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

*Паскар Дмитро В’ячеславович*

Період практики з “20 липня” 2020 р. по “7 серпня” 2020р.

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

[Лабораторна 1 7](#_Toc46936881)

[Завдання 1 8](#_Toc46936882)

[ПОСТАНОВКА ЗАДАЧІ 8](#_Toc46936883)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 8](#_Toc46936884)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 8](#_Toc46936885)

[Висновок 8](#_Toc46936886)

[Лабораторна 2 10](#_Toc46936887)

[Завдання 1 11](#_Toc46936888)

[ПОСТАНОВКА ЗАДАЧІ 11](#_Toc46936889)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 11](#_Toc46936890)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 11](#_Toc46936891)

[Висновок 11](#_Toc46936892)

[Завдання 2 12](#_Toc46936893)

[ПОСТАНОВКА ЗАДАЧІ 12](#_Toc46936894)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 12](#_Toc46936895)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 12](#_Toc46936896)

[Висновок 12](#_Toc46936897)

[Лабораторна 3 14](#_Toc46936898)

[Завдання 1 15](#_Toc46936899)

[ПОСТАНОВКА ЗАДАЧІ 15](#_Toc46936900)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 15](#_Toc46936901)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 15](#_Toc46936902)

[Висновок 15](#_Toc46936903)

[Завдання 2 16](#_Toc46936904)

[ПОСТАНОВКА ЗАДАЧІ 16](#_Toc46936905)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 16](#_Toc46936906)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 16](#_Toc46936907)

[Висновок 16](#_Toc46936908)

[Завдання 3 17](#_Toc46936909)

[ПОСТАНОВКА ЗАДАЧІ 17](#_Toc46936910)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 17](#_Toc46936911)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 17](#_Toc46936912)

[Висновок 17](#_Toc46936913)

[Лабораторна 4 19](#_Toc46936914)

[Завдання 1 20](#_Toc46936915)

[ПОСТАНОВКА ЗАДАЧІ 20](#_Toc46936916)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 20](#_Toc46936917)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 20](#_Toc46936918)

[Висновок 20](#_Toc46936919)

[Завдання 2 21](#_Toc46936920)

[ПОСТАНОВКА ЗАДАЧІ 21](#_Toc46936921)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 21](#_Toc46936922)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 21](#_Toc46936923)

[Висновок 21](#_Toc46936924)

[Лабораторна 5 23](#_Toc46936925)

[Завдання 1 24](#_Toc46936926)

[ПОСТАНОВКА ЗАДАЧІ 24](#_Toc46936927)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 24](#_Toc46936928)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 24](#_Toc46936929)

[Висновок 24](#_Toc46936930)

[Лабораторна 6 26](#_Toc46936931)

[Завдання 1 27](#_Toc46936932)

[ПОСТАНОВКА ЗАДАЧІ 27](#_Toc46936933)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 27](#_Toc46936934)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 27](#_Toc46936935)

[Висновок 27](#_Toc46936936)

[Завдання 2 28](#_Toc46936937)

[ПОСТАНОВКА ЗАДАЧІ 28](#_Toc46936938)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 28](#_Toc46936939)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 28](#_Toc46936940)

[Висновок 29](#_Toc46936941)

[Лабораторна 7 30](#_Toc46936942)

[Завдання 1 31](#_Toc46936943)

[ПОСТАНОВКА ЗАДАЧІ 31](#_Toc46936944)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 31](#_Toc46936945)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 31](#_Toc46936946)

[Висновок 31](#_Toc46936947)

[Завдання 2 32](#_Toc46936948)

[ПОСТАНОВКА ЗАДАЧІ 32](#_Toc46936949)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 32](#_Toc46936950)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 32](#_Toc46936951)

[Висноков 33](#_Toc46936952)

[Лабораторна 8 34](#_Toc46936953)

[Завдання 1 35](#_Toc46936954)

[ПОСТАНОВКА ЗАДАЧІ 35](#_Toc46936955)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 35](#_Toc46936956)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 35](#_Toc46936957)

[Висновок 35](#_Toc46936958)

[Лабораторна 9 37](#_Toc46936959)

[Завдання 1 38](#_Toc46936960)

[ПОСТАНОВКА ЗАДАЧІ 38](#_Toc46936961)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 38](#_Toc46936962)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 38](#_Toc46936963)

[Висновок 38](#_Toc46936964)

[Лабораторна 10 40](#_Toc46936965)

[Завдання 1 41](#_Toc46936966)

[ПОСТАНОВКА ЗАДАЧІ 41](#_Toc46936967)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 41](#_Toc46936968)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 41](#_Toc46936969)

[Висновок 41](#_Toc46936970)

[Лабораторна 11 43](#_Toc46936971)

[Завдання 1 44](#_Toc46936972)

[ПОСТАНОВКА ЗАДАЧІ 44](#_Toc46936973)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 44](#_Toc46936974)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 44](#_Toc46936975)

[Висновок 45](#_Toc46936976)

[Завдання 2 48](#_Toc46936977)

[ПОСТАНОВКА ЗАДАЧІ 48](#_Toc46936978)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 48](#_Toc46936979)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 48](#_Toc46936980)

[Висноков 48](#_Toc46936981)

[Лабораторна 12 50](#_Toc46936982)

[Завдання 1 51](#_Toc46936983)

[ПОСТАНОВКА ЗАДАЧІ 51](#_Toc46936984)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 51](#_Toc46936985)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 51](#_Toc46936986)

[Висновок 51](#_Toc46936987)

[Лабораторна 13 55](#_Toc46936988)

[Завдання 1 56](#_Toc46936989)

[ПОСТАНОВКА ЗАДАЧІ 56](#_Toc46936990)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 56](#_Toc46936991)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 56](#_Toc46936992)

[Висновок 56](#_Toc46936993)

[Завдання 2 58](#_Toc46936994)

[ПОСТАНОВКА ЗАДАЧІ 58](#_Toc46936995)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 58](#_Toc46936996)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 58](#_Toc46936997)

[Висноков 58](#_Toc46936998)

[Лабораторна 14 60](#_Toc46936999)

[Завдання 1 61](#_Toc46937000)

[ПОСТАНОВКА ЗАДАЧІ 61](#_Toc46937001)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 61](#_Toc46937002)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 61](#_Toc46937003)

[Висновок 61](#_Toc46937004)

[Завдання 2 62](#_Toc46937005)

[ПОСТАНОВКА ЗАДАЧІ 62](#_Toc46937006)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 62](#_Toc46937007)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 62](#_Toc46937008)

[Висновок 63](#_Toc46937009)

[Завдання 3 65](#_Toc46937010)

[ПОСТАНОВКА ЗАДАЧІ 65](#_Toc46937011)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 65](#_Toc46937012)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 65](#_Toc46937013)

[Висновок 65](#_Toc46937014)

# 

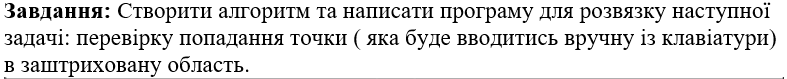
# Лабораторна 1

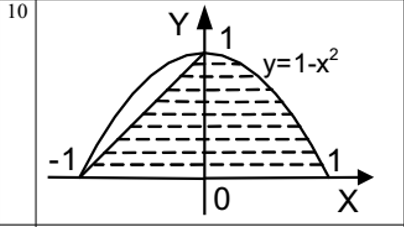
## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання





### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв'язвання задачі я скористався стандартними функціями введення-виведення(scanf,printf) для введення та виведення даних та умовним оператором if…else для визначення чи належить введена точка площі яка дана за умовою задачі.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

##### Підключені бібліотеки

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

### Висновок

Програма була успішно написана і може визначати чи належить введена точка площі

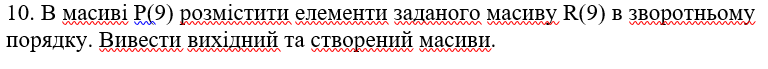
# 

# Лабораторна 2

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв'язвання задачі я скористався циклом for, функцію srand для того, щоб кожного запуску масиви заповнювалися новими випадковими числами.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdlib.h>  
   #include <stdio.h>  
   #define N 9

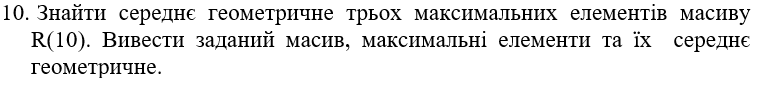
### Висновок

Програгама успішно записую і виводить перший масив у зворотньому порядку

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Заповнював масив за допомогою циклу for та за допомогою функції псевдовипадкових чисел rand та функцією srand для того, щоб кожного запуску масиви заповнювалися новими випадковими числами.Також використовував умовні оператори if…else.

### 

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

#include <stdlib.h>  
#include <stdio.h>  
#include <stdbool.h>  
#include <math.h>  
#include <time.h>  
#define N 10

### Висновок

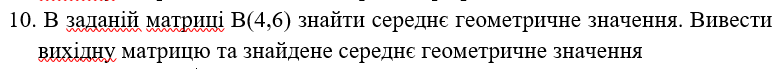
Програма успішно знаходить 3 максимальні числа та рахує їх середнє геометричне.

# Лабораторна 3

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я використав цикл for, стандартну функцію виведення та функцію pow для піднесення до степеня.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdlib.h>  
   #include <stdio.h>  
   #include <math.h>  
   #include <time.h>  
   #define N 4  
   #define M 6

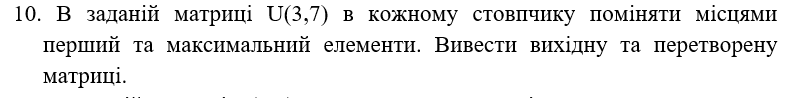
### Висновок

Програма успішно виводить середнє геометричне значення матриці

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’звання цієї задачі я використав цикл for та стандартну функцію виведення printf.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

#include <stdlib.h>  
#include <stdio.h>  
#include <math.h>  
#include <time.h>  
#define N 4  
#define M 6

### Висновок

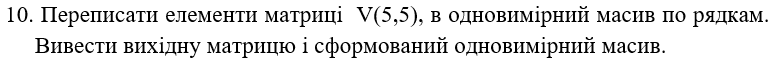
Програма успішно міняє місцями перший та максимальний елемент

##### Додаток(б)

## Завдання 3

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Для розв’звання цієї задачі я використав цикл for та стандартну функцію виведення printf.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

#include <stdlib.h>  
#include <stdio.h>  
#include <time.h>  
#define N 5

### Висновок

Програма успішно переписує вміст матриці у одновимірний масив.

# 

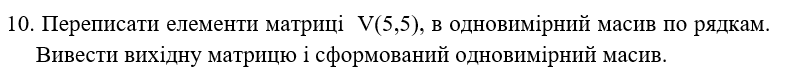
# Лабораторна 4

## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Для розв’звання цієї задачі я використав цикл for та стандартну функцію виведення printf.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdlib.h>  
   #include <stdio.h>  
   #include <time.h>  
   #define N 5

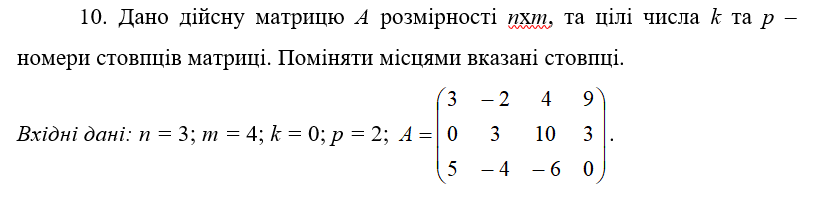
### Висновок

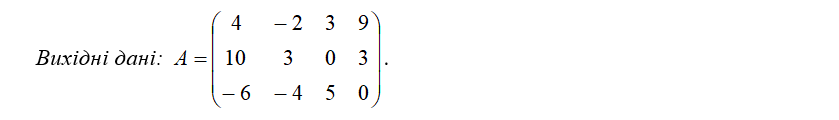
1. Програма успішно переписує вміст матриці у одновимірний масив.

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання





### 

### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для того, щоб поміняти місцями стовпці я скористався циклом for та на кожній ітерації міняв місцями елементи відповідних рядків

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

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

### Висновок

Програма успішно виводить спочатку масив який заповнюється випадковими числами потім вона виводить другий масив у якого вказані вище стовпці

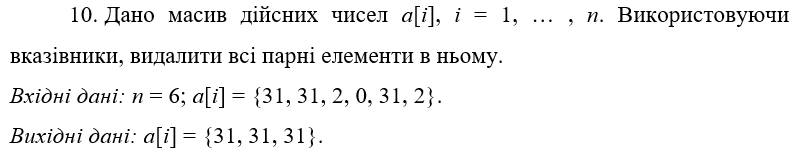
# Лабораторна 5

## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Я використав вказівники для переміщення по масиву. Якщо у на є масив int A[20] і ми захочимо, наприклад, надрукувати перший елемент то нам досить написати printf(“%d”,\*A)(\* - розіменування, щоб отримати значення) тобто наш масив і є вказівником, а щоб вивести інші елементи треба до вказівника додати зміщення тому, щоб вивести другий елемент треба до вказівника додати 1: \*A + 1.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

##### Підключені бібліотеки

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

### Висновок

Програма успішно видаляє всі парні елементи у масиві.

# 

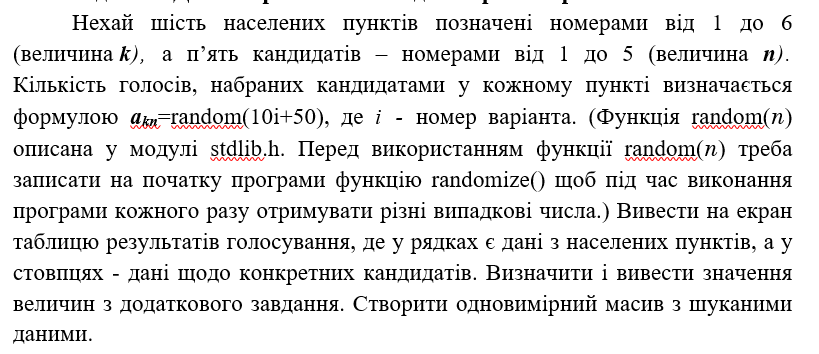
# Лабораторна 6

## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання





### ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Для розв’звання цієї задачі я використав звичайний алгоритм знаходження найменшого елемента Sum = 2000000 тому що це число буде точно більше за всі «голоси» за якого відддали виборці, далі порівнюєм голоси і записуєм id кондидата

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

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

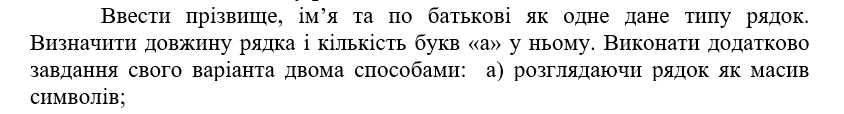
### Висновок

1. Програма успішно знаходить число голосів яке було віддано кандидатів та знаходить кандидата, який набрав менше всього голосів.

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання





### ТЕОРЕТИЧНІ ВІДОМОСТІ

Рядок у сі – масив символів тому можна перевіряти кожний символ

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

#include <stdio.h>  
#include <string.h>  
#include <stdbool.h>

Використані функції

**Strpbrk** – знаходить перше входження симовла у рядок.

### Висновок

Програма успішно виводить скільки букв «а» є у рядку та його дожину. Двома способами визначає чи починається хоч одне слово у рядку букою «М»

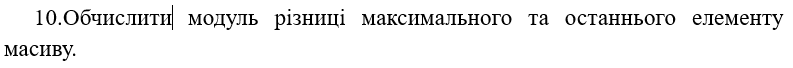
# 

# Лабораторна 7

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для вирішення цієї задачі я використав функції, кожна з яких отримувала вказівник на масив після чого виконувала свою функцію

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdlib.h>  
   #include <stdio.h>  
   #include <math.h>  
   #include <time.h>

###### Використані функції

filling – заповнення масива випадковими числами.

module – повертає модуль різниці максимального та останнього чисел масива.

abs – модуль

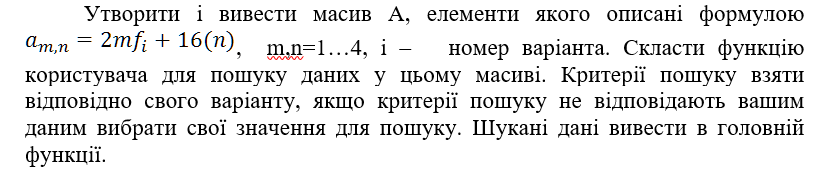
### Висновок

Програма успішно знаходить максимальний елемент, його різницю з останнім елементом та за допомогою функції abs – модуль.

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання





### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для вирішення цієї задачі я скористався динамічним двомірним масивом. Щоб його створити треба спочатку створити вказівник на вказівник який буде масивом, потім кожен елемент якого стає також масивом.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

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

Використані функції

filling – заповнення.

product – рахує добуток чисел менше 10 та більше двох.

### Висноков

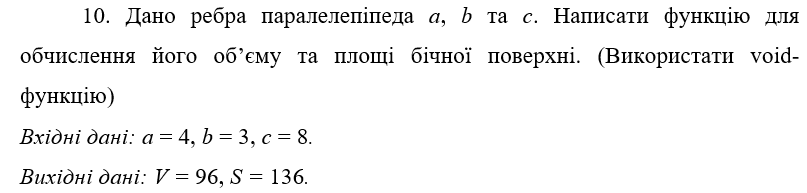
Програма успішно знаходить слова, які більші за 2 та менші за 10, перемножує їх та виводить результат у консоль.

# Лабораторна 8

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я скористався void-функціями (функції, які нічого не повертають) для обчислення та виведення результату

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdio.h>

###### Використані функції

volume – визначення і виведення у консоль об’єм.

area – визначення площі бічної поверхні.

### Висновок

Програма успішно рахує та виводить об’єм та площу бічної поверхні через void-фнкції

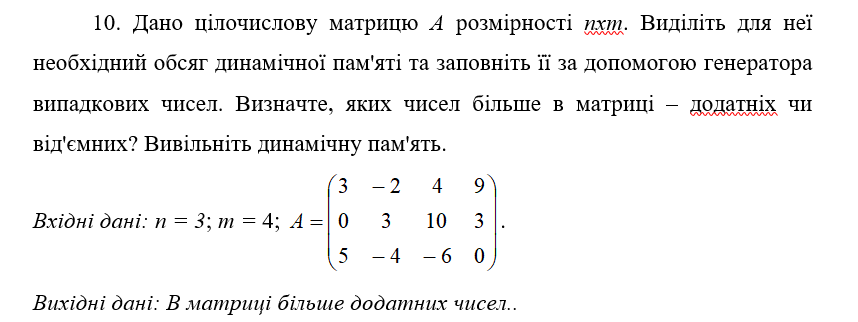
# 

# Лабораторна 9

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я скористався функцією malloc. За допомогою цієї функції я створив динамічний масив з N елементів. Далі кожен елемент цього масиву сам став масивом, таким чином і вийшла динамічна матриця. За допомогою команди free я спочатку вивільнив кожен елемент першого масиву потім, потім і сам масив.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

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

###### Використані функції

malloc – створення динамічного масиву

free – вивільнення динамічної пам’яті.

### Висновок

Програма успішно створює динамічний масив, заповнює його за допомогою функції псевдовипадкових чисел, рахує додатні та від’ємні порівнює їх та виводить остаточний результат.

# Лабораторна 10

## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання

1. 10. Є інформація про телевізори:
2. – фірма-виробник;
3. – розмір екрану в дюймах;
4. – вартість у гривнях.
5. У структурі створити функцію, що визначає, чи є у списку інформація про телевізор вартістю не більше 5,5 тис. грн. і розміром екрана щонайменше 17 дюймів. Якщо так, то вивести всі відомості про нього.

### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я скористався структурами. У структурі я вписав всі необхідні поля. Далі створив масив структур у якому користувач через консоль вводить інформацію N телевізорів. Був використаний прототип функцій для того тіло функції я зміг написати під функцією main.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

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

###### Використані функції

outTV – виводить назву ціну та розмір телевізорів ціна яких менша за 5500 грн., а розмір менший за 17 дюймів.

### Висновок

Програма успішно аналізує дані які ввів користувач і виводить телевізори, які коштують менше 5500 грн,і розміром більше 17 дюймів.

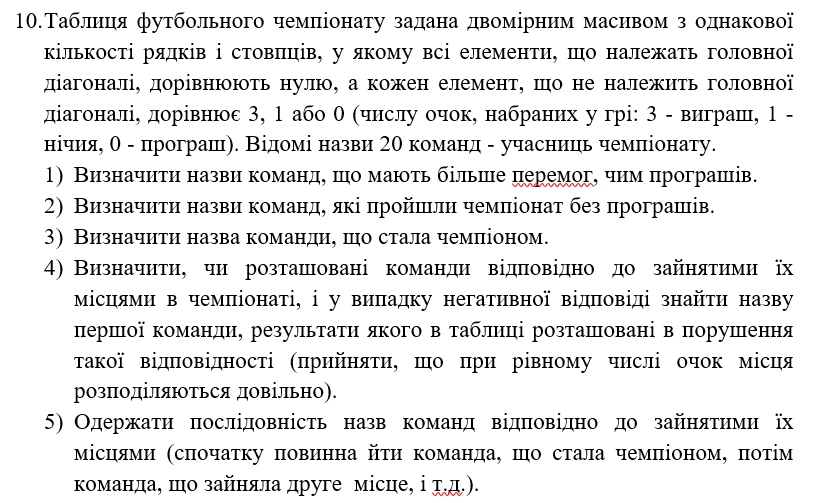
# Лабораторна 11

## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я заповнив масив випадковими числами від 0 до 3, але якщо випадкове число 2 починається цикл у якому змінній надається знову і знову випадкові значення поки воно буде відмінне від 2.Якщо ітератор першого циклу дорівнює ітератору другого то значення надається 0, бо коли i = j – це головна діагональ матриці. Для знаходження чемпіона турніру треба знайти суму очок кожної команди і порівняти їх щоб отримати послідовність команд за рейтингом я скористався сортуванням бульбашкою порівнюючи сумарну кількість очок кожної команди.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdlib.h>  
   #include <stdio.h>  
   #include <string.h>  
   #include <stdbool.h>  
   #include <time.h>  
   #define N 20

###### Використані функції

strcpy – функція, яка копіює рядок з одного масиву символів в інший.

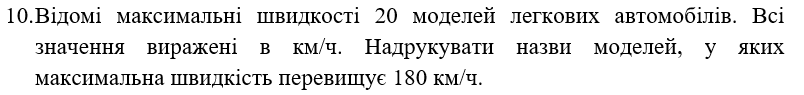
### Висновок

Програма успішно знаходить всі потрібні значення

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я використав структуру де у полях записав назву та швидкість. У функцію out як аргумент я подаю масив структур де він перебирається і друкується.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

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

Використані функції

out – друкує назви машин у яких швидкість перевищу 180 км.

### Висноков

Програма успішно знаходить і друкує машини у яких максимальна швидкість більша ніж 180 км/год через функцію out.

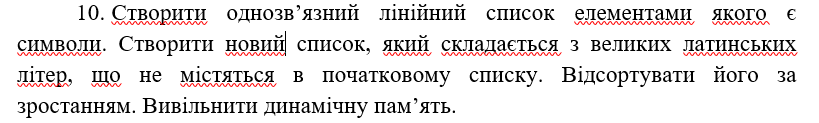
# Лабораторна 12

## 

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання задачі я скористався однозв’язним списком: список кожен елемент якого містить у собі вказівник на наступний а останній елемент має нульовий вказівник. Для того щоб додати елемент треба виділити місце під вказівник за допомогою malloc чи calloc запивнити та вказати вказівник на елемент який ми вказали останнім, потім записати вказівник на нього як перший елемент, щоб потім ми могли вільно перебігатись по ньому.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

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

###### Використані функції

push – додати новий елемент у список.

printList – вивести список.

deleteList – видалити(вивільнити) список.

length – повертає кількість елементів у списку

fillingIn – записує дані з списка у масив.

fillingout – записує дані з масива у список.

Free ­– вивільнення динамічної пам’яті.

### Висновок

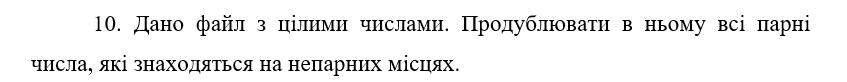
Програма успішно створює два списки в кожному з яких є ті символи, яких немає у іншому

# Лабораторна 13

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі треба використати функції які дають змогу працювати з файлами. Fopen має декілька режимів роботи, я використав r+: режим у якому можна читати файл і додавати якісь записи.Також не треба забувати закривати файл, це може призвести до поганих наслідків.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

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

###### Використані функції

atoi – функція, яка як аргумент ми подаємо адрес символа, а повертає число типу int.

Fopen – функція для того, щоб відкрити файл.

Rewind ­– переміщує курсор на початок файлу

Fclose – закриває файл

Fgetc – зчитує файл посимвольно.

Fprintf – записує у файл дані.

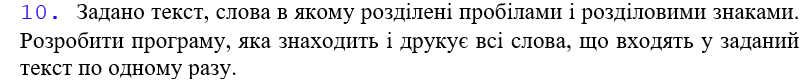
### Висновок

Програма успішно зчитує та записує у файл всю необхідну інйормацію.

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розв’язання цієї задачі я використав функцію strtok для того щоб розбити рядок на лексеми. Для введення рядка я використав fscanf(“%255[^\n]”) що можна читати як: зчитати 255 символів без врахування \n

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

#include <stdio.h>  
#include <string.h>  
#define N 256  
#define WORDS 30

Використані функції

Strtok ­– роділення слів на лексеми.

Strcmp – повертає 0 якщо рядки однакові

### Висноков

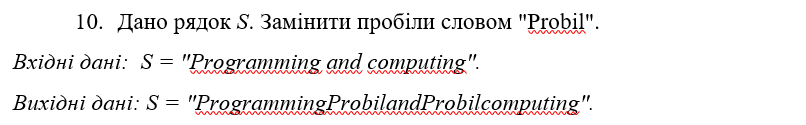
Програма успішно виводить слова які були введені по одному разу

# Лабораторна 14

## Завдання 1

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### ТЕОРЕТИЧНІ ВІДОМОСТІ

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

###### Підключені бібліотеки та макроси

1. #include <stdio.h>  
   #include <string.h>  
   #include <locale.h>

###### Використані функції

Memset – для очищення масиву.

Strcat ­– для конкатенації рядків

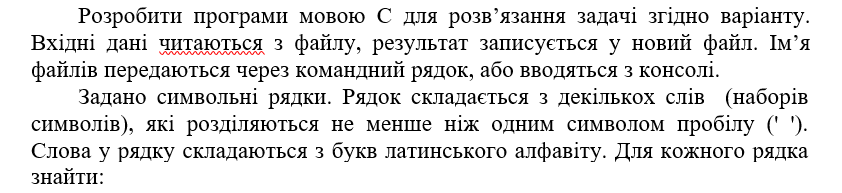
### Висновок

Програма успішно заміняє пробіл на словов ‘Probil’

## Завдання 2

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання





### ТЕОРЕТИЧНІ ВІДОМОСТІ

Весь рядок я розбив на окремі слова, якщо у слова перший символ був 'а', 'b', 'c', 'd' чи 'e', то я його виводжу.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

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

Використані функції

Fopen – функція для того, щоб відкрити файл.

Rewind ­– переміщує курсор на початок файлу

Fclose – закриває файл

Fgets – зчитує рядок з файлу

Fprintf – записує у файл дані.

Fflush – для очистки буфера.

Fputs – записати у файл

Feof – якщо вказівник знаходиться не в кінці файла то функція повертає ненульове значення

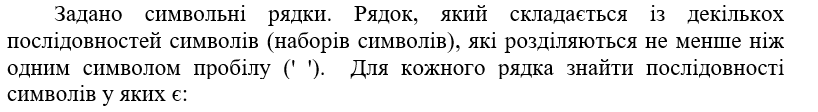
### Висновок

Програма успішно записує всі слова в один файл а слова які починаються на 'а', 'b', 'c', 'd' чи 'e' у інший

## Завдання 3

### ПОСТАНОВКА ЗАДАЧІ

##### Опис завдання



### 

### ТЕОРЕТИЧНІ ВІДОМОСТІ

Щоб розв’язвати цю задачу, рядок розбивається на слова та перевіряються останні символи.

### СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Підключені бібліотеки та макроси

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

Використані функції

Strtok – розбиває слова на лексеми.

Strlen – повертає довжину рядка.

### Висновок

Програма успішно знаходить всі слова які мають закінчення less, ing, ed.