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»

204групи

*(Дармограй Анастасії Сергіївни)*

Період практики з “*20.07* ” 20*20* р. по “*07.08*” 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 7](#_Toc46935607)

[ПОСТАНОВКА ЗАДАЧІ 8](#_Toc46935608)

[Опис завдання 8](#_Toc46935609)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 8](#_Toc46935610)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 8](#_Toc46935611)

[Список підключених бібліотек: 8](#_Toc46935612)

[Висновок 8](#_Toc46935613)

[Додаток(а) 9](#_Toc46935614)

[Додаток(б) 9](#_Toc46935615)

[Завдання 2 10](#_Toc46935616)

[ПОСТАНОВКА ЗАДАЧІ 11](#_Toc46935617)

[Опис завдання 11](#_Toc46935618)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 11](#_Toc46935619)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 11](#_Toc46935620)

[Список підключених бібліотек: 11](#_Toc46935621)

[Висновок 12](#_Toc46935622)

[Додаток(а) 12](#_Toc46935623)

[Додаток(б) 15](#_Toc46935624)

[Завдання 3 16](#_Toc46935625)

[ПОСТАНОВКА ЗАДАЧІ 17](#_Toc46935626)

[Опис завдання 17](#_Toc46935627)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 17](#_Toc46935628)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 18](#_Toc46935629)

[Список підключених бібліотек: 18](#_Toc46935630)

[Список використаних функцій: 18](#_Toc46935631)

[Висновок 18](#_Toc46935632)

[Додаток(а) 19](#_Toc46935633)

[Додаток(б) 24](#_Toc46935634)

[Завдання 4 26](#_Toc46935635)

[ПОСТАНОВКА ЗАДАЧІ 27](#_Toc46935636)

[Опис завдання 27](#_Toc46935637)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 27](#_Toc46935638)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 27](#_Toc46935639)

[Список підключених бібліотек: 27](#_Toc46935640)

[Висновок 28](#_Toc46935641)

[Додаток(а) 28](#_Toc46935642)

[Додаток(б) 30](#_Toc46935643)

[Завдання 5 31](#_Toc46935644)

[ПОСТАНОВКА ЗАДАЧІ 32](#_Toc46935645)

[Опис завдання 32](#_Toc46935646)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 32](#_Toc46935647)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 32](#_Toc46935648)

[Список підключених бібліотек: 32](#_Toc46935649)

[Список використаних функцій: 32](#_Toc46935650)

[Висновок 33](#_Toc46935651)

[Додаток(а) 33](#_Toc46935652)

[Додаток(б) 35](#_Toc46935653)

[Завдання 6 36](#_Toc46935654)

[ПОСТАНОВКА ЗАДАЧІ 37](#_Toc46935655)

[Опис завдання 37](#_Toc46935656)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 37](#_Toc46935657)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 38](#_Toc46935658)

[Список підключених бібліотек: 38](#_Toc46935659)

[Висновок 38](#_Toc46935660)

[Додаток(а) 39](#_Toc46935661)

[Додаток(б) 44](#_Toc46935662)

[Завдання 7 45](#_Toc46935663)

[ПОСТАНОВКА ЗАДАЧІ 46](#_Toc46935664)

[Опис завдання 46](#_Toc46935665)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 46](#_Toc46935666)

[Список підключених бібліотек: 46](#_Toc46935667)

[Список використаних функцій: 47](#_Toc46935668)

[Висновок 47](#_Toc46935669)

[Додаток(а) 48](#_Toc46935670)

[Додаток(б) 53](#_Toc46935671)

[Завдання 8 54](#_Toc46935672)

[ПОСТАНОВКА ЗАДАЧІ 55](#_Toc46935673)

[Опис завдання 55](#_Toc46935674)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 55](#_Toc46935675)

[Список підключених бібліотек: 55](#_Toc46935676)

[Список використаних функцій: 55](#_Toc46935677)

[Висновок 55](#_Toc46935678)

[Додаток(а) 56](#_Toc46935679)

[Додаток(б) 57](#_Toc46935680)

[Завдання 9 58](#_Toc46935681)

[ПОСТАНОВКА ЗАДАЧІ 59](#_Toc46935682)

[Опис завдання 59](#_Toc46935683)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 59](#_Toc46935684)

[Список підключених бібліотек: 59](#_Toc46935685)

[Список використаних функцій: 59](#_Toc46935686)

[Висновок 60](#_Toc46935687)

[Додаток(а) 60](#_Toc46935688)

[Додаток(б) 62](#_Toc46935689)

[Завдання 10 63](#_Toc46935690)

[ПОСТАНОВКА ЗАДАЧІ 64](#_Toc46935691)

[Опис завдання 64](#_Toc46935692)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 64](#_Toc46935693)

[Список підключених бібліотек: 64](#_Toc46935694)

[Висновок 65](#_Toc46935695)

[Додаток(а) 65](#_Toc46935696)

[Додаток(б) 67](#_Toc46935697)

[Завдання 11 68](#_Toc46935698)

[ПОСТАНОВКА ЗАДАЧІ 69](#_Toc46935699)

[Опис завдання 69](#_Toc46935700)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 69](#_Toc46935701)

[Список підключених бібліотек: 70](#_Toc46935702)

[Список використаних функцій: 70](#_Toc46935703)

[Висновок 70](#_Toc46935704)

[Додаток(а) 71](#_Toc46935705)

[Додаток(б) 75](#_Toc46935706)

[Завдання 12 78](#_Toc46935707)

[ПОСТАНОВКА ЗАДАЧІ 79](#_Toc46935708)

[Опис завдання 79](#_Toc46935709)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 79](#_Toc46935710)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 79](#_Toc46935711)

[Список підключених бібліотек: 79](#_Toc46935712)

[Список використаних функцій: 79](#_Toc46935713)

[Висновок 79](#_Toc46935714)

[Додаток(а) 79](#_Toc46935715)

[Завдання 13 80](#_Toc46935716)

[ПОСТАНОВКА ЗАДАЧІ 81](#_Toc46935717)

[Опис завдання 81](#_Toc46935718)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 81](#_Toc46935719)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 81](#_Toc46935720)

[Список підключених бібліотек: 81](#_Toc46935721)

[Висновок 81](#_Toc46935722)

[Додаток(а) 81](#_Toc46935723)

[Додаток(б) 82](#_Toc46935725)

[Завдання 14 83](#_Toc46935727)

[ПОСТАНОВКА ЗАДАЧІ 84](#_Toc46935728)

[Опис завдання 84](#_Toc46935729)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 84](#_Toc46935730)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 84](#_Toc46935731)

[Список підключених бібліотек: 84](#_Toc46935732)

[Висновок 84](#_Toc46935734)

[Додаток(а) 84](#_Toc46935735)

[Додаток(б) 84](#_Toc46935736)

Посилання на Google сайт <https://sites.google.com/view/nastia204/%D0%B3%D0%BB%D0%B0%D0%B2%D0%BD%D0%B0%D1%8F-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0?authuser=1>

Посилання на репозиторій

<https://github.com/Nastia204Darmohrai/Hello-world>

Посилання на папку

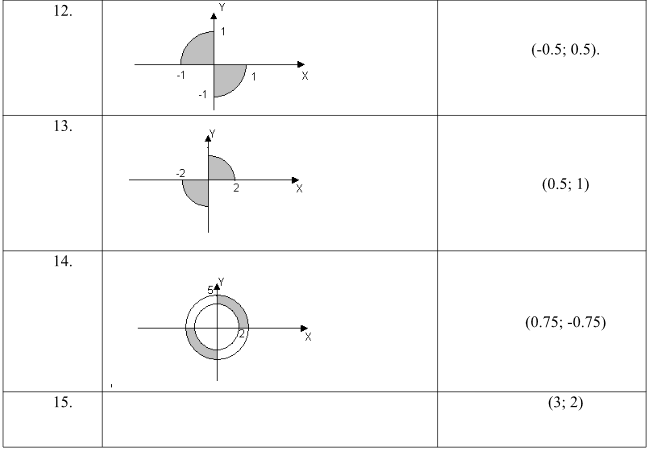
<https://drive.google.com/drive/folders/1l3vN1uJP6hmrJ6HKbEY8QYgz-8RiGPMi>

# Завдання 1

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

1. Варіант 10
2. Створити алгоритм та написати програму для розвязку наступної задачі: перевірку попадання точки ( яка буде вводитись вручну із клавіатури) в заштриховану область.



## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if;
* математичні знання щодо визначення координат точки

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>

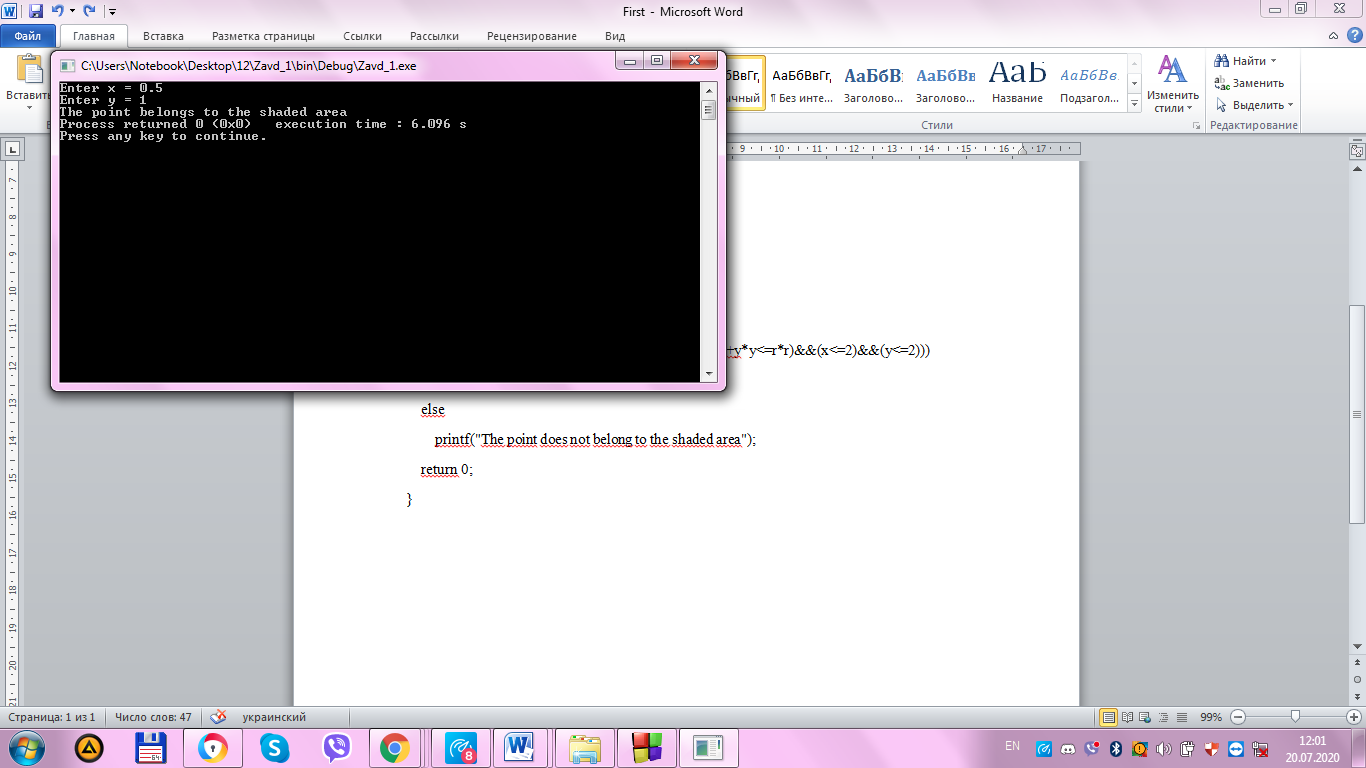
## Висновок

Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме при введенні координат точки знати чи належить вона заштрихованій області, що дана в умові. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити координати і відразу бачити результат.

## Додаток(а)

1. *#include <stdio.h>*
2. *#include <stdlib.h>*
3. *int main()*
4. *{*
5. *float x,y,r=2;*
6. *printf("Enter x = ");*
7. *scanf("%f", &x);*
8. *printf("Enter y = ");*
9. *scanf("%f", &y);*
10. *if (((x\*x+y\*y<=r\*r)&&(x<=-2)&&(y<=-2)) || ((x\*x+y\*y<=r\*r)&&(x<=2)&&(y<=2)))*
11. *printf("The point belongs to the shaded area");*
12. *else*
13. *printf("The point does not belong to the shaded area");*
14. *return 0;*
15. *}*

## Додаток(б)



# Завдання 2

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

1. **2.1** Знайти мінімальний та максимальний елементи заданого масиву К(9) і поміняти їх місцями. Вивести мінімальний і максимальний елементи, початковий та перетворений масиви.
2. **2.2** Вибрати з масиву D(12) вiд'ємнi елементи, вiдсортувати їх за спаданням та занести у масив Z. Вивести обидва масиви.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

1. **2.1** При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if, за допомогою якого визначала максимальний та мінімальний елементи
* оператор while для перевірки кількості елементів
* рандомне заповнення масиву за допомогою функції rand()%

1. **2.2** При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* спосіб сортування бульбашкою
* відбір від’ємних елементів за допомогою оператора вибору if

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>

## Висновок

**2.1**Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме визначати максимальний та мінімальний елементи масиву і міняти їх місцями. У результаті виведено конвертований масив, з вже поміняними елементами.

При створенні даної програми було розглянуто чимало інформації пов’язаної з масивами і роботи з ними і т.п.

Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

**2.2** Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме визначати від’ємні елементи масиву і згодом іх сортувати за спаданням від найбільшого до найменшого . У результаті виведено конвертований масив, вже з відсортованими значеннями.

При створенні даної програми було розглянуто чимало інформації пов’язаної з масивами і роботи з ними і т.п.

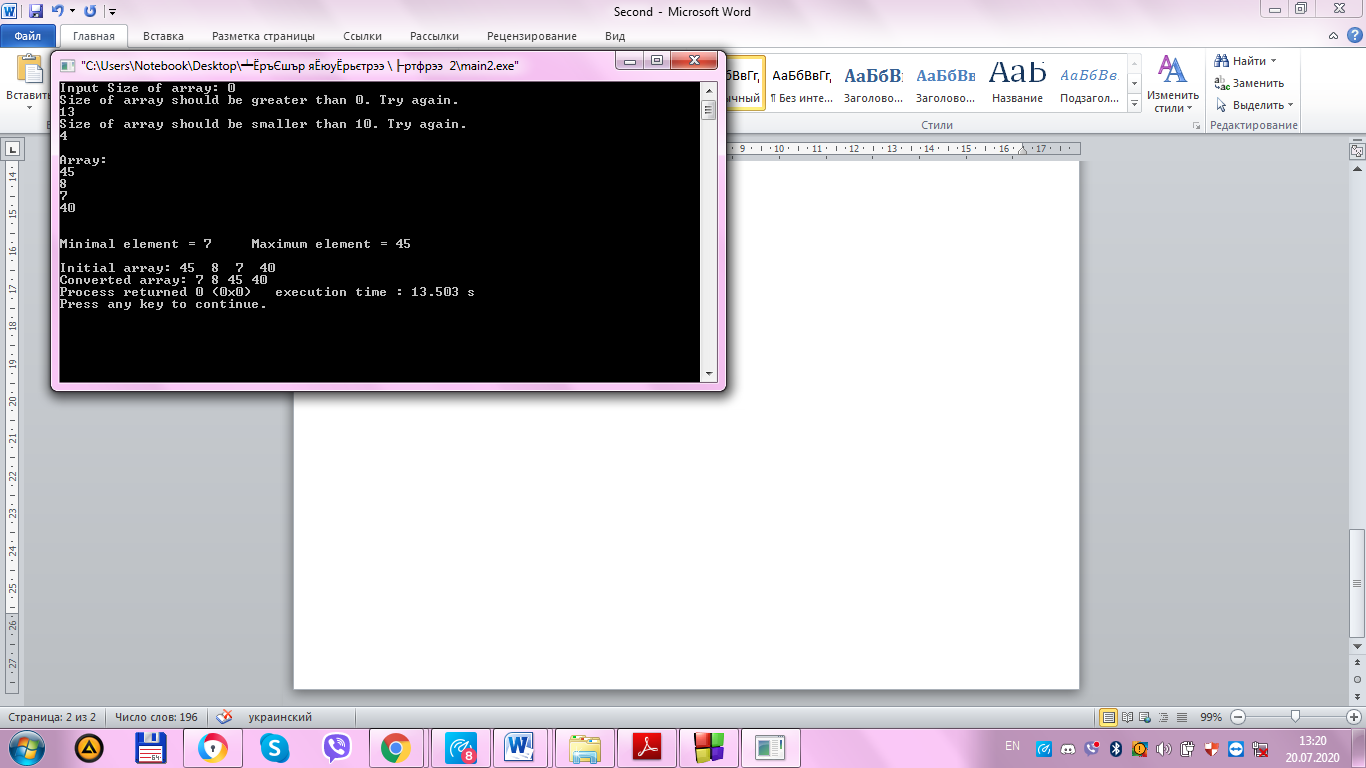
Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

## Додаток(а)

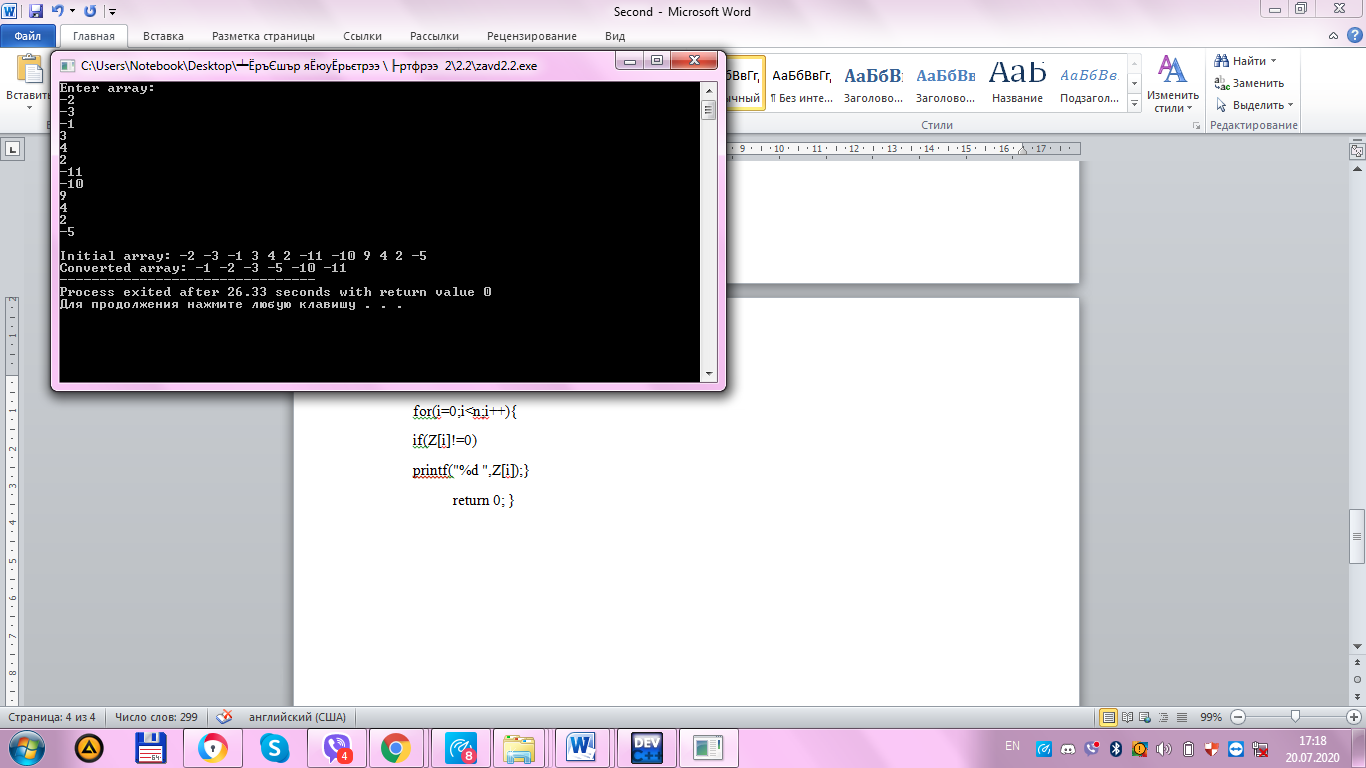
1. ***2.1***
2. *#include <stdio.h>*
3. *#include <stdlib.h>*
4. */\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/*
5. *int main(int argc, char \*argv[]) {*
6. *int K[9], i, Size, buf, min\_index=0, max\_index=0 ;*
7. *printf("Input Size of array: ");*
8. *scanf("%d", &Size);*
9. *while (Size < 1)*
10. *{*
11. *printf("Size of array should be greater than 0. Try again.\n");*
12. *scanf("%d", &Size);*
13. *}*
14. *while (Size > 9)*
15. *{*
16. *printf("Size of array should be smaller than 10. Try again.\n");*
17. *scanf("%d", &Size);*
18. *}*
19. *printf("\nArray:");*
20. *printf("\n");*
21. *srand(time(NULL));*
22. *for (i = 0; i < Size; i++)*
23. *{ K[i] = rand()%50;*
24. *printf("%d\n", K[i]); } printf("\n");*
25. *for (i = 0; i < Size; i++){*
26. *if (K[i]< K[min\_index])*
27. *min\_index=i;}*
28. *printf("\nMinimal element = %d", K[min\_index]);*
29. *for (i = 0; i < Size; i++){*
30. *if (K[i]> K[max\_index])*
31. *max\_index = i;}*
32. *printf("\tMaximum element = %d", K[max\_index]); printf("\n");*
33. *printf("\nInitial array: ");*
34. *for (i = 0; i < Size; i++)*
35. *{ printf("%d ", K[i]);}*
36. *printf("\nConverted array: ");*
37. *buf = K[min\_index];*
38. *K[min\_index] = K[max\_index];*
39. *K[max\_index] = buf;*
40. *for (i=0; i<Size; i++)*
41. *printf("%d ", K[i]);*
42. *return 0; }*
43. ***2.2***
44. *#include <stdio.h>*
45. *#include <stdlib.h>*
46. */\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/*
47. *int main(int argc, char \*argv[]) {*
48. *const int n=12;*
49. *int D[n];*
50. *int Z[n];*
51. *int i, j, buf=0;*
52. *printf("Enter array:");*
53. *for(i=0;i<n;i++){*
54. *scanf("%d", &D[i]);*
55. *Z[i]=0;*
56. *}*
57. *printf("\nInitial array: ");*
58. *for(i=0;i<n;i++){*
59. *printf("%d ", D[i]);*
60. *}*
61. *for(i=0; i<n; i++){*
62. *if(D[i]<0){*
63. *Z[i]=D[i];}}*
64. *for(i=0; i<n; i++)*
65. *for( j=0; j<n; j++){*
66. *if (Z[i]>Z[j]){*
67. *buf=Z[i];*
68. *Z[i]=Z[j];*
69. *Z[j]=buf;}}*
70. *printf("\nConverted array: ");*
71. *for(i=0;i<n;i++){*
72. *if(Z[i]!=0)*
73. *printf("%d ",Z[i]);}*
74. *return 0; }*

## Додаток(б)

**2.1**



**2.2**



# Завдання 3

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

1. **3.1** В заданій матриці L(5,4) знайти середнє арифметичне найбільшого та найменшого елементів. Вивести вихідну матрицю, найбільший, найменший елементи та їх середнє
2. арифметичне.
3. **3.2** 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.

**3.3** 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. **3.1** При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* рандомне заповнення масиву за допомогою функції rand()%

1. **3.2** При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if для знаходження максимального значення
* засоби роботи із масивом

**3.3**

1. При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if для знаходження максимального значення
* засоби роботи із масивами одновимірними та двовимірним

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <time.h>

### Список використаних функцій:

**3.1**

*void Init(int a[][N], int m, int n) –* заповнює двовимірний масив рандомними числами від -1 до 10.

*void Output(int a[][N], int m, int n) –* виводить масив на екран у вигляді так званої таблички.

*void Max\_Min\_Sered(int a[][N], int m, int n) –* знаходить максимальне і мінімальне значення масиву, вивовить їх. Також шукає середнє значення максимального і мінімального елементів, виводить його на екран*.*

## Висновок

1. **3.1**Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме визначати максимальний та мінімальний елементи масиву. А у результаті визначати і виводити їхнє середнє значення на екран .
2. **3.2**Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме визначати найбільший елемент серед парних додатних елементів, що розміщуються в стовпчиках з парними індексами.
3. **3.3**Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією з двовимірним та одновимірними масивами.
4. Загалом при створенні даних програм було розглянуто чимало інформації пов’язаної з масивами і роботи з ними і т.п.
5. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

## Додаток(а)

**3.1**

*# include<stdio.h>*

*#include <stdlib.h>*

*#include <time.h>*

*#define M 5*

*#define N 4*

*void Init(int a[][N], int m, int n)*

*{*

*int i, j;*

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

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

*a[i][j] = rand()%10-1;*

*}*

*void Output(int a[][N], int m, int n)*

*{*

*int i, j;*

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

*{*

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

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

*printf("\n");*

*}*

*}*

*void Max\_Min\_Sered(int a[][N], int m, int n)*

*{int i,j,max,min;*

*float ser,f;*

*max = a[0][0];*

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

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

*if(a[i][j] > max)*

*max = a[i][j];*

*printf("\nMax = %d", max);*

*min = a[0][0];*

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

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

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

*min = a[i][j];*

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

*f=max+min;*

*ser=f/2;*

*printf("\nAverage = %f",ser);*

*}*

*int main()*

*{*

*int a[M][N];*

*srand(time(NULL));*

*Init(a, M, N);*

*printf("\nMatrix:");*

*printf("\n");*

*Output(a, M, N);*

*Max\_Min\_Sered(a,M,N);*

*return 0;}*

**3.2**

*#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[]) {*

*int K[5][3];*

*int i, j,b,l, max=0;*

*printf("Enter numbers of matrix: ");*

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

*{*

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

*{ scanf("%d", &K[i][j]);}*

*}*

*printf("\nMatrix");*

*printf("\n");*

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

*{*

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

*{*

*printf("%d ", K[i][j]);*

*}*

*printf("\n");*

*}*

*max=K[0][0];*

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

*{*

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

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

*{ if(K[i][j] > max)*

*max = K[i][j];*

*b=i;*

*l=j;}*

*}}*

*printf("Maximum=%d[%d][%d]", max,b, l);*

*return 0;}*

**3.3**

*#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[]) {*

*int K[8][8], S1[8], S2[30];*

*int i, j, k1=0,k2=0;*

*srand(time(NULL));*

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

*{*

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

*{*

*K[i][j] = rand()%50;*

*}}*

*printf("\nMatrix");*

*printf("\n");*

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

*{*

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

*{*

*printf("%d ", K[i][j]);*

*}*

*printf("\n");*

*}*

*printf("\n");*

*printf("Array S1 of elements above the main diagonal\n");*

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

*{*

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

*{*

*if (i<j){*

*k1=k1+1;*

*S1[k1]=K[i][j];*

*printf("%d ", S1[k1]);}*

*}}*

*printf("\n");*

*printf("\n");*

*printf("Array S2 of elements below the main diagonal\n");*

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

*{*

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

*{*

*if (i>j){*

*k2=k2+1;*

*S2[k2]=K[i][j];*

*printf("%d ", S2[k2]);}*

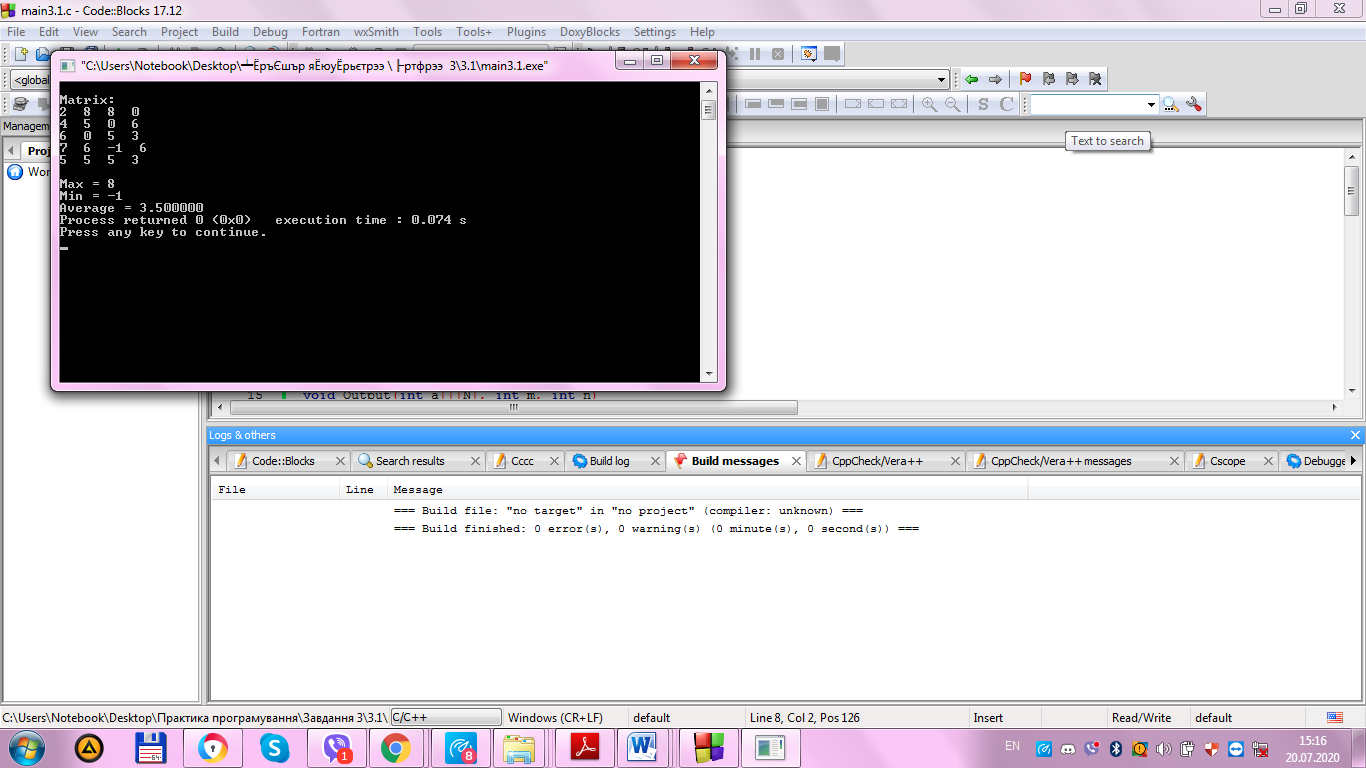
*}}*

*return 0;*

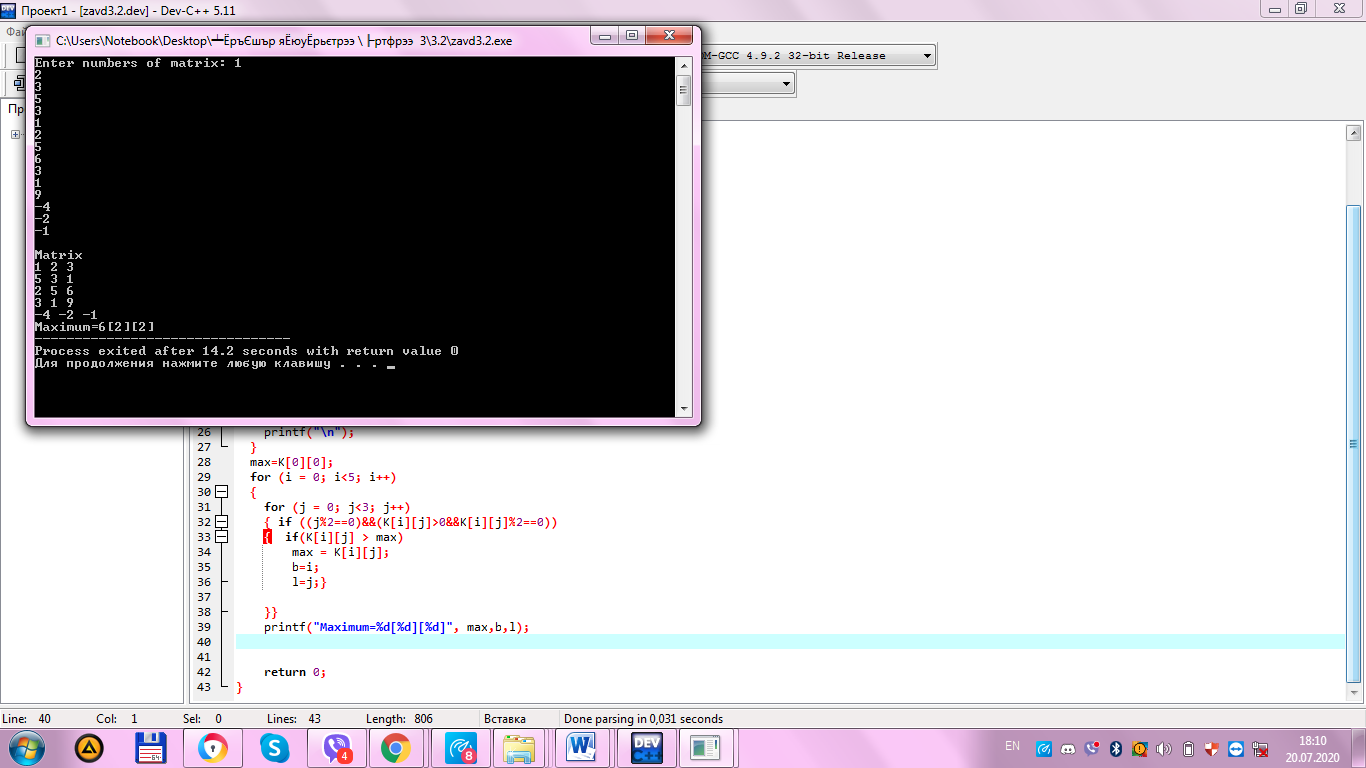
*}*

## Додаток(б)

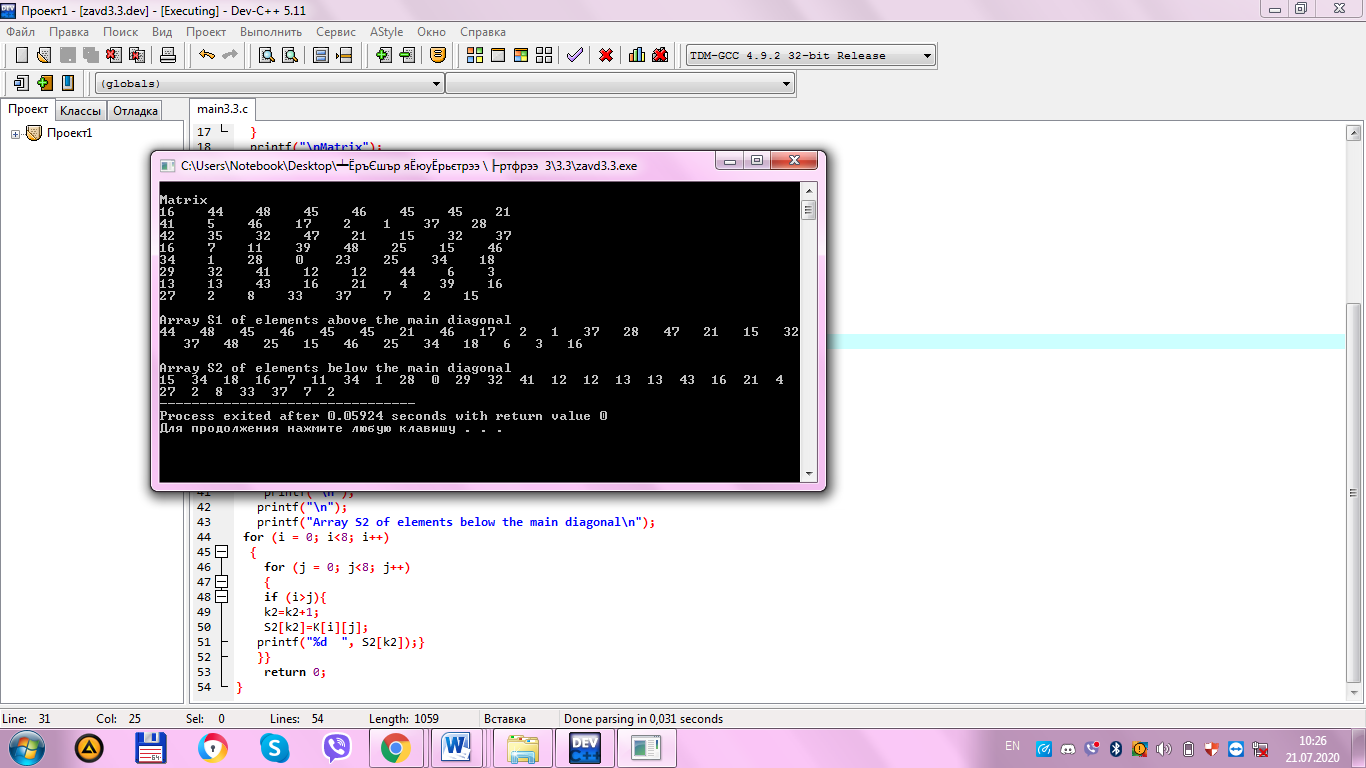
**3.1**



**3.2**



**3.3**



# Завдання 4

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант13

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

## ТЕОРЕТИЧНІ ВІДОМОСТІ

1. При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* рандомне заповнення масиву за допомогою функції rand()% в межах від -7 до 10
* засоби роботи із двовимірним масивом

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <time.h>

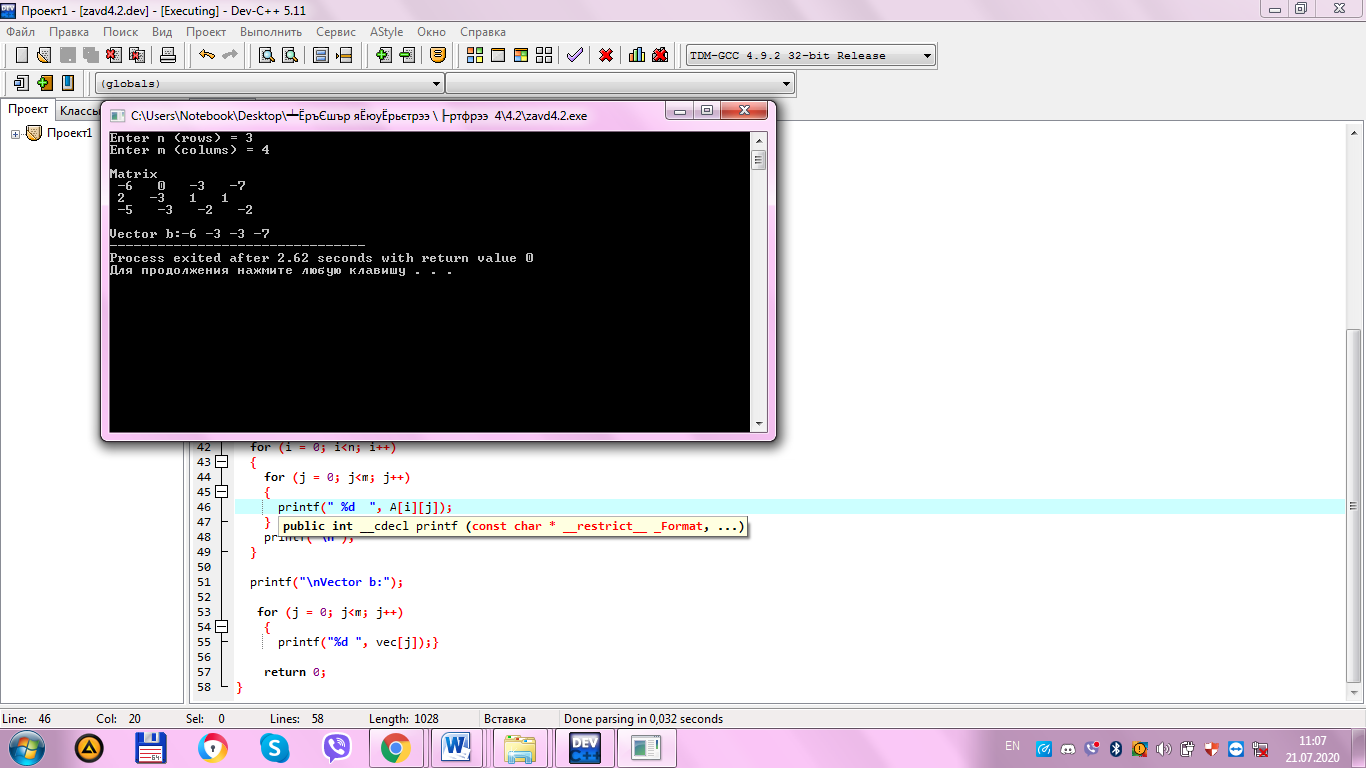
## Висновок

1. Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме визначати мінімальний елементи в кожному із стовпців масиву . У результаті виведено конвертований одновимірний масив, з цих мінімальних елементів
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з масивами і роботи з ними і т.п.
3. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

## Додаток(а)

1. #include <stdio.h>
2. #include <stdlib.h>
3. /\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/
4. int main(int argc, char \*argv[]) {
5. int i, j,b, n, m,vec[50], min=10000;
6. int A[100][100];
8. printf("Enter n (rows) = ");
9. scanf("%d",&n);
10. printf("Enter m (colums) = ");
11. scanf("%d",&m);
12. for (i = 0; i<n; i++)
13. {
14. for (j = 0; j<m; j++)
15. {
17. A[i][j] = rand()%10-7;
18. }
19. }
21. for ( j = 0; j < m; j++)
22. {
23. for ( i = 0; i < n; i++)
24. {
25. if (A[i][j] < min)
26. {
27. min = A[i][j];
28. }
29. }
30. vec[j] = min;
31. min = 10000;
32. }
34. printf("\nMatrix");
35. printf("\n");
36. for (i = 0; i<n; i++)
37. {
38. for (j = 0; j<m; j++)
39. {
40. printf(" %d ", A[i][j]);
41. }
42. printf("\n");
43. }
45. printf("\nVector b:");
46. for (j = 0; j<m; j++)
47. {
48. printf("%d ", vec[j]);}
49. return 0;
50. }

## Додаток(б)



# Завдання 5

## ПОСТАНОВКА ЗАДАЧІ

### 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.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* рандомне заповнення масиву за допомогою функції rand()% в межах 20
* засоби роботи із двовимірним масивом
* динамічна пам’ять та елементи роботи із нею

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <time.h>

### Список використаних функцій:

*double \*A(int n) –* виділення пам’яті під матрицю

*void I(int \*a, int n) –* заповнення матриі рандомними числами в межах 20

*void Print(int \*a, int n) –* виведення матриці на екран

*void Minimum (int \*a, int n) –* функція для знаходження мінімального елемента вище головної діагоналі

*void Sumadiah(int \*a, int n) –* функція знаходження суми елементів побічної діагоналі

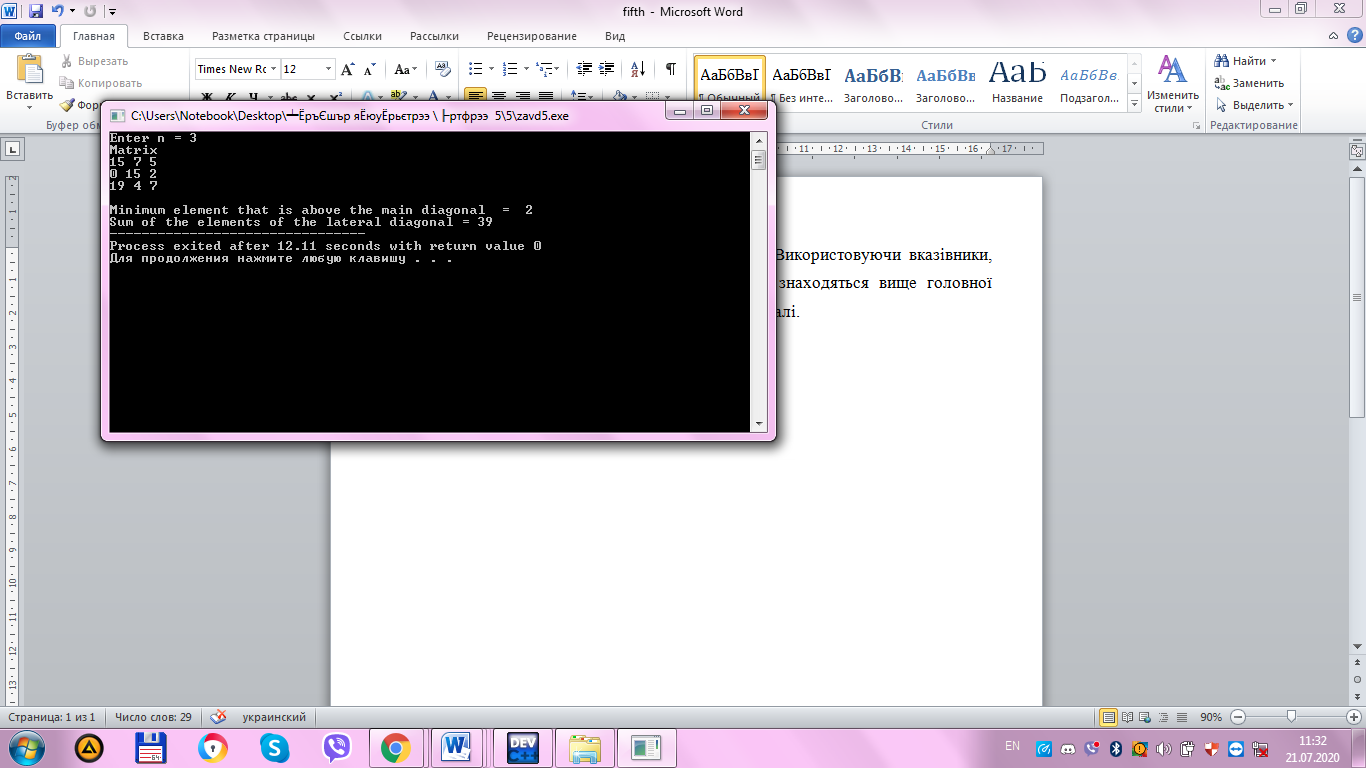
## Висновок

1. Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме визначати мінімальний елемент над головною діагоналлю та суму елементів побічної діагоналі .
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з масивами і роботи з нимиб також способи роботи із динамічною пам’яттю та елементами і т.п.
3. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

## Додаток(а)

1. *#include <stdio.h>*
2. *#include <stdlib.h>*
3. *#include <time.h>*
4. *double \*A(int n) {*
5. *return (double \*)malloc(n\*n\*sizeof(double));}*
6. *void I(int \*a, int n) {*
7. *int i, j;*
8. *for(i = 0; i < n; i++)*
9. *for(j = 0; j < n; j++)*
10. *\*(a + i\*n + j) = rand()%20;}*
11. *void Print(int \*a, int n) {*
12. *int i, j;*
13. *for (i = 0; i < n; i++){*
14. *for (j = 0; j < n; j++)*
15. *printf("%d ", \*(a + i\*n + j));*
16. *printf("\n"); } }*
17. *void Minimum (int \*a, int n) {*
18. *int i, j, min=0;*
19. *for (i = 0; i < n; i++){*
20. *for (j = 0; j < n; j++){*
21. *if(i!=j&&i<j){*
22. *min=\*(a+1);*
23. *if(\*(a + i\*n + j)<min){ min=\*(a + i\*n + j);}*
24. *}}}*
25. *printf("\nMinimum element that is above the main diagonal = %d",min);*
26. *}*
27. *void Sumadiah(int \*a, int n) {*
28. *int i, j, sum=0;*
29. *for (i = 0; i < n; i++){*
30. *for (j = 0; j < n; j++){*
31. *if(i==(n-1)-j){*
32. *sum=sum+\*(a + i\*n + j); }}}*
33. *printf("\nSum of the elements of the lateral diagonal = %d ",sum);}*
34. *int main()*
35. *{ int \*a;*
36. *srand(time(NULL));*
37. *int n;*
38. *printf("Enter n = ");*
39. *scanf("%d", &n);*
40. *printf("Matrix");*
41. *printf("\n");*
42. *a = A(n);*
43. *if (a) {*
44. *I(a, n);*
45. *Print(a, n);*
46. *Minimum(a, n);*
47. *Sumadiah(a, n);*
48. *free(a);}*
49. *return 0;}*

## Додаток(б)



# Завдання 6

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

**6.1**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?

**6.2** 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.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**6.1**

При розв’язанні поставленої задачі були використані наступні засоби:

* оператор вибору if
* динамічне виділення пам‘яті під структуру
* засоби роботи зі структурою та її гарне оформлення

**6.2**

При розв’язанні поставленої задачі були використані наступні засоби:

* оператор вибору if
* засоби роботи зі строками; розглядання їх як і через масив так і з допомогою стандартних функцій

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <time.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

## Висновок

1. **6.1**Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме реалізувати задачу за допомогою структури.
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з структурами роботи із динамічною пам’яттю та елементами і т.п.
3. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

**6.2**

1. **1**Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. В якій ми можемо оперувати інформацією, а саме працювати зі строками.
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з роботою із символьними масивами та строками і т.п.
3. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити строку бажаного розміру і відразу бачити результат.

## Додаток(а)

**6.1**

*#include <time.h>*

*#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 \*/*

*typedef struct {*

*char town\_name[64],candidate\_name[64];*

*}infa;*

*int main(void) {*

*infa \*candidate;*

*int A[100][100];*

*int Array[100],f;*

*int i,j,n,k;*

*printf("Enter n = ");*

*scanf("%d", &n);*

*candidate = (infa \*) calloc (n,sizeof(infa));*

*printf("Enter k = ");*

*scanf("%d", &k);*

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

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

*{ A[i][j] = rand()%150; }}*

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

*printf("Town ");*

*fflush(stdin);*

*gets(candidate[i].town\_name);}*

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

*printf("Name ");*

*fflush(stdin);*

*gets(candidate[i].candidate\_name);}*

*printf("\n \t");*

*for(i=0;i<n+1;i++){*

*printf("\t%s",candidate[i].candidate\_name);}*

*printf("\n");*

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

*{printf("\t%s ",candidate[i].town\_name);*

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

*{*

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

*}*

*printf("\n");*

*}*

*int max=0;*

*int max1=0;*

*int s,d,x,z;*

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

*{*

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

*{*

*if (i == 1 && A[i][j]>max) {max=A[i][j]; s=j;x=i;}*

*if (i == 2 && A[i][j]>max1) {max1=A[i][j]; d=j;z=i;}*

*}*

*}*

*printf("max = %d Candidate - %s ",max,candidate[s].candidate\_name);*

*printf("\nmax = %d Candidate - %s ",max1,candidate[d].candidate\_name);*

*printf("\n");*

*for (f=0;f<2;f++){*

*Array[0]=candidate[s].candidate\_name;*

*Array[1]=candidate[d].candidate\_name;*

*}*

*printf("Array : ");*

*for (f=0;f<2;f++){*

*printf("%s ",Array[f]);*

*}*

*return 0;*

*}*

**6.2**

*#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[]) {*

*char Array[100];*

*char findL='a';*

*int i,k,n;*

*k=0;*

*printf("Enter line:");*

*gets(Array);*

*n=strlen(Array);*

*printf("Line length = %d",n);*

*for(i=0; i<strlen(Array); i++)*

*if(Array[i]==findL) k++;*

*printf("\nQuantity of Letter a = %d", k);*

*return 0;*

*}*

**а)**

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <string.h>*

*int main()*

*{ char PIB[100];*

*int first\_name,last\_name,zero=0,count=0;*

*printf("Type surname name : ");*

*gets(PIB);*

*for (int i=0;i<strlen(PIB);i++)*

*{*

*if (PIB[i]==' ')*

*zero++;*

*if (PIB[i]==' ' && PIB[i+1]!=' ')*

*first\_name=i-1; }*

*if (zero<2)*

*{ printf("\nError");*

*return 1; }*

*int i;*

*for ( i=first\_name;i>0;i--)*

*if (PIB[i]==' ' && PIB[i-1]!=' ')*

*{last\_name=i+1;}*

*for (i=first\_name;i>=last\_name;i--)*

*printf("%c",PIB[i]);*

*return 0;*

*}*

**б)**

*#include <stdio.h>*

*#include <string.h>*

*#define STOP ",. ;:!?"*

*int main()*

*{*

*char line[100], \*potochw, \*last;*

*int i;*

*fgets(line, 100, stdin);*

*for(potochw = strtok(line, STOP), i = 0; potochw; potochw = strtok(NULL, STOP), i++){*

*if(i){*

*last = potochw + strlen(potochw) - 1;*

*while(last >= potochw)*

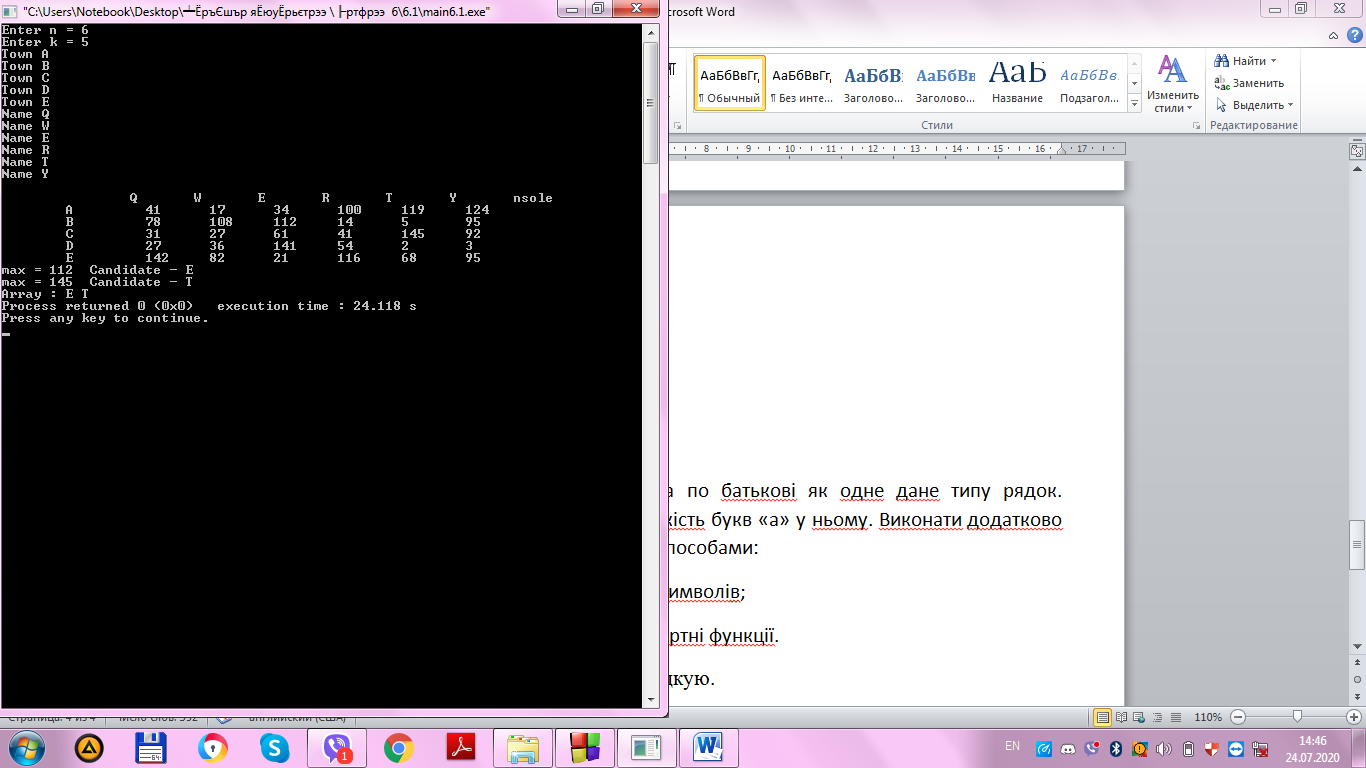
*putchar(\*last--);*

*break;}*

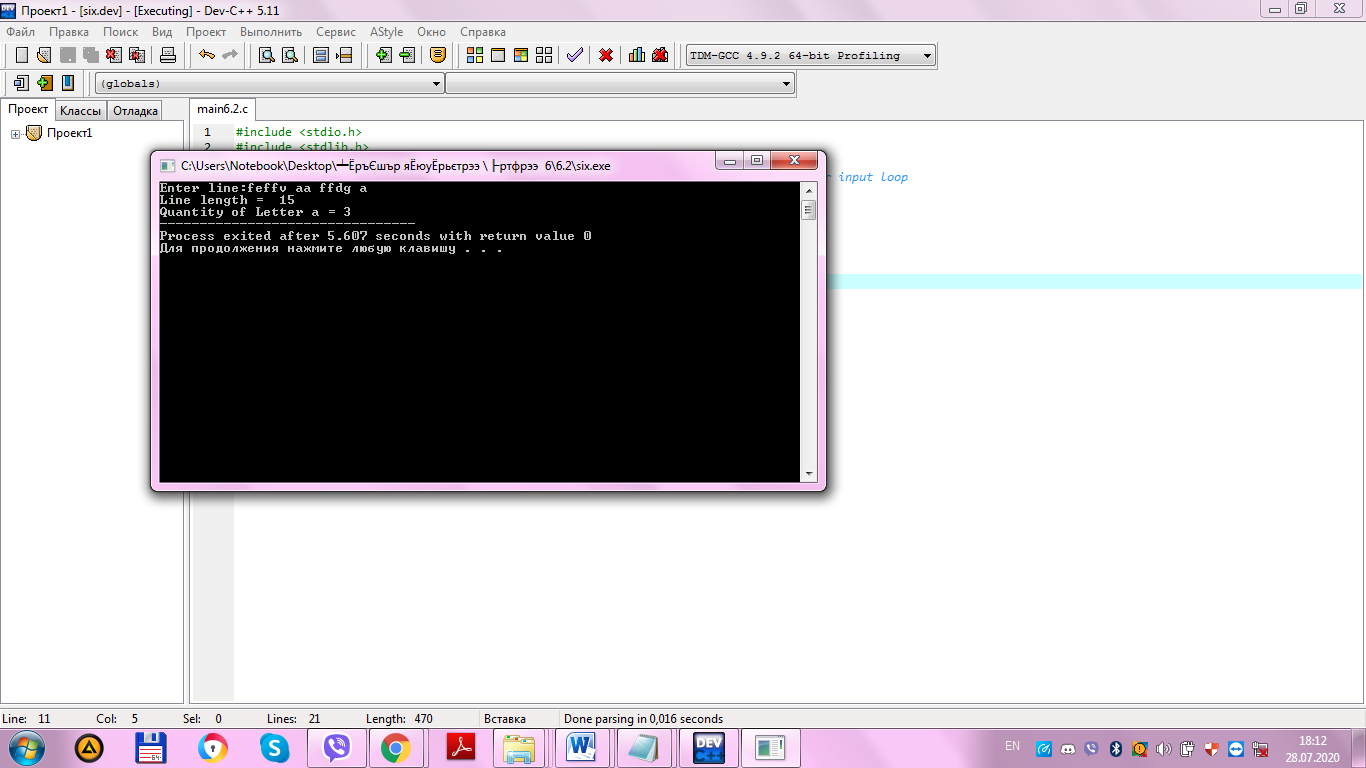
*} return 0; }*

## Додаток(б)

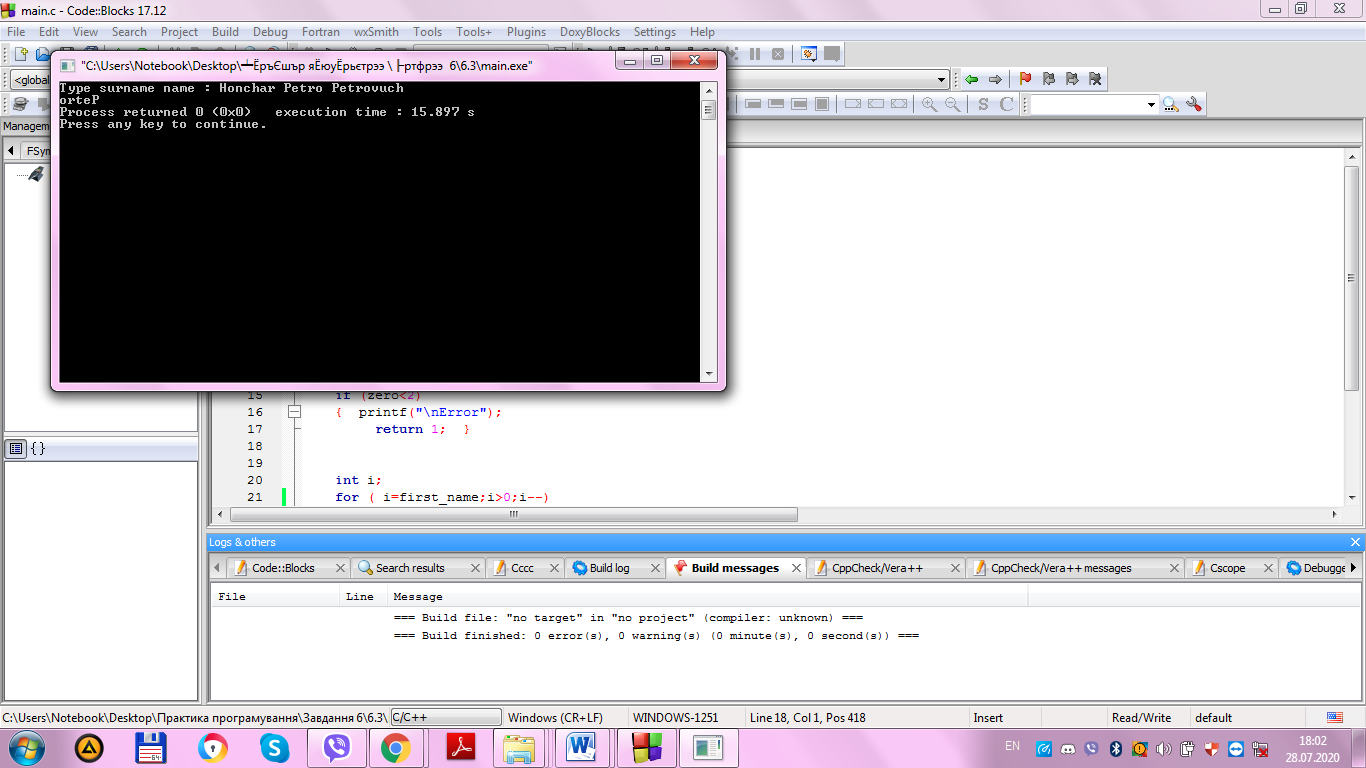
**6.1**



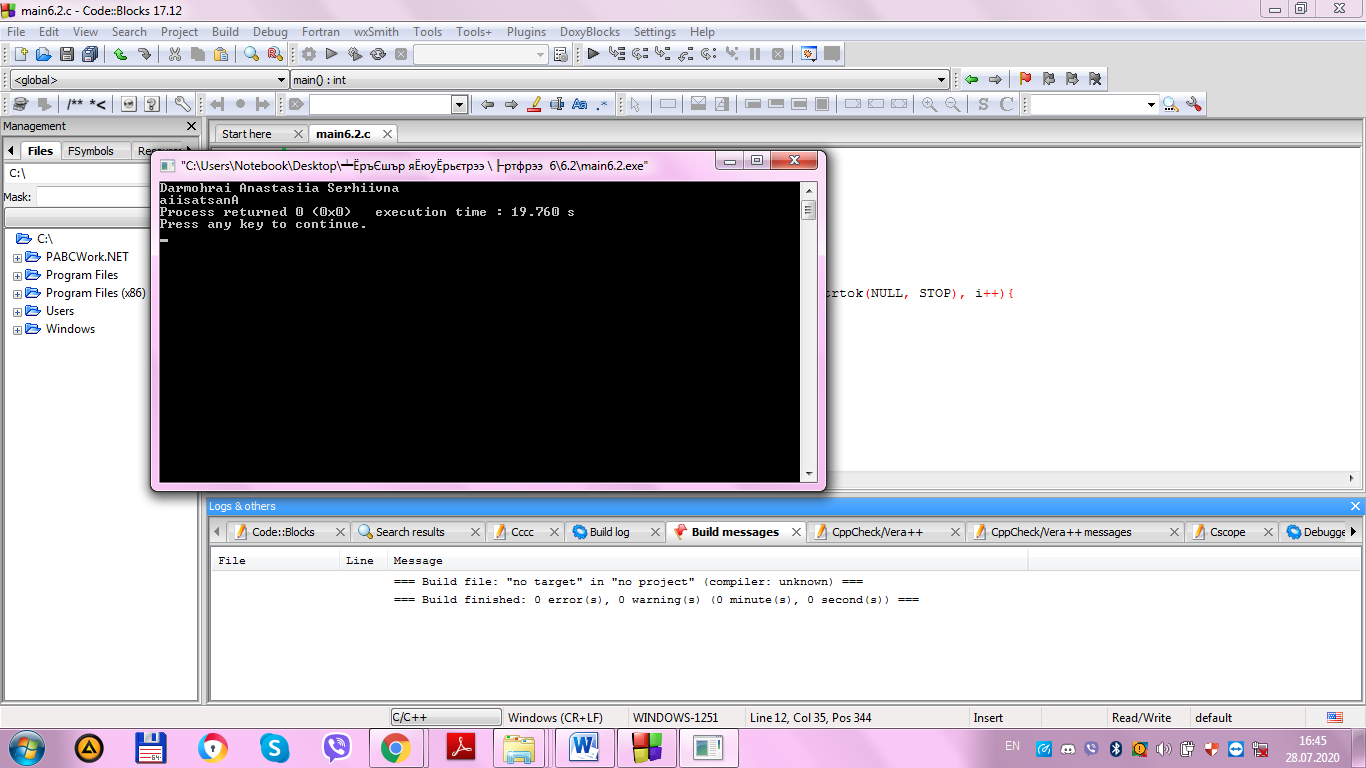
**6.2**

****

**а)**



**б)**



# Завдання 7

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

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

**7.2**Обчислити суму елементів масиву над головною діагоналлю. Визначити індекси мінімального елемента.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

1. **7.1** При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* рандомне заповнення масиву за допомогою функції rand()%
* засоби роботи із динамічним масивом

1. **7.2** При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if для знаходження індексів максимального значення
* засоби роботи із динамічним двовимірним масивом
* використала спосіб сортування бульбашкою

**СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

### Список підключених бібліотек:

1. *#include <stdio.h>*
2. *#include <stdlib.h>*
3. *#include <time.h>*

### Список використаних функцій:

**7.1**

*double \*A(int n) –* динамічне виділення пам’яті для одновимірного масиву

*void Init(int \*a, int n) –* рандомне заповнення масиву

*void print(int \*a, int n) –* виведення масиву на екран

*void F(int \*a, int n) –* функція для знаходження індексу мінімального елемента сортировка масиву за спаданням

**7.2**

*double \*\*Allocate(int m, int n) –* динамічне виділення пам’яті для двовимірного масиву

*void Init(double \*\*a, int m, int n)* *–* рандомне заповнення двовимірного масиву

*void Print(double \*\*a, int m, int n)* *–* виведення масиву на екран у вигляді таблиці

*void My\_func(double \*\*a, int m, int n) –* функція для знаходження індексу мінімального елемента і суми елементів масиву над головною діагоналлю

## Висновок

1. **7.1** Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. У результаті виведено індекс мінімального елементу і відсортований масив.
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з масивами і роботи з ними і т.п.
3. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.
4. **7.1** Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. У результаті виведено індекс мінімального елементу і сума елементів над головною діагоналлю.
5. При створенні даної програми було розглянуто чимало інформації пов’язаної з масивами, динамічним виділенням пам’яті і роботи з ними і т.п.
6. Створено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводити масив бажаного розміру і відразу бачити результат.

## Додаток(а)

**7.1**

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <time.h>*

*double \*A(int n) {*

*return (double \*)malloc(n\*sizeof(double));*

*}*

*void Init(int \*a, int n) {*

*int i;*

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

*\*(a + i) = rand()%13\*1.12;}}*

*void print(int \*a, int n) {*

*int i;*

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

*printf("%d ", \*(a + i));}}*

*void F(int \*a, int n) {*

*int min = 999;*

*int k = 0,i, t,j;*

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

*{*

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

*{*

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

*k = i;*

*}}*

*printf("\n");*

*printf("\nIndex of minimal element = %d",k);*

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

*for (i = 0; i < n-1;i++) // n - размерность массива*

*{*

*if (\*(a+i)<\*(a+i+1)) {*

*t=\*(a+i);*

*\*(a+i) = \*(a+i+1);*

*\*(a+i+1)=t;*

*}}}*

*printf("\nSorted array: ");*

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

*printf("%d ",\*(a+i));}}*

*int main(){*

*int \*a;*

*int n;*

*printf("Enter n = ");*

*scanf("%d", &n);*

*a = A(n);*

*Init(a, n);*

*print(a, n);*

*F(a, n);*

*return 0;*

*}*

**7.2**

*#include<stdio.h>*

*#include<stdlib.h>*

*double \*\*Allocate(int m, int n)*

*{*

*double \*\*a;*

*int i;*

*a = (double \*\*)malloc(m\*n\*sizeof(double) + m\*sizeof(double \*));*

*if (a)*

*{*

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

*a[i] = (double \*)(a + m) + i\*n;*

*return a;*

*}*

*else return NULL;*

*}*

*void Init(double \*\*a, int m, int n)*

*{*

*int i, j;*

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

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

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

*}*

*void Print(double \*\*a, int m, int n)*

*{*

*int i, j;*

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

*{*

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

*printf("%.1f ", a[i][j]);*

*printf("\n");*

*}*

*}*

*void My\_func(double \*\*a, int m, int n) {*

*int i, j,k,f;*

*int min = 999;*

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

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

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

*{*

*min = a[i][j];*

*k = i;*

*f=j;*

*}}}printf("\nIndex of min value = [%d][%d]",k,f);*

*double sum=0;*

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

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

*if(i<j){*

*sum=sum + a[i][j];*

*}}}*

*printf("\nSum of elements above the main diagonal = %.2f ",sum);*

*}*

*int main()*

*{*

*double \*\*a;*

*int m, n;*

*printf("Enter m: ");*

*scanf("%d", &m);*

*printf("Enter n: ");*

*scanf("%d",&n);*

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

*a = Allocate(m, n);*

*if (a)*

*{*

*Init(a, m, n);*

*Print(a, m, n);*

*My\_func(a,m,n);*

*free(a);*

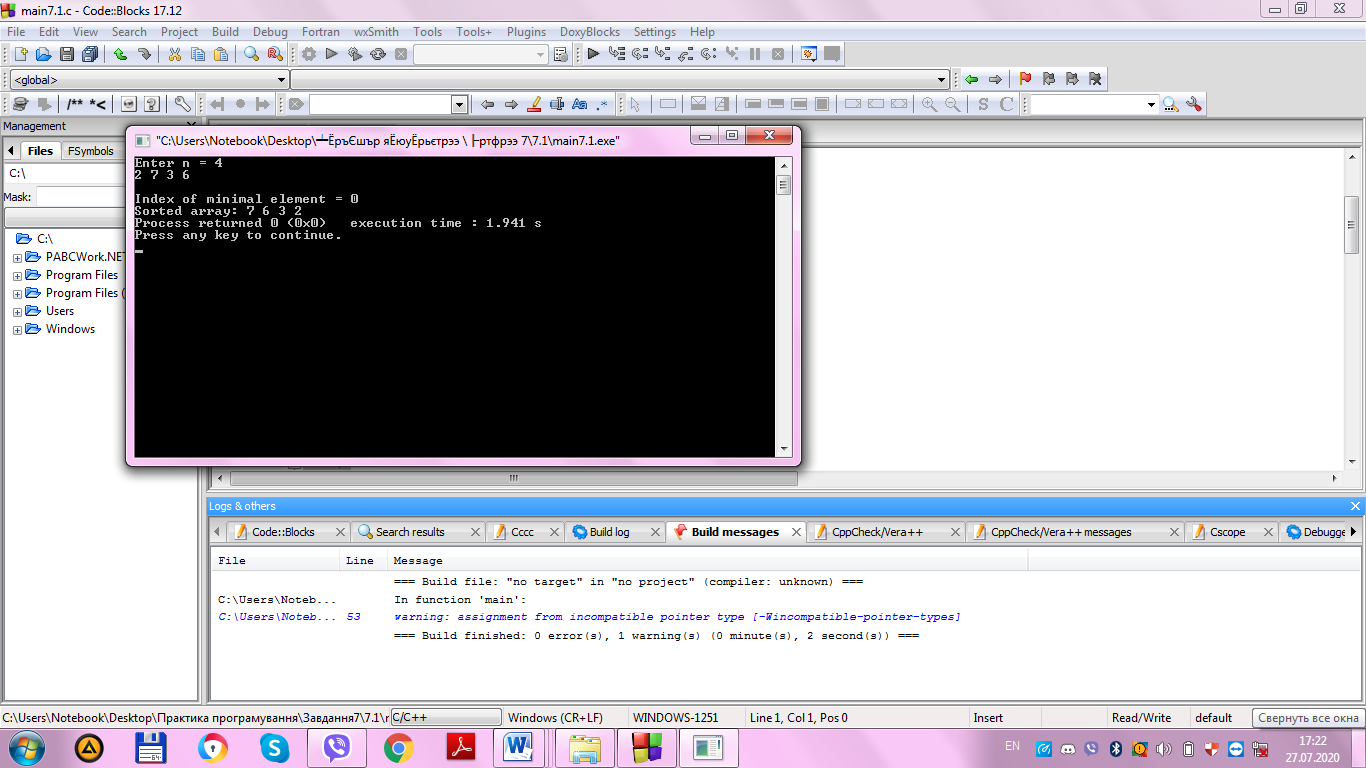
*}*

*return 0;*

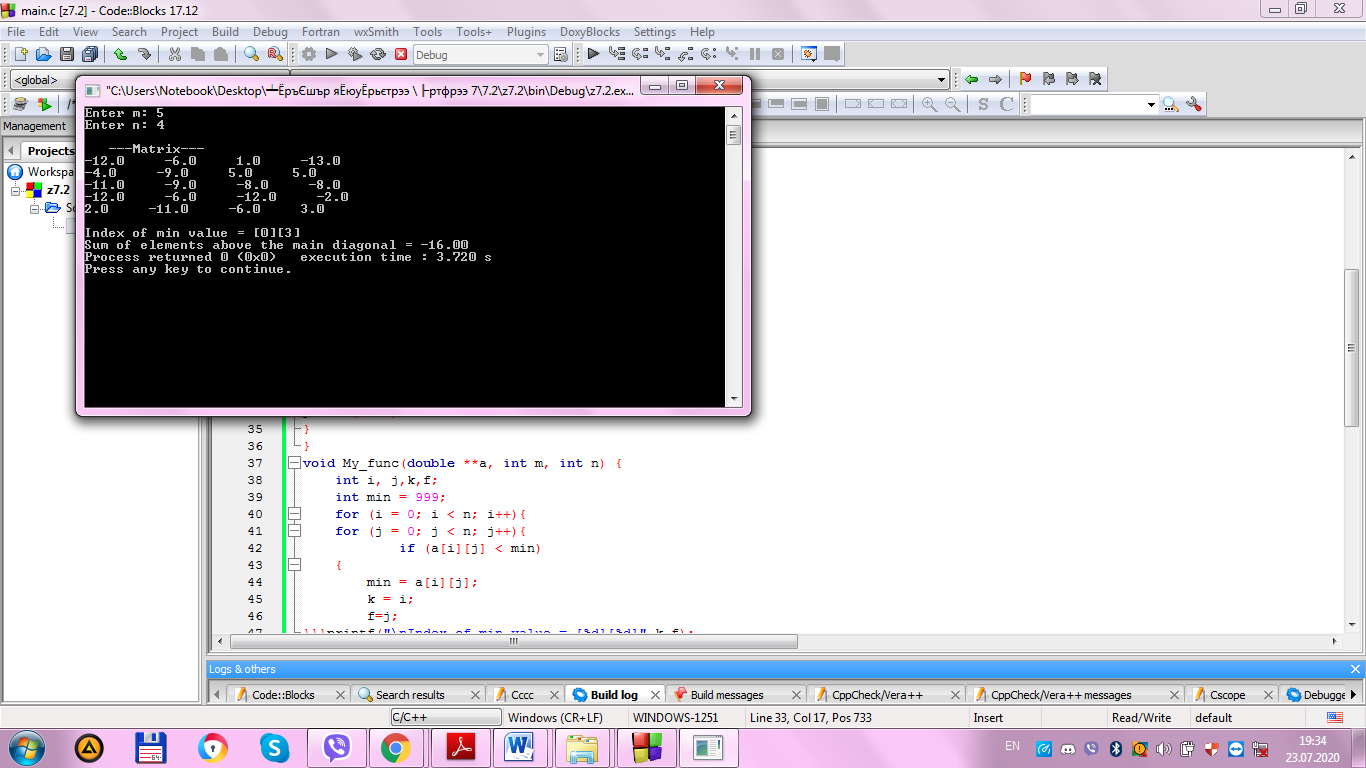
*}*

## Додаток(б)

**7.1**



**7.2**



# Завдання 8

## ПОСТАНОВКА ЗАДАЧІ

### 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

1. При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* знання умови паралельності прямих(геометрія)

**СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>

### Список використаних функцій:

*Paral(int a, int c) –* функція для визначення паралельні прямі чи ні

## Висновок

1. Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. Вона визначає чи паралельні прямі. Для розв’язку цієї задачі розглянула і пригадала матеріал з геометрії і планіметрії.
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з математикою.
3. Розроблено зручний інтерфейс для користувача, за допомогою якого він може комфортно вводить дані і відразу бачить результат.

## 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 \*/*

*Paral(int a, int c){*

*int b, d;*

*if(a/c==b/d||a==c&&b!=d){*

*printf("The lines are parallel");*

*}else*

*printf("The lines are not parallel");*

*}*

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

*int a,b,c,d;*

*printf("We have to lines : \n ax + b=0 and cx + d = 0");*

*printf("\nEnter a = ");*

*scanf("%d",&a);*

*printf("Enter c = ");*

*scanf("%d",&c);*

*printf("Enter b = ");*

*scanf("%d",&b);*

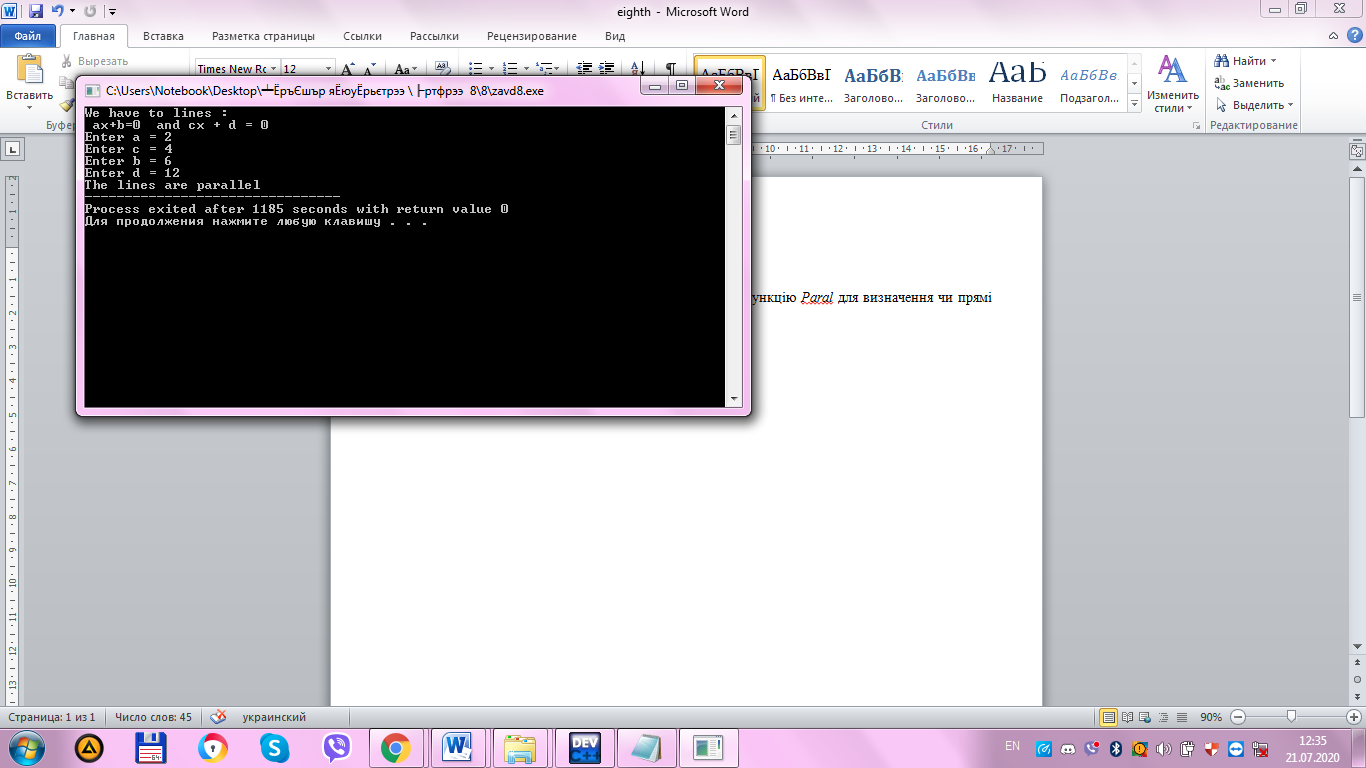
*printf("Enter d = ");*

*scanf("%d",&d);*

*Paral(a,c);*

*return 0;}*

## Додаток(б)

****

# Завдання 9

## ПОСТАНОВКА ЗАДАЧІ

### 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.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

1. При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* засоби роботи із динамічною пам’яттю
* засоби роботи із динамічними масивами

**СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

### Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <time.h>

### Список використаних функцій:

*double \*Allocate(int n)* – виділення динамічної пам’ятф під матрицю

*void Init(int \*a, int n)* – заповнення матриці елементами в межах 9

*void Print(int \*a, int n)* – вивидення матриці на екран

*void Suma(int \*a, int n)* – знаходження діагоналі де максимальна кількість однакових елементів

## Висновок

1. Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. Вона побудована за допомогою динамічної матриці.
2. При створенні даної програми було розглянуто чимало інформації пов’язаної з матрицями та динамічною пам’яттю..

Розроблено зручний інтерфейс для користувача, за допомогою якого він

комфортно вводить дані і відразу бачить результат та швидко може знайти потрібну інформацію.

## Додаток(а)

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <math.h>*

*#include <time.h>*

*double \*Allocate(int n) { return (double \*)malloc(n\*n\*sizeof(double));}*

*void Init(int \*a, int n) {*

*int i,j;*

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

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

*\*(a + i\*n + j) = rand()%9;}}}*

*void Print(int \*a, int n) {*

*int i,j;*

*for(i = 0; i < n; i++){printf("\n");*

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

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

*void Suma(int \*a, int n) {*

*int i,j,z,k1 = 1, k2 = 1, line\_where\_maxkil\_of\_same = -1;;*

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

*{*

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

*{ z = j+1;*

*while (z < n)*

*{*

*if (\*(a + i\*n + j) == \*(a + i\*n + z))*

*{*

*k1++;*

*}*

*z++;*

*}*

*}*

*if (k1 > k2)*

*{*

*k2 = k1;*

*line\_where\_maxkil\_of\_same = i;*

*}*

*k1 = 1;*

*}*

*if (line\_where\_maxkil\_of\_same == -1)*

*{*

*printf("No identical elements");*

*}*

*else*

*{*

*printf("\nIndex = %d",line\_where\_maxkil\_of\_same);*

*}}*

*int main(){*

*int \*a;*

*int n;*

*printf("Enter n:");*

*scanf("%d", &n);*

*a = Allocate(n);*

*if (a) {*

*printf("Matrix:\n");*

*Init(a,n);*

*Print(a,n);*

*Suma(a,n);*

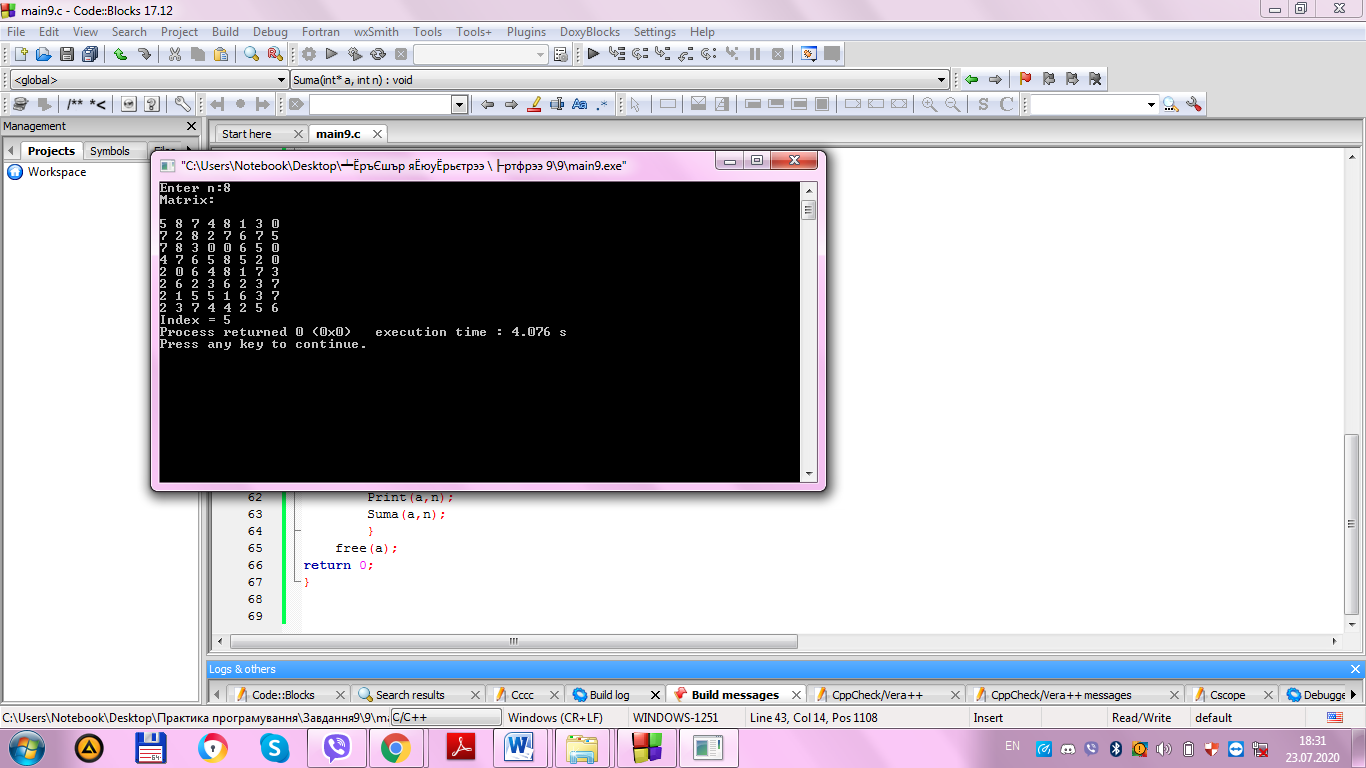
*}*

*free(a);*

*return 0;*

*}*

## Додаток(б)



# Завдання 10

## ПОСТАНОВКА ЗАДАЧІ

### 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

1. При розв’язанні поставленої задачі були використані наступні засоби:

* умовний оператор вибору if
* засоби роботи із структурами
* виділення динамічної пам’яті структурі

**СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

### Список підключених бібліотек:

1. #include <stdio.h>
2. #include <stdlib.h>

#include <string.h>

## Висновок

1. Під час написання цієї роботи було розроблено програму. В результаті виконання отримано працюючу консольну програму. Вона побудована за допомогою структури. ЇЇ можна використовувати при підрахунку середньої вартості та знаходити час звучання.
2. При створенні даної програми було розглянуто чимало інформації пов’язаної зі структурами та динамічною пам’яттю..

Розроблено зручний інтерфейс для користувача, за допомогою якого він

може комфортно вводить дані і відразу бачить результат та швидко може знайти потрібну інформацію.

## Додаток(а)

*#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 \*/*

*typedef struct {*

*char media[64];*

*char name[64];*

*char performer[64];*

*int time, num\_works, price;*

*} musical\_works;*

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

*int i,n=3;*

*musical\_works \*mus\_work;*

*mus\_work = (musical\_works \*) calloc (n,sizeof( musical\_works));*

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

*printf("Enter tipe of media (gramophone record, audio cassette, laser disk):");*

*fflush(stdin);*

*gets(mus\_work[i].media);*

*printf("Enter name of composition: ");*

*fflush(stdin);*

*gets(mus\_work[i].name);*

*printf("Enter performer (surname, name): ");*

*fflush(stdin);*

*gets(mus\_work[i].performer);*

*printf("Enter sound time(minutes) = ");*

*scanf("%d",&mus\_work[i].time);*

*printf("Enter number of works = ");*

*scanf("%d",&mus\_work[i].num\_works);*

*printf("Enter the price according to the catalog:");*

*fflush(stdin);*

*scanf("%d",&mus\_work[i].price);*

*printf("\n");*

*}*

*float suma, average\_value;*

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

*suma = suma + mus\_work[i].price;*

*}*

*average\_value = 1.00\*suma/n;*

*printf("The average price according to the catalog = %.2f", average\_value);*

*int S;*

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

*S=mus\_work[i].num\_works\*mus\_work[i].time;*

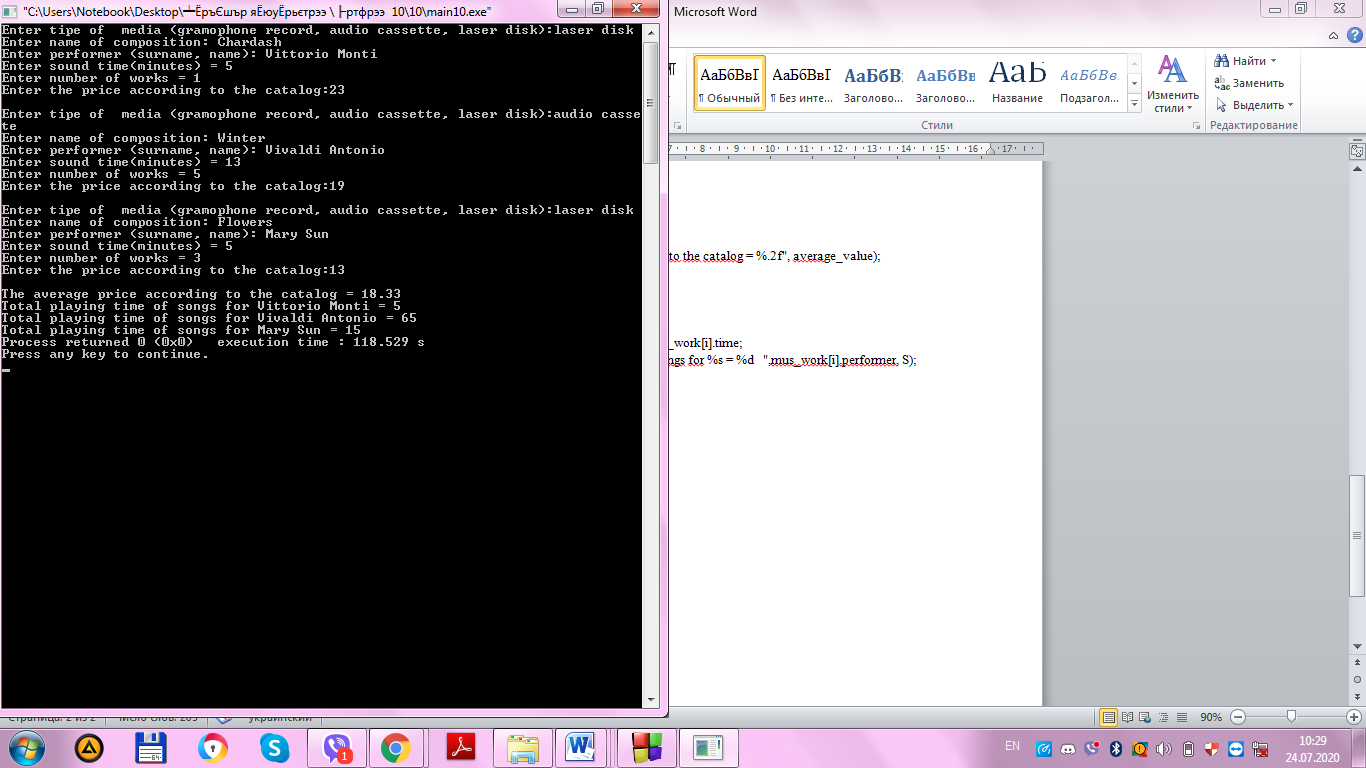
*printf("\nTotal playing time of songs for %s = %d ",mus\_work[i].performer, S);*

*}*

return 0;

}

## Додаток(б)



# Завдання 11

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

**11.1**

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.

**11.2**

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

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**11.1**

При розв’язанні поставленої задачі були використані наступні засоби:

* Оператор вибору Switch.. Оператор Switch я використала для визначення формату номера для введення
* оператор вибору if
* динамічне виділення пам‘яті під структуру
* форматоване введеня і вивидення

**11.2**

При розв’язанні поставленої задачі були використані наступні засоби:

* оператор вибору if для визначення статі
* динамічне виділення пам‘яті під структуру

**СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ**

### Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

### Список використаних функцій:

**11.1**

*void print(sts fil) –* виводить фамілію ім’я та адресу учня

## Висновок

**11.1**

Під час написання цієї роботи було розроблено консольну програму. В якій ми можемо оперувати інформацією, а саме визначати учнів у яких немає телефона або в номері присутня цифра 7.

В ході розробки програми були вивченні та поглиблені знання з динамічної пам‘яті та роботи зі структурами.

Було забезпечено написання базових операцій з даними :

* + Введення даних
  + Виведення всіх даних на екран.

Ця програма може використовуватися вчителями для збору і розподілення контактної інформації про учнів.

**11.2**

Під час написання цієї роботи було розроблено консольну програму. В якій ми можемо оперувати інформацією, а саме знаходити середній ріст саме чоловіків.

В ході розробки прграмибули вивченні та поглиблені знання з динамічної пам‘яті та роботи зі структурами.

Було забезпечено написання базових операцій з даними :

* + Введення даних
  + Виведення всіх даних на екран.

Ця програма може використовуватися вчителями для визначення росту учнів.

## Додаток(а)

**11.1**

*#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 \*/*

*typedef struct {*

*char surname[64];*

*char name[64];*

*char address[64];*

*char tel1[6];*

*char w;*

*int ist;*

*} sts;*

*void print(sts fil)*

*{*

*printf("\nSurname:%s\tname:%s\tAddress:%s ",fil.surname,fil.name,fil.address);*

*}*

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

*int i,j,n,ist;*

*sts \*st; n=4; int k=0;*

*st = (sts \*) calloc (n,sizeof(sts));//виділяю пам'ять під масив зі структурних елементів*

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

*printf("\n");*

*printf("Enter surname: ");*

*fflush(stdin);*

*gets(st[i].surname);*

*printf("Enter name:");*

*fflush(stdin);*

*gets(st[i].name);*

*printf("Enter addresa:");*

*fflush(stdin);*

*gets(st[i].address);*

*printf("Telefone?y/n: ");*

*scanf("%c",&st[i].w);*

*if (st[i].w=='n'){*

*printf("No number\n"); }*

*else{*

*printf("\n");*

*printf("Choose standart of telephone");*

*printf("\n1 - \*\*\*\*\*\*");*

*printf("\n2 - (80372) - \*\*-\*\*-\*\* ");*

*printf("\n3 - \*\*-\*\*-\*\*");*

*printf("\n");*

*scanf( "\n%d", &ist );*

*switch ( ist ) {*

*case 1:*

*printf("number: \*\*\*\*\*\* \n");*

*scanf("%s",&st[i].tel1);*

*break;*

*case 2:*

*printf("number: (80372)- \*\*-\*\*-\*\* \n"); printf("(80372) - ");*

*scanf("%s",&st[i].tel1);*

*break;*

*case 3:*

*printf("number: \*\*-\*\*-\*\* \n");*

*scanf("%s",&st[i].tel1);*

*break;*

*}*

*printf("\n");}*

*}*

*printf("\nStudents with no phone:\n");*

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

*if (st[i].w=='n')*

*print(st[i]); }*

*printf("\n");*

*printf("\nStudents with 7 in phone:\n");*

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

*for (j=0;j<10;j++) {*

*if (st[i].tel1[j] == '7'){*

*print(st[i]);*

*}}}*

*printf("\n");*

*return 0;}*

**11.2**

*#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 \*/*

*typedef struct {*

*float height;*

*char s;*

*}man;*

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

*man \*men;*

*int i, n=22, k=0;*

*float ser;*

*men = (man \*) calloc (n,sizeof(man));*

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

*{*

*printf("Enter height = ");*

*scanf("%f",&men[i].height );*

*printf("Enter pol(m/f) = ");*

*fflush(stdin);*

*scanf("%c",&men[i].s );*

*}*

*float sum=0;*

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

*if ( men[i].s == 'm'){*

*sum = sum + men[i].height;*

*k++;}}*

*printf("Suma = %.2f Quantity = %d",sum,k);*

*ser =(sum/k);*

*if (k==0){*

*printf("No men");}*

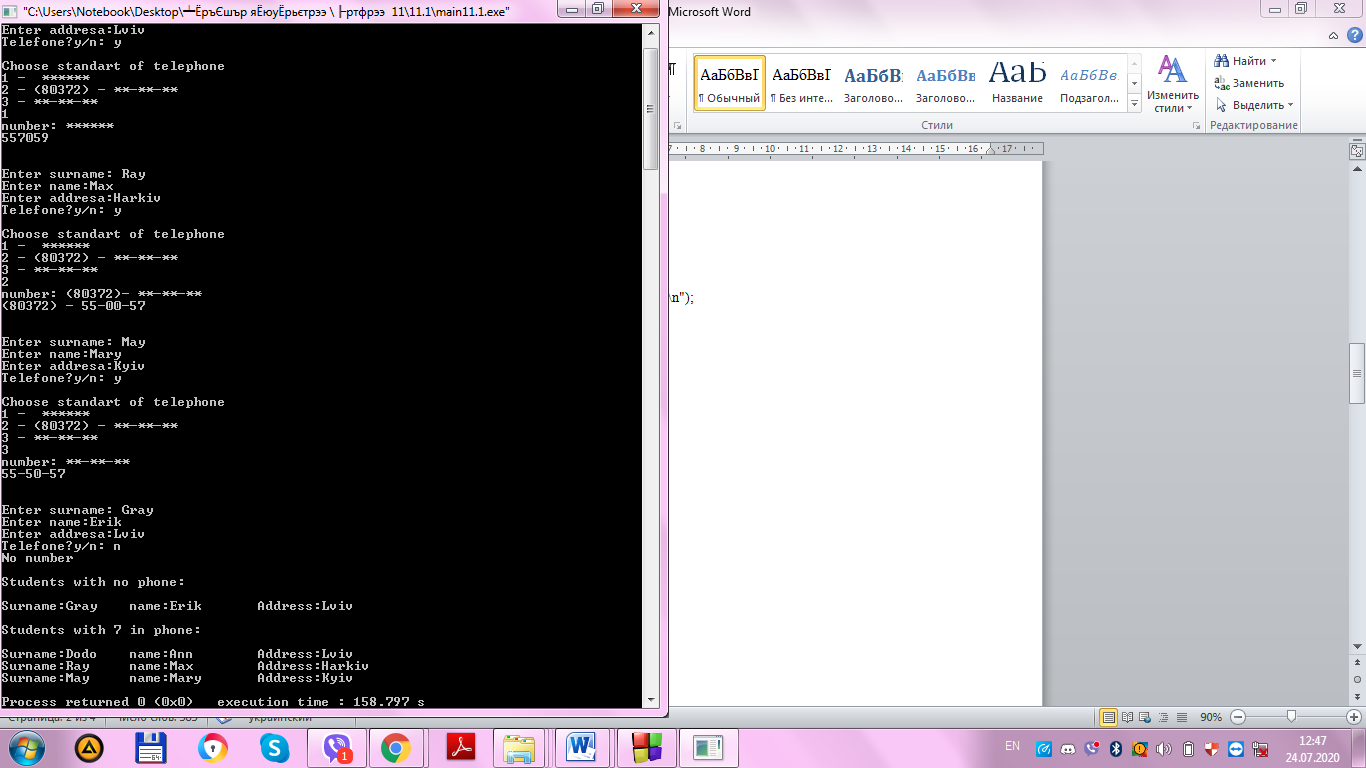
*else*

*printf("\nAverage height of men = %.2f", ser);*

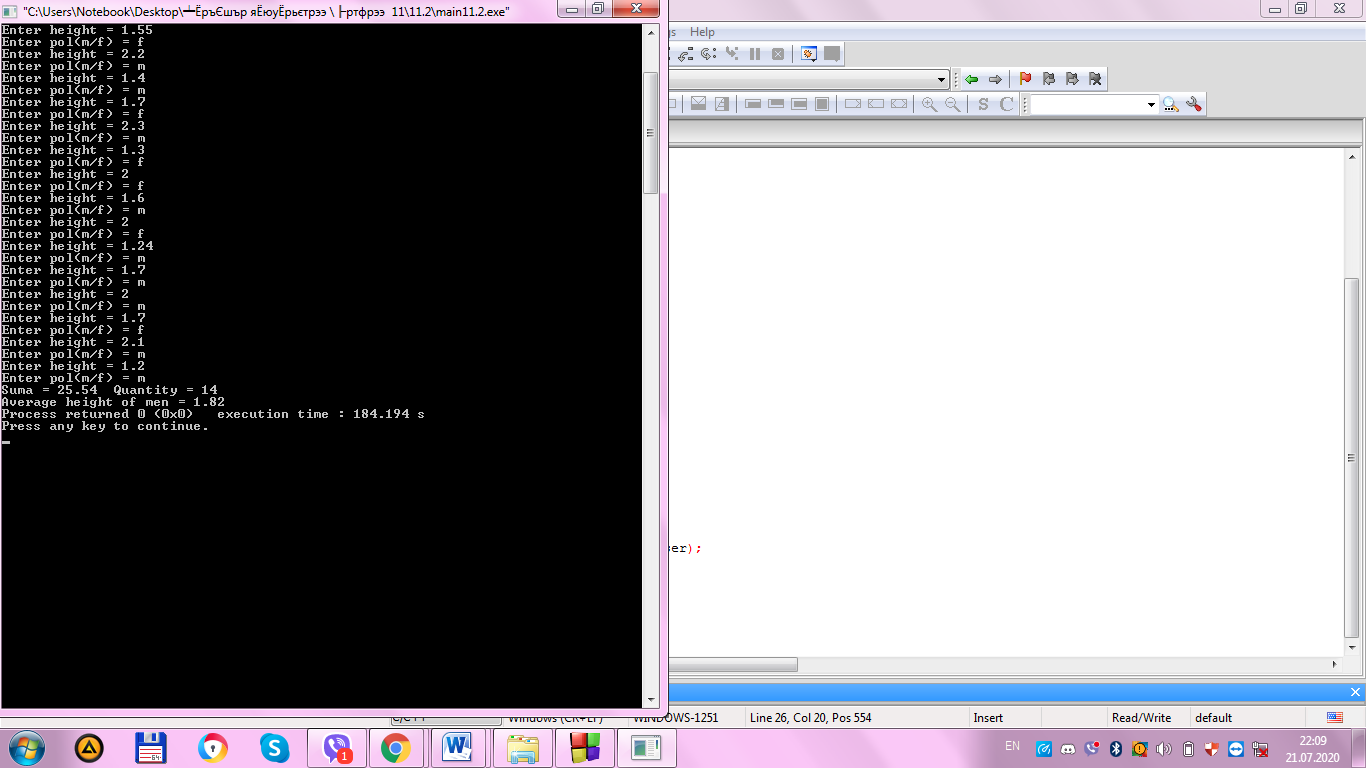
*return 0;}*

## Додаток(б)

**11.1**



**11.2**



# Завдання 12

## ПОСТАНОВКА ЗАДАЧІ

### 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

При розв’язанні поставленої задачі були використані наступні засоби:

* оператор вибору if
* динамічне виділення пам‘яті під структуру та елементи списку
* форматоване введеня і вивидення

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

#include <string.h>

## Висновок

Під час написання цієї роботи було розроблено консольну програму. В якій ми можемо оперувати інформацією, а саме знаходити кількість різних елементів списку.

В ході розробки прграмибули вивченні та поглиблені знання з динамічної пам‘яті та роботи зі структурами та списками.

Було забезпечено написання базових операцій з даними :

* + Введення даних
  + формаування даних
  + Виведення всіх даних на екран.

## Додаток(а)

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <time.h>*

*#include <math.h>*

*#include <string.h>*

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

*typedef struct Node {*

*char value;*

*struct Node \*next;*

*};*

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

*int listSize = 0, uniqueAmount = 0, i = 0;*

*Node \*head = NULL, \*iterator = NULL, \*tmpPtr = NULL, \*uniqueHead = NULL;*

*printf("Input N:\n");*

*scanf("%d", &listSize);*

*while (listSize < 1) {*

*printf("Size of list should be greater than 0. Try again.\n");*

*scanf("%d", &listSize);*

*}*

*printf("Type list of %d characters:\n", listSize);*

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

*tmpPtr = (Node\*) malloc(sizeof(Node));*

*tmpPtr->next = NULL;*

*scanf("%s", &(tmpPtr->value));*

*if (head == NULL) {*

*head = tmpPtr;*

*} else {*

*iterator->next = tmpPtr;*

*}*

*iterator = tmpPtr;*

*}*

*iterator = head;*

*while (iterator != NULL) {*

*if (uniqueHead == NULL) {*

*uniqueHead = (Node\*) malloc(sizeof(Node));*

*uniqueHead->next = NULL;*

*uniqueHead->value = iterator->value;*

*uniqueAmount++;*

*iterator = iterator->next;*

*continue;*

*}*

*tmpPtr = uniqueHead;*

*while (tmpPtr != NULL) {*

*if (tmpPtr->value == iterator->value || tmpPtr->next == NULL) { break; }*

*tmpPtr = tmpPtr->next;*

*}*

*if (tmpPtr->value != iterator->value) {*

*tmpPtr->next = (Node\*) malloc(sizeof(Node));*

*tmpPtr->next->value = iterator->value;*

*tmpPtr->next->next = NULL;*

*uniqueAmount++;*

*}*

*iterator = iterator->next;*

*}*

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

*while (head != NULL) {*

*printf("%c ", head->value);*

*iterator = head;*

*head = head->next;*

*free(iterator);*

*}*

*printf("\n\n------Unique Characters List-------\n");*

*while (uniqueHead != NULL) {*

*printf("%c ", uniqueHead->value);*

*iterator = uniqueHead;*

*uniqueHead = uniqueHead->next;*

*free(iterator);*

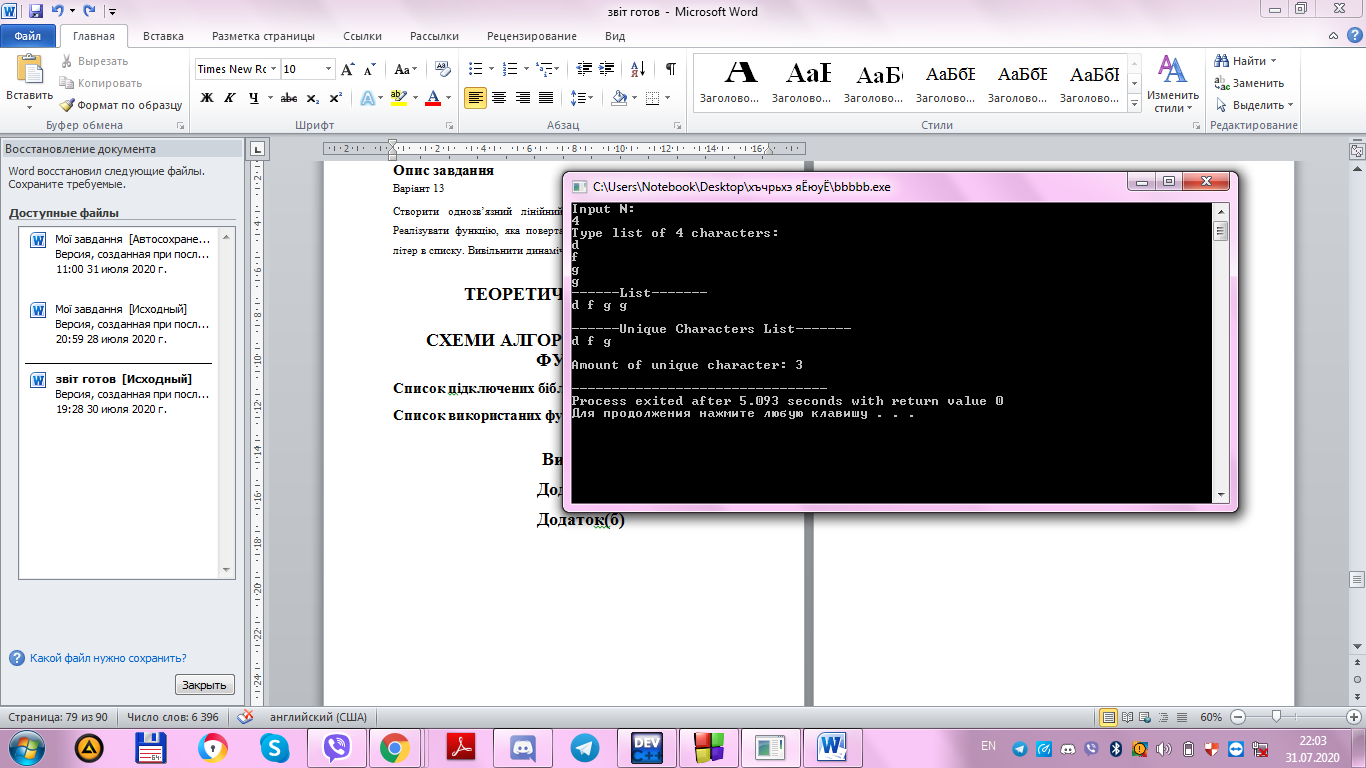
*}*

*printf("\n\nAmount of unique character: %d \n", uniqueAmount);*

*return 0;*

*}*

## Додаток(б)



# Завдання 13

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

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

**13.2** Розробити дві консольні 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 читаютє з файлу, результат записується у новий файл. Ім’я файлів передаються через командний рядок, або вводяться з консолі.

Задано два тексти, слова в яких розділені пробілами і розділовими знаками. Розробити програму, яка створює третій текст із слів першого тексту, які не входять у другий текст, розділяючи їх пробілами.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

* оператор вибору if
* функції роботи з файлами
* засоби роботи із сроками в файлах
* стандартні функції роботи зі строками

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <stdio.h>

#include <string.h>

## Висновок

Під час написання цієї роботи було розроблено консольні програми. В яких ми можемо оперувати інформацією, а саме працювати із файлами та строками що записані у них.

В ході розробки програми були вивченні та поглиблені знання для робот з файлами і строками.

Було забезпечено написання базових операцій з даними :

* + Введення даних
  + Виведення всіх даних на екран.

## Додаток(а)

## 13.1

*#include <stdio.h>*

*#include <string.h>*

*#define N 100*

*main() {*

*FILE \*f;*

*char s[N],a[N];*

*unsigned short ff, i;*

*f = fopen("text.txt","r");*

*printf("Array: ");*

*while (fgets(s, N, f) != NULL) {*

*if (s[i] == ' ') ff = 0;*

*a[i]=strlen(s);*

*printf("%d ",a[i]-1);*

*}*

*fclose(f);*

*}*

**13.2**

*#include <stdio.h>*

*#include <string.h>*

*#include <stdlib.h>*

*#define MAXCHAR 1000*

*typedef struct Character {*

*char value;*

*Character \*next;*

*};*

*typedef struct Node {*

*struct Character \*value;*

*struct Node \*next;*

*int size = 0;*

*int hashCode = 0;*

*};*

*int getHashcode(Character \*character, int size) {*

*int hash = 0; // added for clarity*

*int i = 0;*

*while (character != NULL) {*

*hash = 31 \* hash + character->value;*

*character = character->next;*

*}*

*return hash;*

*}*

*Node\* composeWord(Character \*characterHead) {*

*Character \*iterator = characterHead;*

*int wordSize = 0;*

*while (iterator != NULL) {*

*wordSize++;*

*iterator = iterator->next;*

*}*

*Node \*node = (Node\*) malloc(sizeof(Node));*

*node->value = characterHead;*

*node->size = wordSize;*

*node->hashCode = getHashcode(characterHead, wordSize);*

*node->next = NULL;*

*return node;*

*}*

*void printWord(Node \*node, FILE \*fp) {*

*Character \*iterator = node->value;*

*char chars[node->size];*

*int i = 0;*

*while (iterator != NULL) {*

*chars[i++] = iterator->value;*

*iterator = iterator->next;*

*}*

*fprintf(fp, chars);*

*fprintf(fp, " ");*

*}*

*Node\* readFile(char\* filename) {*

*FILE \*fp;*

*Node \*head = NULL, \*iterator = NULL, \*tmpPtr = NULL;*

*char ch;*

*int a = 97, A = 65, z = 122, Z = 90;*

*fp = fopen(filename, "r");*

*if (fp == NULL) {*

*printf("Could not open file %s", filename);*

*return NULL;*

*}*

*Character \*character = NULL, \*characterHead = NULL;*

*Node \*wordHead = NULL, \*word = NULL;*

*Character \*nextCharacter = (Character\*) malloc(sizeof(Character));*

*while ((nextCharacter->value = fgetc(fp)) != EOF) {*

*if ((nextCharacter->value >= A && nextCharacter->value <= Z) || (nextCharacter->value >= a && nextCharacter->value <= z)) {*

*if (characterHead == NULL) {*

*characterHead = nextCharacter;*

*character = characterHead;*

*character->next = NULL;*

*nextCharacter = (Character\*) malloc(sizeof(Character));*

*} else {*

*character->next = nextCharacter;*

*character->next->next = NULL;*

*character = character->next;*

*nextCharacter = (Character\*) malloc(sizeof(Character));*

*}*

*} else if (characterHead != NULL) {*

*if (wordHead == NULL) {*

*wordHead = composeWord(characterHead);*

*word = wordHead;*

*} else {*

*word->next = composeWord(characterHead);*

*word = word->next;*

*}*

*characterHead = NULL;*

*}*

*}*

*if (characterHead != NULL) {*

*if (wordHead == NULL) {*

*wordHead = composeWord(characterHead);*

*word = wordHead;*

*} else {*

*word->next = composeWord(characterHead);*

*word = word->next;*

*}*

*characterHead = NULL;*

*}*

*return wordHead;*

*}*

*bool equals(Node \*a, Node \*b) {*

*return a->hashCode == b->hashCode;*

*}*

*int main()*

*{*

*Node \*source = readFile("input1.txt");*

*Node \*stopWords = readFile("input2.txt");*

*Node \*differentHead = NULL, \*differentWord = NULL;*

*Node \*sourceIterator = source, \*stopIterator = stopWords;*

*bool equal = false;*

*FILE \*fp = fopen("output.txt", "w");*

*while (sourceIterator != NULL) {*

*while (stopIterator != NULL) {*

*if (equals(sourceIterator, stopIterator)) {*

*equal = true;*

*break;*

*}*

*stopIterator = stopIterator->next;*

*}*

*stopIterator = stopWords;*

*if (!equal) {*

*printWord(sourceIterator, fp);*

*} else {*

*equal = false;*

*}*

*sourceIterator = sourceIterator->next;*

*}*

*fclose(fp);*

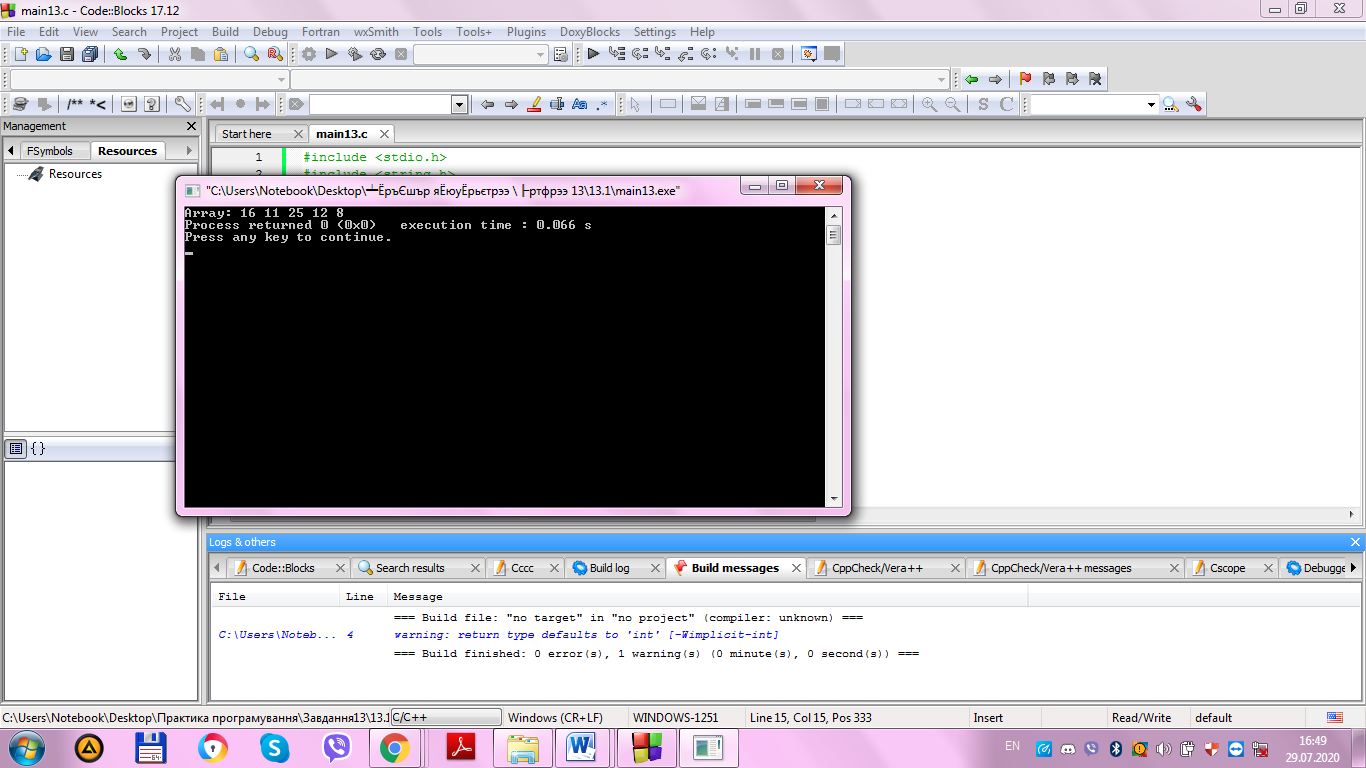
*return 0;*

*}*

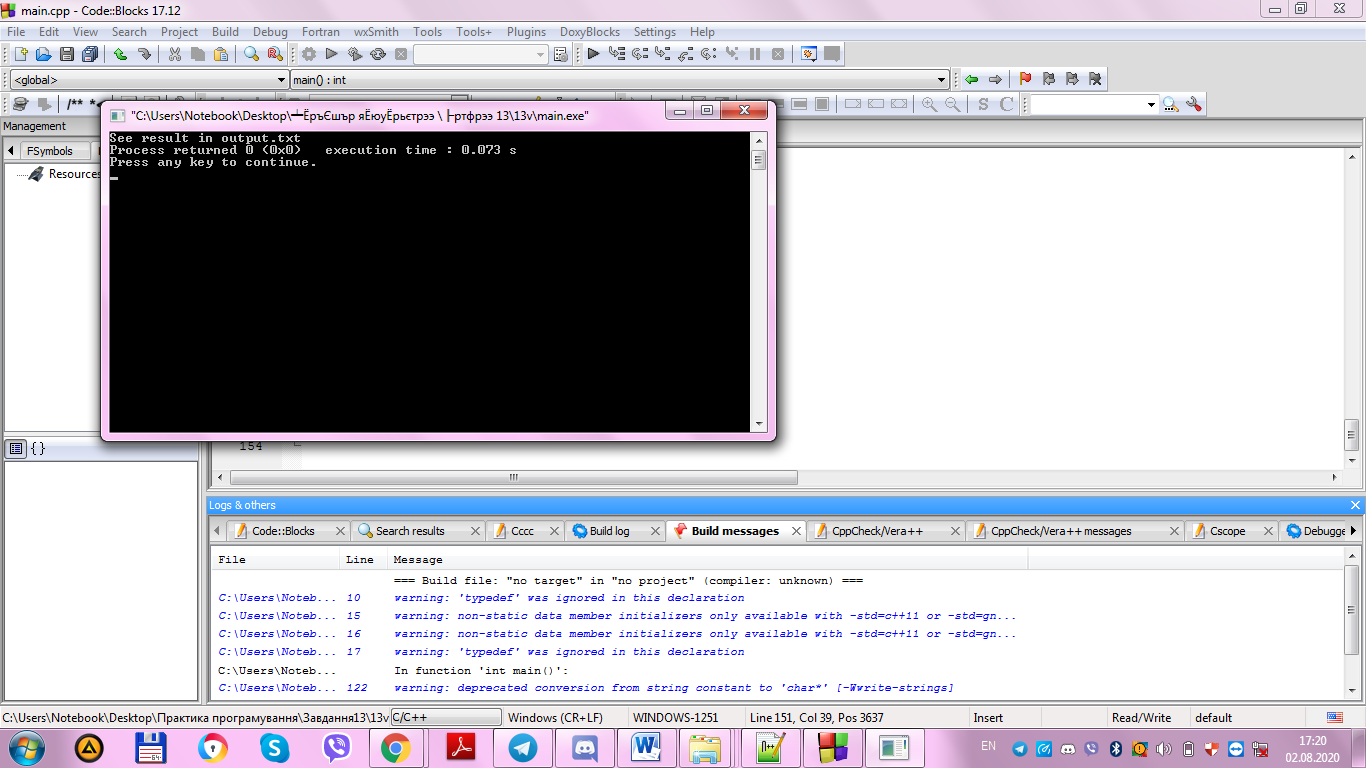
## 

## Додаток(б)

## 13.1



## 13.2



# Завдання 14

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Варіант 13

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

**14.2** 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.

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

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

## ТЕОРЕТИЧНІ ВІДОМОСТІ

При розв’язанні поставленої задачі були використані наступні засоби:

* оператор вибору if
* функції роботи з файлами
* засоби роботи із сроками в файлах
* стандартні функції роботи зі строками

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

### Список підключених бібліотек:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

## Висновок

Під час написання цієї роботи було розроблено консольну програму. В якій ми можемо оперувати інформацією, а саме працювати із файлами та строками що записані у них.

В ході розробки програми були вивченні та поглиблені знання для робот з файлами і строками.

Було забезпечено написання базових операцій з даними :

* + Введення даних
  + Виведення всіх даних на екран.

## Додаток(а)

**14.1**

*#include <stdio.h> #include <stdlib.h>*

*#include <string.h>*

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

*{ char S1[20];*

## *char S2[20];*

## *int i, j;*

## *printf("Enter the line: ");*

## *gets(S1);*

## *j=0;*

## *for(i = 0; i<strlen(S1); i++)*

## *if(S1[i] != ' ') {*

## *S2[j++] = S1[i]; }*

## *int k=0;*

## *printf("\nConverted line:");*

## *for(i =0; i<strlen(S2)-1; i++) {*

## *if (S2[i] == 'a'||S2[i] == 'e'||S2[i] == 'i'||S2[i] == 'o'||S2[i] == 'u')*

## *{ printf("%c%c",S2[i],S2[i]); }*

## *else printf("%c",S2[i]);*

## *} return 0; }*

**14.2**

**а.)**

*#include <string.h>*

*#include <stdio.h>*

*int main (void)*

*{*

*FILE \*in;*

*FILE \*out;*

*int k=0, j,i;*

*char s[100] ;*

*in = fopen ("in.txt","r");*

*out = fopen ("5.txt","w");*

*while(1){*

*if(NULL!= fgets(s,100,in)){ fprintf(out,"\n");for(i=0;i<strlen(s);i++)*

*{ if(s[i]!=' '){k++;}else{k=0;}*

*if((s[i]=='a')||(s[i]=='c')||(s[i]=='d')||(s[i]=='i')||(s[i]=='e')){fprintf(out," ");*

*for (j=i-k+1;j<strlen(s);j++) {if(s[j]!=' '){fprintf(out,"%c",s[j]);}else{i=j;j=strlen(s);}*

*}*

*}*

*}*

*} else{break;}}*

*printf ("Already compiled \n");*

*return 0;}*

**б)**

*#include <string.h>*

*#include <stdio.h>*

*int main (void)*

*{*

*FILE \*in;*

*FILE \*out;*

*int k=0, j,i;*

*char s[256] ;*

*in = fopen ("in.txt","r");*

*out = fopen ("6.txt","w");*

*while(1){*

*if(NULL!= fgets(s,100,in)){ fprintf(out,"\n");for(i=0;i<strlen(s);i++)*

*{ if(s[i]!=' '){k++;}else{k=0;}*

*if((s[i]=='1')||(s[i]=='2')||(s[i]=='3')||(s[i]=='4')||(s[i]=='6')||(s[i]=='7')||*

*(s[i]=='8')||(s[i]=='9')||(s[i]=='<')|| (s[i]=='=')||(s[i]=='>')){fprintf(out," ");*

*for (j=i-k+1;j<strlen(s);j++) {if(s[j]!=' '){fprintf(out,"%c",s[j]);}else{i=j;j=strlen(s);}*

*}*

*}*

*}*

*}*

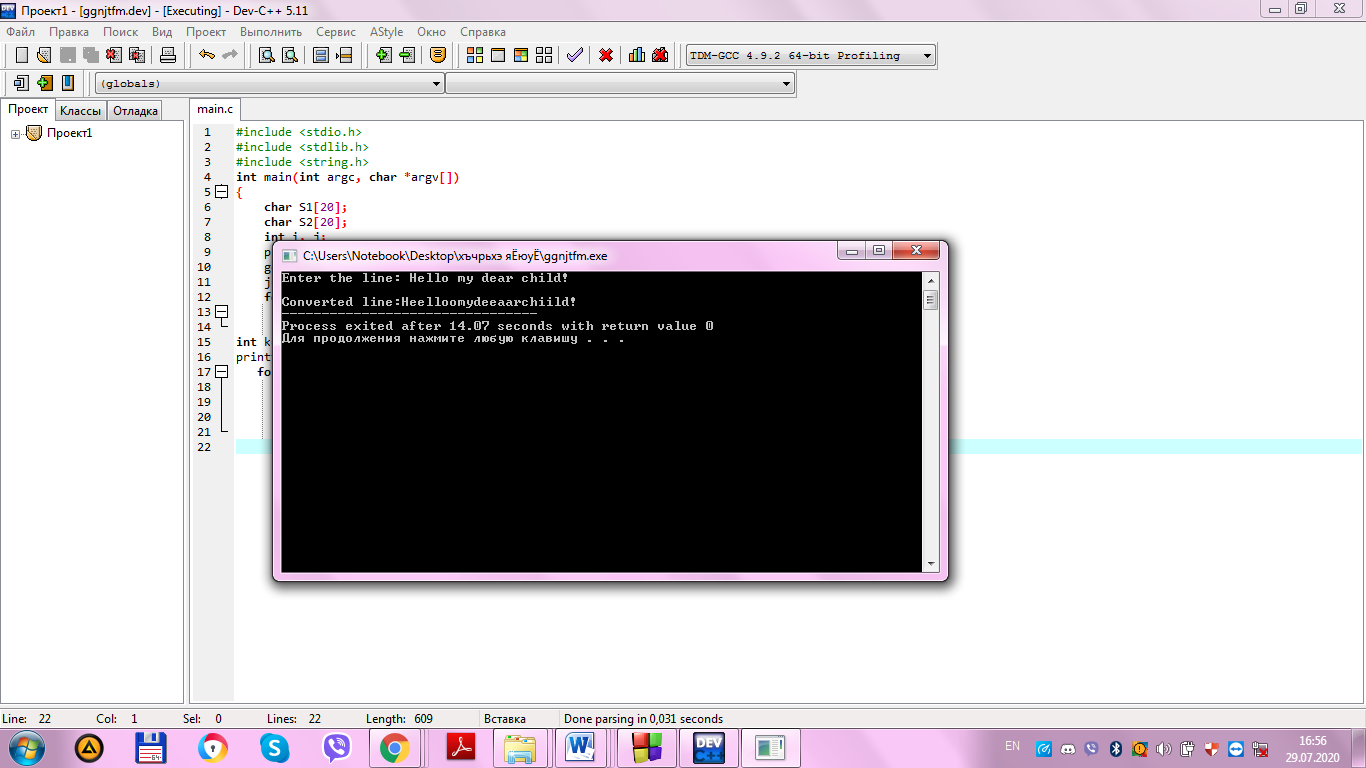
*else{break;}}*

*printf ("Already compiled \n");*

*return 0;}*

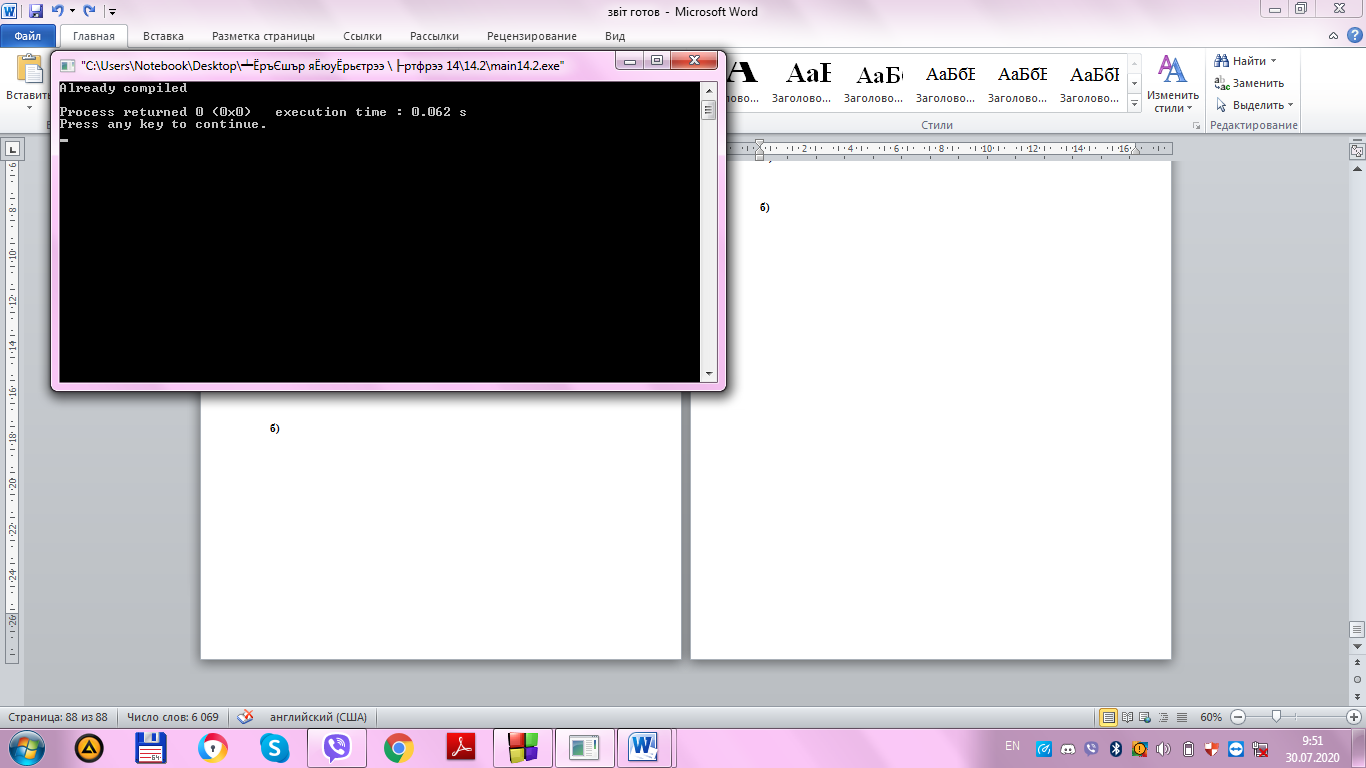
## Додаток(б)

**14.1**



**14.2**

**а.)**



**б)**

