sum;

if(grn>= 0)

sum = grn\*100+cop;

else

sum = grn\*100-cop;

grn=sum/100;

cop=sum%100;

cout<<grn<<","<<cop<<endl;

}

};

#endif

(main.cpp)

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

#include <conio.h>

#include "Groshi.h"

using namespace std;

int main()

{

Groshi sum1,sum2;

while(666)

{

int menu;

cout<<"1)From Keyboard\n2)From File\n0)Exit\n";

cin>>menu;

if(menu==0)

break;

switch(menu)

{

case 1:

while(666)

{

cout<<"1)Input first sum from KB\n2)Input second sum from KB\n3)First sum + Second sum\n";

cout<<"4)First sum - Second sum\n5)To cop\n6)From cop\n0)Exit\n";

cin>>menu;

if(menu==0)

break;

switch(menu)

{

case 1:

sum1.InputFromKB();

sum1.OutputToScreen();

break;

case 2:

sum2.InputFromKB();

sum2.OutputToScreen();

break;

case 3:

(sum1+sum2).OutputToScreen();

break;

case 4:

(sum1-sum2).OutputToScreen();

break;

case 5:

sum1.ToCop();

sum2.ToCop();

(sum1+sum2).ToCop();

(sum1-sum2).ToCop();

break;

case 6:

sum1.FromCop();

sum2.FromCop();

(sum1+sum2).FromCop();

(sum1-sum2).FromCop();

break;

}

}

case 2:

while(666)

{

cout<<"1)Input first sum from File\n2)Input second sum from File\n3)First sum + Second sum\n";

cout<<"4)First sum - Second sum\n5)To cop\n6)From cop\n0)Exit\n";

cin>>menu;

if(menu==0)

break;

switch(menu)

{

case 1:

sum1.InputFromFile();

sum1.OutputToFile();

break;

case 2:

sum2.InputFromFile();

sum2.OutputToFile();

break;

case 3:

(sum1+sum2).OutputToScreen();

(sum1+sum2).OutputToFile();

break;

case 4:

(sum1-sum2).OutputToScreen();

(sum1-sum2).OutputToFile();

break;

case 5:

sum1.ToCop();

sum2.ToCop();

(sum1+sum2).ToCop();

(sum1-sum2).ToCop();

break;

case 6:

sum1.FromCop();

sum2.FromCop();

(sum1+sum2).FromCop();

(sum1-sum2).FromCop();

break;

}

}

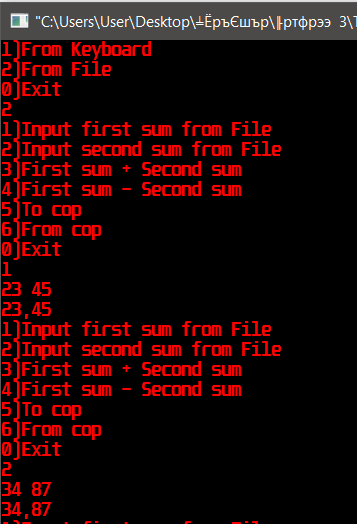
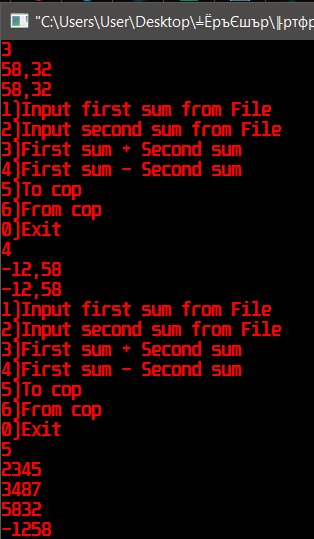
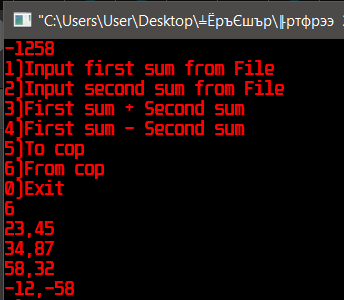
}

}

getch();

return 0;

}

Аналогічно для вводу з клавіатури

**Рисунки 3.1, 3.2,3.3 - Результати роботи програми**

**Завдання №2**: Створити програму, що зчитує елементи масивів x та y з файлів x.txt та y.txt відповідно, після чого опрацьовує масивixта формує масив z за відповідним алгоритмом. Функції для вводу/виводу масивів та їх опрацюваннянеобхіднореалізувати за допомогою шаблонів функцій.

a)збільшує парні елементи масивуixвдвічі;

b)zi =xi\*yi, де і від 1 до N;

**Алгоритм:** Оголошуємо масиви х і у в main. Створюємо шаблони функцій для зчитування масивів x і у, якщо розмір масиву у не рівний масиву х, просимо задати його ще раз.Створюємо шблон дя збільшення парних елементів масиву х вдвічі, парність елементів перевіряємо через остачу ділення на 2.Створюємо шаблон для створення і заповнення масиву z, множимо выдповыдны елементи масивыв х і у.

**Лістинг програми:**

**#include <iostream>**

**#include <conio.h>**

**#include <windows.h>**

**#include <fstream>**

**using namespace std;**

**template <typename T>**

**void scanfArrayX(T\* array, int size)**

**{**

**ifstream fin("x.txt");**

**if (!fin.is\_open())**

**cout << "File can not be open!\n";**

**else**

**{**

**cout<<"Array Xi:"<<endl;**

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

**{**

**fin>>array[i];**

**cout<<"Array["<< i << "] = " << array[i]<<endl;**

**}**

**}**

**cout<<endl;**

**fin.close();**

**}**

**template <typename T>**

**void doubleEvenElemens(T\* array,int size)**

**{**

**printf("Array Xi with doubled even elements:\n");**

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

**{**

**if(array[i]%2 == 0)**

**{**

**array[i] \*=2;**

**}**

**cout<<"Array["<< i << "] = " << array[i]<<endl;**

**}**

**}**

**template <typename T>**

**void scanfArrayY(T\* array,int size)**

**{**

**ifstream fin("y.txt");**

**if (!fin.is\_open())**

**cout << "File can not be open!\n";**

**else**

**{**

**cout<<"Array Yi:"<<endl;**

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

**{**

**fin>>array[i];**

**cout<<"Array["<< i << "] = " << array[i]<<endl;**

**}**

**}**

**cout<<endl;**

**fin.close();**

**}**

**template <typename T>**

**void printfArrayZ(T\* array1, T\* array2, int size)**

**{**

**T arrayZ[size] = {};**

**cout<<"ArrayZ (Array Xi \* Array Yi):\n";**

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

**{**

**arrayZ[i] = array1[i] \* array2[i];**

**cout<<"Array["<< i << "] = " << arrayZ[i]<<endl;**

**}**

**}**

**int main()**

**{**

**int sizeX, arrayX[25], sizeY, arrayY[25], arrayX2[25];**

**cout<< "Input size of ArrayX:\n";**

**cin>> sizeX;**

**scanfArrayX(arrayX,sizeX);**

**doubleEvenElemens(arrayX,sizeX);**

**while(666)**

**{**

**cout<< "Input size of ArrayY:\n";**

**cin>> sizeY;**

**if(sizeX == sizeY)**

**{**

**break;**

**}**

**else**

**{**

**cout<< "Size of ArrayY have to equal size of ArrayX!Try again.\n";**

**}**

**}**

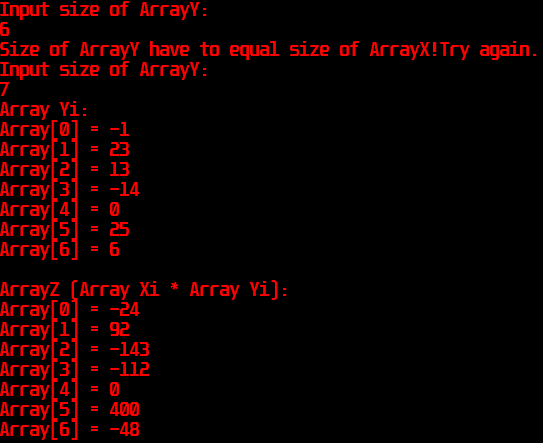
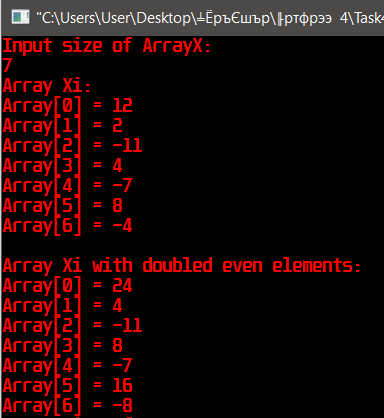
**scanfArrayY(arrayY,sizeY);**

**printfArrayZ(arrayX,arrayY,sizeY);**

**getch();**

**return 0;**

**}**



**Рисунки 4.1, 4.2 - Результати роботи програми**

**Завдання №5:** Дана послідовність, що містить від 2 до 25 слів, у кожному з яких є від 1 до 7 рядкових латинських букв, між сусідніми словами не менш одного пробілу, за останнім словом крапка. Надрукувати ті слова послідовності, які відмінні від останнього слова й задовольняють наступній властивості: а) перша буква слова входить у нього ще раз; б) слово збігається з початковим відрізком латинського алфавіту ( а, ab, abc і т.д.); в) довжина слова мінімальна; г) у слові немає букв, що повторюються. Послідовність символів читати з файлу

**Алгоритм:**

А)Берем першу букву, порівнюємо її з кожною наступною, якщо дорівнює виводимо все слово на екран, переходимо до наступного, якщо ж не дорівнюємо просто перехлдимо до наступного

Б)Робимо цикл з двома умовами і змінними. Перша переберає букви в слові до його кінця інша букви від а до z. Якщо букви співпадають дивимось наступну, якщо ні берем наступне слово, і так до кінця слова, якщо слово закінчилось без переходу на інше друкуємо його

В)Надаємо змінній min довжину першого слова, перевыряэмо з довжиною наступного, якщо воно менше записуэмо в min, в кінці виводимо слова з довжиною min

Г)Алгорим схожий до випадку а , тыльки для всых букв, якщо знаходиться співпадання відразу переходимо до наступного слова, якщо ні, то друкуємо.

**Лістинг програми:**

**#include <iostream>**

**#include <fstream>**

**#include <cstring>**

**#include <algorithm>**

**using namespace std;**

**void A(char\* words[7],int N){**

**cout<<"a) ";**

**string str;**

**for(int i=0;i<N;i++){**

**str=words[i];**

**for(int j=1;j<str.length();j++){**

**if(str[0]==str[j]){**

**cout<<str<<" ";**

**break;**

**}**

**}**

**}**

**cout<<"."<<endl;**

**}**

**void B(char\* words[7],int N){**

**cout<<"b) ";**

**string str;**

**for(int i=0;i<N;i++){**

**str=words[i];**

**char k='a';**

**int j=0;**

**string stralf="";**

**for(j=0,k='a';j<str.length();j++,k++){**

**stralf+=k;**

**}**

**if(str==stralf)**

**cout<<str<<" ";**

**}**

**cout<<"."<<endl;**

**}**

**void C(char\* words[7],int N){**

**cout<<"c) ";**

**string str;**

**int min;**

**for(int i=0;i<N;i++){**

**str=words[i];**

**if(str.length()<min){**

**min=str.length();**

**}**

**}**

**for(int i=0;i<N;i++){**

**str=words[i];**

**if(str.length()==min){**

**cout<<str<<"\t";**

**}**

**}**

**cout<<"."<<endl;**

**}**

**void D(char\* words[7],int N){**

**cout<<"d) ";**

**string str;**

**bool fl;**

**for(int i=0;i<N;i++){**

**str=words[i];**

**fl=true;**

**for(int j=0;j<str.length();j++){**

**for(int k=0;k<str.length();k++){**

**if(str[j]==str[k] && j!=k){**

**fl=false;**

**break;**

**}**

**}**

**}**

**if(fl)**

**cout<<str<<" ";**

**}**

**cout<<"."<<endl;**

**}**

**int main(){**

**ifstream in("data.txt");**

**string put;**

**getline(in,put);**

**char\* input=new char[put.length() + 1];**

**strcpy(input, put.c\_str());**

**in.close();**

**char\* words[7];**

**char\* point;**

**string str;**

**point=strtok(input," ,.");**

**int i=0;**

**while(point!=NULL){**

**words[i]=point;**

**point=strtok(NULL," ,.");**

**i++;**

**}**

**int N=i;**

**A(words,N);**

**B(words,N);**

**C(words,N);**

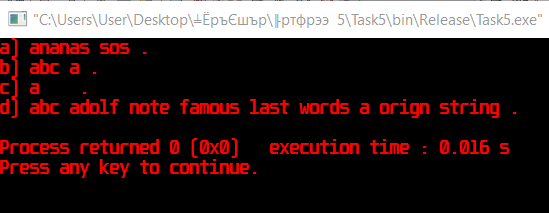
**D(words,N);**

**delete [] input;**

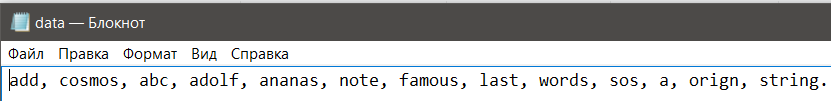
**delete [] point;**

**return 0;**

**}**



**Рисунок 4.1 – робота програми**



**Рисунок 4.2 – дані з файлу**