МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КОЛЕДЖ ЧЕРНІВЕЦЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

**ЗВІТ**

про навчальну практику

студент(а/ки) II курсу, спеціальності

«Комп`ютерні науки»

204(214) групи

*Рущак Юрій Сергійович*

Період практики з “\_\_\_\_\_\_\_\_\_\_” 20\_\_ р. по “\_\_\_\_\_\_\_\_\_” 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)

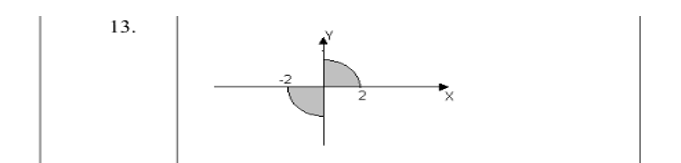
# **Завдання 1**

## **ПОСТАНОВКА ЗАДАЧІ**

### **Опис завдання**

Варіант 13

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



## 

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання задачі я додав дві змінні “x” і “y” за допомогою **int**, а також вивів їх на екран.

Моя програма працює таким чином, що якщо число, яке буде введене в консолі, буде поза меж заштрихованої області, то програма видасть **“no**”(так, як введене число знаходиться поза межами заштрихованого кола), і навпаки, якщо введене число знаходиться в заштрихованому колі, тоді програма видасть **“yes”**.

Щодо формули:

if (x\*x+y\*y<=4) , то вона саме обчислює, чи попаде число в заштриховане коло, чи ні, і так як річ іде про коло то цифра **4** означає його радіус.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

**Список підключених бібліотек**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

## **Висновок**

## Було розроблено алгоритм, який працює методом виключення(тобто якщо введене число знаходиться в заштрихованій області то воно проходить “yes”, а якщо ні, то аналогічно “no)”.

## **Додаток(а)**

**1.c**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

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

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

int x,y;

printf("vvedit x");

scanf("%d",&x);

printf("vvedit y");

scanf("%d",&y);

if (x\*x+y\*y<=4){

if (x\*y>0)

printf("yes");

else

printf("no"); }

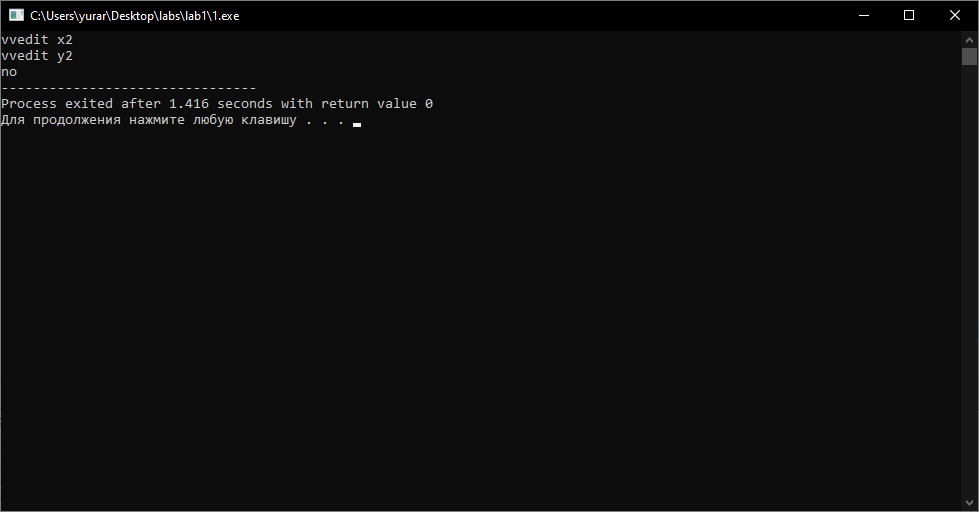
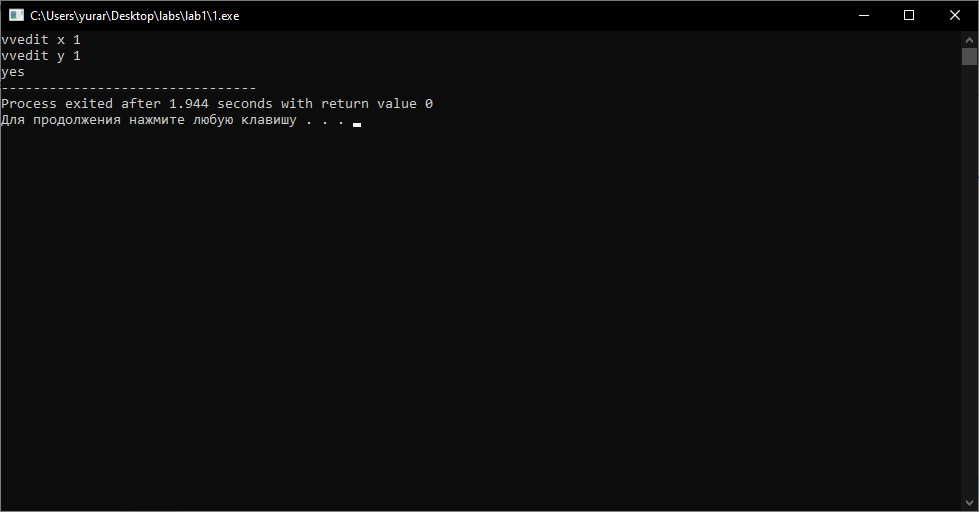
else

printf("no");

return 0;

}

## **Додаток(б)**



# **Завдання 2**

## **ПОСТАНОВКА ЗАДАЧІ**

### **Опис завдання**

Варіант 13

**1.Виконання обчислень.**

Знайти мінімальний та максимальний елементи заданого масиву К(9) і поміняти їх місцями. Вивести мінімальний і максимальний елементи, початковий та перетворений масиви.

**2.Упорядкування елементів масиву**

Вибрати з масиву D(12) вiд'ємнi елементи, вiдсортувати їх за спаданням та занести у масив Z. Вивести обидва масиви.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1.Щоб виконати цю задачу, впершу чергу я створив масив К, також в другій половині задачі я додав дві змінні **pos\_min** і **pos\_max**, за допомогою яких я впершу чергу служили для пошуку мінімального і максимального елементів масиву, а по друге завдяки цим змінним я облегшив собі завдання тим в зміні місцями знайдені елементи, тим саме зробив код більш компактним.

2.Щоб виконати цю задачу, мені потрібно було спочатку оголосити два масива та задати їм розмір 12. Порівняно з попередньою задачею, зміні **pos\_min** і **pos\_max** тут не потрібні, але лишилася змінна **buff**, яка служить в цій задачі буфером. Також завдяки циклу **for\if**(методом виключення я зробив так що числа сортуються за спаданням).

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>

#include <time.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

Buff – буфер обміну данних, та їх сортування.

## **Висновок**

В результаті , завдання 2-ї лабораторної роботи були майже схожими між собою тому я використовував ті ж самі змінні і функції. В першій задачі я додав дві змінні **pos\_min** і **pos\_max**, за допомогою яких я впершу чергу служили для пошуку мінімального і максимального елементів масиву, а по друге завдяки цим змінним я облегшив собі завдання тим в зміні місцями знайдені елементи, тим саме зробив код більш компактним. В другій задачі я лишив лиш buff , так як в нічого міняти місцями не треба було.

## **Додаток(а)**

**1.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int K[9];

int i,max,min,pos\_max,pos\_min,buff;

printf("Vhidnyi array\n");

for(i=0;i<9;i++){

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

printf("%d\t",K[i]);}

printf("\n");

max= K[0];

pos\_max=0;

for (i=0;i<9;i++)

{ if(max <K[i]){

max = K[i];

pos\_max= i;

}}

printf("max=%d",max);

printf("\n");

min= K[0];

pos\_min=0;

for (i=0;i<9;i++)

{ if(min >K[i])

{

min = K[i];

pos\_min= i;

}

}

printf("min=%d",min);

buff=K[pos\_min];

K[pos\_min]=K[pos\_max];

K[pos\_max]=buff;

printf("\nresult array\n");

for(i=0;i<9;i++){

printf("%d\t",K[i]);}

printf("\n");

return 0;

}

**2.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int D[12],Z[12];

int i,j;

j=0;

printf("Vhidnyi array\n");

for(i=0;i<12;i++){

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

printf("%d\t",D[i]);}

printf("\n");

for(i=0;i<12;i++){

if(D[i]<0) {

Z[j]=D[i];

j++;

}

}

printf("\n");

int k,buff;

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

for(i=0;i<j-k;i++)

{if (Z[i]<Z[i+1])

{

buff=Z[i];

Z[i]=Z[i+1];

Z[i+1]=buff;

}}

}

printf("\nresult array\n");

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

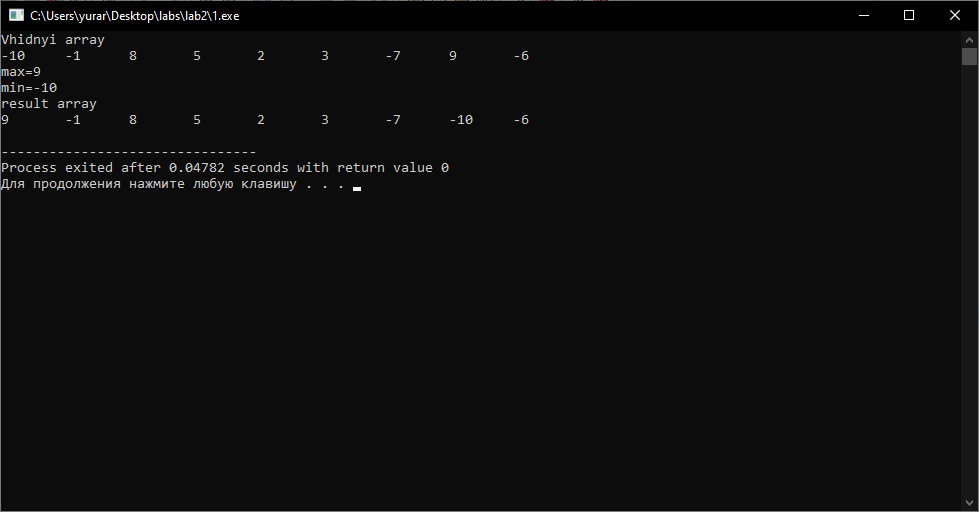
printf("%d\t",Z[i]);}

printf("\n");

return 0;

}

## **Додаток(б)**



# **Завдання 3**

## **ПОСТАНОВКА ЗАДАЧІ**

### **Опис завдання**

13 Варіант

1. В заданій матриці L(5,4) знайти середнє арифметичне найбільшого та найменшого елементів. Вивести вихідну матрицю, найбільший, найменший елементи та їх середнє арифметичне.

2. В заданій матриці К(5,3) визначити найбільший елемент серед парних додатних елементів, що розміщуються в стовпчиках з парними індексами. Вивести найбільший елемент i його індекси.

3. Із заданої матриці S(8,8) переписати елементи, розташовані вище головної діагоналі в одновимірний масив S1, а елементи, розташовані нижче головної діагоналі – в одновимірний масив S2. Вивести вихiдну матрицю і два сформованих масиви.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1.Для виконання цієї задачі було створено зміну **av** для пошуку середнього арифметичного для найбільшого і найменшого елементів. Аналогічно до другої лабораторної я зробив той же самий спосіб обчислення чисел L[i][j]=rand()%20-10.

2.Для виконання другої задачі, я скопіював попередню для економії часу і замість того щоб шукати середнє арифметичне я шукав парні числа серед додатніх. Було створено нові зміні **imax** і **jmax**, які являють собою максимальні елементи матриці масиву.

3. Для виконання цієї задачі було створенно два одновимірні масиви S1 і S2 по 32. В цій задачі я використовував зміну **av** , а також завдяки **for** ввів декілька формул для обчислення.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>

#include <time.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

Float - це властивість CSS для позиціонування елементів.

Av – середнє арифметичне.

Rand- генерує випадкові числа, повертає псевдовипадкове ціле число в діапазоні від 0 до RAND\_MAX .

## **Висновок**

Задачі по своїй структурі були майже ідентичні. Я використав ті ж самі зміні і функції.

Перша задача виводить вихідну матрицю, друга знаходила найбільший елемент серед парних додатних елементів та їх індекси, в третій задачі треба було з двовимірного масиву перенести елементи в два одновимерні масиви і в вивести вихiдну матрицю і два сформованих масиви.

## **Додаток(а)**

## **1.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int L[5][4];

int i,max,min,j;

float av;

printf("Vhidnyi array\n");

for(i=0;i<5;i++){

for(j=0;j<4;j++){

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

printf("%d\t",L[i][j]);}

printf("\n");}

printf("\n");

max= L[0][0];

for(i=0;i<5;i++){

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

{ if(max <L[i][j])

max = L[i][j];

}}

printf("max=%d",max);

printf("\n");

min= L[0][0];

for(i=0;i<5;i++){

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

{ if(min>L[i][j])

min = L[i][j];}}

printf("min=%d",min);

av=((max+min)/2.0);

printf("\naverage=%0.3f\t",av);

printf("\n");

return 0;

}

**2.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int K[5][3];

int i,max,min,j,imax,jmax;

float av;

printf("Vhidnyi array\n");

for(i=0;i<5;i++){

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

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

printf("%d\t",K[i][j]);}

printf("\n");}

printf("\n");

max= -999;

for(i=0;i<5;i++){

for(j=0;j<3;j=j+2)

{ if ((K[i][j]>0)&&(K[i][j]%2==0))

{

if(max<K[i][j])

{max = K[i][j];

imax=i;

jmax=j;

}}}}

printf("max=%d",max);

printf("\n");

printf("imax=%d",imax);

printf("\n");

printf("jmax=%d",jmax);

printf("\n");

return 0;

}

**3.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int S[8][8];

int i,S1[32],S2[32],j;

float av;

printf("Vhidnyi array\n");

for(i=0;i<8;i++){

for(j=0;j<8;j++){

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

printf("%d\t",S[i][j]);}

printf("\n");}

printf("\n");

int l=0;

printf("Vushe golovna diagonal\n");

for(i=0;i<8;i++){

for(j=i+1;j<8;j++)

{ S1[l]=S[i][j];

printf("%d\t",S1[l]);

l++;}}

printf("\n");

l=0;

printf("Nusahe golovna diagonal\n");

for(i=0;i<8;i++){

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

{ S2[l]=S[i][j];

printf("%d\t",S2[l]);

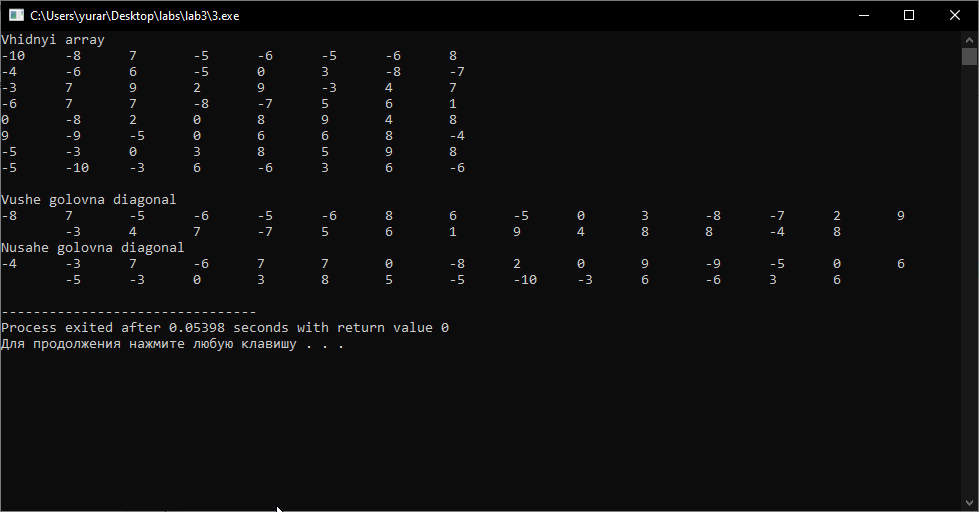
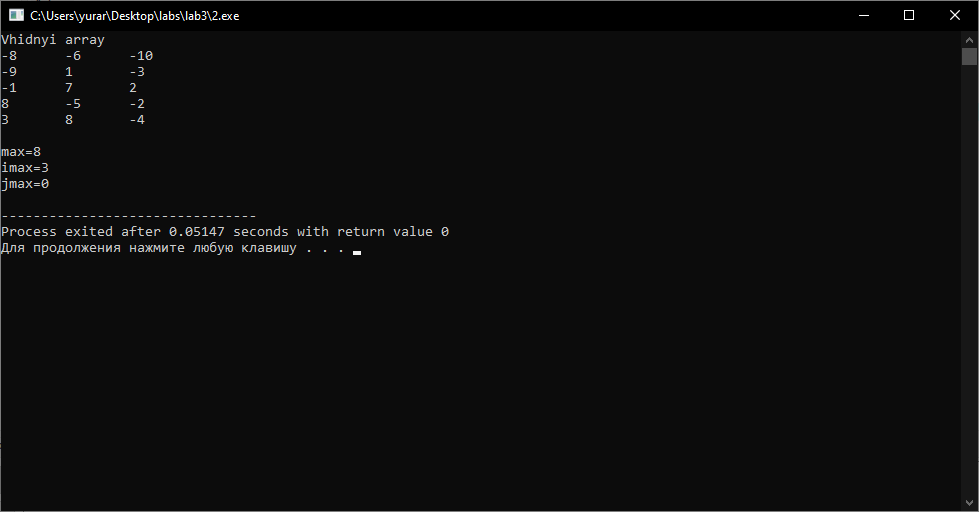
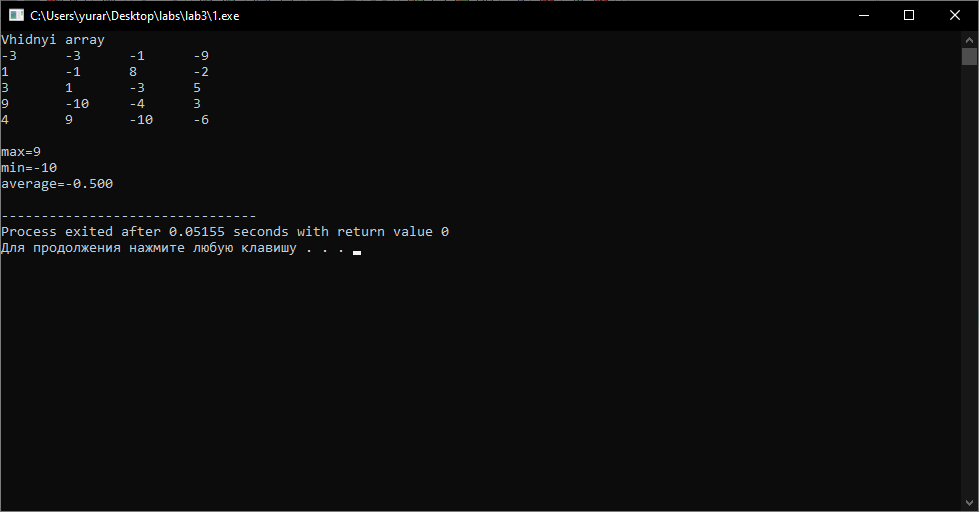
l++;}}

printf("\n");

return 0;

}

## **Додаток(б)**



# **Завдання 4**

## **ПОСТАНОВКА ЗАДАЧІ**

### **Опис завдання**

Варіант 13

1.Із заданої матриці S(8,8) переписати елементи, розташовані вище головної діагоналі в одновимірний масив S1, а елементи, розташовані нижче головної діагоналі – в одновимірний масив S2. Вивести вихiдну матрицю і два сформованих масиви.

2.Дано дійсну матрицю *А* розмірності *n*x*m*. Вивести на екран вектор *b*, елементи якого є мінімальними елементами кожного зі стовпців.

*Вхідні дані:* *n =* 3; *m =* 4; .

*Вихідні дані:* .

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1. Для виконання цієї задачі було створенно два одновимірні масиви S1 і S2 по 32. В цій задачі я використовував зміну **av** , а також завдяки **for** ввів декілька формул для обчислення.

2.Для виконання цієї задачі я взяв за шаблон попередні задачі з вхідними масивами і переробив їх так, що з крім масивів і воно шукало кількість сповпчиків.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>

#include <time.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

Rand- генерує випадкові числа, повертає псевдовипадкове ціле число в діапазоні від 0 до RAND\_MAX .

## **Висновок**

Ця лабораторна робота далась мені легше інших, так як 1-ше завдання є аналогічно схожим на 3-тє завдання 3-ї лабораторної, тому не буду довго на ній зупинятися. Щодо другої задачі то там треба було створити матрицю та вивести її з вектором на екран.

## **Додаток(а)**

**1.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int S[8][8];

int i,S1[32],S2[32],j;

float av;

printf("Vhidnyi array\n");

for(i=0;i<8;i++){

for(j=0;j<8;j++){

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

printf("%d\t",S[i][j]);}

printf("\n");}

printf("\n");

int l=0;

printf("Vushe golovna diagonal\n");

for(i=0;i<8;i++){

for(j=i+1;j<8;j++)

{ S1[l]=S[i][j];

printf("%d\t",S1[l]);

l++;}}

printf("\n");

l=0;

printf("Nusahe golovna diagonal\n");

for(i=0;i<8;i++){

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

{ S2[l]=S[i][j];

printf("%d\t",S2[l]);

l++;}}

printf("\n");

return 0;

}

**2.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int A[50][50];

int i,b[50],n,m,j;

printf("Kilkist riadkiv: ");

scanf("%d",&n);

printf("Kilkist stovpstiv: ");

scanf("%d",&m);

printf("Vhidnyi array\n");

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

for(j=0;j<m;j++){

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

printf("%d\t",A[i][j]);}

printf("\n");}

printf("\n");

int min;

for(j=0;j<m;j++){

min=A[0][j];

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

{ if(min>A[i][j])

{min=A[i][j];}

} b[j]=min;}

printf("\n");

printf("Rezultat\n");

for(j=0;j<m;j++)

{ printf("%d\t",b[j]);

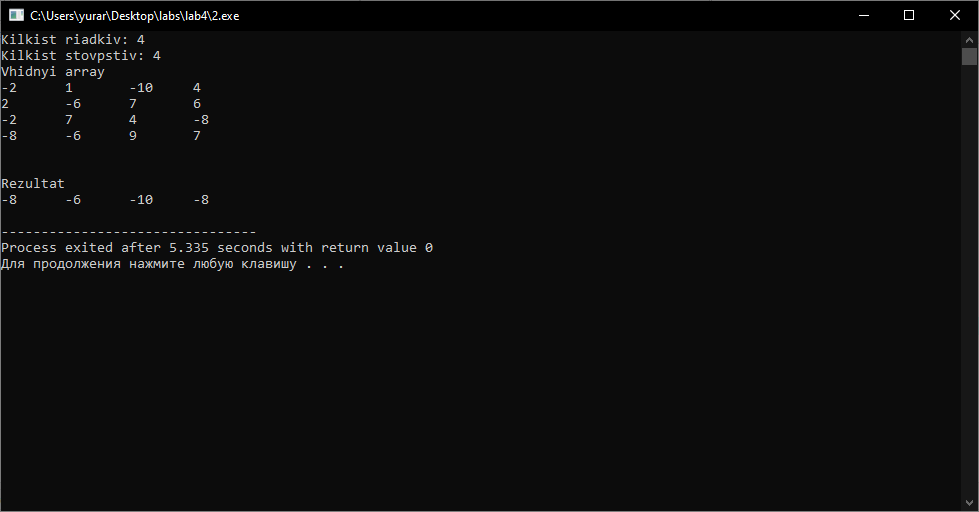
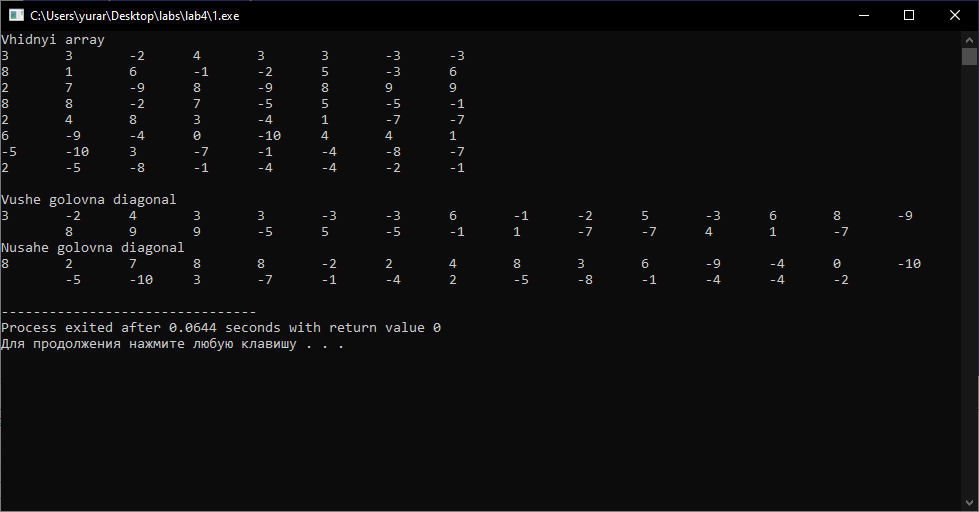
}

printf("\n");

return 0;

}

## **Додаток(б)**

****

# **Завдання 5**

### **Опис завдання**

Варіант 13

Дано матрицю *А* розмірності *nxп*. Використовуючи вказівники, знайти мінімальний елементів масиву, які знаходяться вище головної діагоналі та суму всіх елементів бічної діагоналі.

*Вхідні дані:* *n = 3*; .

*Вихідні дані:* *Мінімальний елемент =* -6.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання цієї задачі я створив матрицю А, використовуючи вказівники **malloc** і **sizeof** я зміг знайти мінімальний елемент масива А, а також завдяки зміній **sum** я зміг знайти суму всіх елементів бічної діагоналі.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>

#include <time.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

Rand- генерує випадкові числа, повертає псевдовипадкове ціле число в діапазоні від 0 до RAND\_MAX .

Malloc - повертає адресу на перший байт області пам'яті розміром size байт

sizeof - це унарний оператор, який повертає довжину в байтах змінної або типу, поміщених в дужки.

## **Висновок**

Порівняно з попередніми задачами, ця задача була однією з перших, де я застосовував більше вказівників, і саме з цієї лабораторної на мою думки почалися важчі завдання. Я виконав задачу через malloc, тому що мені так легше(хіба міг зробити по іншому).

## **Додаток(а)**

**1.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int A[50][50];

int i,b[50],n,m,j,\*min,\*sum;

min=(int\*)malloc(sizeof(int));

sum=(int\*)malloc(sizeof(int));

printf("Kilkist riadkiv i stovpstiv : ");

scanf("%d",&n);

printf("Vhidnyi array\n");

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

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

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

printf("%d\t",A[i][j]);}

printf("\n");}

printf("\n");

min=&A[0][0];

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

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

if(\*min>A[i][j]){min=&A[i][j];}

}}

printf("MINIMALNE=%d\t",\*min);

printf("\n");

\*sum=0;

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

\*sum+=A[i][n-i-1];

}

printf("SUMA=%d\t",\*sum);

return 0;

}

## **Додаток(б)**



# **Завдання 6**

### **Опис завдання**

Варіант 13

**Завдання:**

Для наведених нижче задач

1. зробити математичну постановку;
2. скласти алгоритм програми;
3. скласти програму для обчислення;
4. проаналізувати результати роботи програми;
5. оформити протокол.

**Задача 1. Двовимірні масиви. Задача про вибори.**

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.

Хто з кандидатів набрав найбільше голосів у другому і третьому населених пунктах?

**Задача 2. Дані типу рядки.**

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;

Вивести ім’я у зворотному порядкую.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1.Для виконання 1-ї задачі я вирішив ввести оператор malloc для роботи з динамічними масивами і пам’ятю. Суть задачі полягала у тому, щоб вивело на екран кандидата за якого було віддану найбільшу кількість голосів і в якому наслеленому пункті їх було найбільше.

2.В 2-ї задачі треба було зробити так, щоб програма вивела на екран прізвище, ім’я, по батькові як одне слово.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <time.h>
5. #include <string.h>
6. **Список використаних функції:**
7. strlen - функція стандартної бібліотеки, для повернення довжини нуль-терминированной рядки без символу закінчення рядка (нуля).
8. gets - функція, що входить в Стандартну бібліотеку мови Сі, оголошується в заголовки stdio. h, яка зчитує рядок стандартного вводу і поміщає її в буфер, створений викликає функцією.
9. char – проста зміна, яка працює зі стовпчиками.
10. break - оператор завершує виконання найближчого включає циклу або умовного оператора, в якому він відображається.
11. Malloc - повертає адресу на перший байт області пам'яті розміром size байт
12. sizeof - це унарний оператор, який повертає довжину в байтах змінної або типу, поміщених в дужки.
13. Rand- генерує випадкові числа, повертає псевдовипадкове ціле число в діапазоні від 0 до RAND\_MAX .
14. srand(time(NULL)); - встановлює в якості бази поточний час.
15. **Висновок**

В цій лабораторній роботі було застосовано велику кількість операторів та змінних, в цей раз завдання не були сильно схожі між собою, тому прийшлося довго розбиратися з ними. Також це одна з перших лабораторних робіт, яка в мене застосовує бібліотеку **#include <string.h>**.

## **Додаток(а)**

**1.c**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int \*\*A;

int i,b[50],n,m,j,\*min,\*sum,\*max,\*l;

max=(int\*)malloc(sizeof(int));

l=(int\*)malloc(sizeof(int));

A=(int\*\*)malloc(5\*sizeof(int\*));

for(i=0;i<6;i++)

A[i]=(int\*)malloc(6\*sizeof(int));

printf("Golosuvannia\n");

for(i=0;i<6;i++){

for(j=0;j<5;j++){

A[i][j]=rand()%180;

printf("%d\t",A[i][j]);}

printf("\n");}

printf("\n");

max=&A[1][0];

\*l=1;

for(j=0;j<5;j++){

if(\*max<A[1][j]){max=&A[1][j]; \*l=(j+1);}

}

printf("V 2 misti naybilse nabrav %d kandudat %d golosiv",\*l,\*max);

printf("\n");

max=&A[2][0];

\*l=1;

for(j=0;j<5;j++){

if(\*max<A[2][j]){max=&A[2][j]; \*l=(j+1);}

}

printf("V 3 misti naybilse nabrav %d kandudat %d golosiv",\*l,\*max);

return 0;

}

**2.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

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

char s[255];

int n, i;

printf("Enter text\n");

gets(s);

n=strlen(s);

int kil=0;

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

{

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

}

printf("Kilkist a: %d",kil);

int k1,k2;

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

{

if(s[i]==' ') {k1=i; break;}

}

for(i=n;i>k1;i--)

{

if(s[i]==' ') {k2=i; break;}

}

printf("\n");

printf("Name\n");

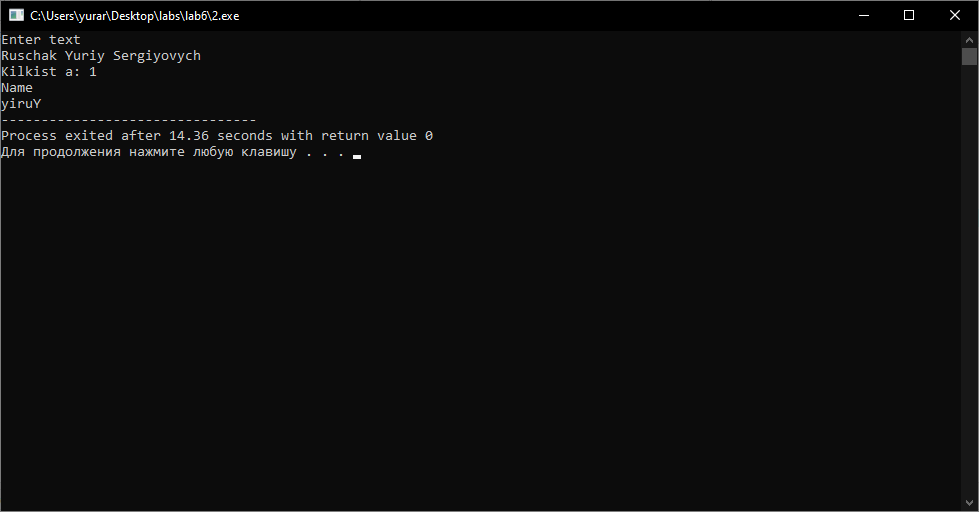
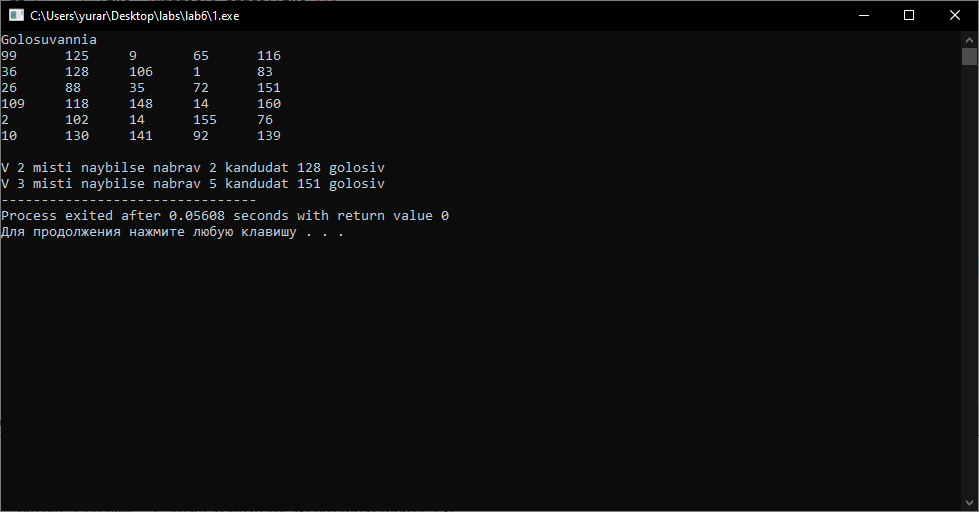
for(i=k2-1;i>k1;i--)

printf("%c",s[i]);

return 0;

}

## **Додаток(б)**



# **Завдання 7**

### **Опис завдання**

Варіант 13

Для наведених нижче задач

1. зробити математичну постановку;
2. скласти алгоритм програми;
3. скласти програму для обчислення;
4. проаналізувати результати роботи програми;
5. оформити протокол.

**Задача 1. Функції користувача.**

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.



Знайти індекс мінімального значення масиву. Після цього впорядкувати масив за спадання значень масиву.

**Задача 2. . Функції користувача.**

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.



Обчислити суму елементів масиву над головною діагоналлю.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1.Для виконання цієї задачі за застосував функцію **void.** В цій задачі треба було знайти індекс мінімального числа. Також в цій задачі я використав свою змінну buff, про яку я писав в попередніх лабораториних роботах.

2.Для виконання цієї задачі я взяв за основу першу задачу і трохи переробив її. Було створено дві нові змінні i\_idx(стопчик),j\_idx(рядок). Завдяки цим змінним можна знайти мінімальний індекс стовпчика і рядка.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <time.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

Rand- генерує випадкові числа, повертає псевдовипадкове ціле число в діапазоні від 0 до RAND\_MAX .

Void- універсальний" покажчик, який просто відповідає вказівником на будь-яку адресу в пам'яті комп'ютера, безвідносно типу даних, на які вказує цей покажчик.

## **Висновок**

Ця лабораторна була чимось схожа на попередні, там де були діагоналі, буфери обміну та ін.

Порівняно з попередньою лабораторною роботою ця в декілька разів легше в плані поставленої задачі. Дві задачі були майже ідентичні, правда в першій був одновимірний масив, а в другій двовимірний.

## **Додаток(а)**

**1.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

void func(int n,int \*a)

{ srand(time(NULL));

int i;

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

{a[i]=rand()%23-13;

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

}

}

void index(int n,int \*a)

{

int i, idx, min=a[0];

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

{

if(min<a[i]) { min=a[i]; idx=i; }

}

printf("\nIndex of minimal: %d",idx);

printf("\n");

int k,buff;

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

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

{if (a[i]<a[i+1])

{

buff=a[i];

a[i]=a[i+1];

a[i+1]=buff;

}}

}

printf("\nSort array\n");

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

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

printf("\n");

}

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

srand(time(NULL));

int n;

printf("Size of array\n");

scanf("%d",&n);

int a[n];

printf("Array:\n");

func(n,a);

index(n,a);

return 0;

}

**2.c**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

void func(int n,int m,int a[n][m])

{ srand(time(NULL));

int i,j;

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

{for(j=0;j<m;j++)

{a[i][j]=2\*m\*(rand()%13-13)+16\*n;

printf("%d\t",a[i][j]);}

printf("\n");}

}

void suma (int n, int m,int a[n][m])

{ int i,j;

int sum=0;

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

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

sum+=a[i][j];

}

printf("SUMA: %d\t",sum);

}

void index(int n,int m,int a[n][m])

{

int i,j, i\_idx,j\_idx, min=a[0][0];

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

{for (j=0;j<m;j++)

if(min>a[i][j]) { min=a[i][j]; i\_idx=i;j\_idx=j; }

}

printf("\nIndex of minimal: %d\t%d",i\_idx+1,j\_idx+1);

printf("\n");

}

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

srand(time(NULL));

int n,m;

printf("Size of matrix\n");

scanf("%d",&n);

scanf("%d",&m);

int a[n][m];

printf("Matrix:\n");

func(n,m,a);

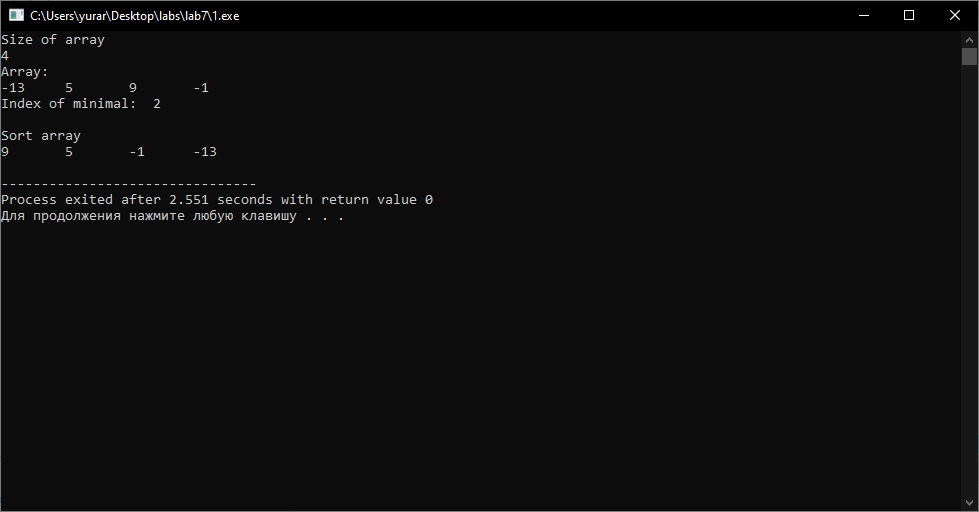
suma(n,m,a);

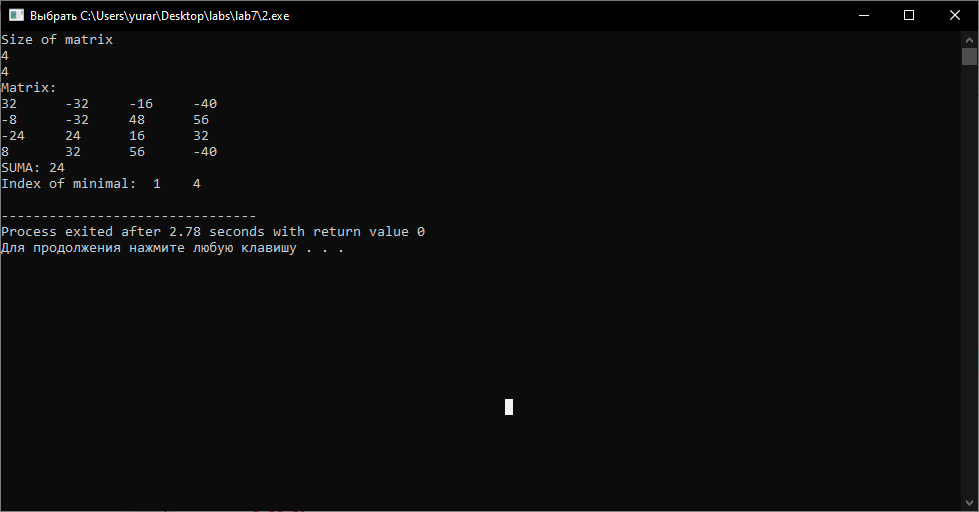
index(n,m,a);

return 0;

}

## **Додаток(б)**





# **Завдання 8**

### **Опис завдання**

Варіант 13

Дано дві прямі: *ax* + *b* = 0; *cx* + *d* = 0. Написати функцію *Paral* для визначення чи прямі паралельні, чи ні.

*Вхідні дані:* *a =* 2, *b =* 6, *c =* 4, *d =* 12*.*

*Вихідні дані:* *«Прямі є паралельними».*

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання цієї задачі, мені довелося підключити бібліотеку #include <math.h>, так як за умовою треба визначити чи паралельні прямі чи ні. Також завдляки if\else(методом виключення) можна дізнатися чи паралельні прямі чи ні.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <time.h>
5. #include <math.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

## **Висновок**

Це одна з перших задач практики де я використав нові бібліотеки для пошуку паралельних і не паралельних прямих було дано дві прямі: *ax* + *b* = 0; *cx* + *d* = 0.

Моя програма працює таким чином, що якщо введене число відповідає написаному циклу if\else то воно паралельне, а якщо ні, то буде писати, що число не паралеьне.

## **Додаток(а)**

**1.с**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

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

void Paral(int a,int b,int c,int d)

{ srand(time(NULL));

if ((1.0\*a/c)==(1.0\*b/d)){

printf("Paralel");

}

else

printf("Ne Paralel");

}

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

int a,b,c,d;

printf("a=\n");

scanf("%d",&a);

printf("b=\n");

scanf("%d",&b);

printf("c=\n");

scanf("%d",&c);

printf("d=\n");

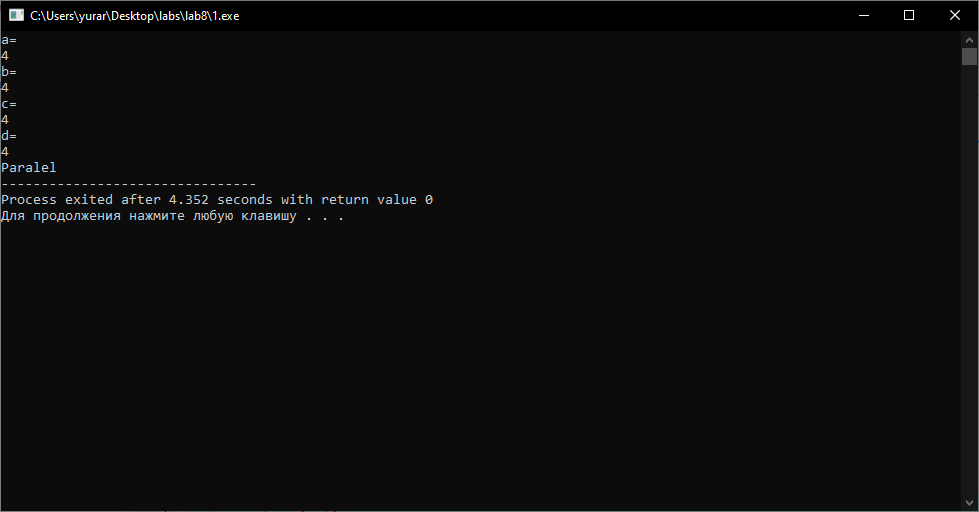
scanf("%d",&d);

Paral(a,b,c,d);

return 0;

}

**Додаток(б)**



# **Завдання 9**

### **Опис завдання**

Варіант 13

Дано цілочислову матрицю *А* розмірності *nxn*. Виділіть для неї необхідний обсяг динамічної пам'яті та знайдіть номер останнього з її рядків, що містить максимальну кількість однакових елементів. Вивільніть динамічну пам'ять.

*Вхідні дані:* *n = 3*; .

*Вихідні дані:* *Номер рядка =* 2.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання цієї задачі я застосовував функцію malloc, для визначення динамічної пам’яті, я ввів матрицю , і вивів на екран найбільшу кількість повторів, і в кінці вивільнив масив за допомогою **free()**.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <time.h>

### **Список використаних функції:**

srand(time(NULL)); - встановлює в якості бази поточний час.

1. Malloc - повертає адресу на перший байт області пам'яті розміром size байт
2. sizeof - це унарний оператор, який повертає довжину в байтах змінної або типу, поміщених в дужки.

Rand- генерує випадкові числа, повертає псевдовипадкове ціле число в діапазоні від 0 до RAND\_MAX .

Free - повертає пам'ять, на яку вказує параметр , тому в купу.

## **Висновок**

1. В цій задачі було використано динамічну пам’ять. Завдяки rand ренерквалось випадкове число.
2. Після знаходження елементів матриці, я використав функція для вивільнення пам’яті з масиву.

## **Додаток(а)**

**1.c**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

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

srand(time(NULL));

int \*\*A;

int i,b[50],n,m,j;

printf("Enter n= ");scanf("%d",&n);

printf("Enter m= "); scanf("%d",&m);

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

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

A[i]=(int\*)malloc(m\*sizeof(int));

printf("Matrix\n");

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

for(j=0;j<m;j++){

A[i][j]=rand()%20-13;

printf("%d\t",A[i][j]);}

printf("\n");}

printf("\n");

int max,kil,k,idx;

max=0;

idx=n;

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

kil=0;

for(j=0;j<m;j++){

for(k=j+1;k<m;k++){ if (A[i][j]==A[i][k]){ kil++; break;}

}}

printf("Kil= %d\n",kil);

if(kil>max)

{

max=kil;

idx=i;

}

}

printf("\n");

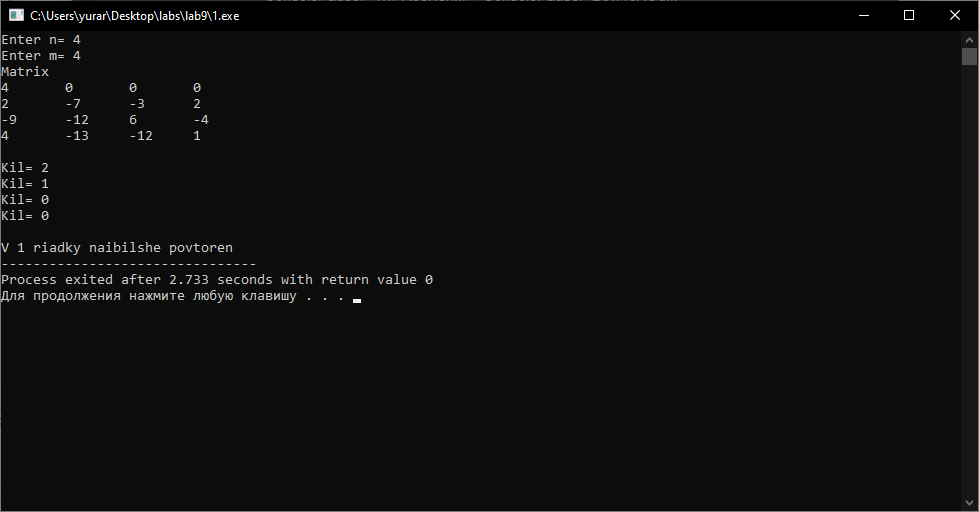
printf("V %d riadky naibilshe povtoren",idx+1);

free(A);

return 0;

}

## **Додаток(б)**



# **Завдання 10**

### **Опис завдання**

Варіант 13

Доступна така інформація про музичні твори:

– носій (грамплатівка, аудіокасета, лазерний диск);

– назва;

– виконавець (прізвище, ім'я);

– час звучання;

– кількість творів;

– ціна по каталогу.

Створити в структурі функції, що визначають: а) загальний час звучання пісень для заданого виконавця; б) середню вартість всіх аудіокасет.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання цієї задачі я застосував багато змінних і функцій серед них була strcpy(копіює значення рядка),stmp(здійснює лексикографічну перевірку двох рядків).Av\_price – середня ціна.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <string.h>

### **Список використаних функції:**

1. sizeof - це унарний оператор, який повертає довжину в байтах змінної або типу, поміщених в дужки.
2. Void- універсальний" покажчик, який просто відповідає вказівником на будь-яку адресу в пам'яті комп'ютера, безвідносно типу даних, на які вказує цей покажчик.
3. char – проста зміна, яка працює зі стовпчиками.
4. break - оператор завершує виконання найближчого включає циклу або умовного оператора, в якому він відображається.
5. switch case -це зручна заміна длинної конструкції if-else, яка порівняє переменную з декількома константними знаннями.
6. struct - совокупність змінних, об'єднаних одних імен.
7. strcpy- використовується для копіювання вмісту з одного масиву в інший.
8. strcmp- здійснює лексикографічну перевірку двох рядків.

## **Висновок**

В цій задачі мені довелося створити декілька змінних і застосувати змінних і функцій, було створено список в якому можна вказати носій, ім’я пісні, виконавця пісні, кількість пісень, і загальний час.

## **Додаток(а)**

**1.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

struct music

{

char nosiy[6];

char name [255];

char artist[255];

int tim;

int kil;

int price;

};

void Artist(int n, struct music \*M)

{ int i,sum=0;

char a[50];

printf("Enter you Artist\n");

scanf\_s("%s",&a,sizeof(a));

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

{

if(strcmp(M[i].artist,a)==0) {sum+=M[i].tim;}

}

if(sum!=0){ printf("All time of this Artist: %d", sum); }

else printf("We don't have information of this artist");

}

void Av\_Price(int n, struct music \*M)

{ int i,sum=0,kil=0;

float av;

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

{

if(strcmp(M[i].nosiy,"tape")==0) {sum+=M[i].price;kil++; }

}

av=1.0\*sum/kil;

printf("Avarage sume of all tape %.3f",av);

}

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

int n,i;

printf("Enter n="); scanf("%d",&n);

struct music M[n];

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

{

printf("Enter informasion of %d music\n ",i+1);

printf("Type: ");

printf("1-gramp\n 2-tape\n 3-disc\n");

int t;

scanf("%d",&t);

switch(t)

{

case 1: { strcpy(M[i].nosiy,"gram");

break;

}

case 2: { strcpy (M[i].nosiy,"tape");

break;

}

case 3: { strcpy (M[i].nosiy,"disc");

break;

}

}

printf("Name:"); scanf\_s("%s",&M[i].name,sizeof(M[i].name));

printf("Artist:"); scanf\_s("%s",&M[i].artist,sizeof(M[i].artist));

printf("All time:"); scanf("%d", &M[i].tim);

printf("Kil traks:"); scanf("%d", &M[i].kil);

printf("Price:"); scanf("%d", &M[i].price);

printf("-----------------------------------------------------------\n");

}

Artist(n,M);

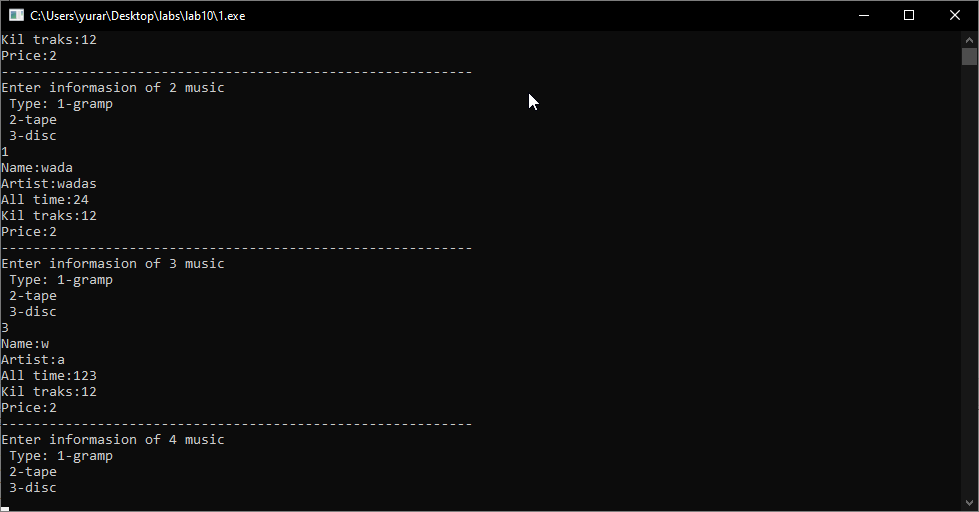
printf("\n");

Av\_Price(n,M);

return 0;

}

## **Додаток(б)**



# **Завдання 11**

### **Опис завдання**

Варіант 13

1. Відомі дані про 25 учнів класу: прізвище, ім'я, по батькові, адреса та домашній телефон, якщо він є. Вивести на екран прізвище, ім'я та адресу кожного учня, у якого немає домашнього телефону та учня в номері, якого присутня цифра «7» . Розглянути три випадки:
   1. телефон заданий у вигляді 6 – значного числа: 557059;
   2. телефон заданий у вигляді всеукраїнського стандарту: (80372) – 55–00–57
   3. телефон заданий у вигляді: 55-50-57.
2. **Прості задачі.**

Відомий ріст і підлога кожного з 22 чоловік. Знайти середній ріст чоловіків.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1.Для виконання першої задачі я створив структуру де треба було написати ім’я, фамілію, по батьку, пошту, і адресу. Також в умові задачі треба було дізнатися чи є у школяра телефон чи ні.

2.Для виконання цієї задачі я структуру для пошуку за ростом і статтю. Якщо написати 4 рази , то видасть середній ріст у чоловіків.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <string.h>

### **Список використаних функції:**

1. char – проста зміна, яка працює зі стовпчиками.
2. break - оператор завершує виконання найближчого включає циклу або умовного оператора, в якому він відображається.
3. switch case -це зручна заміна длинної конструкції if-else, яка порівняє переменную з декількома константними знаннями.
4. Void- універсальний" покажчик, який просто відповідає вказівником на будь-яку адресу в пам'яті комп'ютера, безвідносно типу даних, на які вказує цей покажчик.
5. struct - совокупність змінних, об'єднаних одних імен.
6. strcpy- використовується для копіювання вмісту з одного масиву в інший.
7. strcmp- здійснює лексикографічну перевірку двох рядків.
8. double- типи даних з плаваючою точкою зберігаються в пам'яті комп'ютера інакше, ніж цілочисельні.

## **Висновок**

## **Додаток(а)**

**1.с**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

struct pupil {

char surname[50];

char name[50];

char fname[50];

char email[50];

char adress[50];

char pnumb[11];

};

void Inf(int n, struct pupil \*M)

{

int i;

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

{

if(strcmp(M[i].pnumb,"-")==0)

{

printf("%s\t%s\t%s",M[i].surname,M[i].name,M[i].adress);

printf("\n");

}

}

}

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

int n,i;

printf("Enter n="); scanf("%d",&n);

struct pupil M[n];

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

{

printf("Enter informasion of %d pupil\n ",i+1);

printf("Surname:"); scanf\_s("%s",&M[i].surname,sizeof(M[i].surname));

printf("Name:"); scanf\_s("%s",&M[i].name,sizeof(M[i].name));

printf("FName:"); scanf\_s("%s", &M[i].fname,sizeof(M[i].fname));

printf("EMail:"); scanf\_s("%s", &M[i].email,sizeof(M[i].email));

printf("Adress:"); scanf\_s("%s", &M[i].adress,sizeof(M[i].adress));

printf("PNumb:");

printf("Do you have a phone?\n");

printf("1-yes\n2-no\n");

int t;

scanf("%d",&t);

switch (t)

{

case 1: {

scanf\_s("%s",&M[i].pnumb,sizeof(M[i].pnumb));

break;

}

case 2:{

strcpy(M[i].pnumb,"-");

break;

}

}

printf("-----------------------------------------------------------\n");

}

printf("Pupil no phone:\n");

Inf(n,M);

return 0;

}

**2.с**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

struct people {

char sex[5];

int rist;

};

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

int n,i;

printf("Enter n="); scanf("%d",&n);

struct people M[n];

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

{

printf("Enter informasion of %d human\n ",i+1);

printf("Sex:");

printf("1-man\n2-woman\n");

int t;

scanf("%d",&t);

switch (t)

{

case 1: {

strcpy(M[i].sex,"man");

break;

}

case 2:{

strcpy(M[i].sex,"woman");

break;

}

}

printf("Rist:"); scanf("%d", &M[i].rist);

printf("-----------------------------------------------------------\n");

}

printf("Average of man:\n");

int sum=0, kil=0;

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

{

if (strcmp(M[i].sex,"man")==0) {kil++; sum+=M[i].rist; }

}

double av;

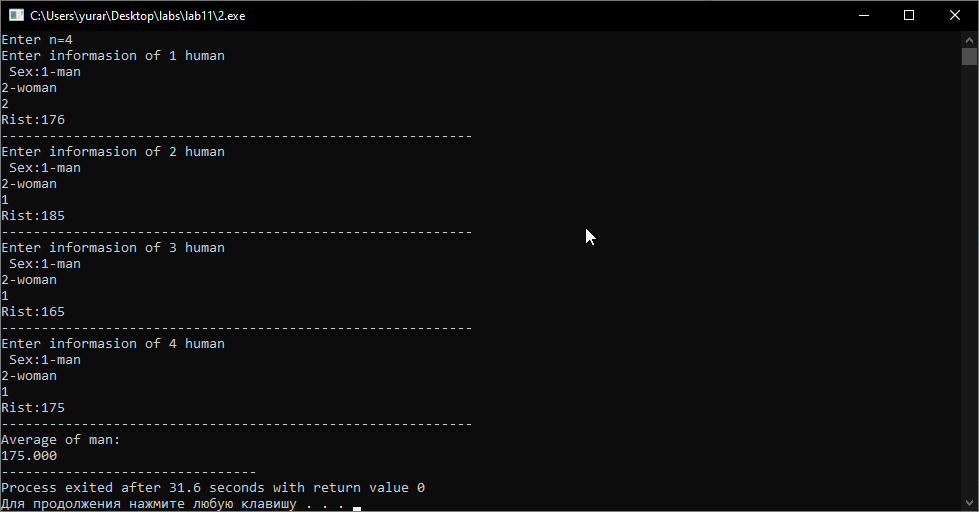
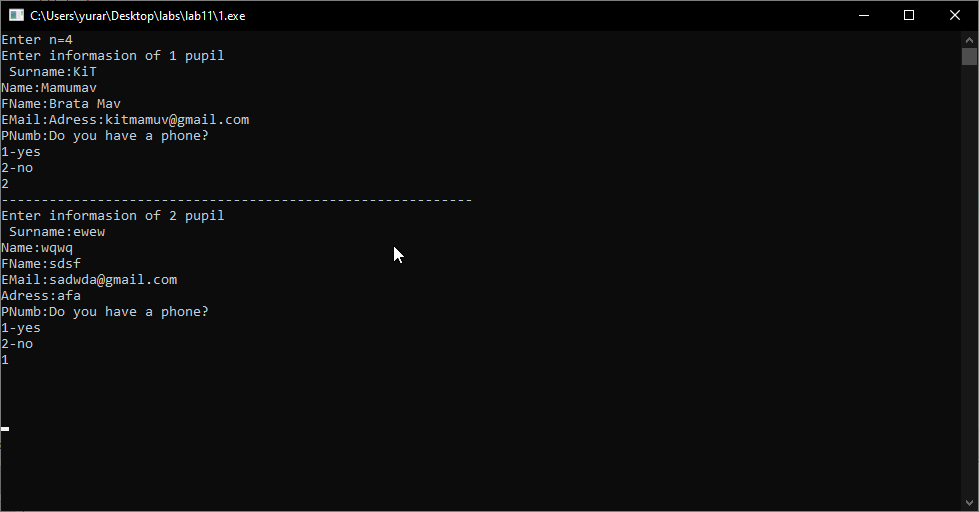
av=1.0\*sum/kil;

printf("%0.3f",av);

return 0;

}

## **Додаток(б)**



# **Завдання 12**

### **Опис завдання**

Варіант 13

Створити однозв’язний лінійний список елементами якого є символи. Реалізувати функцію, яка повертає кількість різних маленьких латинських літер в списку. Вивільнити динамічну пам’ять.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання цієї задачі було створенно велику кількість змінних, таких як PElem(елементи),st(старт) і тд.

В цій задачі треба було створити лінійний список де вводиш кількість(до прикладу 4), і треба буде написати 4 букви, і задача порахує кількість.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <conio.h>
4. #include <malloc.h>
5. #include <string.h>

### **Список використаних функції:**

1. char – проста зміна, яка працює зі стовпчиками.
2. Void- універсальний" покажчик, який просто відповідає вказівником на будь-яку адресу в пам'яті комп'ютера, безвідносно типу даних, на які вказує цей покажчик.
3. struct - совокупність зиінні, об'єднаних одних імен.
4. typedef struct-це ім’я нового типу.
5. sizeof - це унарний оператор, який повертає довжину в байтах змінної або типу, поміщених в дужки.

While- це цикл з умовою поста, де істинність виразу.

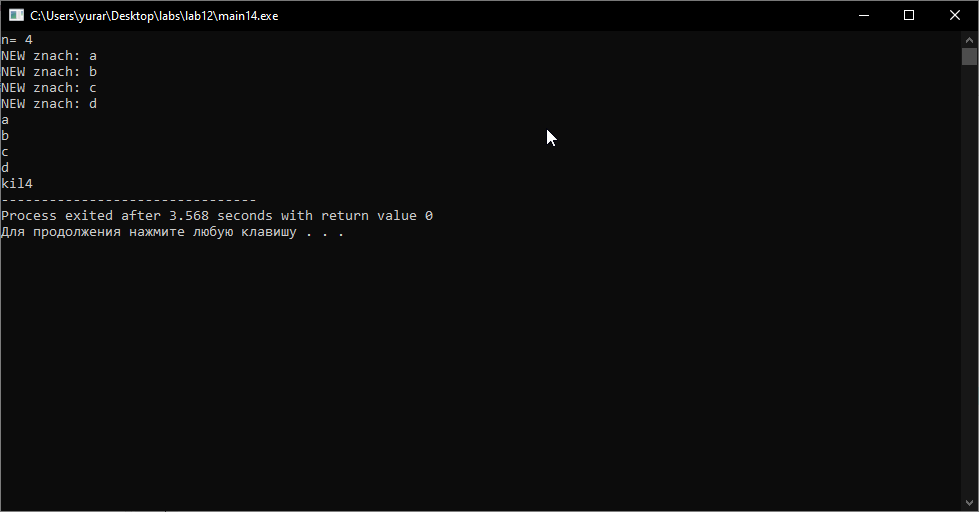
## **Висновок**

В цій задачі я підклчив більше всього бібліотек, і як на мене це одна з самих складних лабораторних робіт під час літньої практики.

## **Додаток(а)**

1. **main14.с**
2. #include <stdio.h>
3. #include <conio.h>
4. #include <malloc.h>
5. #include <string.h>
6. typedef struct PElem {
7. char inf;
8. struct PElem \*Next;
9. }TElem;
10. TElem \*st, \*pt, \*dod,\*pt1;
11. void funk (TElem \*pt){
12. int total=0;
13. pt1=pt;
14. while (pt != NULL)
15. { if (pt->inf>96&&pt->inf<123)
16. { int kil=0;
18. while (pt1 != NULL)
19. {
20. if (pt->inf==pt1->inf) kil++;
21. pt1=pt1->Next; }
22. if (kil==1) {total++;}}
23. pt1=st;
24. pt = pt->Next;
25. }
26. printf("kil%d",total);
27. }
28. int main(int argc, char \*argv[]) {
29. int n, i;
30. printf("n= ");
31. scanf("%d", &n);
32. st= NULL, pt=NULL;
33. for(i=0; i<n; i++){
34. dod=(TElem\*)calloc(1,sizeof(TElem));
35. printf("NEW znach: ");
36. scanf("%s",&dod->inf);
37. dod->Next=NULL;
38. if(st==NULL)st=dod;
39. else pt->Next=dod;
40. pt=dod;
41. }
42. pt = st; i = 0;
43. pt1=pt;
44. int total=0;
45. while (pt != NULL)
46. { printf("%c\n",pt->inf);
47. pt = pt->Next;}
48. pt = st;
49. funk(pt);
50. return 0;
51. }

## **Додаток(б)**



# **Завдання 13**

### **Опис завдання**

Варіант 13

Дано текстовий файл. Створити на його основі цілочисловий масив *а*[*i*], який складається з чисел, що визначають кількість символів в кожному з рядків.

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Для виконання цієї задачі я вирішив поексперементувати і написати програму на с++ для економії часу.

Я створив текстовий текстовий документ і записав там дані, якщо запустити програми то дані з текстового документа будуть виведені на екран.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <iostream>
3. #include <fstream>

### **Список використаних функції:**

using namespace - надає можливість використовувати все ім’я

setlocale- позволяет запрашивать или устанавливать определенные параметры, зависящие от географического положения.

Ifsteam- для введення (читання) даних з файлу

Getline-призначена для введення даних з потоку, наприклад, для введення даних з консольного вікна.

Cout -аналог printf.

## **Висновок**

Це одна з тих лабораторних де я вперше спробував написати код на с++, думаю вийшло добре.Також я створив файл, дані якого будуть виводитися на екран.

## **Додаток(а)**

**Lab13.1.cpp**

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

ifstream fin;

int temp = 0, b[100];

fin.open("Test.txt");

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

cout << "Îøèáêà îòêðûòèÿ ôàéëà";

}else{

string str;

while(!fin.eof()){

str = " ";

getline(fin, str);

cout << str << endl;

b[temp] = str.length();

temp++;

}

}

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

cout << b[i] << ' ';

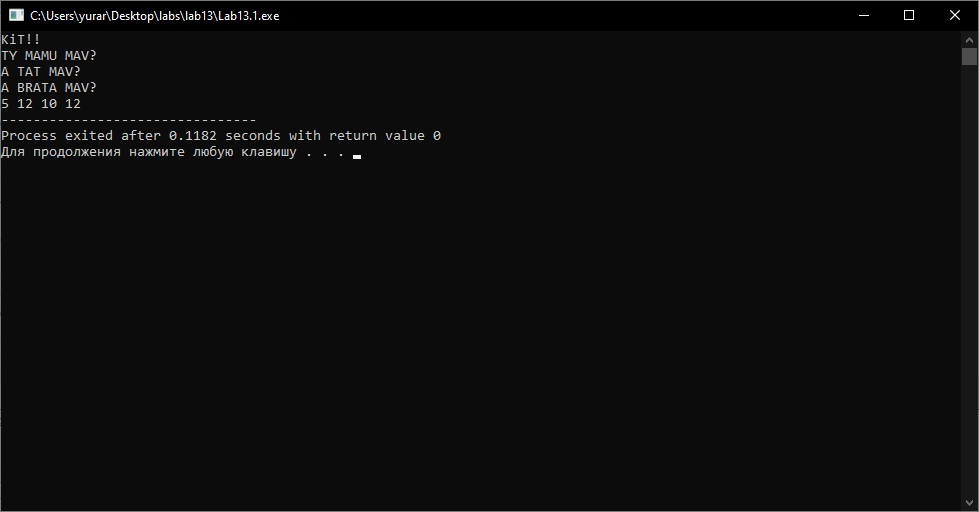
}

fin.close();

return 0;

}

## **Додаток(б)**



# **Завдання 14**

### **Опис завдання**

Варіант 13

1. Дано рядок *S*. Подвоїти входження кожної голосної букви і видалити всі пробіли.

*Вхідні дані:* *S = "Mathematic – a queen of the science"*.

*Вихідні дані:* *S = "Maatheemaatiic – aa quueeeen oof thee sciieencee"*.

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:

2.слова, в яких є хоча б одна з букв: 'а', 'b', 'c', 'd' i 'e';

3.цифри ('0','1',...'9') та знаки: '>', '<', '=';

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

1.Для виконнаня першої задачі я повернувся до задачі з лабораторних робіт, де були лексимуми. В цій задачі повторюються деякі букви словосполучення *Mathematic – a queen of the science.*

2. В другій же задачі словосполучення *Mathematic – a queen of the science* було розділено і кожне слово було написано з нового рядка.

3.В третій задачі записав числа від 1 до 8 , останній елемент буде видалений.

## **СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

1. **Список підключених бібліотек**
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <string.h>

### **Список використаних функції:**

Gets - функція, що входить в Стандартну бібліотеку мови Сі, оголошується в заголовки stdio. h, яка зчитує рядок стандартного вводу і поміщає її в буфер, створений викликає функцією.

Strlen – довжина рядка.

1. char – проста зміна, яка працює зі стовпчиками.

## **Висновок**

Всі ці задачі об’єднує те , що вони за структурою написання схожі між собою, і те що їх об’єднує тема лексимумів.

## **Додаток(а)**

**1.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

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

char s[255];

int n, i;

printf("Enter text\n");

gets(s);

n=strlen(s);

int kil=0;

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

{

if((s[i]=='a')||(s[i]=='i')||(s[i]=='o')||(s[i]=='y')||(s[i]=='e')||(s[i]=='u')) {printf("%c%c",s[i],s[i]); }

else printf("%c",s[i]);}

return 0;

}

**2.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

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

char s[255];

int n, i,j;

printf("Enter text\n");

gets(s);

n=strlen(s);

int kil=0;

int k1=0,k2;

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

{ kil=0;

if(s[i]==' ') {k2=i;

for(j=k1;j<k2;j++)

if((s[j]=='a')||(s[j]=='b')||(s[j]=='c')||(s[j]=='d')||(s[j]=='e')) {kil++; }

if(kil!=0)

{

for(j=k1;j<k2;j++)

printf("%c",s[j]);

printf("\n");

}

k1=k2+1;}

}

return 0;

}

**3.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

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

char s[255];

int n, i,j;

printf("Enter text\n");

gets(s);

n=strlen(s);

int kil=0;

int k1=0,k2;

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

{ kil=0;

if(s[i]==' ') {k2=i;

for(j=k1;j<k2;j++)

if(((s[j]>='1')&& (s[j]<='9'))||(s[j]=='>')||(s[j]=='<')||(s[j]=='=')) {kil++; }

if(kil!=0)

{

for(j=k1;j<k2;j++)

printf("%c",s[j]);

printf("\n");

}

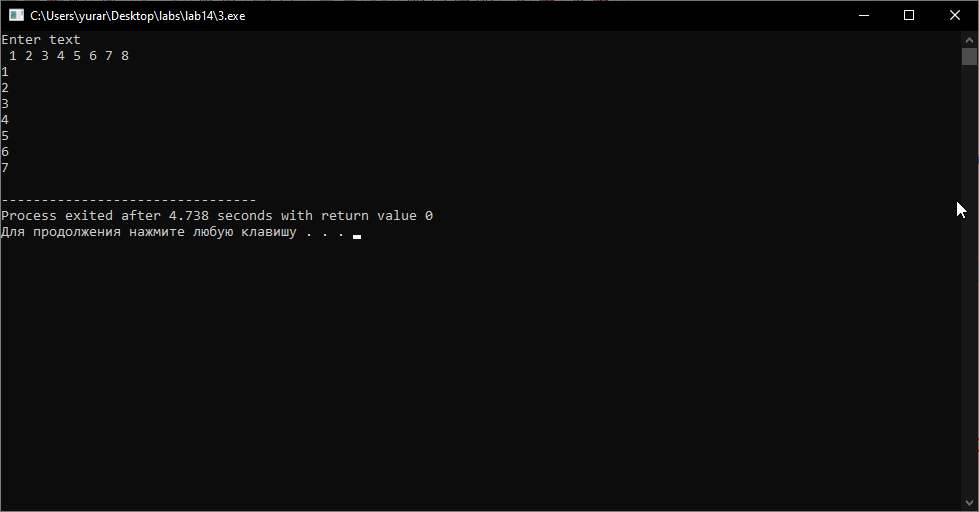
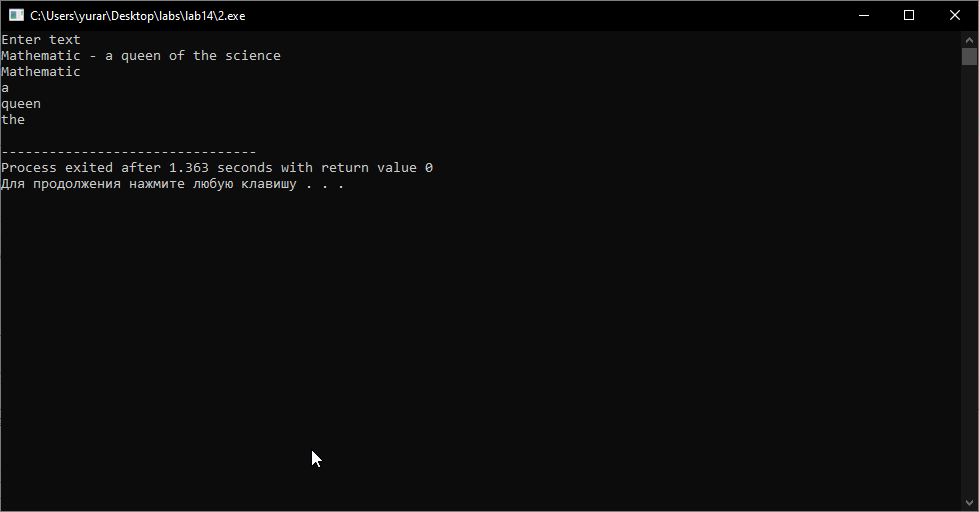
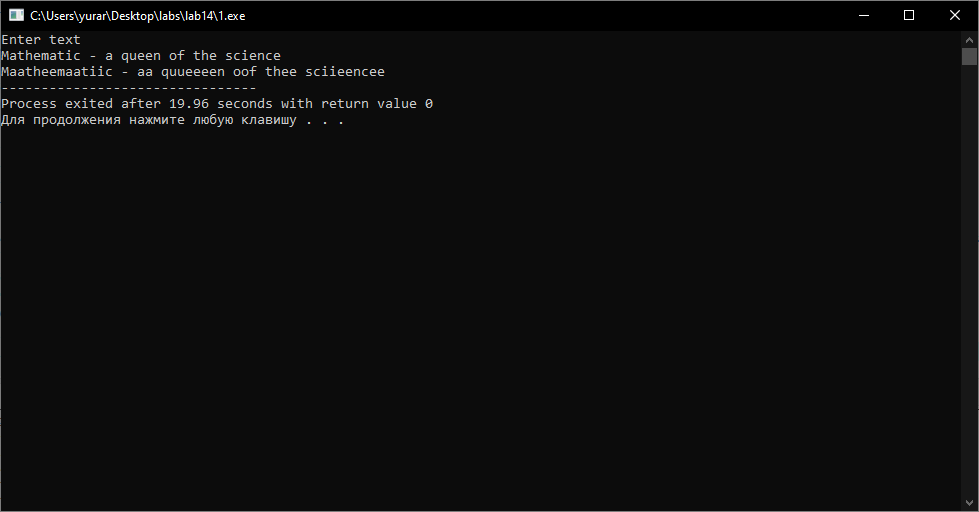
k1=k2+1;}

}

return 0;

}

## **Додаток(б)**

****