



Національний університет
водного господарства та природокористування

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій

04-03-150

Методичні вказівки
до виконання лабораторних робіт №24-26
з навчальної дисципліни

“Інформатика та комп'ютерна техніка”

для студентів спеціальностей:

151 “Автоматизація та комп'ютерно-інтегровані технології”,

141 “Електроенергетика, електротехніка та електромеханіка”

денної та заочної форм навчання

Рекомендовано науково-
методичними комісіями:
спеціальності 151
“Автоматизація та
комп'ютерно-інтегровані
технології” протокол № 4
від 10 жовтня 2016 р. ;
спеціальності 141
“Електроенергетика,
електротехніка та
електромеханіка”
протокол №3
від 14 вересня 2016 р.

Рівне – 2016



Методичні вказівки до виконання лабораторних робіт №24-26 з навчальної дисципліни “Інформатика та комп’ютерна техніка” для студентів спеціальностей: 151 “Автоматизація та комп’ютерно-інтегровані технології”, 141 “Електроенергетика, електротехніка та електромеханіка” денної та заочної форм навчання / А.П.Сафоник – Рівне: НУВГП, 2016. – 37 с.

Упорядник: А.П. Сафоник, доктор технічних наук, професор кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій НУВГП.



Відповідальний за випуск: В.В. Древецький, д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій.

| | |
|-----------------------------|----|
| Лабораторна робота №24..... | 3 |
| Лабораторна робота №25..... | 10 |
| Лабораторна робота №26..... | 24 |



Розробка програм з даними структурного типу

24.1. Мета роботи

Ознайомитись з поняттям структури і структурної змінної. Навчитися створювати масиви структур і працювати з вкладеними структурами

24.2. Теоретичні відомості

Структура - це об'єднання одного або більше об'єктів (змінних, масивів, покажчиків, інших структур). Як і масив, вона являє собою сукупність даних, але відрізняється від нього тим, що до її елементів необхідно звертатися на ім'я, а її різні елементи не обов'язково має належати до одного типу.

Структури зручно використовувати там, де різноманітні дані, що відносяться до одного і того ж об'єкту, необхідно об'єднувати. Наприклад, учня середньої школи характеризують такі дані: прізвище, ім'я, дата народження, клас, вік.

Оголошення структури здійснюється за допомогою ключового слова `struct`, за яким слід її тип, список елементів, укладених у фігурні дужки. Її можна представити в наступному загальному вигляді:

```
struct тип {тип елемента 1 ім'я елемента 1;  
тип елемента n ім'я елемента n;};
```

Іменем елемента може бути будь-який ідентифікатор. В одному рядку можна записувати через кому кілька ідентифікаторів одного типу.

Наприклад:

```
struct date {int day;  
int month;  
int year;};
```

Російські букви використовувати в ідентифікатор у мові СІ не можна.

Слідом за фігурною дужкою, що закінчує список елементів, можуть записуватися змінні даного типу, наприклад:

```
struct date {...} a, b, c;
```

При цьому виділяється відповідна пам'ять.

Виведене ім'я типу можна використовувати для оголошення запису, наприклад: `struct date day;`. Тепер мінлива `day` має тип `date`.

Дозволяється вкладати структури одна на іншу. Для кращого сприйняття структури використовуємо російські букви в ідентифікаторах, в мові СІ цього робити не можна.

Наприклад:

```
struct УЧЕНЬ {char Прізвище [15];  
ім'я [15];  
struct DATA ДАТА НАРОДЖЕННЯ;
```



```
int клас, вік,};
```

Певний вище тип DATA включає три елементи: День, Місяць, Рік, що містять цілі значення (int). Запис УЧЕНЬ включає елементи: ПРИЗВИЩЕ [15]; ІМ'Я [15]; ДАТА НАРОДЖЕННЯ, КЛАС, ВІК. ПРИЗВИЩЕ [15] і ІМ'Я [15] - це символічні масиви з 15 компонент кожен. Змінна ДАТА НАРОДЖЕННЯ представлена складовим елементом (вкладеної структурою) ДАТА. Будь-яку дату народження відповідають день місяця, місяць і рік. Елементи КЛАС і ВІК містять значення цілого типу (int). Після введення типів ДАТА і УЧЕНЬ можна оголосити змінні, значення яких належать цим типам.

Наприклад:

```
struct УЧЕНЬ УЧНІ [50];
```

масив УЧНІ складається з 50 елементів типу УЧЕНЬ.

У мові Сі дозволено використовувати масиви структури; записи можуть складатися з масивів та інших записів.

Щоб звернутися до окремого компоненту структури, необхідно вказати її ім'я, поставити крапку і відразу за нею написати ім'я потрібного елемента.

Наприклад:

```
Учні [1]. КЛАС = 3;
```

```
Учні [1]. ДАТА НАРОДЖЕННЯ. ДЕНЬ = 5;
```

```
Учні [1]. ДАТА НАРОДЖЕННЯ. МІСЯЦЬ = 4;
```

```
Учні [1]. ДАТА НАРОДЖЕННЯ. РІК = 1979;
```

Перший рядок вказує, що 1-й учень навчається в третьому класі, а наступні рядки - його дату народження: 5.04.79.

Кожен тип елемента структури визначається відповідним рядком оголошення в фігурних дужках. Наприклад, масив УЧНІ має тип УЧЕНЬ, рік є цілим числом. Оскільки кожен елемент запису відноситься до певного типу, його складене ім'я може з'являтися скрізь, де дозволено використовувати значення цього типу. Розглянемо приклад програми:

```
/* Демонстрація запису */  
# Include <stdio.h>  
struct computer {int mem;  
int sp;  
char model [20];};  
/* Оголошення записи типу computer, що складається з трьох  
елементів: mem, sp, model */  
struct computer pibm =  
{512, 1, "ПЕОМ ЄС 1840.05 "}  
/* Оголошення і ініціалізація змінної pibm типу computer */  
main ()
```



```
{Printf ("персональна ЕОМ% s \ n \ n", ribm.model);  
printf ("обсяг оперативної пам'яті -% d До байт \ n", ribm.mem);  
printf ("продуктивність -% d млн. операцій в секунду \ n", ribm.sp);  
/* Висновок на екран значень елементів структури */  
}
```

У даній програмі оголошується запис `computer`, яка складається з трьох елементів: `mem` (пам'ять ЕОМ), `sp` (швидкодія), `model` [20] (модель ПЕОМ). Змінна `ribm` має тип `computer` і є глобальною. Рядки `ribm.model`, `ribm.mem`, `ribm.sp` в операторі `printf` викликають звернення до відповідних елементів запису `ribm` типу `computer`, яким раніше були присвоєні певні значення.

Результат роботи програми має вигляд:
персональна ЕОМ ПЕОМ ЄС 1840.05
обсяг оперативної пам'яті - 512 До байт
продуктивність - 1 млн. операцій у секунду
Розглянемо використання в програмі вкладених структур:

```
/* Демонстрація вкладених структур */
```

```
# Include <stdio.h>
```

```
struct date {int day;
```

```
int month;
```

```
int year;};
```

```
/* Оголошення записи типу date */
```

```
struct person {char fam [20];
```

```
char im [20];
```

```
char ot [20];
```

```
struct date fl;};
```

```
/* Оголошення структури типу person; одним з елементів запису person
```

є запис `fl`

```
типу date */
```

```
main ()
```

```
{Struct person ind1;
```

```
/* Оголошення змінної ind1 типу person */
```

```
printf ("Вкажіть прізвище, ім'я, по батькові, день, \ n місяць"
```

```
"I рік народження громадянина ind1 \ n");
```

```
scanf ("% S% S% S% d% d", & ind1.fam, & ind1.im, & ind1.ot,
```

```
& ind1.fl.day, & ind1.fl.month, & ind1.fl.year);
```

```
/* Введення відомостей про громадянина ind1 */
```

```
printf ("Прізвище, ім'я, по батькові:% S% S% S \ n", ind1.fam, ind1.im,
```

```
ind1.ot);
```

```
printf ("Рік народження -% d \ n ", ind1.fl.year);
```

```
printf ("Місяць народження -% d - й \ n ", ind1.fl.month);
```



```
printf ("День народження -% d-й \n", ind1.fl.day);  
/* Висновок відомостей про громадянина ind1 */  
}
```

Структура типу date (дата) містить три елементи: day (день), month (місяць), year (рік). Структура типу person (чоловік) містить чотири елементи: fam [20] (прізвище), im [20] (ім'я), ot [20] (по батькові), fl (дата народження). Останній з них (fl) - це вкладена запис типу date.

Результати роботи програми:

Вкажіть прізвище, ім'я, по батькові, день, місяць і рік народження громадянина ind1

```
Алексеев  
Сергій  
Петрович  
3  
5  
1978
```

Підкреслена інформація вводиться користувачем.

Відомості про громадянина ind1

Прізвище, ім'я, по батькові: Алексеев Сергій Петрович

Рік народження - 1978

Місяць народження - 5-й

День народження - 3-й

У наступній програмі розглянемо використання структури у вигляді елементів масиву ribm. Кожен елемент складається з наступних компонентів: mem (пам'ять), sp (обсяг вінчестера), model [20] (модель ПЕОМ):

```
/* Масиви записів */  
# Include <stdio.h>  
struct computer {int mem, sp;  
char model [20];  
ribm [10];};  
/* Оголошення запису типу computer;  
оголошення масиву ribm типу computer */  
main ()  
{int i, j, k, priz;  
for (i = 0; i <10; i ++)  
{Printf ("Введіть відомості про ПЕОМ% d і ознака (0-кінець;  
\n інша цифра-продовження) \n ", i);  
printf ("модель ПЕОМ - ");  
scanf ("% S", & ribm [i]. model);  
printf ("обсяг оперативної пам'яті - ");
```



```
scanf ("%d", & pibm [i]. mem);
printf ("обсяг вінчестера - ");
scanf ("%d", & pibm [i]. sp);
printf ("ознака -");
scanf ("%d", & priz);
k = i;
if (! priz) break;}
/* Тут! Priz - операція заперечення priz; break - вихід з циклу for, якщо
priz = 0 */
for (i = 0; i <10, i + +);
{
printf ("\n Про яку ПЕОМ Ви хочете отримати відомості? \n (Введіть
номер від 0 до 9) \n");
scanf ("%d", & j);
if (j> k)
{Printf ("Немає відомостей про цю ПЕОМ \n");
continue;}
printf ("персональна ЕОМ% s \n ", pibm [j]. Model);
printf ("обсяг оперативної пам'яті -% d Мб \n", pibm [j]. mem);
printf ("обсяг вінчестера -% d Мб \n", pibm [j]. sp);
printf ("ознака -");
scanf ("%d", & priz);
if (! priz) break;}
/* Введення відомостей про ПЕОМ та занесення в масив pibm записів
типу computer (перший цикл for); висновок на екран відомостей
про ПЕОМ (другий цикл for) */
}
```

Результати роботи програми:

Введіть відомості про ПЕОМ та ознака (0-кінець; інша цифра -
продовження)

модель ПЕОМ - АТ 486 SX
обсяг оперативної пам'яті - 32
обсяг вінчестера - 4 Гбайта
ознака - 1

Введіть відомості про ПЕОМ та ознака (0-кінець; інша цифра -
продовження)

модель ПЕОМ - АТ 386 DX
обсяг оперативної пам'яті - 64
обсяг вінчестера - 14 Гбайт
ознака - 0



9)

1

модель ПЕОМ - АТ 386 DX
обсяг оперативної пам'яті - 16 Мб
обсяг вінчестера - 2,5 Гбайт
ознака – 0

24.3. Програма роботи

1. Ознайомитись з особливостями опису, виклику нестандартних функцій та підпрограм.
2. Написати та відлагодити програми для виконання завдання, згідно варіанту.

24.4. Контрольні питання

1. Що таке структура?
2. Які типи даних може містити структура?
3. Як оголошується структурований тип даних?
4. Для чого використовується структура?
5. Як оголошується структура?

Варіанти:

1. Опишіть запис СТУДЕНТ і помістіть в неї наступну інформацію: П.І.Б., оцінки (математика, фізика, креслення, хімія, опір матеріалів). Визначте, скільки студентів мають незадовільну оцінку з математики.
2. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте, скільки студентів мають незадовільну оцінку хоча б з одного предмета.
3. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте, скільки студентів склали всі іспити на 5.
4. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте середній бал групи з фізики.
5. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте кількість відмінних оцінок, отриманих групою з усіх предметів.
6. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте, скільки студентів мають середній бал від 4 до 5.
7. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте, яка кількість незадовільних оцінок отримано з усіх предметів.
8. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте, який з предметів був зданий групою найкраще.



9. Скориставшись записом СТУДЕНТ з варіанту № 1, визначте, скільки студентів не мають заборгованостей.
10. Опишіть запис АНКЕТА і помістіть в неї наступну інформацію: П.І.Б. (прізвище, ім'я, по батькові), адресу (вулиця, номер будинку, номер квартири), стать, вік. Визначте, скільки осіб жіночої та скільки чоловічої статі проживають в одному будинку.
11. Скориставшись записом АНКЕТА з варіанту № 10, визначте, скільки осіб чоловічої статі у віці старше 18 років і молодше 60 проживають на одній вулиці.
12. Скориставшись записом АНКЕТА з варіанту № 10, визначте, скільки осіб жіночої статі у віці старше 30 років проживають в одному будинку.
13. Скориставшись записом АНКЕТА з варіанту № 10, визначте, скільки дітей до 7 років проживають на одній вулиці.
14. Скориставшись записом АНКЕТА з варіанту № 10, визначте, скільки осіб чоловічої статі та жіночого у віці до 50 років проживають на одній вулиці.
15. Скориставшись записом АНКЕТА з варіанту № 10, визначте, скільки дітей від 1 року до 5 проживають в одному будинку.
16. Опишіть запис ТРАНСПОРТ і помістіть в неї наступну інформацію: П.І.Б. (Прізвище, ім'я, по батькові пасажир), багаж (кількість речей, вага в кг). Визначити кількість пасажирів, вага багажу яких перевищує 30 кг.
17. Скориставшись записом ТРАНСПОРТ з варіанту № 16, визначте, чи є пасажир, багаж якого складається з однієї речі вагою в 20 кг.
18. Скориставшись записом ТРАНСПОРТ з варіанту № 16, визначте середню вагу багажу
19. Скориставшись записом ТРАНСПОРТ з варіанту № 16, визначте кількість пасажирів, вага багажу яких перевершує середній.
20. Скориставшись записом ТРАНСПОРТ з варіанту № 16, визначте кількість пасажирів, що мають більше трьох речей.



25.1. Мета роботи

Вивчити функції та алгоритми для організації роботи з файлами.

25.2. Теоретичні відомості

Бібліотеки введення/виводу і робота з файлами мовою C. Операції введення/виводу в мові C організовані за допомогою бібліотечних функцій. Потрібно сказати, що система Borland C++ наслідує стандарт ANSI, названому також буферизованим (buffered) або форматованим (formatted) введенням/виводом.

В той же час система Borland C++ підтримує й інший метод вводу/виводу, так званий UNIX-подібний, або неформатований (небуферизований) ввід/вивід.

Ми приділимо головну увагу першому методу - стандарту ANSI.

Мова C++ підтримує ще і власний об'єктно-орієнтований ввід/вивід.

Важливо зрозуміти, що таке файл (file) і потік (stream) і яке розходження між цими поняттями. Система введення/виводу мови C підтримує інтерфейс, що не залежить від того, який в дійсності використовується фізичний пристрій вводу/виводу, тобто є абстрактний рівень між програмістом і фізичним пристроєм. Ця абстракція і називається потоком. Спосіб же збереження інформації на фізичному пристрої називається файлом.

Незважаючи на те що пристрої дуже різні (термінал, дискет, магнітна стрічка й ін.), стандарт ANSI мови C зв'язує кожен з пристроїв із логічним пристроєм, названим потоком. Оскільки, потоки не залежать від фізичних пристроїв, то та сама функція може записувати інформацію на диск, на магнітну стрічку або виводити її на екран.

У мові C існує два типи потоків: текстовий (text) і двійковий (binary).

Текстовий потік – це послідовність символів. Однак взаємооднозначної відповідності між символами, що подаються в потоці і виводяться на екран, може не існувати.

Двійковий потік – це послідовність байтів, що взаємооднозначно відповідають тому, що знаходиться на зовнішньому пристрої.

Файл у мові C - це поняття, що може бути застосоване до усього від файла на диску до терміналу. Потік може бути зв'язаний із файлом за допомогою оператора відкриття файла. Як тільки файл відкритий, то інформація може передаватися між ним і вашою програмою.

Не всі файли однакові. Для прикладу з файла на диску ви можете вибрати 5-ий запис або замінити 10-ий запис. У той же час у файл,



зв'язаний із друкувальним пристроєм, інформація може передаватися тільки послідовно в тому ж порядку. Це ілюструє саме головне розходження між потоками і файлами: усі потоки однакові, що не можна сказати про файли.

Операція відкриття файла зв'язує потік із визначеним файлом. Операція закриття файла розриває цей зв'язок. Якщо потік був відкритий для виводу, то при виконанні операції закриття файла відповідний буфер записується на зовнішній пристрій. Якщо програма закінчила роботу нормальним способом, усі файли автоматично закриваються.

Кожний потік, зв'язаний із файлом, має керуючу структуру, названу FILE. Вона описана в заголовному файлі STDIO. H.

Ввід/вивід на консоль. До цих операцій відносяться операції введення з клавіатури і виводу на екран. Технічні функції, що здійснюють ці операції, зв'язують консоль із стандартними потоками введення/виводу. У багатьох системах стандартне введення/вивід може бути перенаправлене (у тому числі й у MS DOS). Однак ми будемо для простоти припускати, що стандартне введення - це введення з клавіатури, а стандартний вивід - це вивід на екран.

Найпростіша функція введення `getche()`, що читає символи з клавіатури. Функція очікує, поки не буде натиснута клавіша, і повертає код, що відповідає символу. Одночасно відбувається відображення введеного символу на екран. Її прототип

```
int getche(void);
```

 знаходиться у файлі CONIO.H.

Найпростіша функція виводу - `putchar`. Вона виводить символ, що є її аргументом, на екран у поточну позицію курсору. Прототип цієї функції

```
int putchar(int c);
```

 знаходиться в STDIO.H.

Двома найбільше важливими аналогами функції `getche()` є `getchar()` і `getch()`. Функція `getchar()` робить буферизований ввід, але вимагає натискання клавіші Enter. Прототипи цих функцій описані у файлі STDIO. H. Функція `getch()` діє так само, як `getche()`, але не виводить символ на екран, її прототип знаходиться в CONIO. H. Функцію `getch()` часто використовують для припинення дії програми до натискання якоїсь клавіші саме тому, що вона не видає echo на екран.

Функції `gets()` і `puts()` здійснюють відповідно введення і вивід на консоль рядка символів, прототип `gets()` має вигляд

```
char &gets(char *s);
```

де `s` - вказівник на масив символів, що заповнюється введеними з клавіатури символами. Закінчення введення здійснюється натисканням клавіші Enter. Символ повернення каретки в масив не записується, зате заноситься символ `'\0'`, що завершує рядок.

Функція `puts()` виводить на екран рядок. Її прототип -



```
int puts(char *);
```

Ця функція, так само як і `printf()`, розпізнає спеціальні символи, наприклад символ табуляції `\t`. Функція `puts()` на відміну від `printf()` може виводити тільки рядок, зате працює швидше і її запис коротший, чим у `printf()`. У результаті дії функції `puts()` завжди відбувається перехід на новий рядок. Якщо вивід успішно завершений, то функція повертає нульове значення, у протилежному випадку повертає символ EOF. Прототипи функцій `puts()` і `gets()` знаходяться у файлі `STDIO.H`.

Вказівник на файлову змінну. Сполучною ланкою між файлом і потоком у системі введення/виводу стандарту ANSI мови C є вказівник на файл (file pointer). Вказівник на файл - це вказівник на інформацію, що визначає різні сторони файла: ім'я, статус, поточну позицію. Вказівник файла визначає ім'я файла на диску і його використання в потоці, асоційованим з ним. Вказівник файла - це вказівник на структуру типу `FILE`, яка визначена у файлі `STDIO.H`. У файлі `STDIO.H` визначені також наступні функції:

Функція Дія функції

| | |
|-------------------------|---|
| <code>fopen()</code> | Відкрити файл |
| <code>fclose()</code> | Закрити файл |
| <code>putc()</code> | Записати символ у потік |
| <code>getc()</code> | Прочитати символ із потоку |
| <code>fseek()</code> | Змінити вказівник позиції файла на зазначене місце |
| <code>fprintf()</code> | Форматний запис у файл |
| <code>fscanf()</code> | Форматне читання з файла |
| <code>feof()</code> | Повертає значення "істинно", якщо досягнутий кінець файла |
| <code>ferror</code> | Повертає значення "хибно", якщо виявлена помилка |
| <code>fread()</code> | Читає блок даних із потоку |
| <code>fwrite()</code> | Пише блок даних у потік |
| <code>rewind()</code> | Встановлює вказівник позиції файла на початок |
| <code>remove()</code> | Знищує файл |

Щоб оголосити вказівник на файл, використовується оператор `FILE *fput;`

Розглянемо більш докладно перераховані вище функції.

Функція `fopen()` виконує дві дії: по-перше, відкриває потік і зв'язує файл на диску з цим потоком; по-друге, повертає вказівник, асоційований із цим файлом. Прототип функції

```
FILE *fopen(char filename, char mode);
```

де `mode` - це рядок, що містить режим що відкриває файл. Можливі режими відкриття файлів перераховані нижче:



| | |
|-------|---|
| "r" | Відкрити для читання |
| "w" | Створити для запису |
| "a" | Відкрити для додавання в існуючий файл |
| "rb" | Відкрити двійковий файл для читання |
| "wb" | Відкрити двійковий файл для запису |
| "ab" | Відкрити двійковий файл для додавання |
| "r+" | Відкрити файл для читання і запису |
| "w+" | Створити файл для читання і запису |
| "a+" | Відкрити файл для додавання або створити для читання і запису |
| "r+b" | Відкрити текстовий файл для читання і запису |
| "w+b" | Створити двійковий файл для читання і запису |
| "a+b" | Відкрити двійковий файл для додавання або створити для читання і запису |
| "rt" | Відкрити текстовий файл для читання |
| "wt" | Створити текстовий файл для запису |
| "at" | Відкрити текстовий файл для додавання |
| "r+t" | Відкрити текстовий файл для читання і запису |
| "w+t" | Створити текстовий файл для читання і запису |
| "a+t" | Відкрити текстовий файл для додавання або створити для читання і запису |

Якщо ви збираєтеся відкрити файл з ім'ям test для запису, то досить написати

```
FILE* fp;  
fp=fopen("test", "w");
```

Однак рекомендується використовувати наступний спосіб відкриття файла:

```
FILE *fp;  
If((fp=fopen("test", "w"))==NULL)  
{  
    puts("Не можу відкрити файл \n");  
    exit(1);  
}
```

Цей метод визначає помилку при відкритті файла. Константа NULL визначена в STDIO. Н. Функція exit(), яку ми використовували, має прототип у файлі STDLIB.H

```
void exit(int val);
```



і припиняє виконання програми, а розмір val повертає в операційну систему (програму , що викликає). При цьому перед припиненням роботи програма закриває усі відкриті файли, звільняє буфери, зокрема виводячи всі необхідні повідомлення на екран. Крім цього, існує функція abort() із прототипом void abort(int val);

Ця функція негайно припиняє виконання програми без закриття файлів і звільнення буферів. У потік stderr вона направляє повідомлення "abnormal program termination".

Якщо файл відкритий для запису, то існуючий файл знищується і створюється новий файл. При відкритті файла для читання потрібно, щоб він існував. У випадку відкриття для читання і запису існуючий файл не знищується, однак створюється, якщо його не існує.

Запис функції в потік робиться функцією putc() із прототипом:

```
int putc(int ch, FILE *fptr);
```

Якщо операція була успішною, то повертається записаний символ. У випадку виникнення помилки повертається EOF.

Функція getc() зчитує символ із потоку, відкритого для читання функцією fopen().Прототип функції getc()

```
int getc(FILE *fptr);
```

Історично склалося так, що getc() повертає значення int. Те ж саме можна сказати про аргумент ch в описі функції putc(). Використовується ж в обох випадках тільки молодший байт. Функція повертає символ EOF, якщо досягнутий кінець файла або відбулася помилка при читанні файла. Щоб прочитати текстовий файл, можна використовувати конструкцію

```
ch=getc(fptr);  
while(ch!=EOF) { ch=getc(fptr) };
```

Коли зчитується двійковий файл, то визначити наявність кінця файла, так само як при читанні текстового файла, не вдасться. Для визначення кінця текстового файла служить функція feof() із прототипом

```
int feof(FILE *fptr).
```

Функція повертає значення "істинно", якщо кінець файла досягнутий, і "нуль" - якщо ні. Наступна конструкція читає двійковий файл до кінця файла:

```
while(! feof(fptr)) { ch=getc(fptr); }
```

Для текстових файлів ця конструкція також може бути застосована. Функція fclose(), оголошена у виді

```
int fclose(FILE * fptr);
```

повертає "нуль", якщо операція закриття файла була успішно. Інше значення означає помилку. При успішній операції закриття файла відповідні дані з буфера зчитуються у файл, відбувається звільнення блока керування файлом, асоційованого з потоком, і файл стає доступним для



Якщо відбулася помилка читання або запису текстового файлу, то відповідна функція повертає EOF. Щоб визначити, що ж у дійсності відбулося, служить функція `ferror()` із прототипом

```
int ferror(FILE *fptr);
```

який повертає значення "істинно" при виконанні останньої операції з файлами і значення "хибно" у протилежному випадку. Функція `ferror()` повинна бути виконана безпосередньо після кожної операції з файлами, інакше її повідомлення про помилку може бути втрачено.

Функція `rewind()` встановлює індикатор позиції файлу на початок файлу, визначеного як аргумент функції, прототип цієї функції має вид

```
void rewind(FILE *fptr);
```

Borland C++ визначає ще дві функції буферизованого введення /виводу: `putw()` і `getw()`. Ці функції не входять у стандарт ANSI мови C. Вони використовуються для читання і запису цілих чисел. Ці функції працюють точно так само, як `putc()` і `getc()`.

Стандарт ANSI мови C включає також функції `fread()` і `fwrite()`, що використовуються для читання і запису блоків даних:

```
unsigned fread(void *buf,int bytes, int c, FILE *fptr);
```

```
unsigned fwrite(void *buf,int bytes, int c, FILE *fptr);
```

де `buf` - вказівник на область пам'яті, звідки буде відбуватися обмін інформацією; `c` - кількість одиниць запису, кожна довжиною `bytes` байтів буде зчитано (записано); `bytes` - довжина кожної одиниці запису в байтах; `fptr` - вказівник на відповідний файл.

Читання і запис у файл необов'язково робити послідовно, можна це робити безпосередньо доступом до потрібного елемента файлу за допомогою функції `fseek()`, що встановлює вказівник позиції файлу в потрібне місце. Прототип цієї функції -

```
int fseek(FILE *fptr,long numbytes, int origin);
```

де `fptr` - вказівник на відповідний файл; `numbytes` - кількість байтів від точки відліку для встановлення поточної позиції вказівника файлу, `origin` - один із макросів, визначених у `STDIO.H`:

| Точка відліку | Макрос | Значення |
|-----------------------|----------|----------|
| Початок файлу | SEEK SET | 0 |
| Поточна позиція файлу | SEEK CUR | 1 |
| Кінець файлу | SEEK END | 2 |

Коли починається виконання програми, автоматично відкривається 5 визначених потоків. Перші три з них - стандартний (`stdin`), стандартний



вивід (stdout) і стандартний потік помилок (stderr). У звичайній ситуації вони зв'язані з консоллю, однак можуть бути перенаправлені на інший потік. Можна використовувати stdin, stdout і stderr як вказівники файлів в усіх функціях, що застосовують тип FILE.

Крім того, Borland C++ відкриває потоки stderr і stderr, асоційовані відповідно з принтером і послідовним портом комп'ютера. Ці потоки відкриваються і закриваються автоматично.

Стандарт ANSI включає також функції fprintf() і fscanf(), що працюють аналогічно функціям printf() і scanf(), за тим винятком, що зв'язані з файлами на диску. Прототипи цих функцій відповідно

```
int fprintf(FILE *fptr, const char*string,...);  
і int fscanf(FILE *fptr, const char*string,...);
```

де fptr - вказівник на файл, що повертається функцією fopen().

Функція remove() знищує зазначений файл. Прототип цієї функції :

```
int remove(char *filename);
```

Функція повертає значення 0 при успішній операції і ненульове значення в іншому випадку.

Оскільки мова C пов'язана з операційною системою UNIX, то в системі Borland C++ створена друга система введення/виводу. Ця система відповідає стандарту UNIX. Прототипи функцій знаходяться у файлі IO.H. Цими функціями є:

```
read() - читає буфер даних,  
write() - пише в буфер даних,  
open() - відкриває файл,  
close() - закриває файл,  
fseek() - пошук визначеного байта у файлі,  
unlink() - знищує файл.
```

Опис цих функцій можна знайти в документації посібників Borland C++.

Поради

Складаючи програми для роботи з файлами слід пам'ятати, що:

- в програмі, яка виконує операції читання з файла, або запису в файл повинен бути оголошений вказівник на тип FILE;
- для того, щоб файл був доступним, його необхідно відкрити, вказавши, для виконання якої дії відкривається файл: читання, запису чи поновлення даних, а також тип файла (двійковий чи текстовий)ж
- при роботі з файлами можливі помилки, тому рекомендується при допомозі функції feof перевіряти результат виконання операцій з файлами (fopen());
- читання даних з текстового файла можна виконувати з



допомогою функції `fscanf()`, запис – `fprintf()`;

□ після завершення роботи з файлом потрібно його обов'язково закрити (функція `fclose()`);

25.3 Опис лабораторного обладнання

1. Персональний комп'ютер.
2. Програмне забезпечення: Borland C++ 3.1.

25.4 Програма роботи

1. Запустити середовище Borland C++.
2. Скласти алгоритм програми до завдання 1 (додаток 1) згідно свого варіанту.
3. Написати програму до завдання 1 (додаток 1) згідно свого варіанту.
4. Скласти алгоритм програми створення та читання текстових файлів (додаток 2) згідно свого варіанту.
5. Скласти програму створення та читання текстових файлів (додаток 2) згідно свого варіанту.

Вимоги до програм

- вхідні дані ввести оператором введення ;
- привести результат виконання програм згідно завдання.

Вимоги до звіту

Звіт повинен включати в себе:

- Титульний лист із зазначенням номеру варіанту
- Мету роботи
- Програму роботи
- Завдання №1, текст програми, результат виконання програми, скопійований з монітора комп'ютера
- Завдання №2, текст програми, результат виконання програми, скопійований з монітора комп'ютера

25.5 Контрольні запитання

1. Які методи введення/виводу підтримує середовище Borland C.
2. Що називають файлом і потоком.
3. Які Ви знаєте типи потоків.
4. Які Ви знаєте функції введення.
5. В яких заголовних файлах знаходяться прототиби функцій



введення/виводу.

6. Які Ви знаєте функції виводу.
7. Які функції здійснюють введення/вивід рядка символів.
8. Що таке вказівник на файл, його прозначення.
9. Перерахуйте основні функції роботи з файлами.
10. Які Ви знаєте режими відкриття файлів, яким чином вони задаються.
11. Яке призначення функції exit().
12. Яка функція призначена для закривання файлів.
13. Які функції призначені для читання і запису блоків даних.
14. Яка функція встановлює доступ до конкретного елемента файла.





Варіанти:

- Компонентами файла g є натуральні числа від 96 до 158. Знайти:
 - кількість парних чисел серед його компонент;
 - кількість непарних чисел серед його компонент.
- Дано файл f , компоненти якого є натуральні числа від 1 до 150. Записати у файл g всі компоненти файла f , які є парними числами.
- Дано файл f , компоненти якого є натуральні числа від 1 до 150. Записати у файл g всі компоненти файла f , які діляться на 3 і не діляться на 7.
- Дано файл f , компоненти якого є натуральні числа від 1 до 150. Записати у файл g всі компоненти файла f , які є повними квадратами.
- Дано файл f , компоненти якого A_1, \dots, A_n ($n=9$) отримуються за формулою: $A_i = 1.5 * i$, ($i=1 \dots 9$). Створити файл f .
- Дано файл даних f : (1,3,-4,-21,5,9,-3,11,-20). Переписати компоненти файла f у файл g так, щоб у файлі g спочатку йшли додатні елементи, а потім від'ємні.
- Дано натуральне число $n=10$. Записати в файл g послідовність чисел b_1, b_2, \dots, b_n , визначених за формулою $b_i = 2^i / i!$, $i=1, 2, 3, \dots, n$.
- Послідовність x_1, x_2, \dots, x_n утворена за законом $x_i = (i - 0.1) / i^3 + tg 2i$ ($i=1 \dots 9$). Дано дійсне число $\varepsilon = 0.05$. Записати у файл f ті члени послідовності, для яких виконується умова $x_i < a$.
- У файлі даних 1, 2, 3, 8, 10, 12, 17.7, 14, 5.6 підрахувати кількість елементів та обчислити суму їх квадратів.
- У файлі даних 7,6,4,3,2,1,0,14,8,19,41,105 підрахувати кількість елементів та знайти їх середнє значення.
- Дано числовий файл f : (7,6,5,4,-8,5,3,2,1). Записати в числовий файл g компоненти файла f у зворотному порядку.
- Дано символний файл f : смородина, яблука, груші, сливи, абрикоси, порічки. Записати його компоненти в файл g у зворотному порядку.
- Дано числовий файл f : (3,4,5,2,4,3,4,3,7). Вияснити, чи є перші два компоненти файла парними чи непарними.
- Записати в файл f послідовність чисел Фібоначі: U_1, U_2, \dots, U_n ($n=12$). Послідовність чисел Фібоначі: U_1, U_2, \dots, U_n утворюється за законом $U_0 = 0, U_1 = 1, \dots, U_i = U_{i-1} + U_{i-2}; (i = 2, 3, \dots)$.



15. Дано числа 7,-5,3,4,6,1,8,100. Записати їх у файл f. Поміняти місцями найбільший найменший елементи в даному файлі. Записати ці дані в новий файл g.

16. Записати наступні дані у файл послідовного доступу: (7.8, -5.23, 1.2, 0.71,43,7.8,8,9.76). Знайти: а) суму компонент файла; б) добуток компонент файла; в) останню компоненту файла.

17. Записати наступні дані у файл послідовного доступу: (-55,14.2,70.7, 3.3,-8.9,17.05,-0.6,-10). Знайти: а) найменше значення з парними індексами; б) різницю першої та останньої компоненти файла.

18. Записати наступні дані у файл послідовного доступу: (-5,51.2,26.1,4.3, -7.8,5611.03,-0.65,610). Записати найбільше із значень модулів компонент з парними індексами.

19. Записати наступні дані у файл послідовного доступу: (-5.5,3.2,12.1,-4.3, 7.9,14.03,5.5,34). Знайти: а) кількість чисел файла, які знаходяться в інтервалі [0.2; 0.9]; б) максимальне значення компонент файла, які стоять на парних місцях.

20. Записати символічні дані у файл g: золото, срібло, платина, цинк, алюміній. Отримати копію даного файла g.

21. Наступні числові дані записати в два файли: 1) - 4.76,11.8,0.44, 45.6, 23.6,1.3 --> файл f1; 2)4.5,56.7,3.6,4.6,47.5,-5.4 --> файл f2. Записати дані файла f2 в кінець файла f1.

22. Дано два списки: 1) каучук, нейлон, поролон, капрон; 2) залізо, олово, свинець, мідь. Записати їх в текстові файли f1, f2. Дописати компоненти файла f1 у кінець файла f2.

23. Наступні дані записати у файл f: 1,3,4,2,5,7,8,16,44,77,10. Записати у файл g1 всі парні числа файла f, а у файл g2 - всі непарні числа файла f. При цьому зберегти порядок слідування чисел.

24. Дано файл даних: 4,7,9,44,18,21,43,72,108,14,66. Записати у масив А ті дані, які кратні 3, та знайти їх суму.

25. Наступні дані записати у два файли:
1) 7,11,0.6,1.33,40,13,17,20 --> файл f1;
2) 0.4,33,1.2,0.5,43,78,90 ---> файл f2.
Записати елементи файла f2 у файл f1 після 3-го елемента.

26. В файлі даних: (7,44,17,33,46,74,21,100,43) поміняти місцями 2-й і передостанній компоненти.

27. Компоненти файла g є натуральні числа від 96 до 158. Знайти: а) кількість парних чисел; б) кількість подвоєних непарних чисел серед його компонент.

28. Дано файл f, компоненти якого є натуральними числами від 1 до 150. Отримати у файлі g всі компоненти файла f, які є парними числами .



29. При наборі даних текстового файлу g, який складається з 12 прізвищ, оператор допустив помилку, набираючи крізь замість символу “o” цифру “0”. Виправити помилку, допущену оператором.

30. В файлі F зберігаються дані : задовільно, добре, задовільно, відмінно, добре, незадовільно, задовільно, добре. Виправити другу оцінку “задовільно” на “добре”.

31. Компоненти файлу g натуральні числа від 1 до 100. Записати у файл f “решето” Ератосфена (всі прості числа).

Додаток 2

Варіанти:

1. Сформувати файл “BOOK”, який містить інформацію про книги вашої бібліотеки. Інформація повинна містити прізвище автора, назву книги, видавництво та рік видання. Використовуючи сформований файл, роздрукувати інформацію про книги, видані у видавництві “Просвіта”.

2. Сформувати файл , який містить інформацію про побутові холодильники: назва холодильника, вартість, об’єм холодильної камери, завод виготовлювач. Використовуючи сформований файл, роздрукувати інформацію про холодильники, вартість яких понад 55000 гривень.

3. Під час футбольної гри формується файл, який включає прізвище гравця та кількість набраних за гру балів. Використовуючи сформований файл, роздрукувати прізвища 3 найрезультативніших гравців команди.

4. Сформувати файл, який містить інформацію про потяги, які слідують до Києва (номер потяга, повна назва, час в дорозі). Використовуючи сформований файл, роздрукувати інформацію про потяги, час перебування в дорозі яких не перевищує 6 годин.

5. Сформувати файл , який містить інформацію про потяги, які відправляються зі станції Здолбунів (враховуючи транзитні): номер потяга, станцію призначення, час відправки, час в дорозі. Використовуючи сформований файл, роздрукувати інформацію про потяги, які йдуть до Львову.

6. Сформувати файл, що містить інформацію про студентів, які народилися влітку (червень, липень, серпень).

7. Сформувати файл “stud”, який має наступну структуру: прізвище студента, рік народження, стать. Роздрукувати, використовуючи файл, список студентів чоловічого роду та вказати їх вік. В кінці списку надрукувати середній вік студентів.

8. Сформувати файл “EXAM” за результатами екзаменаційної сесії (три екзамени). Інформація про студентів вводиться у символічному вигляді



в наступному порядку: прізвище **N1**N2**N3, де Ni-оцінка. Використовуючи файл роздрукувати результати сесії у вигляді таблиці. Передбачити друк шапки таблиці з назвами дисциплін.

9. Протокол лижних гонок записати у файл "SCI". Для кожного учасника ввести: прізвище, час старту та фінішу (година, хвилина, секунда). Використовуючи сформований файл, роздрукувати прізвища учасників, котрі виконали норми ГПО.

10. Сформувати файл "CAR", який містить інформацію про власників автомобілів: прізвище, марка, колір. Використовуючи сформований файл, роздрукувати відомості про власників, які мають автомобілі "Жигулі" зеленого кольору.

11. Сформувати файл, який містить дані про книги вашої особистої бібліотеки: прізвище автора, назва книги, видавництво, рік видання кількість сторінок. Використовуючи сформований файл, роздрукувати інформацію про книги, випущені у видавництві "Мир", а також підрахувати загальну кількість таких книг.

12. Сформувати файл, який містить інформацію про магнітофони: марка, його вартість, клас. Використовуючи сформований файл, роздрукувати інформацію про магнітофони першого класу.

13. Сформувати файл - телефонний довідник. Інформація повинна містити прізвище, ім'я та по-батькові абонента, номер телефону. Роздрукувати весь довідник.

14. Сформувати файл "FRIEND" із прізвищ і дат народження ваших друзів. Використовуючи сформований файл, роздрукувати прізвища тих, хто народився взимку.

15. У шаховому турнірі приймають участь 10 шахістів. Сформувати файл який включає прізвища та результати ігор (перемога - 1, нічия -1, програш - 0). Використовуючи сформований файл, обробити результати чемпіонату і роздрукувати назви команд, які зайняли призові місця а також кількість перемог кожної команди.

16. В чемпіонаті з футболу приймає участь 16 команд. Сформувати файл команд і результатів матчу (виграш - 2 очка, нічия -1, програш - 0). Використовуючи сформований файл, обробити результати чемпіонату і роздрукувати назви команд, які зайняли призові місця, а також кількість перемог кожної команди.

17. В журналі обліку відвідувань щодня по кожному з предметів відмічають пропуск заняття студентами. Сформувати файл, який включає прізвище та дату відвідування занять з одного предмета (1 - присутній, 0 - відсутній) кожним студентом групи. Використовуючи сформований файл, сформувати список тих студентів, які мають більше 5 пропусків.



18. Сформувати файл, який включає прізвища та оцінки студентів на протязі семестру з дисципліни “Обчислювальна техніка”. Використовуючи сформований файл, роздрукувати прізвища тих студентів, в котрих середній бал з дисципліни - 4.

19. До 20 спортивних журналістів звернулись із проханням назвати 3 кращих футболістів сезону. Сформувати файл, який включає прізвища футболістів, кількість набраних очків від кожного журналіста (3 - перше місце, 2 - друге місце, 1 - третє місце). Використовуючи сформований файл, визначити 3 кращих гравців.

20. Сформувати файл, який включає прізвища та посади викладачів, які викладають дисципліну “Обчислювальна техніка” на всіх факультетах інституту. Використовуючи сформований файл, роздрукувати прізвища тих викладачів, які починаються на букву “В”.

21. Сформувати файл, який містить прізвища та імена студентів своєї групи. Використовуючи сформований файл, роздрукувати прізвища тих студентів, які починаються на букву “К”.

22. Сформувати файл, який містить прізвища та ініціали викладачів, які викладають у вашій групі, та відповідні предмети. Роздрукувати прізвища викладачів з кафедри електротехніки і автоматики.

23. Сформувати файл, який містить результати вашої атестації. Використовуючи створений файл, роздрукувати ті предмети, з яких ви отримали “5”. Якщо ви не отримали жодної п’ятірки, то роздрукуйте предмети, з яких ви отримали оцінку “4” або “3”.



26.1. Мета роботи

Вивчити організацію однотипних даних у вигляді двовимірних масивів на мові C, їх оголошення, способи доступу до елементів та програмування алгоритмів обробки.

26.2. Теоретичні відомості

Масивом називають кінцеву сукупність елементів одного типу.

Оголошення k-мірного масиву здійснюється згідно наступного формату:

<тип> <ім'я масиву> [i1][i2]*...*[ik]

де ім'я масиву є правильним ідентифікатором, i1, i2,...,in – максимальна кількість елементів масиву по кожному виміру.

Кожен індекс масиву повинен записуватися в окремих квадратних дужках []. Кількість індексів масиву необмежена. За замовчуванням загальний розмір масиву не повинен перевищувати одного сегменту (64К). Для оголошення масиву, більшого за 64К, між типом та ім'ям масиву вказується модифікатор huge.

Індекс масиву по кожному виміру змінюється від 0 до ім-1, де m=1,...,k.

В якості типу масиву можуть вказуватися:

- 1) один із основних типів (int, char, float, double або синоніми);
- 2) тип іншого масиву;
- 3) вказівник, у тому числі вказівник на функцію;
- 4) структура (structure);
- 5) об'єднання (union).

Наприклад:

```
int mas[10]; /*масив із 10 цілих */
char line[80]; /*масив із 80 символів*/
float a[5][4]; /*двовірний масив із 5x4 елементів типу float*/
double b[7][7][7]; /*трьохвимірний масив із 7x7x7 елементів типу
double*/
int *x[10]; /*масив із 10 вказівників на тип int*/
```

При оголошенні масиву допускається його ініціалізація. Початкові значення елементів повинні задаватися через кому у фігурних дужках { }. Значення елементів по кожному окремому виміру може охоплюватися фігурними дужками, між якими ставиться кома, наприклад:



```
int year[12]={31,28,31,30,31,30,31,31,30,31,30,31};
float mas[2][3]={0.0,0.1,0.2},{1.0,1.1,1.2};
Остання ініціалізація еквівалентна наступній:
float mas[2][3]={0.0,0.1,0.2,1.0,1.1,1.2}.
```

Кількість елементів не може бути більшою від вказаної розмірності. Якщо кількість елементів менша від вказаної розмірності, то присвоєння відбувається тільки для початкових елементів, а інші елементи приймають нульові значення, якщо масив оголошений як зовнішній або статичний, і приймають невизначені значення в інших випадках:

```
static int vec[10]={1,8,3};/*Перші три елементи приймуть задані значення, а наступні сім – нульові значення*/
auto float matr[3][4]={0.0}, {1.0,1.1},{2.0,2.1,2.2}};
/*елементи з індексами [0][0],[1][0],[1][1],[2][0],[2][1],[2][2] приймуть значення, а всі інші – невизначені значення*/.
```

При явній ініціалізації масиву, його розмірності можна не вказувати. Тоді розмірність масиву визначається кількістю заданих елементів, наприклад:

```
int i_array []={4,-2,7,10,-3,5,3}
/*оголошений та ініціалізований масив із семи цілих чисел*/
```

При явній ініціалізації двовимірних масивів перший індекс може бути опущений, а другий вказується обов'язково:

```
float f_mas[][4]={1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

В даному прикладі оголошений масив, що складається з 3-х рядків і 4-стовпчиків. Наведене оголошення еквівалентне наступному:

```
float f_mas[][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

Згідно ANSI-стандарту ініціалізувати масив можна будь-де, у тому числі і всередині функції для старих компіляторів. Щоб ініціалізувати масив в середині функції, його потрібно оголосити як static:

```
static float vec[]={5.2,6.3,8.4};
```

У пам'яті елементи масиву розміщуються у неперервній області, так що правий індекс змінюється першим.

Приклад розміщення у пам'яті масиву int matrix[2][3]

| [0][0] | [0][1] | [0][2] | [1][0] | [1][1] | [1][2] |
|--------|--------|--------|--------|--------|---------|
| 0 | 2 | 4 | 6 | 8 | 12 байт |

посилання на елемент масиву відбувається вказуванням імені масиву і його індексів. Індеси повинні бути константами, змінними, або виразами цілого типу. Кожен індекс береться в окремі квадратні дужки, наприклад:

```
i_array[3]; /*звертання до четвертого елемента масиву i_array*/
```



`matrix[1][2]; /*звертання до елемента масиву matrix, який знаходиться на перетині другого рядка і третього стовпчика*/`

Інший спосіб доступу до елементів масиву полягає у використанні вказівників.

Ім'я одного масиву є вказівником на його нульовий елемент, наприклад:

```
int vec{5};
vec---→
vec[0]      vec[1]      vec[2]      vec[3]      vec[4]
vec=&vec[0];
vec+i=&vec[i];
*vec=vec[0];
*(vec+i)=vec[i];
```

Відповідно до оголошення двовимірного масиву `int matrix[2][3];` позначення `matrix[i]`, де `i=0,1`, визначає адресу розміщення в пам'яті `i`-го рядка, а ім'я масиву `matrix` визначає вказівник на початок масиву `int matrix[2][3]`. Ім'я масиву `matrix` є сумісне з типом вказівника на масив з трьох цілих чисел: `int (*vec)[3];`

| вказівними на рядки | Елементи масиву | | |
|------------------------|---------------------|---------------------|---------------------|
| <code>matrix[0]</code> | <code>[0][0]</code> | <code>[0][1]</code> | <code>[0][2]</code> |
| <code>matrix[1]</code> | <code>[1][0]</code> | <code>[1][1]</code> | <code>[1][2]</code> |

Значення вказівників `matrix` та `*matrix` є однакові, але не сумісні по типу. Правила сумісності та еквівалентності приведені нижче:

```
matrix=&matrix[0]=matrix[][m];
matrix+i=&matrix[i];
*matrix=matrix[0]=&matrix[0][0];
*(matrix+i)=matrix[i]=&matrix[i][0];
*(matrix+i)+j=matrix[i]+j=&matrix[i][j];
**matrix=*matrix[0]=matrix[0][0];
**(matrix+i)=*matrix[i]=matrix[i][0];
*(*(matrix+i)+j)=*(matrix[i]+j)=matrix[i][j];
```

Аналогічно для оголошення трьохмірного масиву `int fx [5][4][8];`

позначення `fx[i]` визначає адресу розміщення в пам'яті `i`-го двовимірного масиву розмірністю `4x8` елементів, а позначення `fx[i][j]` визначає адресу `8`-елементного `j`-го рядка `i`-го двовимірного під масиву.

Враховуючи, що перші `p` індексів масиву (`p<k`) визначають адресу відповідного підмасиву, для звертання до елементів масиву чи їх вказівників допускається форма, аналогічна адресуванню комірок пам'яті через вказівники:



Наприклад:

(ip)[ім'я масиву [i1][i2]/ *...*/[i(p-1)]]. де ip - змінна або вираз цілого типу, але не константа.

Наприклад:

(i)[vec] /* еквівалентно звертанняу vec[i] */ (j)[matrix[i]] /* еквівалентно звертанняу matrix[i][j]*/.

При роботі з масивами потрібно пам'ятати, що ім'я масиву є константою, тому не дозволяються операції модифікації значення, яке визначається цією константою, наприклад:

```
int a[5],y,n,  
int *z; /* наступні операції не дозволяються *  
a=y;  
a++;  
a+=n;  
z=&a;
```

На відміну від мови Паскаль, на мові Сі не можна присвоїти цілком один масив іншому такого ж типу та розмірності. Копіювання масивів здійснюється поелементно за допомогою операторів циклу, масивів вказівників

На мові Сі допускається використання масивів вказівників на всі типи даних, наприклад:

```
int *r[3];
```

де r[3] - масив, що містить адреси трьох елементів типу int.

Для доступу до значення, яке знаходиться за адресою r[i], необхідно використати операцію розіменування індексованого вказівника *r[i]. Робота із масивом вказівників демонструється наступним прикладом:

```
int i;  
float x1=1, x2=2, x3=3, sum=0, *r[3];  
r[0]=&x1;  
r[1]=&x2;  
r[2]=&x3;  
for(i=0;i<3;i++) sum = sum+*r[i];  
printf("Сума = %fn",sum);
```

1) Ввід-вивід масивів. Ввід та вивід числових масивів даних здійснюється поелементно за допомогою функцій форматowanego вводу-виводу.

Для вводу елементів масиву використовується бібліотечна функція scanf, яка має наступний формат:

```
int scanf("форматний рядок", список вказівників на елементи масиву);
```



Для виводу елементів масиву використовується бібліотечна функція `printf`, яка має такий формат:

```
int printf( "форматний рядок", список елементів масиву);
```

Форматний рядок для вводу-виводу повинен містити специфікації форматів списку даних. Специфікація формату повинна починатися із символу процента (%), після якого вказується один із символів:

і або d - для вводу-виводу десяткового числа зі знаком;

u - для вводу-виводу десяткового без знаку;

v - для вводу-виводу вісімкового без знаку;

x - для вводу шістнадцяткового зі знаком і виводу шістнадцяткового без знаку;

f- для вводу-виводу числа у форматі з фіксованою крапкою; ,

e або E - для вводу-виводу числа у форматі з плаваючою крапкою;

g або G - для вводу-виводу числа у форматі з фіксованою або плаваючою крапкою в залежності від можливої точності представлення.

Для вводу довгого цілого (long) або дійсного з подвійною точністю (double) перед символом формату вказується символ l. Для вводу типу long double перед символом формату вказується символ L.

У форматний рядок можуть входити інші символи таблиці ASCII.

Якщо такі символи включені у форматний рядок функції `scanf`, то вони повинні зустрітися у вхідному потоці. Якщо ж додаткові символи включені у форматний рядок функції `printf`, то вони будуть поміщені у вихідний потік. Таким чином можна виводити допоміжні повідомлення. Приклад вводу-виводу елементів одновимірною масиву:

```
#include <stdio.h>
#define N 10
main( )
{int i;
 float vec[N];          // Ввід
 for(i=0;i<N;i++) scanf("%f",&vec[i]);
 .....//Обробка за програмою
 for(i=0;i<N;i++) printf("%f",vec[i]); // Вивід
 printf("\n");
}
```

Приклад вводу-виводу елементів двовірною масиву:

```
#include <stdio.h>
#define N 3
#define M 4
main()
```



```
int i,j,mas[N][M];
// Ввід
for(i=0;i<N;i++)
for(j=0;j<M;j++)
scanf("%d",&mas[i][j]);
{ } // Обробка по програмі
// Вивід по рядках
for(i=0;i<N;i++)
{
for(j=0;j<M;j++)
printf(" %d ",mas[i][j]);
}
printf("\n");
}
```

26.3. Опис лабораторного обладнання

3. Персональний комп'ютер.
4. Програмне забезпечення: GCC MinGW 3.1.

26.4. Програма роботи

6. Запустити середовище GCC MinGW.
7. Скласти алгоритм програми до завдання (додаток 1) згідно свого варіанту.
8. Написати програму для завдання (додаток 1) згідно свого варіанту.
9. Скласти алгоритм програми для визначення числових характеристик матриці (додаток 2) згідно свого варіанту.
10. Скласти програму для визначення числових характеристик матриці (додаток 2) згідно свого варіанту.

Вимоги до програм:

- вхідні дані ввести оператором введення;
- на друк вивести параметри масивів згідно завдання.

Вимоги до звіту

Звіт повинен включати в себе:

- Титульний лист із зазначенням номеру варіанту
- Мету роботи
- Програму роботи



- Завдання №1, текст програми, результат виконання програми, скопійований з монітора комп'ютера
- Завдання №2, текст програми, результат виконання програми, скопійований з монітора комп'ютера

26.5. Контрольні запитання

1. Як оголошується масив?
2. Які можливі типи масивів?
3. Який вигляд буде мати масив, якщо при його ініціалізації вказати кількість елементів, меншу за розмірність масиву?
4. Які є способи доступу до елемента масиву?
5. Як скопіювати один масив в інший?
6. Як описати масив вказівників?
7. Для чого здійснюється розіменування?
8. Як здійснюється ввід-вивід масивів?
9. Назвіть відомі вам специфікатори формату.





Варіанти:

1) Задана матриця

$$A = \begin{pmatrix} 2,1 & 1,5 & 0,3 \\ 0,2 & 0,5 & 0,4 \\ 1,3 & 2,0 & 3,1 \end{pmatrix}.$$

Обчислити елементи нової матриці, які визначаються з допомогою формули

$$x_{ij} = \frac{a_{ij}}{S},$$

де S - слід матриці A (слідом квадратної матриці називають суму значень її елементів, які стоять на головній діагоналі).

2) Для заданого двомірного масиву знайти середнє арифметичне \bar{x} і середнє квадратичне відхилення за формулами

$$\bar{x} = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m x_{ij}; \quad \delta = \sqrt{\frac{1}{mn-1} \sum_{i=1}^n \sum_{j=1}^m (x_{ij} - \bar{x})^2};$$

$$x = \begin{pmatrix} 1,4 & 1,2 & 1,1 & 1,5 \\ 1,7 & 1,5 & 1,3 & 1,6 \end{pmatrix}.$$

3) Знайти найбільший елемент матриці $z = \{z_{ij}\}$, елементи якої обчислюються за формулою

$$z_{ij} = x_i y_j, \quad i = \overline{1,5}, \quad j = \overline{1,3},$$

$$x = (2,1; 0,5; -1,3; 4,2; 0,1), \quad y = (0; -0,2; 1,5).$$

4) В матриці B знайти суму елементів, що лежать вище головної діагоналі:

$$B = \begin{pmatrix} 3 & 4 & -1,5 & 0 \\ 2 & -1,2 & 3 & 1 \\ 1 & 2 & 0,3 & 4 \\ 1,1 & 2,5 & 0 & 1 \end{pmatrix}.$$

5) В матриці B (див. приклад 4) знайти добуток елементів головної діагоналі.

6) Знайти квадрат найбільшого елемента масиву:



$$Y = \begin{pmatrix} 3 & 0 & 0,1 & 2,5 \\ -1 & 2 & 5 & 1,3 \\ 4 & -2,5 & 3 & 0,5 \end{pmatrix}.$$

7) Знайти суму двох матриць:

$$A = \begin{pmatrix} 3 & 1 & 5 \\ -2 & 0 & 3 \end{pmatrix}, B = \begin{pmatrix} 1 & 3 & 4 \\ 3 & 0 & 2 \end{pmatrix},$$

користуючись формулою

$$c_{ij} = a_{ij} + b_{ij}, \quad i = \overline{1,2}, \quad j = \overline{1,3}.$$

8) В матриці В поміняти місцями 1-й і 2-й стовпці:

$$B = \begin{pmatrix} 2 & 0 & 3 & 4 \\ 1 & 2 & 0 & 5 \\ -2 & 3 & 4 & 5 \end{pmatrix}.$$

9) Знайти суму елементів матриці $A = \{a_{ij}\}$, що задовольняють умові:

$-1 \leq a_{ij} \leq 1$. Матриця А задана:

$$A = \begin{pmatrix} -0,7 & 1,2 & 3,5 & 0,3 \\ 0,5 & 0,7 & -1,1 & -3,5 \\ 0,9 & 0,8 & 2,7 & 1 \end{pmatrix}.$$

10) Задана матриця $z = \begin{pmatrix} -1,5 & 2 & 3,5 \\ 0,5 & 0 & 1,2 \\ -2 & 1 & 0,7 \end{pmatrix}$.

Знайти $c = \frac{z_{\min} + z_{\max}}{2}$, де $z_{\min} = \min\{z_{ij}\}$, $z_{\max} = \max\{z_{ij}\}$, $i = \overline{1,3}$, $j = \overline{1,3}$.

11) Скласти таблицю значень функції $z = x^3 e^{-yx} \sin x$ для таких значень аргументів: $x = \{1; 1,2; -0,3; 0,7\}$; $y = \{3; 8; 2; -1,5\}$.

12) В квадратній матриці С підрахувати і вивести на друк окремо суму додатних та від'ємних елементів, підраховавши при цьому їх кількість.

$$C = \begin{pmatrix} -1 & 2,5 & 1,6 & -4 \\ 0 & 0,5 & 0 & 1,3 \\ 1 & -3 & 0,4 & 0 \end{pmatrix}.$$

13) В заданому масиві підрахувати окремо суму кожного стовпця. З отриманих сум сформувати масив X.



$$A = \begin{pmatrix} 3 & 4 & 1,3 & 0,7 \\ 1 & 3 & 2,2 & 0,8 \\ 5 & 2 & 3,5 & 0,6 \end{pmatrix}.$$

14) Обчислити елементи матриці $c = \{c_{ij}\}$,

$$\text{де } c_{ij} = \begin{cases} a_{ij} & , \text{ якщо } |a_{ij}| > |b_{ij}| \\ b_{ij} & , \text{ якщо } |a_{ij}| \leq |b_{ij}| \end{cases},$$

$$A = \begin{pmatrix} 2,5 & -3 & 6,2 \\ -0,1 & 4,2 & 2,8 \\ 2 & 7,5 & -1,8 \end{pmatrix}, \quad B = \begin{pmatrix} -2 & 2,3 & -7,1 \\ 0,8 & 3,6 & -5,6 \\ 3 & 6 & 4 \end{pmatrix}.$$

15) Провести перетворення матриці А таким чином: додатні елементи замінити на число 2, від'ємні - на 1, нульові - на 3.

$$A = \begin{pmatrix} -1,5 & 0,6 & -2,1 \\ 0,5 & 0 & 0,7 \\ 0 & -1,3 & 0 \end{pmatrix}.$$

16) Транспонувати матрицю (зробити рядки стовпцями).

$$Z = \begin{pmatrix} 2 & 5 & 1 & 0 \\ 0 & 3 & -2 & 4 \\ 1 & 2 & 0 & 3 \end{pmatrix}.$$

17) Знайти добуток двох матриць $A = \{a_{ij}\} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 2 & 1 \end{pmatrix}$ і

$$B = \{b_{ij}\} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 5 \end{pmatrix} \text{ користуючись формулою } c_{ik} = \sum_{j=1}^3 a_{ij} b_{jk}.$$

18) Вивести на друк у вигляді масиву X всі невід'ємні елементи матриці А.

$$A = \begin{pmatrix} -2 & 0 & 1 & 5 \\ 1,3 & -1 & 3 & -2 \\ 2,1 & 5 & 0 & 1 \end{pmatrix}.$$

19) Обчислити компоненти вектора $C = A \cdot b$, де $A = \{a_{ij}\}$,

$$b = \{b_j\}, \quad c_i = \sum_{j=1}^3 a_{ij} b_j,$$



$$A = \begin{pmatrix} 1,8 & 0,1 & 1,3 \\ 0,9 & -1 & 3,1 \\ -1,5 & 0,5 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 3,2 \\ 2 \end{pmatrix}.$$

20) Задана матриця А. Утворити нову матрицю $X = \{x_{ij}\}$, кожен елемент якої дорівнює добутку відповідного елемента матриці А на номер рядка: $x_{ij} = i \cdot a_{ij}$.

$$A = \begin{pmatrix} 1 & 0,7 & 3,2 \\ -2 & 0,3 & 1,2 \\ 2 & 3,7 & 0,5 \end{pmatrix}.$$

21) Перетворити матрицю, замінивши додатні елементи номера рядка, в якому вони знаходяться, від'ємні елементи - номером стовпчика, в якому вони знаходяться, нульові елементи - сумою відповідного рядка і стовпця.

$$B = \begin{pmatrix} -1 & 2 & 0 & 4 \\ 3 & -5 & 4 & 0 \\ 0 & 1 & -1 & 2 \end{pmatrix}.$$

22) Сформувати масив X з квадратів тих елементів матриці, які стоять на побічній діагоналі.

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix}.$$

23) Всі елементи другого стовпця матриці В збільшити в 2 рази.

$$B = \begin{pmatrix} 3,1 & 1,2 & 0,5 & -1,3 \\ 2 & -1,5 & 0,3 & 2,1 \\ 1,5 & 0,5 & 0 & 1,8 \end{pmatrix}.$$

Вивести на друк стару і нову матрицю.

24) В матриці Y знайти найбільший елемент і вивести на друк разом з його індексами.

$$Y = \begin{pmatrix} 3,1 & 2,5 & 6 \\ -1 & 0 & 1 \\ 2,5 & 0,5 & 3 \end{pmatrix}.$$



25) Задана матриця А. Отримати нову матрицю шляхом ділення всіх її елементів на найбільший по модулю елемент.

$$A = \begin{pmatrix} 3 & 1,5 & -1,3 \\ -8 & 5 & 0 \end{pmatrix}.$$

26) Знайти найменшу компоненту вектора $\bar{a} = \{a_i\}$, де $a_i = \sum_{j=1}^4 x_{ij}$,

$$X = \begin{pmatrix} 3,2 & 0,7 & 1,3 & 1,5 \\ 0,4 & -2,1 & 2,5 & 1,2 \\ 1,3 & 0,4 & 3,1 & 1,8 \end{pmatrix}.$$

27) Задана матриця А. Сформувати нову матрицю В, кожен елемент якої $b_{ij} = a_{ij} \cdot a_{\max}^2$, де a_{\max} – найбільший елемент другого рядка.

$$A = \begin{pmatrix} 1 & 0,5 & 6 & 2 \\ 0,3 & 2 & -1 & 0 \\ 4 & -1 & 0 & 3 \end{pmatrix}.$$

28) Провести перетворення заданої матриці таким чином: кожен додатній елемент замінити літерою А, нульовий - D, від'ємний - O.

$$A = \begin{pmatrix} 1 & 0 & 2 \\ -3 & 0 & 5 \end{pmatrix}.$$

29) Обчислити елементи матриці $C = \{c_{ij}\}$, що є різницею двох матриць, користуючись формулою $c_{ij} = a_{ij} - b_{ij}$,

$$A = \begin{pmatrix} -1 & 2 & 3,1 \\ 0 & 4 & 5,2 \\ 3 & 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1,5 & 6 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

30) Впорядкувати елементи другого стовпця матриці по зростанню. Вивести стару і нову матриці на друк.

$$A = \begin{pmatrix} 0 & 2 & 1 \\ 3 & -1 & 2 \\ 1 & 0 & 5 \end{pmatrix}.$$



Варіанти:

1. Із матриці $A(n,n)$ ($n \leq 6$) отримати нову матрицю $B(n,n)$ шляхом ділення всіх елементів матриці A на її максимальний по модулю елемент.
2. В матриці $A(6,8)$ необхідно поміняти рядок, який містить мінімальний елемент, на рядок що містить максимальний елемент. Вважати, що ці елементи єдині.
3. Із матриці $A(m,n)$ ($m \leq 5, n < 6$) отримати числа a_1, \dots, a_m , де a_i - значення першого по порядку додатного елемента i -го рядка.
4. Транспонувати матрицю $A(m,n)$ ($m \leq 4, n < 6$) і роздрукувати отриману матрицю.
5. Координати m векторів задані матрицею $A(m,n)$ ($m \leq 6, n < 7$). Необхідно обчислити довжини цих векторів, роздрукувати значення і серед цих знайти і вказати номер вектора мінімальної довжини.
6. Провести таке перетворення матриці $A(m,n)$ ($n \leq 4$) при якому всі додатні елементи замінюються на суму відповідних індексів, а від'ємні - на добуток індексів.
7. Знайти мінімальний елемент матриці $A(m,n)$ ($m \leq 5$) і роздрукувати його. Елементи матриці, що лежать нижче головної діагоналі, замінити мінімальним елементом.
8. В матриці $A(6,6)$ знищити 4-й рядок і роздрукувати отриману матрицю.
9. Провести таке перетворення матриці $A(m,n)$ ($m \leq 5, n < 7$), при якому останній стовпчик займе місце першого, а всі інші змістяться на один стовпчик вправо.
10. Координати n векторів задані матрицею $A(m,n)$ ($m \leq 5, n \leq 6$). Обчислити довжини цих векторів, роздрукувати і нанести їх значення одновимірний масив. Серед елементів масиву знайти максимальний елемент та його номер.
11. В матриці $A(3,7)$ знищити 5-й стовпчик і роздрукувати отриману матрицю.
12. Матриця $A(m,n)$ ($m \leq 4, n \leq 3$) містить додатні та від'ємні елементи. Сформулювати з елементів даної матриці два масиви: B - що містить додатні елементи, та C - від'ємні. Підрахувати кількість елементів в даних масивах.
13. Цілочисельна матриця $A(n,n)$ ($n \leq 5$). Знайти найменше із значень елементів стовпчика, який має максимальну суму модулів елементів. Вказати номер стовпчика.



14. Провести таке перетворення матриці $A(m,n)$ ($m \leq 7$, $n \leq 3$), при якому останній рядок поміняється місцями з першим, передостанній з другим і т.д. Роздрукувати перетворену матрицю.

15. В матриці $B(m,n)$ ($m \leq 4$, $n \leq 6$), всі елементи якої різні. В кожному рядку вибирається елемент з найменшим значенням, потім серед цих чисел вибирається найбільше. Роздрукувати знайдений елемент.

16. Піднести до квадрату всі непарні елементи матриці $A(m,n)$ ($m \leq 5$, $n \leq 4$) і сформулювати із цих квадратів одновимірний масив.

17. У матриці $A(m,n)$ ($m \leq 5$, $n \leq 5$) замінити елемент головної діагоналі елементами побічної.

18. У матриці $A(m,n)$ ($m \leq 5$, $n \leq 4$) знайти суму елементів, які обрамляють дану матрицю і поміняти місцями мінімальний елемент лівої сторони на максимальний елемент правої.

19. Зробити таке перетворення матриці $A(m,n)$ ($m \leq 5$, $n \leq 5$):

а) елементи, які стоять над побічною діагоналлю замінити систематичними їм відносно побічної діагоналі;

б) замінити елементи, які стоять над побічною діагоналлю, їм симетричними відносно головної діагоналі.

20. У матриці $A(m,n)$ ($m \leq 5$, $n \leq 5$) замінити елементи, які стоять над побічною діагоналлю, їм симетрично відносно цієї діагоналі.

21. У матриці $A(m,n)$ ($m \leq 5$, $n \leq 5$) поміняти елементи діагоналі, яка прилягає до головної діагоналі згори, на елемент діагоналі, яка прилягає до головної діагоналі знизу.

22. У матриці $A(m,n)$ ($m \leq 5$, $n \leq 4$) впорядкувати двох останніх рядків по спаданню.

23. Впорядкувати елементи матриці $A(m,n)$ ($m \leq 5$, $n \leq 4$), які стоять вище головної діагоналі по зростанню і записати їх нижче головної діагоналі по рядках.

24. Впорядкувати елементи матриці $A(m,n)$ ($m \leq 5$, $n \leq 4$) по спаданню і розмістити їх по рядках.