МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КОЛЕДЖ ЧЕРНІВЕЦЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

**ЗВІТ**

про навчальну практику

Студента II курсу, спеціальності

«Комп`ютерні науки»

204 групи

Гажука Богдана Георгійовича

Період практики з “\_\_\_\_Липень\_\_\_\_\_” 20 20\_ р. по “\_\_\_Серпень\_\_\_\_” 20\_20\_ р.

Керівник практики від коледжу:

викл. Ковдриш В.В. \_\_\_\_\_\_\_\_\_\_

/підпис/

Оцінка:\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

/підпис/

ЧЕРНІВЦІ, 2020

ТАБЛИЦЯ ОЦІНЮВАННЯ

|  |  |  |
| --- | --- | --- |
| Завдання | Оцінка за завдання | Підпис керівника практики |
| Завдання 1. |  |  |
| Завдання 2. |  |  |
| Завдання 3. |  |  |
| Завдання 4. |  |  |
| Завдання 5. |  |  |
| Завдання 6. |  |  |
| Завдання 7. |  |  |
| Завдання 8. |  |  |
| Завдання 9. |  |  |
| Завдання 10. |  |  |
| Завдання 11. |  |  |
| Завдання 12. |  |  |
| Завдання 13. |  |  |
| Завдання 14. |  |  |
| Оформлення звіту | |  |
| Захист звіту | |  |
| Підсумкова оцінка за навчальну практику | |  |

Зміст

[Завдання 1 5](#__RefHeading___Toc2200_2096317517)

[ПОСТАНОВКА ЗАДАЧІ 6](#__RefHeading___Toc2302_2096317517)

[Опис завдання 6](#__RefHeading___Toc3512_2096317517)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 6](#__RefHeading___Toc3404_2096317517)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 6](#__RefHeading___Toc3409_2096317517)

[Висновок 6](#__RefHeading___Toc3716_2096317517)

[Додаток(а) 6](#__RefHeading___Toc3718_2096317517)

[Додаток(б) 6](#__RefHeading___Toc3720_2096317517)

[Завдання 2 7](#__RefHeading___Toc3737_2096317517)

[ПОСТАНОВКА ЗАДАЧІ 8](#__RefHeading___Toc2302_20963175171)

[Опис завдання 8](#__RefHeading___Toc3512_20963175171)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 8](#__RefHeading___Toc3404_20963175171)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 8](#__RefHeading___Toc3409_20963175171)

[Висновок 8](#__RefHeading___Toc3716_20963175171)

[Додаток(а) 8](#__RefHeading___Toc3718_20963175171)

[Додаток(б) 8](#__RefHeading___Toc3720_20963175171)

[Завдання 3 9](#__RefHeading___Toc3760_2096317517)

[ПОСТАНОВКА ЗАДАЧІ 10](#__RefHeading___Toc2302_20963175172)

[Опис завдання 10](#__RefHeading___Toc3512_20963175172)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 10](#__RefHeading___Toc3404_20963175172)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 10](#__RefHeading___Toc3409_20963175172)

[Висновок 10](#__RefHeading___Toc3716_20963175172)

[Додаток(а) 10](#__RefHeading___Toc3718_20963175172)

[Додаток(б) 10](#__RefHeading___Toc3720_20963175172)

[Завдання 4 11](#__RefHeading___Toc3762_2096317517)

[ПОСТАНОВКА ЗАДАЧІ 12](#__RefHeading___Toc2302_20963175173)

[Опис завдання 12](#__RefHeading___Toc3512_20963175173)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 12](#__RefHeading___Toc3404_20963175173)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 12](#__RefHeading___Toc3409_20963175173)

[Висновок 12](#__RefHeading___Toc3716_20963175173)

[Додаток(а) 12](#__RefHeading___Toc3718_20963175173)

[Додаток(б) 12](#__RefHeading___Toc3720_20963175173)

[Завдання 5 13](#__RefHeading___Toc3764_2096317517)

[ПОСТАНОВКА ЗАДАЧІ 14](#__RefHeading___Toc2302_20963175174)

[Опис завдання 14](#__RefHeading___Toc3512_20963175174)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 15](#__RefHeading___Toc3404_20963175174)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 16](#__RefHeading___Toc3409_20963175174)

[Список підключених бібліотек: 16](#__RefHeading___Toc3772_2096317517)

[Список використаних класів: 16](#__RefHeading___Toc3774_2096317517)

[Список використаних функції: 16](#__RefHeading___Toc3776_2096317517)

[Висновок 17](#__RefHeading___Toc3716_20963175174)

[Додаток(а) 18](#__RefHeading___Toc3718_20963175174)

[main.cpp 18](#__RefHeading___Toc3778_2096317517)

[vidomist.cpp 18](#__RefHeading___Toc3780_2096317517)

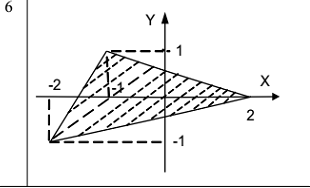
[vidomist.h 19](#__RefHeading___Toc3782_2096317517)

[Додаток(б) 29](#__RefHeading___Toc3720_20963175174)

# Завдання 1

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання: Створити алгоритм та написати програму для розвязку наступної задачі: перевірку попадання точки ( яка буде вводитись вручну із клавіатури) в заштриховану область.



## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завдання використовуються логічні оператори та оператори вибору.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати логічний оператор IF та оператор вибору SWITCH за допомогою перевірки умови.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float x, y;

printf("Vvedit koord tocky(x,y)\n");

scanf("%f%f",&x ,&y);

if((x>=-4)&&(x<=4)&&(y>=-4)&&(y<=4))

{if(( sqrt(pow(x1-x2,2)+(pow(y1-y2,2)+(pow(z1-z2,2)) >16))

printf("Tocka vhodyt v vydilenu obl");

else

printf("Tocka ne vhodyt v vydilenu obl");

}

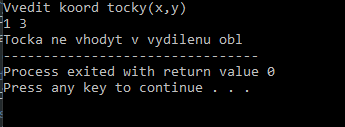
else

printf("Tocka ne vhodyt v vydilenu obl");

return 0;

}

## Додаток(б)



# Завдання 2

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### 6. Обчислити та вивести середн∙ арифметичне парних елементiв масиву N(11).

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних задачах я використовував одновимірні масиви.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <iostream>

## Висновок

За ці завдання я навчився використовувати одновимірні масиви для розвязку задачі з розробленням блок-схем та програм.

## Додаток(а)

#include <iostream>

using namespace std;

double sr(int a[], int n)

{ double s=0; int i, kol=0;

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

if (a[i]) { s+=a[i]; kol++; }

if (kol) s/=kol;

return s;

}

int main ()

{ setlocale(0,".1251");

int a[10], i;

cout<<" Введiть 10 цiлих чисел:\n";

for(i=0; i<10; i++) cin >> a[i];

cout<<"\n Середнє арифметичне "<<10<<" елементiв - " <<sr(a,10) <<endl;

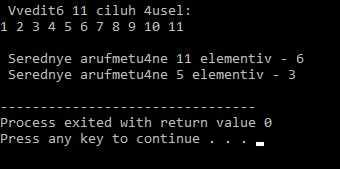
cout<<" Середнє арифметичне "<< 5 << " елементiв - " << sr(a,5) <<endl;

system ("pause>>void");

return 0;

}

## Додаток(б)



# Завдання 2(1)

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання: 6 Для заданого масиву Х(10) визначити, чи упорядкований він за спаданням, тобто для кожної пари сусідніх елементів виконується умова: Х(і)> Х(і+1), де і=1,2,…9. Вивести на екран заданий масив та відповідне повідомлення.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних задачах я використовував одновимірні масиви.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <iostream>

## Висновок

За ці завдання я навчився використовувати одновимірні масиви для розвязку задачі з розробленням блок-схем та програм.

## Додаток(а)

⦁ // сортування масиву методом вставки

⦁ int A[10];

⦁ int i,j; // допоміжні змінні - лічильники

⦁ int t; // допоміжна змінна

⦁

⦁ // ввід масиву A

⦁ // ...

⦁

⦁ // сортування

⦁ for (i=0; i<9; i++)

⦁ for (j=i; j>=0; j--)

⦁ if (A[j]<A[j+1])

⦁ {

⦁ // поміняти місцями A[j] та A[j+1]

⦁ t = A[j];

⦁ A[j] = A[j+1];

⦁ A[j+1] = t;

⦁ }

Додаток(б)

# Завдання 3

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання: 6. У заданiй матрицi A(6,4) знайти значення найбiльшого за модулем елемента матрицi, а також iндекси цього елемента. Вивести вихiдну матрицю, найбiльший за модулем елемент та його iндекси.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завданнях я використовував двовимірні масиви та роботу з матрицями.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <iostream>

## Висновок

В данній лабораторній пізнав нові можливості роботи двовимірних масивів та матриць. А також обробку рядків і стовпчиків матриці.

## Додаток(а)

#include <iostream>

using namespace std;

double MIN(double a[4][3], int& imin, int& jmin)

{

double min=a[0][0];

imin=jmin=0;

for(int i=0; i<4; i++)

for(int j=0; j<3; j++)

if(a[i][j]<min)

{ min=a[i][j];

imin=i; jmin=j;

}

imin++; jmin++;

return min;

}

int main ()

{

setlocale(0,".1251");

double a[4][3], m;

int i, j, ind\_i, ind\_j;

cout<<" Введiть матрицю з 4-х рядкiв i 3-х стовпцiв:"<<endl;

for(i=0; i<4; i++)

for(j=0; j<3; j++) cin>>a[i][j];

m = MIN(a, ind\_i, ind\_j);

cout<< "\n Мiнiмальний елемент " << m << "\n розміщено у "

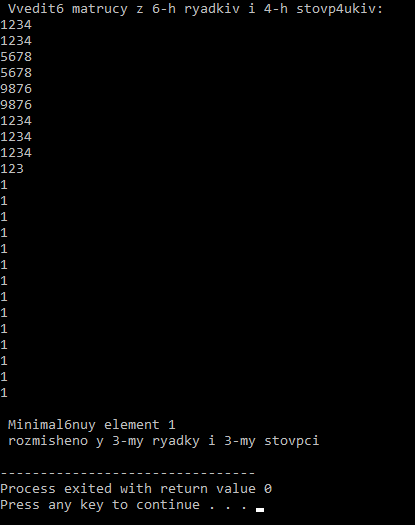
<< ind\_i <<"-му рядку i " << ind\_j <<"-му стовпці" << endl;

system ("pause>>void");

return 0;

}

## Додаток(б)



# Завдання 3(1)

## ПОСТАНОВКА ЗАДАЧІ

Опис завдання: 6. Сформувати масив з добуткiв вiд'ємних елементiв кожного стовпчика заданої матрицi F(5,5). В одновимiрному масивi знайти мiнiмальний елемент та його iндекс. Вивести вихiдний і створений масиви, мiнiмальний елемент та його iндекс.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завданнях я використовував двовимірні масиви та роботу з матрицями.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <iostream>

## Висновок

В данній лабораторній пізнав нові можливості роботи двовимірних масивів та матриць. А також обробку рядків і стовпчиків матриці.

Додаток(а)

#include <iostream>

using namespace std;

double MIN(double a[2][2], int& imin, int& jmin)

{

double min=a[0][0];

imin=jmin=0;

for(int i=0; i<2; i++)

for(int j=0; j<2; j++)

if(a[i][j]<min)

{ min=a[i][j];

imin=i; jmin=j;

}

imin++; jmin++;

return min;

}

int main ()

{

setlocale(0,".1251");

double a[2][2], m;

int i, j, ind\_i, ind\_j;

cout<<" Vvedit6 nmeobhidnuy masuv:"<<endl;

for(i=0; i<2; i++)

for(j=0; j<2; j++) cin>>a[i][j];

m = MIN(a, ind\_i, ind\_j);

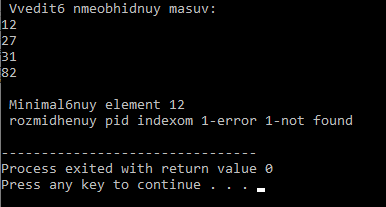
cout<< "\n Minimal6nuy element " << m << "\n rozmidhenuy pid indexom "

<< ind\_i <<"-error " << ind\_j <<"-not found " << endl;

return 0;

}

Додаток(б)



# Завдання 4

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання: ⦁ Дано дійсну матрицю А розмірності nxm. Знайти суму найбільших елементів її рядків.

### Вхідні дані: n = 3; m = 4; .

### Вихідні дані: sum = 24.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завданнях я використовував двовимірні масиви та працював з матрицями.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

## Висновок

В данній лабораторній пізнав нові можливості роботи двовимірних масивів та матриць. А також обробку рядків і стовпчиків матриці.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void Sum(int \*a, int n) { //функція знаходження сліду матриці

int i,j,l;

float max,min;

max=0;

for (i=1; i<=n; i++)

for (j=1; j<=n; j++)

if(\*(a+i\*n+j)>max)

max=(\*(a+i\*n+j));

printf("\nmax element --> %0.1f",max);

min=\*(a+1\*n + 1);

for (i=1; i<=n; i++)

for (j=1; j<=n; j++)

if(\*(a+i\*n+j)<min)

min=(\*(a+i\*n+j));

printf("\nmax element --> %0.1f",min);

}

int main (){

int \*c;

int i,j,n,l;

scanf("%d", &n);

c =(int \*)malloc(n\*n\*sizeof(int)); // виділяю пам'ять на масив

for(i=1; i<=n; i++)

{printf("\n");

for (j=1; j<=n; j++){

\*(c + i\*n + j)=rand()% 9 +1;

printf("%d ", \*(c + i\*n + j));}}

if (c == NULL){{

printf("Error");

return -1;

}}

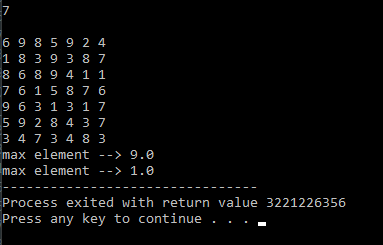
Sum(c,n);

free(c);

return 0;

}

## Додаток(б)



# Завдання 5

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Дано масив цілих чисел a[i], і = 1, … , n. Використовуючи вказівники, знайти номера тих елементів масиву, які більші свого лівого сусіда, і кількість таких елементів.

### Вхідні дані: n = 6; a[i] = {31, 77, 2, 0, 12, 11}.

### Вихідні дані: Номера елементів = 1, 4. Кількість = 2.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Вказівники

2)Динамічна память

3)Двовимірні масиви

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з динамічною памятю та двовимірними масивами.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,s,k=0;

printf("n=");

scanf("%d",&n);

int a[n],\*p;

p=a;

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

scanf("%d",&a[i]);

}

s=\*(p+1);

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

if(s!=a[i-1]){

k++;

}

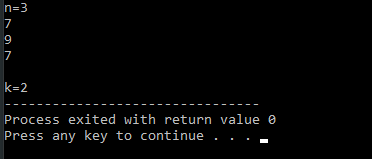
}

printf("\nk=%d",k);

return 0;

}

## Додаток(б)



# Завдання 6

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Нехай шість населених пунктів позначені номерами від 1 до 6 (величина k), а п’ять кандидатів – номерами від 1 до 5 (величина п). Кількість голосів, набраних кандидатами у кожному пункті визначається формулою akn=random(10і+50), де і - номер варіанта. (Функція random(п) описана у модулі stdlib.h. Перед використанням функції random(п) треба записати на початку програми функцію randomize() щоб під час виконання програми кожного разу отримувати різні випадкові числа.) Вивести на екран таблицю результатів голосування, де у рядках є дані з населених пунктів, а у стовпцях - дані щодо конкретних кандидатів. Визначити і вивести значення величин з додаткового завдання. Створити одновимірний масив з шуканими даними.

### ⦁ Скільки виборців взяли участь у голосуванні у кожному населеному пункті?

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Вказівники

2)Динамічна память

3)Двовимірні масиви

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <iostream>

#include <stdlib.h>

#include <locale.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з динамічною памятю та двовимірними масивами.

## Додаток(а)

#include <iostream>

#include <stdlib.h>

#include <locale.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "UKRAINE");

int n, k, max = 0, min = 0, a, b;

int d[10][10];

for (int i = 1; i <= 6; i++)

{

for (int f = 1; f <= 5; f++)

{

d[i][f] = rand() % 210;

cout << d[i][f] << "\t";

if (f == 1) {

max = d[i][f];

min = d[i][f];

}

if (i == 1) {

if (max > d[i][f]) {

max = d[i][f];

a = f;

}

if (min < d[i][f]) {

min = d[i][f];

b = f;

}

}

}

cout << endl;

}

cout << "Kandudat, yakuy nabrav min k-t6 baliv v pershomy misty- " << b << endl;

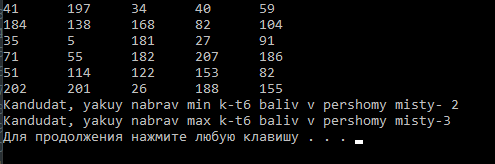
cout << "Kandudat, yakuy nabrav max k-t6 baliv v pershomy misty-" << a << endl;

system("pause");

return 0;

}

## Додаток(б)



# Завдання 6(1)

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Ввести прізвище, ім’я та по батькові як одне дане типу рядок. Визначити довжину рядка і кількість букв «а» у ньому. Виконати додатково завдання свого варіанта двома способами: а) розглядаючи рядок як масив символів;

### б) застосовуючи до рядка стандартні функції вивести ім’я та кількість букв у прізвищі.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Вказівники

2)Динамічна память

3)Двовимірні масиви

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <iostream>

#include <string.h>

#include <locale.h>

#include <string>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з динамічною памятю та двовимірними масивами.

## Додаток(а)

#include <iostream>

#include <string.h>

#include <locale.h>

#include <string>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Ukrainian");

string str\_pib;

char pib[50];

char bat[50];

int a = 0, n = 0, n2 = 0, x = 0;

bool c = false;

cout << "Pravul6no vvedit6 vashe PIB vuk-chu eng." << endl;

cin.getline(pib, 50);

for (int i = 0; i < 50; i++)

{

if (pib[i] == 'a' || pib[i] == 'A')

a++;

}

for (int i = 2; i < 50; i++)

{

if (isupper(pib[i]) == true)

{

c = true;

n = i;

};

if (c == true) break;

}

c = false;

for (int i = n + 1; i < 50;i++)

{

if (isupper(pib[i]) == true)

{

c = true;

n2 = i;

}

if (c == true) break;

}

for (int i = n2; i < 50; i++)

{

bat[x] = pib[i];

x++;

}

str\_pib = pib;

int pos1 = str\_pib.find(" ", 2);

int pos2 = str\_pib.find(" ", pos1 + 1);

string str\_name = str\_pib.substr(pos1 + 1, pos2 - pos1);

int n\_name = str\_name.length();

char ppp = '!';

str\_pib.push\_back(ppp);

int pos3 = str\_pib.find("!", pos2);

str\_pib.erase(pos3, 1);

string str\_last\_name = str\_pib.substr(pos2 + 1, pos3-pos2);

cout << endl;

cout << "Vashe PIB = " << pib << endl;

cout << "Dovzuna ryadka = " << strlen(pib) << endl;

cout << "Kil6kist6 bykv A/a = " << a << endl;

cout << endl;

cout << "Ind-ne zavdannya, vukonane rozglyadayu4u \n stroku, yak masuvy sumvoliv:" << endl;

cout << "Ki-t6 sumvoliv v vashomy imeni = " << ((n2 - n) - 1) << endl;

cout << "Vashe pobat6kovi = " << bat << endl;

cout << endl;

cout << "In-ne zavdannya, vukonani z dopomogoyu vuk-nnya \n da ryadka standartnuh fynkciy:" << endl;

cout << "Kil6kist6 sumvoliv v vashomy imeni = " << n\_name - 1 << endl;

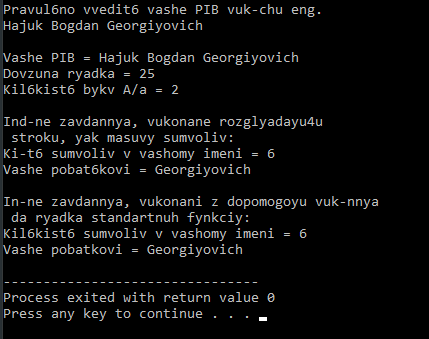
cout << "Vashe pobatkovi = " << str\_last\_name << endl;

system("pause");

return 0;

}

## Додаток(б)



# Завдання 7

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Утворити і вивести масив y з елементами yk=fi+1(k) , де k=1,12, і- номер варіанта. Для розрахунку yk створити функцію користувача. Скласти ще одну функцію користувача для пошуку даних у цьому масиві. Критерії пошуку взяти відповідно свого варіанту. Шукані дані вивести в головній функції.

### ⦁ Скільки було від’ємних значень? Визначити максимальне значення.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Вказівники

2)Динамічна память

3)Двовимірні масиви

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include<iostream>

#include<math.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з динамічною памятю та двовимірними масивами.

## Додаток(а)

#include<iostream>

#include<math.h>

using namespace std;

long double fank (long double k);

int main()

{

setlocale(LC\_ALL,"UKR");

cout<<"Praktuka N1. Laboratorna N7. Zavdannya 1."<<endl;

const int k=12;

long double y[k],sk,d;

int i;

for(i=1,y;i<=k;i++)

{

y[i]=fank(i);

cout<<"X["<<i<<"] "<<y[i]<<endl;

}

for(i=1,y[i],sk=0,d=1;i<=k;i++)

{

y[i]=fank(i);

if(y[i]>5 && y[i]<10)

{

sk+=(y[i]\*y[i]);

d\*=y[i];

}

}

cout<<"Syma kvadrativ "<<sk<<endl;

cout<<"Dobytok "<<d<<endl;

system("pause");

}

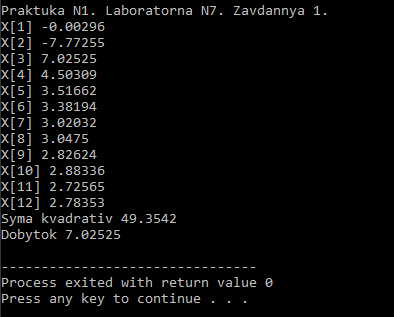
long double fank(long double k)

{

return ((6.3\*k-2\*sin(3\*k))/(1.5\*(3\*k-7))+pow((k-2),1/3));

}

## Додаток(б)



# Завдання 7(1)

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Утворити і вивести масив А, елементи якого описані формулою , m,n=1…4, i – номер варіанта. Скласти функцію користувача для пошуку даних у цьому масиві. Критерії пошуку взяти відповідно свого варіанту, якщо критерії пошуку не відповідають вашим даним вибрати свої значення для пошуку. Шукані дані вивести в головній функції.

### 6. Обчислити добуток квадратів тих елементів масиву, для яких виконується нерівність | am,n |<3 та їх кількість.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Вказівники

2)Динамічна память

3)Двовимірні масиви

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include"iostream"

#include"conio.h”

#include"locale.h"

#include"stdlib.h"

#include<time.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з динамічною памятю та двовимірними масивами.

## Додаток(а)

#include"iostream"

#include"conio.h"

#include"locale.h"

#include"stdlib.h"

#include<time.h>

long double fanky(long double x);

long double fanka(long double y, int n, int m);

using namespace std;

int main()

{

setlocale(LC\_ALL,"UKR");

int n,m;

long double y,c[4][4],x,a,sum;

cout<<"Praktuka N1. Zavdannya N2. Zadacha N2"<<endl;

cout<<"x= ";

cin>>x;

y=fanky(x);

cout<<"Nachal6y masuv"<<endl;

for(n=1,c;n<=4;n++)

{

cout<<n<<" ";

for(m=1,c;m<=4;m++)

{

a=fanka(y,n,m);

c[n][m]=a;

cout<<c[n][m]<<" ";

}

cout<<endl;

}

cout<<"Masuv pislya zaminu(yaksho vona ye)"<<endl;

for(n=1,c;n<=4;n++)

{

cout<<n<<" ";

for(m=1,c;m<=4;m++)

{

a=fanka(y,n,m);

c[n][m]=a;

if(c[n][m]==0)

c[n][m]=1;

cout<<c[n][m]<<" ";

}

cout<<endl;

}

cout<<"Etapu symyvannya(dlya sproshennya testuvannya)"<<endl;

for(n=1,sum=0,c;n<=4;n++)

{

for(m=4,c;m>=n;m--)

{

a=fanka(y,n,m);

c[n][m]=a;

sum+=a;

cout<<sum<<endl;

}

}

cout<<" "<<endl;

cout<<"Suma elementiv pid golocnoyu diagonallyu(i diagonali) v cilomy = "<<sum<<endl;

system("pause");

}

long double fanky(long double x)

{

return(7.4\*(x)+(5.7\*x-3+((4\*x)))/((x-1.2)\*(x+3.5)));

}

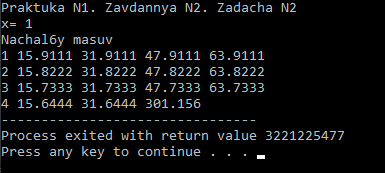
long double fanka(long double y, int m, int n)

{

return(y\*2\*m+16\*n);

}

## Додаток(б)



# Завдання 8

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### ⦁ Дано основи і висоти двох рівнобедрених трапецій. Знайти суми їх периметрів. Знаходження периметру трапеції реалізувати за допомогою функції.

### Вхідні дані: a1 = 4, b1 = 6, h1 = 5, a2 = 3, b2 = 7, h2 = 4.

### Вихідні дані: P1 = 20,2, P2 = 18,94.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Функції

2)Функції користувача

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

## Висновок

В ході розробки практичної роботи були вивченні та поглиблені знання з прорграмування на Сі та С++, а саме я більше дізнався про функції користувача, які відіграли ключову роль при створенні програми.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

float fync(float a1,float b1,float a2,float b2){

float ab;

ab=sqrt((a1-a2)\*(a1-a2)+(b1-b2)\*(b1-b2));

return ab;}

int main(int argc, char \*argv[]) {

float p,x1,x2,x3,y1,y2,y3;

printf("vvedit x i y tocky 1 ");

scanf("%f%f",&x1,&y1);

printf("vvedit x i y tocky 2 ");

scanf("%f%f",&x2,&y2);

printf("vvedit x i y tocky 3 ");

scanf("%f%f",&x3,&y3);

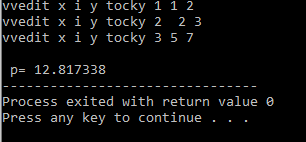
p=fync(x1,y1,x2,y2)+fync(x2,y2,x3,y3)+fync(x3,y3,x1,y1);

printf("\n p= %f",p);

return 0;

}

## Додаток(б)



# Завдання 9

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Дано масив дійсних чисел a[i], і = 1, … , n. Виділіть для нього необхідний обсяг динамічної пам'яті. Створити динамічний масив b[j], j = 1, … , m, в якому повинні розміщуватися елементи масиву a[i], відсортовані за зростанням. Вивільніть динамічну пам'ять.

### Вхідні дані: n = 6, a[i] = {31, -6, 2, 0, 33, 6}.

### Вихідні дані: b[j] = {-6, 0, 2, 6, 31, 33}.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Динамічні структури

2)Стеки

3)Черги

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з динамічними структурами та стеками,чергами.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

float main (void) {

int \*a, \*b, j,l,l1;

int i, n,m;

scanf("%d",&n);

scanf("%d",&m);

a=(int\*) malloc (n\* sizeof(int));

b=(int\*) malloc (m\* sizeof(int));

if (a == NULL){

{

printf("Error");

return -1;

}}

for(i=1; i<=n; i++){

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

printf("%d ",a[i]);

}

for(i=1; i<=n; i++){

if(a[i]%3 != 0)

l=l+1;}

printf("\n");

printf("K-st %d", l);

printf("\n");

l1=0;;

for(i=1; i<=m; i++){

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

printf("%d ",b[i]);

}

for(i=1; i<=m; i++){

if((a[i]%3)== 0)

l1=l1+1;}

printf("\n");

printf("K-st %d ", l1);

printf("\n");

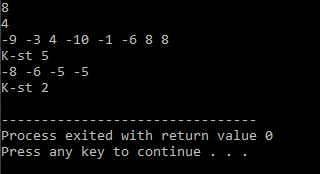
free(a);

free(b);

return 0;

}

## Додаток(б)



# Завдання 10

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### 6. Каталог передплатних видань містить:

### – поштовий індекс видання;

### – назва видання;

### – періодичність видання;

### – передплатну ціну на 12 місяців.

### Підрахувати: а) кількість видань, передплатна ціна яких перевищує задану величину х гривень на рік; б) назву найдешевшого видання та видань, ціна яких знаходиться в межах від a до b гривень

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Структури

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з структурами.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct {

char nomer [100];

char misto [100];

float vid;

float chas1;

float chas2;

float chasy;

float price ;

} poizd;

int main(int argc, char \*argv[]) {

int p;

int i,n=2;

float chas1, chas2;

poizd \* p;

float chas=0;

p = (poizd \*) calloc (n,sizeof(poizd));

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

printf("Nomer:");

fflush(stdin);

gets(p[i].nomer);

printf("Pruznachennya:");

fflush(stdin);

gets(p[i].misto);

printf("Vidstan");

fflush(stdin);

scanf("%f",&p[i].vid);

printf("Chas Vidpravlennya");

fflush(stdin);

scanf("%f",&p[i].chas1);

printf("Chas Prubytia");

fflush(stdin);

scanf("%f",&p[i].chas2);

printf("Chas y dorozi");

fflush(stdin);

scanf("%f",&p[i].chasy);

}

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

if (chasy<=chas)

chas=i;

if (misto=='Kyiv' && chas1>=18 && chas2<=10)

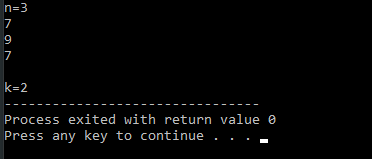
printf("Poizd z naitruv chas = # %c, %c", p[i].misto p[i].nomer );

}

printf("Poizd z naitruv chas = # %c, %c", p[chas].misto p[chas].nomer );

return 0; }

## Додаток(б)



# Завдання 11

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6

### Відома кількість очок, набраних кожної з 19 команд - учасниць першості по футболі. Перелік очок і команд даний у порядку зайнятих ними місць, тобто в порядку спадання кількості набраних очок (жодна пара команд-учасниць не набрала однакової кількості очок). З'ясувалося, що в перелік забули включити ще одну, двадцяту, команду.

### 

### Визначити, яке місце зайняла ця команда (відомо, що вона не стала чемпіоном і не зайняла останнє місце).

### 

### Вивести назви команд, що набрали менше очок, чим ця команда.

### 

### Одержати новий список команд (з урахуванням додаткової команди), у якому команди також розташовані в порядку убування кількості набраних ними очок.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

1)Структури

2)Списки

3)Стеки

4)Черги

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Список підключених бібліотек:

#include <stdio.h>

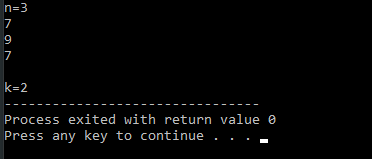
#include <stdlib.h>

## Висновок

Під час написання цієї роботи було збільшення знань взаємодії з списками,структурами та стеками,чергами.

## Додаток(а)

## Додаток(б)



# Завдання 12

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

Варіант 6

Створити однозв’язний лінійний список з 5 елементів, які мають два інформаційні поля: ім’я та зріст людини. При його створенні додавати елементи як з початку списку, так і в кінець. Вивести на екран кількість осіб та всю інформацію про людей, які вище 170 см. Видалити елемент списку, в якому записаний мінімальний ріст людини. Вивільнити динамічну пам’ять.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завдання використовуються стеки та черги.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати стеки та черги за допомогою перевірки умови.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Nomer{

int p;

struct Nomer \*l;

}R;

int Spusok (R \*\*head, int p1){

R \*s = (R\*)malloc(sizeof(R));

s -> p = p1;

s -> l = \*head;

\*head = s;}

int D(R \*head){

int d,i;

int min;

head=head->l;

head=head->l;

d=head->p;

d;}

int Rf(R \*head){

int r,i;

r=head->p;

r;}

int pr(R \*head, int min, int j){

int i;

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

printf("\n\t%d ",head->p);

(head) = (head) -> l;}

if(i==j)

printf("\n\t%d ",min);

}

int main(int argc, char \*argv[]) {

int i;

int p1;

R \*head = NULL;

int d,r, min,max, j;

printf("Vvedit 3 chisla: ");

for (i=0;i<3;i++)

{scanf("%d",&p1);

Spusok(&head,p1);}

d=D(head);

r=Rf(head);

for (i=0;i<3;i++){

if(head->p<min)

min=head->p;

if(head->p>max)

j=i;

if(i==0)

head->p=d;

if(i==2)

head->p=r;

printf("\n\t%d ",head->p);

(head) = (head) -> l;}

pr(head,min,j);

while (head )

{

(head) = (head) -> l;

free(head);

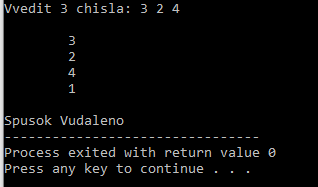
}

printf("\n\nSpusok Vudaleno");

return 0;

}

## Додаток(б)



# Завдання 13

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6.

### Розробити дві консольну програми для розв’язання задачі згідно варіанту. Перша програма вхідні дані читає із стандартного потоку введення, результат записується у стандартний потік виведення. Друга програма вхідні дані читаютає з файлу, результат записується у новий файл. Ім’я файлів передаються через командний рядок, або вводяться з консолі.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завдання використовуються текстові файли.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати текстові файли за допомогою мови програмування.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float x, y;

printf("Vvedit koord tocky(x,y)\n");

scanf("%f%f",&x ,&y);

if((x>=-4)&&(x<=4)&&(y>=-4)&&(y<=4))

{if((x\*x+y\*y>16))

printf("Tocka vhodyt v vydilenu obl");

else

printf("Tocka ne vhodyt v vydilenu obl");

}

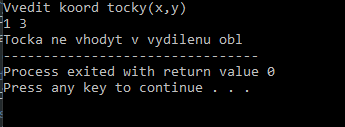
else

printf("Tocka ne vhodyt v vydilenu obl");

return 0;

}

## Додаток(б)



# Завдання 13(1)

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

Варіант 6.

Задано текст, слова в якому розділені пробілами і розділовими знаками. Розробити програму, яка вилучає в кожному слові цього тексту всі попередні входження останньої літери.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завдання використовуються текстові файли.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати текстові файли за допомогою мови програмування.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float x, y;

printf("Vvedit koord tocky(x,y)\n");

scanf("%f%f",&x ,&y);

if((x>=-4)&&(x<=4)&&(y>=-4)&&(y<=4))

{if((x\*x+y\*y>16))

printf("Tocka vhodyt v vydilenu obl");

else

printf("Tocka ne vhodyt v vydilenu obl");

}

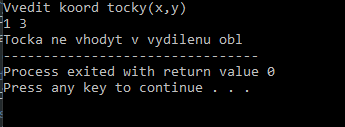
else

printf("Tocka ne vhodyt v vydilenu obl");

return 0;

}

## Додаток(б)



# Завдання 14

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

Варіант 6.

Дано рядок S. Зашифрувати уведений з клавіатури текст заміною вихідних символів на символи з кодом, більшим на дві одиниці.

Вхідні дані: S = "Programming and computing".

Вихідні дані: S = "Stritcookpi cpf eroswvkpi".

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завдання використовуються рядки.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати рядки за допомогою мови програмування.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float x, y;

printf("Vvedit koord tocky(x,y)\n");

scanf("%f%f",&x ,&y);

if((x>=-4)&&(x<=4)&&(y>=-4)&&(y<=4))

{if((x\*x+y\*y>16))

printf("Tocka vhodyt v vydilenu obl");

else

printf("Tocka ne vhodyt v vydilenu obl");

}

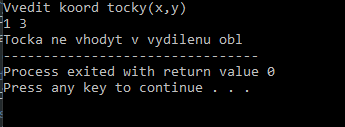
else

printf("Tocka ne vhodyt v vydilenu obl");

return 0;

}

## Додаток(б)



# Завдання 14(1)

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

### Варіант 6.

### Розробити програми мовою С для розв’язання задачі згідно варіанту. Вхідні дані читаються з файлу, результат записується у новий файл. Ім’я файлів передаються через командний рядок, або вводяться з консолі.

### Задано символьні рядки. Рядок складається з декількох слів (наборів символів), які розділяються не менше ніж одним символом пробілу (' '). Слова у рядку складаються з букв латинського алфавіту. Для кожного рядка знайти:

### ⦁ перше найменше за довжиною слово з парною кількістю символів;

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завдання використовуються рядки.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати рядки за допомогою мови програмування.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float x, y;

printf("Vvedit koord tocky(x,y)\n");

scanf("%f%f",&x ,&y);

if((x>=-4)&&(x<=4)&&(y>=-4)&&(y<=4))

{if((x\*x+y\*y>16))

printf("Tocka vhodyt v vydilenu obl");

else

printf("Tocka ne vhodyt v vydilenu obl");

}

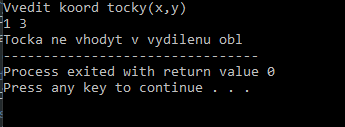
else

printf("Tocka ne vhodyt v vydilenu obl");

return 0;

}

## Додаток(б)



# Завдання 14(2)

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання:

Варіант 6.

Задано символьні рядки. Рядок, який складається із декількох послідовностей символів (наборів символів), які розділяються не менше ніж одним символом пробілу (' '). Для кожного рядка знайти послідовності символів у яких є:

слова, які починаються буквами: 'а', 'b', 'c', 'd' i 'e';

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В данних завданнях використовуються рядки.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include < stdio.h >

#include <stdlib.h>

## Висновок

Під час виконання данної лабораторної роботи я навчився правильно використовувати рядки за допомогою мови програмування.

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float x, y;

printf("Vvedit koord tocky(x,y)\n");

scanf("%f%f",&x ,&y);

if((x>=-4)&&(x<=4)&&(y>=-4)&&(y<=4))

{if((x\*x+y\*y>16))

printf("Tocka vhodyt v vydilenu obl");

else

printf("Tocka ne vhodyt v vydilenu obl");

}

else

printf("Tocka ne vhodyt v vydilenu obl");

return 0;

}

## Додаток(б)

