#### 9. Лабораторна робота №7.

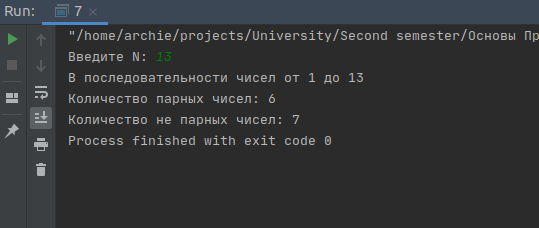
**Обробка числових та символьних послідовностей**

***Мета роботи*:** поглиблення знань та навичок роботи з циклами на прикладі розв’язку задач обробки числових та символьних послідовностей засобами С/C++; ознайомлення з символьним типом даних та типовими засобами їх обробки.

***Завдання:*** Дано натуральне число *n* та послідовність натуральних чисел *a*1, *a*2, …, *an*. Визначити кількість парних і непарних членів заданої послідовності.

***Розв’язання***:

#include <iostream>  
  
using namespace std;  
  
int main() {  
 int doubles = 0;  
 int notDoubles = 0;  
 int n;  
 cout << "Введите N: ";  
 cin >> n;  
  
 for (int i = 1; i <= n; i++) {  
 if (i % 2) {  
 notDoubles++;  
 } else {  
 doubles++;  
 }  
 }  
  
 cout << "В последовательности чисел от 1 до " << n << endl;  
 cout << "Количество парных чисел: " << doubles << endl;  
 cout << "Количество не парных чисел: " << notDoubles;  
  
 return 0;  
}

***Результати роботи:***

***Контрольні запитання:***

1. Що таке послідовність даних?

Послідовність даних — це упорядкована множина даних.

1. Яким чином обробляються послідовності?

Можуть бути виконані за допомогою лише одного циклу, під час введення даних.

1. Як за допомогою послідовностей уникають використання масивів?

Використовують алгоритми накопичення, тобто коли замість зберігання усієї послідовності даних (чисел, символів тощо) у програмі оголошується спеціальна змінна, в якій саме й накопичується результат кожної попередньої ітерації.

1. Що таке кодова таблиця ASCII?

Таблиця ASCII (American Standard Code Information Interchange – американський стандартний код обміну інформацією) алфавіту ПК

1. Яке призначення мають керуючі символи? Які їхні коди?

Використання керуючих символів можливо всюди, де можуть вживатись друковані символи.

1. Як оголошуються та використовуються у програмі С/С++ змінні символьного типу?

Оголошуються за допомогою слова char.

1. Що означають та як застосовуються керуючі послідовності С/С++?

Використання керуючих послідовностей символів можливо всюди, де можуть вживатись друковані символи.

1. Коли і як застосовуються макроси класифікації символів С/С++?

До значень символьного типу корисно застосовувати макроси класифікації символів **is…**, визначені у заголовному файлі ctype.h:

1. Які стандартні функції С/С++ можуть бути застосовані до даних символьного типу?

|  |  |  |  |
| --- | --- | --- | --- |
| Функція | Призначення | Приклад | #include |
| **getc()** | – читання символу з відображенням | c=getc(stdin); | stdio.h |
| **getch()** | – читання символу без відображення | c=getch(); | conio.h |
| **getchar()** | – читання символу з відображенням | c=getchar(); | stdio.h |
| **getche()** | – читання символу з відображенням | c=getche(); | conio.h |
| **putc()** | – відображення символу | putc(c,stdout); | stdio.h |
| **putch()** | – відображення символу | putch(c); | conio.h |
| **putchar()** | – відображення символу | putchar(c); | stdio.h |

#### 10. Лабораторна робота №8.

**Розробка функцій користувача**

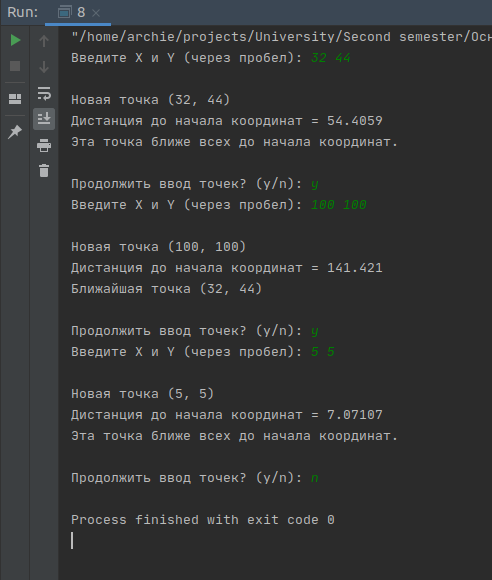
***Мета роботи*:** ознайомлення з означенням і основними характеристиками функцій мови; засвоєння основних прийомів використання в програмах функцій користувача; набуття навичок написання структурованих програм C/C++.

***Завдання:*** На площині задано n точок з дійсними координатами (x1,y1), (x2,y2), … (xn,yn). З’ясувати, яка з них ближча до початку координат, визначивши відповідну функцію. Кожну пару координат точок ввести через пробіл.

***Розв’язання:***

#include <iostream>  
#include <cmath>  
  
using namespace std;  
  
double distance(double x1, double y1, double x2, double y2) {  
 return sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));  
}  
  
int main() {  
 double near\_dist = -1, near\_x, near\_y;  
 double dist, x, y;  
  
 do {  
 cout << "Введите X и Y (через пробел): ";  
 cin >> x >> y;  
 dist = distance(0, 0, x, y);  
  
 cout << endl;  
 cout << "Новая точка (" << x << ", " << y << ")" << endl;  
 cout << "Дистанция до начала координат = " << dist << endl;  
  
 if (dist < near\_dist || near\_dist == -1) {  
 cout << "Эта точка ближе всех до начала координат." << endl;  
 near\_dist = dist;  
 near\_x = x;  
 near\_y = y;  
 } else  
 cout << "Ближайшая точка (" << near\_x << ", " << near\_y << ")" << endl;  
  
 cout << endl;  
 cout << "Продолжить ввод точек? (y/n): ";  
 char c;  
 cin >> c;  
 if (c == 'n')  
 break;  
 } while (true);  
  
 return 0;  
}

***Результати роботи:***

******

***Контрольні запитання:***

1. Що таке функція та яке її призначення в програмі?  
   Функція — це фрагмент програмного коду, до якого можна звернутися з іншого місця програми. Завдяки застосування функцій спрощується процес програмування, а самі програми стають більш структурованими, зрозумілішими та лаконічними.
2. Які види функцій може містити програма С/С++?

Кожна С-програма містить як мінімум одну функцію, а саме функцію main(), та може використовувати інші, наприклад, бібліотечні функції. Для розв’язку у програмі окремої самостійної задачі бажано розробити для неї й окрему функцію (так звану функцію користувача).

1. Що необхідно для використання в програмі функції користувача?

Для використання в С-програмі функції користувача її необхідно:

– оголосити;

– визначити (описати);

– викликати.

1. Що таке прототип, формальні та фактичні параметри (аргументи) функції?

**тип Ім’я\_функції (список\_параметрів);**

Таке оголошення називається прототипом функції,

Списком формальних параметрів позначаються вхідні дані, значення яких передаються функції під час виклику у вигляді фактичних параметрів (аргументів).

1. Яку структуру має функція користувача, та як здійснюється в програмі її виклик?

Прототип функчції **тип Ім’я\_функції (список\_параметрів);**

Функція **тип Ім’я\_функції (список\_параметрів) {**

**//тіло функції**

**return змінна за типом функції;**

**}**

1. Який тип можуть мати функції та їх параметри? Що таке void-функції та void-параметри?

Тип функції вказує тип значення, що повертається функцією в програму. Зазвичай, це один з числових типів (int, float тощо). Іноді функція взагалі не повертаю результату. Така функція має тип void.

1. Яка функція називається рекурсивною та які вимоги для її використання в програмі?

Якщо оператор у функції містить виклик цієї ж функції, така функція називається рекурсивною. Головною вимогою до рекурсивних функцій є необхідність забезпечення коректного виходу з них, аби не призвести до переповнення стеку і «зависанню» системи.

#### 11.Лабораторна робота №9.

#### Створення багатомодульних програм

***Мета роботи*:** ознайомлення з означенням і основними характеристиками функцій мови; засвоєння основних прийомів використання в програмах функцій користувача; набуття навичок написання структурованих програм C/C++.

***Завдання:***

Виконати завдання до лабораторної роботи №8 відповідно до заданого варіанту, організувавши багатомодульну программу з розташуванням функції(й) користувача та відповідних прототипів в окремих файлах.

***Розв’язання:***

*\* main.cpp \**

#include <iostream>  
#include "distance.h"  
  
using namespace std;  
  
  
int main() {  
 double near\_dist = -1, near\_x, near\_y;  
 double dist, x, y;  
  
 do {  
 cout << "Введите X и Y (через пробел): ";  
 cin >> x >> y;  
 dist = distance(0, 0, x, y);  
  
 cout << endl;  
 cout << "Новая точка (" << x << ", " << y << ")" << endl;  
 cout << "Дистанция до начала координат = " << dist << endl;  
  
 if (dist < near\_dist || near\_dist == -1) {  
 cout << "Эта точка ближе всех до начала координат." << endl;  
 near\_dist = dist;  
 near\_x = x;  
 near\_y = y;  
 } else  
 cout << "Ближайшая точка (" << near\_x << ", " << near\_y << ")" << endl;  
  
 cout << endl;  
 cout << "Продолжить ввод точек? (y/n): ";  
 char c;  
 cin >> c;  
 if (c == 'n')  
 break;  
 } while (true);  
  
 return 0;  
}

*\* distance.cpp \**

#include <cmath>  
  
double distance(double x1, double y1, double x2, double y2) {  
 return sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));  
}

*\* distance.h \**

#ifndef DISTANCE\_H  
#define DISTANCE\_H  
double distance(double x1, double y1, double x2, double y2);  
#endif //DISTANCE\_H

***Результати роботи:***

#### 

#### Лабораторна робота №10.

**Робота з одновимірними числовими масивами**

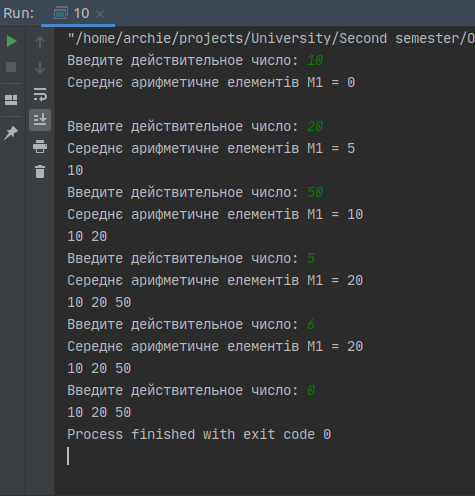
***Мета роботи:*** ознайомлення зі структурами даних, зокрема з одновимірними числовими масивами (векторами); набуття навичок роботи з числовими масивами у програмах C/C++.

***Завдання:*** З довільного одновимірного масиву дійсних чисел М1 сформувати масив М2, що складається з елементів М1, більших за середнє арифметичне елементів М1.

***Розв’язання:***

#include <iostream>  
  
using namespace std;  
  
int main() {  
 const int MAX = 100;  
 int M2[MAX];  
 int M2Length = 1;  
  
 for (int i = 0; i <= MAX; ++i) {  
 int sum = 0, number;  
 cout << "Введите действительное число: ";  
 cin >> number;  
 if (number == 0)  
 break;  
  
 for (int j = 0; j < M2Length-1; j++) {  
 sum += M2[j];  
 }  
  
 cout << "Середнє арифметичне елементів M1 = " << double (sum / M2Length) << endl;  
 for (int k = 0; k < M2Length-1; k++) {  
 cout << M2[k] << " ";  
 }  
 cout << endl;  
 if (number > sum / M2Length) {  
 M2[i] = number;  
 M2Length++;  
 } else  
 i--;  
 }  
 for (int i = 0; i < M2Length-1; i++) {  
 cout << M2[i] << " ";  
 }  
 return 0;  
}

***Результати роботи:***



***Контрольні запитання:***

1. Коли у програмах С/С++ застосовуються масиви?  
   Масиви у С/С++ використовуються для обробки великої кількості однотипних даних.
2. Який синтаксис оголошення масиву у С/С++?  
   Масив створюється як звичайна змінна. Кількість елементів масиву задається при його оголошенні та закривається у квадратні дужки. Для опису елементів масиву при його створенні можна використати фігурні дужки. В них через кому перераховуються елементи масиву. В кінці закриваючої фігурної дужки ставиться крапка з комою. Для формування виразу, що має відношення до конкретного елементу масиву необхідно після імені масиву зазначити індекс обраного елементу (в квадратних дужках). У якості індексу має бути цілий вираз – константа, змінна, результат функції.
3. Що таке розмір та що таке розмірність масиву?  
   Розмір масиву визначається кількістю елементів всередині масиву. Розмірність масиву – це кількість індексів, необхідний для однозначної адресації елементу в рамках масиву. За кількістю використовуваних індексів масиви поділяються на одновимірні, двовимірні, тривимірні і т.д.
4. Як можна ініціалізувати масив у С/С++?  
   Для ініціалізації необхідно оголосити масив (як зазвичай), після чого у фігурних дужках записати (через кому) список значень констант. Під час попередньої ініціалізації масиву можна використовувати sizeof() для коректного визначення числа елементів масиву.
5. Як здійснюється індексація елементів масиву?

Від 0 до необхідної кількості.

1. Як здійснюється звертання до окремих елементів масиву?

При звернені до окремих елементів масиву індекс необхідного

елементу вказується у квадратних дужках.

1. Яким чином можна опрацьовувати одразу декілька елементів масиву?  
   За допомогою використання оператору циклу for.
2. Як здійснюється передача і опрацювання масивів у функціях?  
   Якщо масив використовується в якості аргументу функції, то передається тільки адреса масиву, а не весь масив. При викликанні функції з ім’ям масиву у функцію передається показник на перший елемент масиву. Параметр функції повинен мати тип, сумісний з показником.
3. Яке формулювання задачі пошуку у масиві?  
   Задача пошуку у масиві полягає у визначенні номерів певних елементів масиву або їх значень. Розрізняють задачі пошуку в упорядкованому та неупорядкованому масивах. В неупорядкованому масиві пошук можна здійснити лише за допомогою перегляду всього масиву (лінійний пошук). Якщо масив упорядкований (відсортований), то до нього може бути застосовано спеціальні методи пошуку, наприклад, бінарний пошук тощо.

#### 13. Лабораторна робота №11.

**Робота з числовими матрицями**

***Мета роботи*:** ознайомлення зі структурами даних, зокрема з двовимірними числовими масивами (матрицями); набуття навичок роботи з числовими матрицями у програмах C/C++.

***Завдання:*** Дано натуральні числа *n* та *m*, що визначають відповідно кількість рядків та стовпців у деякій заданій матриці дійсних чисел, а також довільне дійсне число *x*. Знайти серед членів цієї матриці усі, більші за x, елементи та їхні індекси.

***Розв’язання:***

***Контрольні запитання:***

1. Який масив вважається багатовимірним?

В багатовимірному масиві кілька індексів.   
 Прикладом багатовимірного масиву є матриця.

1. Що таке числова матриця?  
   Це двовимірний масив.
2. Який синтаксис оголошення багатовимірного масиву у С/С++?  
   тип ім’я\_масиву[розмір другого виміру][розмір першого виміру] де перший індекс є номер рядка а другий - кількість стовпчиків.
3. Як зберігаються елементи багатовимірного масиву і який обсяг пам’яті вони займають?  
   Збереження багатовимірного масиву та обсяг пам’яті який він займає обчислюється за формулою: число байт = розмір першого виміру\*розмір другого виміру\*sizeof(базовий тип)
4. Як здійснюється індексація елементів багатовимірного масиву?  
   Кожен елемент багатовимірного масиву вписується в квадратні дужки. Ім’я масиву без індексу – автоматичний показник на перший елемент масиву.
5. Яким чином можна ініціалізувати багатовимірний масив у С/С++?  
   Кожен ініціалізатор в списку представляє собою або константу, або, в свою чергу, список ініціалізаторів. Ініціалізатор складається із списку інізіалізаторів, розділених комами та у фігурних дужках. Наявність списку ініціалізаторів в оголошенні масиву дозволяє не вказувати число його першої розмірності. В цьому випадку кількість елементів в списку ініціалізаторів й виявляє кількість елементів за першою розмірністю масива. Тим самим виявляється необхідний розмір пам’яті, необхідний для збереження масиву.
6. Яким чином можуть опрацьовуватися багатовимірні масиви у програмах С/С++?  
   За допомогою вкладених циклів. Перший цикл – по першому індексу( по всім строкам), другий по - другому індексу(по всім стовпчикам)
7. Як здійснюється передача і опрацювання багатовимірних масивів у функціях?  
   Для передачі багатовимірного масиву у якості аргументу функції необхідно задати мінімум інформації, достатньої для перетворення індексних виразів в адреси пам’яті. Зокрема для двовимірного масиву достатньо передати кількість стовпців аби компілятор зміг самостійно обчислити адреси елементів на початку кожного рядка.
8. Наведіть приклади практичного застосування багатовимірних масивів.  
   Матриці доволі часто застосовуються у математиці, особливо під час розв’язку задач, що містять системи алгебраїчних та диференційних рівнянь

#### 14. Лабораторна робота №12.

**Реалізація алгоритмів сортування масивів**

***Мета роботи*:** удосконалювання навичок роботи з одновимірними числовими масивами; набуття навичок розв’язку задач із застосуванням прямих (елементарних) методів сортування масивів у програмах C/C++.

***Контрольні запитання:***

1. Що таке сортування масиву?  
   сортування масиву – це процес розподілення всіх елементів масиву в певному порядку. Це буває дуже корисним при використанні великої кількості елементів.
2. Які відомі методи сортування масивів і як оцінюється їхня швидкодія?   
   Всі методи сортування можна поділити на дві великі групи – прямі (елементарні) та поліпшені (удосконалені) методи. Прямі методи сортування по принципу, який лежить в основі методу, поділяються на три підгрупи:

– сортування вставкою (включенням);

– сортування вибором (виділенням);

– сортування обміном («бульбашкове» сортування).

Поліпшені методи сортування ґрунтуються на тих самих принципах, що і прямі, але використовують деякі оригінальні математичні ідеї для прис-корення процесу сортування. Серед удосконалених методів сортування най-частіше використовуються методи Хоара, Шелла, пірамідальне сортування (за допомогою дерева), метод злиття тощо.

1. Навести алгоритм одного з прямих методів сортування масиву.  
   Починаючи з елементу з індексом 0 іде пошук в масиві з найменшим значенням  
   Знайдене найменше значення міняється місцями з нульовим елементом.  
   Повторення перших двох дій для наступного індексу масиву.
2. Як поліпшити швидкодію методу сортування масиву?  
   Для поліпшення швидкості сортування масиву можна використати більш швидкодіючі методи. Наприклад метод сортування Шелла.
3. На яких засадах ґрунтуються поліпшені методи сортування масивів?  
   Поліпшені методи сортування ґрунтуються на тих самих задах. Що й прямі методи, але в них присутні деякі математичні методи для прискорення сортування.
4. Як може виглядати функція, яка здійснює сортування масиву?  
   Основними вбудованими функціями сортування масиву є qsort та std::sort.  
   qsort:([сортуємий масив], [кількість елементів], sizeof([тип елементу]), [функція для порівняння]);  
   std::sort виглядає так:  
   sort([ітератор початку сортування], [ітератор закінчення сортування]);

15.Лабораторна робота **№13.**

**Обробка текстової інформації**

***Мета роботи*:** подальше ознайомлення зі структурами даних, зокрема з символьними рядками; набуття навичок роботи з рядками та застосування рядкових функцій у програмах C/C++.

***Контрольні запитання:***

1. Що таке символьний рядок?  
   Символьний рядок – особливий вид масиву, елементами якого є значення ASCII-кодів символів. Останнім символом рядка повинен бути так званий нуль-символом (ASCII 0) – ознака кінця рядка.
2. Які типи рядкових змінних використовуються у C/C++?   
   char s[128];

char \*sPtr;

1. Що таке глобальні та локальні рядкові змінні?  
   Рядковий покажчик вказує на масив значень типу char, тобто є адресою місцезнаходження першого символу рядка. Глобальні рядкові змінні оголошуються за межами усіх функцій програми, доступні із усіх її функцій та не потребують ініціалізації. Неініціалізовані локальні рядки доступні лише у своїх функціях та зберігають непередбачувані значення.
2. Як оголошуються та ініціалізуються рядкові змінні різних типів?  
   Символьні рядки можна ініціалізувати безпосередньо під час оголошення. Якщо граничну кількість символів рядка не вказано або рядок оголошено як рядковий покажчик (наприклад, str3, str4 або sPtr, str), то розмір такого рядка встановлюється компілятором автоматично за кількістю елементів-ініціалізаторів з урахуванням нуль-символу
3. Що таке загальна та поточна довжина рядка?  
   Поточна довжина рядка – це кількість символів, яка в ній знаходиться без урахування нуль-символу.  
   Загальна довжина рядка – це кількість символів з урахуванням нуль-символів.
4. Як здійснюється доступ до окремих символів рядка?  
   Доступ до окремих символів рядка здійснюється через індексні вирази, тобто так само, як і до елементів масиву.
5. Як присвоюються значення рядковим змінним?  
   Для присвоєння значень рядкам, оголошеним як символьні масиви, необхідно застосовувати спеціальні бібліотечні функції копіювання рядків
6. За допомогою яких функцій вводяться та виводяться рядки?  
   Для введення символьного рядка застосовуються функції **scanf()** та **gets()**Для виведення символьного рядка застосовуються функції **printf()** та **puts()**
7. Яке призначення має заголовний файл string.h?  
   Це файл стандартної бібліотеки С, що містить функції для роботи з нуль-термінованими строчками та різнимі функції роботи з пам’ятю.
8. Які основні стандартні функції застосовуються для обробки рядків?  
   Основні стандартні функції вміщено в бібліотеках stdlib.h та string.h.
9. Яким чином покажчики використовуються для обробки рядків?  
   З рядковими покажчиками можна працювати так само, як і зі звичай-ними покажчиками.
10. Як використовується динамічна пам'ять для зберігання і обробки символьних рядків?  
    У разі використання динамічної пам’яті (купи) для роботи з рядковими покажчиками необхідно скористатися спеціальними функціями **malloc()** або **calloc() –** длявиділення та **free() –** для вивільнення пам’яті, з прототипами у заголовному файлі stdlib.h. В програмах С++ можна застосовувати також альтернативні оператори керування пам’яттю, а саме **new** – для виділення та **delete** – для вивільнення пам’яті у купі.

16.Лабораторна робота **№14**.

**Робота зі структурами та об’єднаннями**

***Мета роботи*:** подальше ознайомлення зі структурами даних, зокрема зі структурами та об’єднаннями; набуття навичок роботи зі структурами та об’єднаннями у програмах C/C++.

***Контрольні запитання:***

1. Що таке структура та об’єднання? Яка між ними різниця?  
   Структура – це сукупність змінних, що об’єднані одною назвою та представляє собою загальноприйнятий засіб сумісного збереження інформації.   
   Об’єднання – це об’єкт, що дозволяє декільком змінним різних типів займати одну ділянку пам’яті. Об’єднання – це така ж структура, за винятком того, що всі змінні займають один і той самий об’єм пам’яті.
2. Як оголошуються та ініціалізуються структури та об’єднання?  
   Для структури використовується ключове слово struct. При оголошенні структури можна використовувати одну або декілька змінних.   
   Об’єднання оголошується ключовим словом union. Так само, як і для структур, шаблон об’єднання можна оголошувати окремо або одночасно з оголошенням змінних
3. Що таке тег та поле структури (об’єднання)?  
   Тег - це назва структури, що може слідувати за словом struct. Тег дає назву структурі даного виду і може слугувати значенням той частини оголошення, що закрита у фігурні дужки. Поля структури розташовуються у пам’яті саме в тій послідовності, що й в оголошені шаблону. Тип поля може бути простим або складеним. Імена полів в одному шаблоні не можуть збігатися, проте вони можуть збігатися з іменами інших змінних програми або полів інших структур.
4. Як здійснюється доступ до елементів структури (об’єднання)?  
   Для звертання до елементів структури застосовується крапкова нотація (операція «крапка») або операція «стрілка». Для звертання до елементів об’єднання також застосовується крапкова нотація (операція «крапка») або операція «стрілка».
5. Що таке бітові поля?  
   Це тип елементу структури, який може бути цілочисленним або перелічуваним.
6. Як зручніше оголосити масив структур?  
   В
7. Які операції можуть застосовуватися до структур (об’єднань)?  
   Обирання елементу через назву (оператор «крапка»), обирання елементу через вказівник (оператор «стрілка»), присвоєння (оператор «дорівнює»), взяття адреси (операція «амперсанд»)
8. Як передати структуру або окреме її поле у функцію?  
   При передачі в функцію члену структури передається лише його значення. Коли структура використовується в якості аргументу функції, то для передачі цілої структури використовується звичайний метод виклику за значення
9. Чи може бути структура (об’єднання) результатом функції?  
   Так, може.

17. Лабораторна робота **№15**.

## **Реалізація файлового обміну даними**

***Мета роботи*:** подальше ознайомлення зі структурами даних, зокрема з файлами; набуття навичок обробки текстових та двійкових файлів у програмах C/C++.

***Контрольні запитання:***

1. Що таке файл у мові програмування?  
   Файл у мові С/С++ – абстрактне поняття, яким позначається сукупність даних на зовнішньому носії інформації або певний термінальний пристрій (наприклад, клавіатура, дисплей, принтер тощо).
2. Що таке файловий потік і як він створюється засобами С/С++?  
   Послідовність байтів, що надходить від певного файлу (логічного пристрою) або передається до файлу, називається потоком**.** Для здійснення обміну даними між програмою та файлом необхідно насамперед створити відповідний потік у програмі, пов'язаний безпосередньо з цим файлом, тобто відкрити файл для доступу (читання або запису):

**FILE \*f;**

**f=fopen(fileName,fileMode);** де fopen() – стандартна (stdio.h) функція відкриття існуючого (або створення нового) фізичного файлу на ім’я fileName з заданим режимом обміну fileMode та його зв’язування з файловою змінною f.

1. Які основні режими відкриття файлового потоку у програмі С/С++?  
   Відкриття файлу для читання – r, rb, rt  
   Відкриття або створення файлу для запису – w, wb, wt  
   Відкриття файлу для доповнення даними – a, ab, at
2. Як застосовується та працює у програмі С/С++ функція fopen()?  
   Функція fopen() у разі успішного відкриття потоку повертає адресу (покажчик) створеної структури FILE, або повертає порожній покажчик NULL, якщо структуру створити неможливо (невірне ім’я файлу, хибний шлях тощо).

## Як закривається файловий потік у програмі С/С++? Після нормального завершення програми всі відкриті потоки автоматично закриваються, а їх буфери вивільнюються, проте коректніше буде застосування для цього стандартної функції fclose(). Після успішного виконання ця функція повертає нуль, а у разі помилки – макроконстанту EOF ­(End Of File – кінець файлу).

Як засобами С/С++ здійснюються посимвольне та рядкове опрацювання текстових файлів?  
Основними функціями посимвольного зчитування з файлових потоків, відкритих для читання у текстовому режимі (″rt″), є стандартні функції **getc()** і **fgetc()**. У разі успішного виконання ці функції повертають зчитаний з потоку символ або EOF (у разі досягнення кінця файлу або неможливості зчитування даних).

Для посимвольного запису у файлові потоки, відкриті для запису у текстовому режимі (″wt″), використовуються стандартні функції **putc()** і **fputc()**.

1. Як засобами С/С++ здійснюється форматний обмін даними між програмою і файлами?  
   Форматний обмін даними між програмою та файлами здійснюється за допомогою функцій **fscanf()** і **fprintf()** відповідно. Ці функції працюють аналогічно консольним функціям scanf() і printf(),
2. Якими засобами С/С++ здійснюється опрацювання двійкових файлів?  
   Опрацювання двійкових файлів здійснюється за допомогою стандартних функцій блочного обміну даними **fread()** і **fwrite()** відповідно
3. Що таке позиціювання файлових потоків?
4. Які є засоби аналізу помилок файлового обміну у програмі С/С++?  
   Більшість файлових функцій повертають результат, за яким можна перевірити коректність виконаної файлової операції. Проте встановити причину можливого збою можна за допомогою функцій аналізу помилок файлового обміну:  
   feof() – перевіряє досягнення кінця файлу;  
   ferror() – перевіряє ознаку помилки в операціях звертання до потоків;  
   perror() - виводить на екран інформацію про причини виникнення помилки в операціях звернення до потоків.