

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



**ДНІПРОВСЬКА
ПОЛІТЕХНІКА**
1899

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ
ЕЛЕКТРОТЕХНІЧНИЙ ФАКУЛЬТЕТ
Кафедра кіберфізичних та інформаційно-вимірювальних систем**

В.В. ТКАЧОВ, В.В. НАДТОЧИЙ

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ТА ПРОГРАМУВАННЯ

Методичні рекомендації до виконання навчальної практики
для студентів спеціальності
151 «Автоматизація та комп'ютерно-інтегровані технології»

Дніпро
НТУ «ДП»
2020

В.В. Ткачов, В.В. Надточий

Комп'ютерні технології та програмування. Методичні рекомендації до виконання навчальної практики для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». [Електронний ресурс] / В.В. Ткачов, В.В. Надточий; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Електрон. Текст. Дані – Дніпро : НТУ «ДП», 2020. – 45 с.

Автори:

В. В. Ткачов, д.т.н., зав. каф.

В. В. Надточий, старший викладач

Затверджено до видання науково-методичною комісією спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» (протокол №5 від 25.06.2020р.) за поданням кафедри кіберфізичних та інформаційно-вимірjувальних систем (протокол №12 від 09.06.2020 р.).

Подано методичні рекомендації до виконання лабораторних робіт з комп'ютерних технологій та програмування студентами спеціальності 151 «Автоматизації та комп'ютерно-інтегровані технології».

Відповідний за випуск завідувач кафедри кіберфізичних та інформаційно-вимірjувальних систем, д-р техн. наук, проф. В. В. Ткачов.

ЗМІСТ

Вступ.....	5
Практичне завдання №1. Сортування і пошук.....	6
1.1 Організація виконання індивідуального завдання.....	6
1.2 Питання для підготовки до захисту індивідуального завдання	6
Практичне завдання №2. Створювання власних класів.....	10
2.1 Організація виконання індивідуального завдання.	10
2.2 Питання для підготовки до захисту індивідуального завдання.	10
Практичне завдання №3. Створення діалогової прикладної програми для операційної системи Windows	17
3.1 Організація виконання індивідуального завдання.....	17
3.2 Питання для підготовки до захисту індивідуального завдання	20
Практичне завдання №4. Обробка текстових файлів.....	21
4.1 Організація виконання індивідуального завдання.....	21
4.2 Питання для підготовки до захисту індивідуального завдання	21
Перелік літератури	26
Додаток А - Приклад оформлення титульного листа звіту	27
Додаток Б - Опис функцій бібліотеки С++.....	28
Б.1 Математичні функції.....	28
Б.2 Функції переведення.....	29
Б.3 Функції введення-виводу	30
Б.4 Функції роботи з файлами	33
Б.5 Функції роботи з рядками	36
Додаток С - Організація пошуку і сортування	39
С.1 Пошук	39
С.1.1 Лінійний пошук	39
С.1.2 Двійковий пошук	39
С.2 Сортування.....	40
С.2.1 Сортування вставками.....	40
С.2.2 Сортування методом Шелла.....	40
С.2.3 Швидке сортування	42
С.3 Висновки.....	43

ВСТУП

Індивідуальні завдання для навчальної практики з дисципліни «Комп'ютерні технології та програмування» розраховані для студентів групи 151 «Автоматизація та комп'ютерно-інтегровані технології».

Індивідуальні завдання можуть виконуватися в будь-яких версіях середовища програмування Visual C++ і операційної системи Windows.

Перелік індивідуальних завдань включає чотири найменування, до кожного з яких підготовлені методичні вказівки. Номер завдання, яке видається студенту, відповідає номеру, під яким він записаний у списку групи.

За результатами роботи студент оформляє звіт, захищає виконане індивідуальне завдання і відповідає на контрольні питання.

Зміст звіту повинен складатися з:

- титульного аркуша (додаток 1);
- опису завдання, включаючи тему, мету, номер і початкові дані вибраного варіанта;
- алгоритм;
- програми мовою C++;
- результатів обробки тестових даних при запуску програми;
- тестових даних і короткого опису програми;
- завантажувального модуля (архівів проектів).

Звіт надсилається на офіційну пошту перевіряючого викладача.

ПРАКТИЧНЕ ЗАВДАННЯ №1. СОРТУВАННЯ І ПОШУК

Мета: вивчити деякі алгоритми сортування і пошуку, а також навчитися реалізовувати ці алгоритми за допомогою мови програмування C++.

1.1 Організація виконання індивідуального завдання

Для виконання індивідуального завдання необхідно вивчити, використовуючи рекомендовану літературу, конспект лекцій та методичні вказівки, методи сортування і пошуку.

Скласти схему програми і програму мовою C++, що виконує такі дії:

формування одновимірного масиву із заданих елементів двовимірного масиву розміром $n \times n$ (n – задається користувачем). Вибірку елементів проводити у вказаному напрямку (табл. 1.1) та згідно з варіантом. Елементи двовимірного масиву повинні задаватися користувачем або за умовчанням (за допомогою генератора випадкових чисел);

- пошук заданого користувачем елемента в одновимірному масиві, лінійним методом і підрахунок кількості порівнянь;
- сортування вектору, використовуючи методи сортування і пошуку, згідно з варіантом (табл. 1.1);
- пошук заданого користувачем елемента в одновимірному масиві двійковим методом і підрахунок кількості порівнянь.

Кожне завдання повинне бути реалізоване у вигляді функції. Проект програми має бути багато-файловим.

Оцінити ефективність лінійного і двійкового пошуку, а також використовуваного методу сортування.

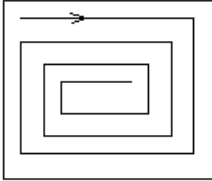
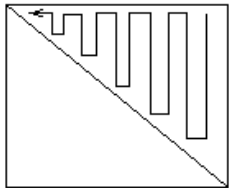
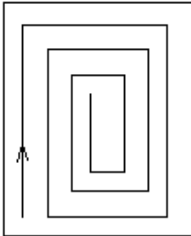
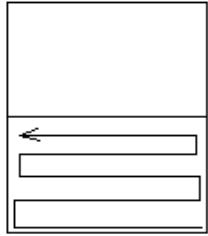
Оформити звіт про виконане завдання.

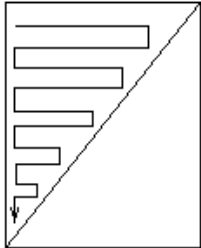
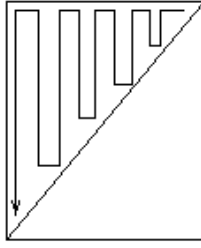
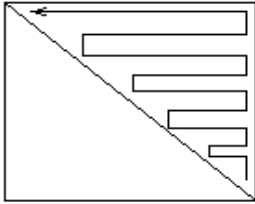
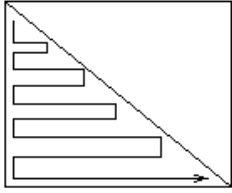
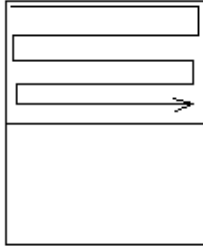
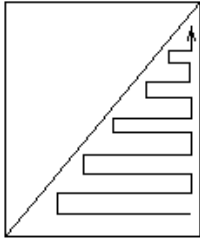

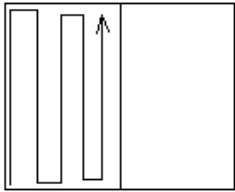
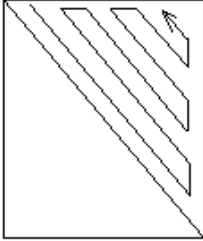
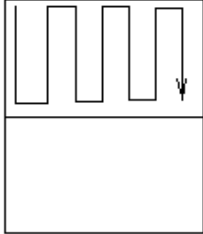
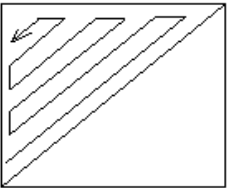
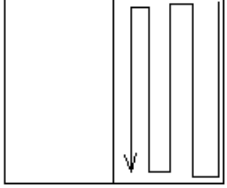
1.2 Питання для підготовки до захисту індивідуального завдання

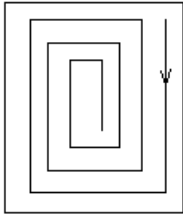
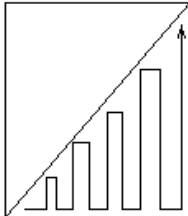
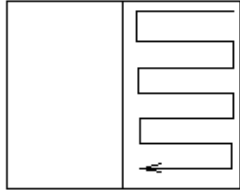
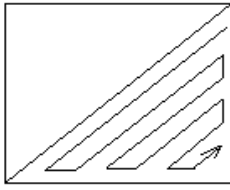
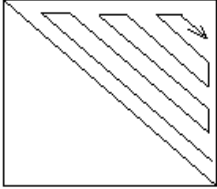
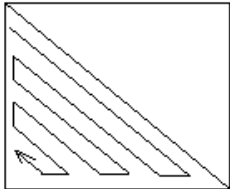
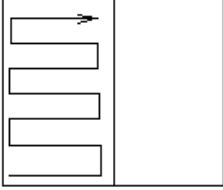
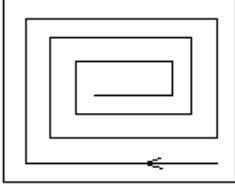
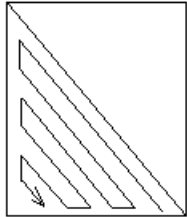
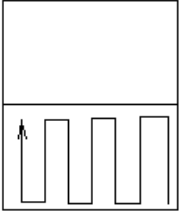
1. У чому полягає лінійний пошук?
2. У чому полягає двійковий пошук?

3. Коли застосовують двійковий пошук?
4. Доведіть, що для двійкового пошуку в масиві з 1023 елементів достатньо 10 порівнянь.
5. Які відомі способи сортування?
6. Який спосіб сортування є найпростішим?
7. У чому полягає спосіб швидкого сортування?
8. У чому полягає спосіб сортування Шелла?
9. У чому полягає сортування вставками?
10. Який спосіб сортування розробив Е. Хоар?
11. При сортуванні вводиться термін „бар’єр”. Що він означає?
12. При якому способі сортування вводиться термін „бар’єр”?
13. Який спосіб сортування належить до методів сортування за місцем?
14. Які методи сортування є стійкими за місцем?

Таблиця 1.1 - Варіанти завдань

№ вар.	Метод сортування масиву	Напрямок вибірки елементів для формування вектору	№ вар.	Метод сортування масиву	Напрямок вибірки елементів для формування вектору
1.	Швидке сортування		7.	Сортування Шелла	
2.	Сортування Шелла		8.	Сортування вставками	

№ вар.	Метод сортування масиву	Напрямок вибірки елементів для формування вектору	№ вар.	Метод сортування масиву	Напрямок вибірки елементів для формування вектору
3.	Сортування вставками		9.	Швидке сортування	
4.	Швидке сортування		10.	Сортування Шелла	
5.	Сортування Шелла		11.	Сортування вставками	
6.	Сортування вставками		12.	Швидке сортування	
13.	Швидке сортування		20.	Сортування Шелла	
14.	Сортування Шелла		21.	Сортування вставками	

№ вар.	Метод сортування масиву	Напрямок вибірки елементів для формування вектору	№ вар.	Метод сортування масиву	Напрямок вибірки елементів для формування вектору
15.	Сортування вставками		22.	Швидке сортування	
16.	Швидке сортування		23.	Сортування Шелла	
17.	Сортування Шелла		24.	Сортування вставками	
18.	Сортування вставками		25.	Швидке сортування	
19.	Швидке сортування		26.	Сортування Шелла	

ПРАКТИЧНЕ ЗАВДАННЯ №2. СТВОРЮВАННЯ ВЛАСНИХ КЛАСІВ

Мета: Ознайомитися з основними принципами об'єктно-орієнтованого програмування, навчитися створювати власні класи мовою програмування C++ і працювати з екземплярами класів.

2.1 Організація виконання індивідуального завдання.

Для виконання індивідуального завдання необхідно вивчити, використовуючи рекомендовану літературу, конспект лекцій та методичні вказівки, основні принципи об'єктно-орієнтованого програмування.

Скласти схему програми і програму мовою C++, що виконує такі дії:

- а) оголошення класу згідно з варіантом (табл. 2.1);
- б) ілюстрування всіх властивостей класу.

Створити багато-файловий проект, тобто оголошення класу, визначення функцій-членів класу, а також розміщення демонстраційної програми в окремих модулях (файлах).

Оформити звіт про виконане завдання.

2.2 Питання для підготовки до захисту індивідуального завдання.

1. Які поняття в мові C++ зв'язуються з класом?
2. Дайте визначення класу в мові C++.
3. Скільки понять у мові C++ зв'язуються з класом?
4. Із скількох частин складається визначення класу в мові C++?
5. Як оголошуються елементи-дані класу?
6. Що задають функції-елементи класу?
7. Як оголошуються функції-елементи класу?
8. Якими властивостями володіє функція-елемент класу?
9. Для чого застосовується засіб утаювання даних?
10. На які частини розбивають тіло класу?
11. Які рівні доступу назначаються для частин тіла класу?

12. Як виконується визначення об'єктів класу?
13. Що таке конструктор у мові C++?
14. Що означає в мові C++ деструктори і деструкція?
15. Наведіть приклад визначення деструкції.

Таблиця 2.1 - Варіанти завдань

№ вар.	Опис класу
1	<p>Дайте визначення класу, що зображує трикутник. Клас має атрибути – координати вершин трикутника. Функції-члени повинні виконувати такі дії: створення об'єкта і його ініціалізація;</p>
2	<p>призначення початкових значень елементам даних; відображення трикутника на екрані за допомогою заданого символу; обчислення довжин сторін трикутника; $S = \frac{abc}{4R}$ обчислення периметра і площі трикутника (де a, b, c – довжини сторін трикутника, R – радіус описаного кола).</p>
3	<p>Дайте визначення класу, що зображує чотирикутник. Елементи даних повинні містити координати точок чотирикутника. Функції-члени повинні виконувати такі дії: створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; відображення чотирикутника на екрані за допомогою заданого символу; обчислення довжини і ширини прямокутника; обчислення периметра і площі чотирикутника.</p>
4	<p>Дайте визначення класу, що зображує коло. Клас має атрибути: радіус кола, який за умовчанням дорівнює 1, і центр кола. Клас має функції-члени, які повинні виконувати такі дії: створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; обчислення довжини кола ($L = 2\pi r$); вивід кола на екран за допомогою заданого символу; знаходження площі, що описується колом ($S = \pi r^2$).</p>

№ вар.	Опис класу
5	<p>Комплексне число складається з двох частин: дійсної та уявної. Одним із способів запису комплексного числа є такий: $(3.0, 4.5i)$. Тут 3.0 – дійсна частина числа, а 4.5 – уявна. Припустімо, що $a=(A, Bi)$ $c=(C, Di)$. Ось деякі операції, виконувані над комплексними числами:</p> <p>додавання: $a + c = (A + C, (B + D) i)$; віднімання: $a - c = (A - C, (B - D) i)$; множення: $a c = (A C - B D, (A D + B C) i)$.</p> <p>Визначте клас, що оперує комплексними числами.</p>
6	<p>Створіть клас “Дата” з такими можливостями:</p> <p>створення об’єкта і його ініціалізація; призначення початкових значень елементам даних; вивід дат у декількох форматах: а) DD YYYY; б) MM/DD/YY; в) місяць, число, рік; визначення дня тижня, якщо відомий один з днів тижня поточного місяця.</p>
7	<p>Розробіть клас “Person”, що містить члени для зберігання імені, дати народження, статі, номера телефону. Функції-члени повинні виконувати такі дії:</p> <p>створення об’єкта і його ініціалізація; призначення початкових значень елементам даних; форматований вивід даних про людину; визначення віку людини.</p>
8	<p>Дайте визначення класу, що зберігає список товарів. Клас має атрибути: назва продукту, його ціна. Функції-члени повинні виконувати такі дії:</p> <p>створення об’єкта і його ініціалізація; призначення початкових значень елементам даних; обчислення ціни продукту з 20%-ною знижкою; вивід на екран величини знижки на продукт; вивід на екран даних про продукт.</p>
9	<p>Дайте визначення класу, що зображує конус. Елементи даних повинні містити радіус основи конуса і його висоту. Функції-члени повинні виконувати такі дії, як:</p> <p>створення об’єкта і його ініціалізація; призначення початкових значень елементам даних; обчислення довжини кола основи $(L = 2\pi r)$, площі бічної поверхні $(S_{біч} = \pi r L)$ та об’єму конуса $\left(V = \frac{1}{3} \pi r^2 h \right)$;</p> <p>вивід усіх відомих параметрів фігури.</p>

№ вар.	Опис класу
10	<p>Дайте визначення класу, що зображує циліндр. Елементи даних повинні містити радіус підстави циліндра і його висоту. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; обчислення довжини кола основи ($L = 2\pi r$), площі бічної поверхні ($S_{бічн} = 2\pi r h$) та об'єму циліндра ($V = \pi r^2 h$); вивід усіх відомих параметрів фігури.
11	<p>Дайте визначення класу, що зображує кулю. Елементи даних повинні містити радіус кулі. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; обчислення площі ($S = 4\pi r^2$) та об'єму кулі ($V = \frac{4}{3}\pi r^3$); вивід усіх відомих параметрів фігури.
12	<p>Дайте визначення класу, що зберігає одновимірний масив цілих чисел. Елементи даних повинні містити кількість елементів масиву, масив цілих чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; інвертування елементів масиву; знаходження суми елементів у масиві; форматований вивід масиву на екран.
13	<p>Дайте визначення класу, що зберігає одновимірний масив дійсних чисел. Елементи даних повинні містити кількість елементів масиву, масив дійсних чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; пошук заданого елемента у масиві; визначення кількості від'ємних елементів у масиві; форматований вивід масиву на екран.
14	<p>Дайте визначення класу, що зберігає одновимірний масив дійсних чисел. Елементи даних повинні містити кількість елементів масиву, масив дійсних чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення кількості додатних елементів у масиві; визначення добутку елементів у масиві; форматований вивід масиву на екран.

№ вар.	Опис класу
15	<p>Дайте визначення класу, що зберігає одновимірний масив цілих чисел. Елементи даних повинні містити кількість елементів масиву, масив цілих чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; сортування елементів масиву за збільшенням; визначення кількості нульових елементів; форматований вивід масиву на екран.
16	<p>Дайте визначення класу, що зберігає одновимірний масив цілих чисел. Елементи даних повинні містити кількість елементів масиву, масив цілих чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення кількості заданих символів у масиві; сортування за спаданням; форматований вивід масиву на екран.
17	<p>Дайте визначення класу, що зберігає одновимірний масив цілих чисел. Елементи даних повинні містити кількість елементів масиву, масив цілих чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення найбільшого спільного дільника елементів масиву; визначення мінімального елемента масиву; форматований вивід масиву на екран.
18	<p>Дайте визначення класу, що зберігає одновимірний масив цілих чисел. Елементи даних повинні містити кількість елементів масиву, масив цілих чисел. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення найменшого спільного кратного елементів масиву; визначення максимального елемента масиву; форматований вивід масиву на екран.
19	<p>Дайте визначення класу, що виконує арифметичні операції над двома цілими числами. Елементи даних повинні містити два цілих числа. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення суми і різниці чисел; ділення чисел за модулем; знаходження заданого відсотка від числа; форматований вивід результату всіх допустимих операцій над двома цілими числами на екран.

№ вар.	Опис класу
20	<p>Дайте визначення класу, що виконує арифметичні операції над дробами. Елементи даних повинні містити два дробових числа. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення найменшого спільного кратного; визначення суми, різниці та добутку чисел; форматований вивід результату всіх допустимих операцій над двома дробовими числами на екран.
21	<p>Дайте визначення класу, що зберігає рядок символів. Елементи даних повинні містити початковий рядок. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних;
22	<ul style="list-style-type: none"> визначення довжини рядка; заміну всіх символів рядка заданим символом; вивід рядка на екран.
23	<p>Дайте визначення класу, що зберігає рядок символів. Елементи даних повинні містити початковий рядок. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; знаходження заданого символу в рядку; вставка символу в кінець рядка, вивід рядка на екран.
23	<p>Дайте визначення класу, що виконує операції над двома цілими числами. Елементи даних повинні містити два цілих числа. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних; визначення найбільшого спільного дільника; визначення найменшого спільного кратного; ділення чисел; форматований вивід результату всіх допустимих операцій над двома цілими числами на екран.
25	<p>Розробіть клас “Студент”, що містить члени для зберігання імені, віку, статі, адреси, року вступу у ВНЗ. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних, зміна даних; форматований вивід даних про студента; визначення року народження студента і місця проживання (тобто чи є студент жителем даного міста).

№ вар.	Опис класу
26	<p>Дайте визначення класу, що являє собою магазин. Клас має атрибути: назва магазину, його адреса, перелік товарів і їх ціна. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкта і його ініціалізація; призначення початкових значень елементам даних, зміна даних; вивід інформації про заданий товар; вивід на екран даних про магазин.
27	<p>Дайте визначення класу, що являє собою каталог фільмів. Клас має атрибути: назва фільму, його жанр, рік випуску. Функції-члени повинні виконувати такі дії:</p> <ul style="list-style-type: none"> створення об'єкту і його ініціалізація; призначення початкових значень елементам даних; зміна даних; вивід на екран даних про фільм.

ПРАКТИЧНЕ ЗАВДАННЯ №3. СТВОРЕННЯ ДІАЛОГОВОЇ ПРИКЛАДНОЇ ПРОГРАМИ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS

Мета: Одержати практичні навички в створенні простої діалогової програми для операційної системи Windows.

3.1 Організація виконання індивідуального завдання

Для виконання індивідуального завдання необхідно ознайомитися, використовуючи рекомендовану літературу, конспект лекцій та методичні вказівки, з основними принципами програмування Windows-програм.

Створити проект «діалогове вікно», при цьому програма повинна мати вигляд, відображений на рисунку 3.1.

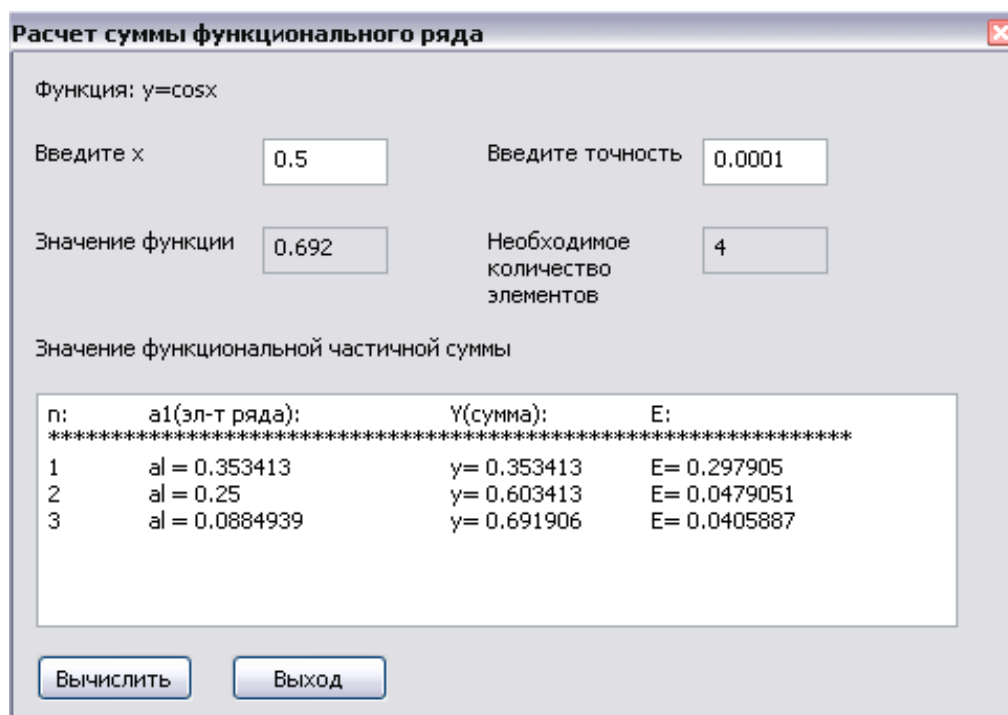


Рисунок 3.1 - Вигляд програми «діалогове вікно»

Програма повинна виконувати такі обчислення:

- значення функціональної часткової суми свого варіанта, виконаної для числа членів i від 1 до n додатків (табл. 3.1), аргумент при цьому змінюється в діапазоні, визначеному у таблиці 3.1;
- значення функції за її формулою (табл. 3.1);

- значення суми ряду $f_n(x)$, функції $f(x)$, що відрізняється від ідеального значення, з необхідною точністю ($\varepsilon=0.0001$);
 - визначення кількості потрібних для цього елементів суми n , а також вивід результатів у вигляді таблиці, зазначивши кількість підсумовуваних елементів, значення суми і її відхилення від функції.
- Оформити звіт про виконане завдання.

Таблиця 3.1 - Функціональні ряди

№ ряду	Функціональні ряди	Діапазон зміни аргументу	Кількість підсумовуваних членів ряду (n)	Функція
1.	$\sum_{i=0}^n \frac{\ln^i 3}{i!} x^i$	[0,1;1]	10	$y=3x$
2.	$\sum_{i=1}^n \frac{\cos ix}{i}$	$[\pi/5; 9\pi/5]$	16	$y=-\ln 2\sin(x/2) $
3.	$\sum_{i=0}^n (-1)^{i+1} \frac{\sin ix}{i}$	$[\pi/5; 4\pi/5)$	18	$y=x/2$
4.	$\sum_{i=0}^n \frac{x^i}{i!}$	[1;2]	15	$y=e^x$
5.	$\sum_{i=0}^n \frac{\cos i \frac{\pi}{4}}{i!} x^i$	[0,1;2]	13	$y=e^x \cos(\pi/4)$ $\cos(x \sin(\pi/4))$
6.	$\sum_{i=0}^n (-1)^i \frac{x^{2i}}{(2i)!}$	[0,1;2]	10	$y=\cos x$
7.	$\sum_{i=0}^n x^i \sin i \frac{\pi}{4}$	[0,1;0,8)	14	$y = \frac{x \sin \frac{\pi}{4}}{1 - 2x \cos \frac{\pi}{4} + x^2}$
8.	$\sum_{i=0}^n \frac{x^{4i+1}}{(2i+1)}$	[-0,3;0,4)	17	$y = \frac{1}{4} \ln \frac{1+x}{1-x} + \frac{1}{2} \operatorname{arctg} x$
9.	$\sum_{i=0}^n \frac{\cos ix}{i!}$	[-0,3;3]	20	$y=e^{\cos x} \cos(\sin x)$

№ ряду	Функціональні ряди	Діапазон зміни аргументу	Кількість підсумовуваних членів ряду (n)	Функція
10.	$\sum_{i=0}^n \frac{2i+1}{i!} x^{2i}$	[-0,3;1,3]	10	$y=(1+2x^2)e^{x^2}$
11.	$\sum_{i=0}^n \frac{1}{2i+1} \left(\frac{x-1}{x+1}\right)^{2i+1}$	[0,2;1]	10	$y=\frac{1}{2}\ln x$
12.	$\sum_{i=0}^n (-1)^i \frac{\cos ix}{i^2}$	$[\pi/5; \pi)$	20	$y=\frac{1}{4}(x^2-\frac{\pi^2}{3})$
13.	$\sum_{i=1}^n (-1)^{i+1} \frac{x^{2i+1}}{4i^2-1}$	[0,1;1]	15	$y=\frac{1+x^2}{2} \arctg x - \frac{x}{2}$
14.	$\sum_{i=0}^n \frac{x^{2i}}{(2i)!}$	[-0,3;1,3]	10	$y=\frac{e^x+e^{-x}}{2}$
15.	$\sum_{i=0}^n \frac{(2x)^i}{i!}$	[-0,3;1,2]	13	$y=e^{2x}$
16.	$\sum_{k=1}^{\infty} \frac{\cos kx}{k}$	$[3;2\pi)$	11	$y=-\ln(2\sin \frac{x}{2})$
17.	$\sum_{k=1}^{\infty} \frac{\cos kx}{k^2}$	$(0;2\pi)$	16	$y=\frac{3x^2-6\pi x+2\pi^2}{12}$
18.	$\sum_{k=1}^{\infty} \frac{\sin kx}{k}$	[2;3,5]	10	$y=\frac{\pi-x}{2}$
19.	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{\cos kx}{k}$	[0;1]	20	$y=\ln(2\cos \frac{x}{2})$
20.	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{\sin kx}{k}$	[0;2]	10	$y=\frac{x}{2}$
21.	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{\cos kx}{k^2}$	[0;2]	12	$y=\frac{\pi^2-3x^2}{12}$
22.	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{\sin kx}{k^3}$	[0;2]	14	$y=\frac{\pi^2 x-x^3}{12}$
23.	$\sum_{k=0}^{\infty} \frac{\cos(2k+1)x}{2k+1}$	[0,5;3]	15	$y=-\frac{1}{2}\ln(\operatorname{tg} \frac{x}{2})$

№ ряду	Функціональні ряди	Діапазон зміни аргументу	Кількість підсумовуваних членів ряду (n)	Функція
24.	$\sum_{k=0}^{\infty} \frac{\sin(2k+1)x}{2k+1}$	[0,6;3]	10	$y = \frac{\pi}{4}$
25.	$\sum_{k=0}^{\infty} \frac{\cos(2k+1)x}{(2k+1)^2}$	[0;□]	15	$y = \frac{\pi^2 - 2\pi x}{8}$

3.2 Питання для підготовки до захисту індивідуального завдання

1. З яких основних етапів складається розробка програм для Windows і в чому їх суть?
2. У чому суть об'єктно-орієнтованого підходу при програмуванні під Windows?
3. У чому суть керованості Windows-програм? Поясніть поняття повідомлення.
4. Структура і призначення карти повідомлень.
5. Розташування і призначення циклу обробки повідомлень.
6. Порядок створення проекту «діалогове вікно».
7. Порядок додавання в «діалогове вікно» кнопки і її обробника.
8. У якому місці проекту ініціалізується і запускається об'єкт «діалогове вікно»?

ПРАКТИЧНЕ ЗАВДАННЯ №4. ОБРОБКА ТЕКСТОВИХ ФАЙЛІВ

Мета: Ознайомитися з методами обробки текстових файлів.

4.1 Організація виконання індивідуального завдання

Використовуючи методичні вказівки, ознайомитися з основними методами роботи з файлами.

Скласти схему програми, для вирішення поставленого завдання згідно з варіантом (табл. 4.1).

Скласти програму мовою C++ для вирішення поставленого завдання:

- написати функцію;
- розробити програму, що демонструє роботу функції;
- створити багато-файловий проект. Основну програму, оголошення, а також визначення функцій розмістити в окремих модулях (файлах).
- оформити звіт про виконане завдання.

4.2 Питання для підготовки до захисту індивідуального завдання

1. Як зберігається текстова інформація в ЕОМ?
2. Що таке файл?
3. Які види файлів бувають?
4. Яка функція застосовується для відкриття файлу?
5. Яка функція застосовується для читання файлу?
6. Які функції застосовуються для запису, оновлення та закриття файлів?
7. Яка функція може перевіряти результат виконання операцій з файлами?
8. Як створити багато-файловий проект?
9. Як додати заголовний файл до проекту?
10. Які дані можуть міститися в заголовному файлі?

Таблиця 1.4 - Варіанти завдань

№ вар.	Опис завдання
1	<p>Скласти програму «Тестування колективу». Нехай цілочисельна матриця розміром $m \times n$ містить інформацію про учнів деякого класу з n осіб. У першому стовпці проставлена маса (кг), у другому – ріст (см), у третьому – успішність (середній бал) і т. д. (використовуйте свої додаткові показники). Учень називається середньостатистичним за k-м параметром (унікальним за k-м параметром), якщо на ньому досягається мінімум (максимум) модуля різниці середнього арифметичного чисел з k-го стовпця і значення k-го параметра цього учня. Учень називається найунікальнішим (середнім), якщо він унікальний (середньостатистичний) з найбільшої кількості параметрів. За даною матрицею визначити найунікальніших і середніх учнів. Інформація про учнів прочитується з початкового файлу, результат роботи програми записується у вихідний файл.</p>
2	<p>Визначити функцію, яка обчислює довжину гіпотенузи правильного трикутника, коли дві інші сторони відомі. Використовуйте цю функцію в програмі для визначення довжини гіпотенузи n трикутників, n задається з клавіатури.</p> <p>Початкові дані прочитуються з файлу, який виглядає так:</p> <pre>1 3.0 4.0 2 5.0 12.0 3 8.0 15.0</pre> <p>Вихідні дані записуються у файл у такому вигляді:</p> <pre>1 5 2 13 3 17</pre> <p>Програма формує початковий і вихідний файли.</p>
3	<p>Скласти програму призначення стипендії студентам за результатами сесії, використовуючи такі правила:</p> <ul style="list-style-type: none"> – якщо всі оцінки 5 – підвищена стипендія; – якщо всі оцінки 4 і 5 – звичайна стипендія; – якщо є оцінка 3, стипендія не призначається. <p>У результаті роботи програми у файлі повинен бути збережений список групи з оцінками і середнім балом кожного студента і два списки прізвищ на підвищену і звичайну стипендію.</p>
4	<p>Підрахувати кількість рядків заданої цілочисельної матриці $N \times N$ чисел, що є перестановкою 1, 2, ..., N (тобто, що містять кожне з чисел 1, 2, ..., N тільки один раз).</p> <p>Матриця прочитується з початкового файлу, результат роботи програми записується у вихідний файл.</p> <p>Програма формує початковий і вихідний файли.</p>

№ вар.	Опис завдання																																				
5	<p>Знайти максимальний елемент серед усіх елементів тих рядків заданої матриці, які впорядковані (або за збільшенням, або за спаданням). Матриця прочитується з початкового файлу, результат роботи програми записується у вихідний файл. Програма формує початковий і вихідний файли.</p>																																				
6	<p>Характеристикою стовпця цілочисельної матриці назвемо суму модулів його від'ємних непарних елементів. Переставляючи стовпці заданої матриці, розташувати їх відповідно до зростання характеристик. Матриця прочитується з початкового файлу, результат роботи програми записується у вихідний файл. Програма формує початковий і вихідний файли.</p>																																				
7	<p>Є база даних, що містить відомості про деяку групу людей (кожен запис містить 5 полів). Скласти програму, яка, використовуючи відомості з бази даних, дозволяє заповнювати запрошення, вписуючи ці відомості в потрібні місця у вказаному документі у відповідних відмінках, особах, годинах і т.д.</p>																																				
8	<p>Ви є власником складу металевих виробів і потребуєте інвентаризації, після якої ви будете знати, скільки всього різних виробів на складі та яка вартість кожного з них. Напишіть програму, яка виконувала б такі дії:</p> <ul style="list-style-type: none"> – дозволяла вводити дані про кожний виріб; – давала можливість друкувати список усіх виробів; – отримувала інформацію про кожний виріб з файлу і могла її змінювати; – виводила інформацію про вироби у файл; – підраховувала загальну вартість виробів. <p>Використовуйте таку інформацію для початку роботи з вашою програмою:</p> <table border="1" data-bbox="279 1444 1449 1881"> <thead> <tr> <th>Номер запису</th> <th>Назва інструмента</th> <th>Кількість, шт.</th> <th>Вартість, грн</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>Шліфувальний верстат</td> <td>7</td> <td>12257,98</td> </tr> <tr> <td>17</td> <td>Молоток</td> <td>76</td> <td>211,99</td> </tr> <tr> <td>24</td> <td>Електро-механічний лобзик</td> <td>21</td> <td>2211,00</td> </tr> <tr> <td>39</td> <td>Газонокосарка</td> <td>3</td> <td>2379,50</td> </tr> <tr> <td>56</td> <td>Електропилка</td> <td>18</td> <td>2499,99</td> </tr> <tr> <td>68</td> <td>Викрутка</td> <td>106</td> <td>126,99</td> </tr> <tr> <td>77</td> <td>Ковальський молот</td> <td>11</td> <td>421,50</td> </tr> <tr> <td>83</td> <td>Гайковий ключ</td> <td>34</td> <td>217,5</td> </tr> </tbody> </table>	Номер запису	Назва інструмента	Кількість, шт.	Вартість, грн	3	Шліфувальний верстат	7	12257,98	17	Молоток	76	211,99	24	Електро-механічний лобзик	21	2211,00	39	Газонокосарка	3	2379,50	56	Електропилка	18	2499,99	68	Викрутка	106	126,99	77	Ковальський молот	11	421,50	83	Гайковий ключ	34	217,5
Номер запису	Назва інструмента	Кількість, шт.	Вартість, грн																																		
3	Шліфувальний верстат	7	12257,98																																		
17	Молоток	76	211,99																																		
24	Електро-механічний лобзик	21	2211,00																																		
39	Газонокосарка	3	2379,50																																		
56	Електропилка	18	2499,99																																		
68	Викрутка	106	126,99																																		
77	Ковальський молот	11	421,50																																		
83	Гайковий ключ	34	217,5																																		

№ вар.	Опис завдання
9	Дані числовий файл. Підрахувати, скільки різних чисел у цьому файлі. Наприклад, у файлі записані числа 8, 7, 8, то різних чисел 2 (8 і 7). Видалити повтори з файлу. Програма формує початковий і вихідний файли.
10	Дані два символні файли, перший з яких складається з 10 компонент, другий – з 20. Сформуйте третій файл за такими правилами: спочатку в результуючий файл записується перша компонента з 1-го файлу, після цього – дві компоненти з другого файлу, потім друга компонента з 1-го файлу і т. д. Програма формує початкові та вихідний файли.
11	Дані два числових файли. Визначити кількість загальних елементів і видати їх у результуючий файл. Програма формує початкові і вихідний файли.
12	У заданому файлі видалити частину тексту, яка поміщена в дужки (разом з дужками). Програма формує початковий і вихідний файли.
13	Написати функцію, що міняє порядок проходження цілих чисел файлу на протилежний. Якщо файл не існує або містить менше двох елементів, то функція не виконує ніяких дій.
14	Дане натуральне число. Написати функцію, яка видаляє з числа всі одиниці, залишивши порядок решти цифр незмінним. Натуральне число прочитується з початкового файлу, а результат записується у вихідний файл. Програма формує початковий і вихідний файли.
15	Здійснити циклічний зсув елементів масиву $T(n)$ на m позицій вліво, тобто одержати масив: $t_{m+1}, \dots, t_n, t_1, \dots, t_m$. При цьому необов'язково, щоб $m < n$. Матриця прочитується з початкового файлу, результат роботи програми записується у вихідний файл. Програма формує початковий і вихідний файли.
16	Дана матриця A розміром $M \times N$ і числа i, j . Написати функцію, яка видаляє з матриці A рядок i і стовпець, що містить елемент $A[i,j]$ (якщо $i > M$ або $j > N$, то матриця не змінюється). Матриця прочитується з початкового файлу, результат роботи програми записується у вихідний файл. Програма формує початковий і вихідний файл.

№ вар.	Опис завдання
17	<p>Написати функцію, що знаходить наближене значення функції $\exp(x)$: $\exp(x) = 1 + x + x^2 / 2! + x^3 / 3! + \dots + x^n / n! + \dots$.</p> <p>У сумі враховувати всі доданки та великі числа ϵ. За допомогою функції знайти наближене значення експоненти для даного x при n різних значеннях ϵ.</p> <p>Значення x, ϵ і n прочитуються з файлу, а результат роботи програми записується у файл.</p>
18	<p>Розглянемо деяке натуральне число n. Якщо воно парне, то розділимо його на 2, якщо ні – помножимо на 3 і додамо 1.</p> <p>Повторюватимемо такі дії (кроки) доти, поки не вийде 1. Одержана послідовність називається послідовністю Хейеса, а найбільше з чисел цієї послідовності – її вершиною. Потрібно написати функцію, що обчислює для заданого n послідовність Хейеса, яка підраховує кількість кроків у ній і знаходить вершину.</p> <p>Число прочитується з початкового файлу, результат роботи програми записується у вихідний файл.</p>

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Шпак З.Я. Програмування мовою С. / З.Я. Шпак. – Л.: Оріяна-Нова, 2006. – 432 с.
2. Ковалюк Т. В. Алгоритмізація та програмування: підручник з грифом МОН України / Т. В. Ковалюк. – Львів : Магнолія-2006, 2013. – 400 с.
3. Кравець П. Об'єктно-орієнтоване програмування: навч. посібник / П.О. Кравець. – Львів: Видавництво Львівської політехніки, 2012. – 624 с.
4. С++. Теорія та практика : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін. ; за ред. О. Г. Трофименко. – 587 с.
5. Шаховська Н. Б. Алгоритми та структури даних / Н. Б. Шаховська, Р.О. Голощук. – Львів : Магнолія-2006. – 2009. – 216 с.
6. Шеховцов В.А. Операційні системи / В.А. Шеховцов. – К. : Видавнича група ВНУ, 2005. – 576 с.

ДОДАТОК А - ПРИКЛАД ОФОРМЛЕННЯ ТИТУЛЬНОГО ЛИСТА ЗВІТУ
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ
ЕЛЕКТРОТЕХНІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КІБЕРФІЗИЧНИХ ТА ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ
СИСТЕМ

НАВЧАЛЬНА ПРАКТИКА

з дисципліни:

«КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ТА ПРОГРАМУВАННЯ»

Виконав студент групи **151-21-1**

Іванов Іван Іваноич

Перевірив: ст. викл. **Надточий В.В.**

Дніпро

2022

ДОДАТОК Б - ОПИС ФУНКЦІЙ БІБЛІОТЕКИ C++

Б.1 Математичні функції

abs, fabs

Синтаксис: **int** *abs* (**int** x) ;
double *fabs* (**double** x) ;

Повертає ціле (*abs*) або дробове (*fabs*) абсолютне значення аргументу, для якого можна використовувати вираз відповідного типу.

Заголовний файл: <math.h>

acos, asin, atan, acosl, asinl, atanl

Синтаксис: **double** *acos* (**double** x) ;
double *asin* (**double** x) ;
double *atan* (**double** x) ;
long double *acosl* (**long double** x) ;
long double *asinl*(**long double** x);
long double *atanl* (**long double** x) ;

Повертає виражену в радіанах величину кута, косинус, синус або тангенс якого переданий відповідній функції як аргумент. Аргумент функції повинен знаходитися в діапазоні від -1 до 1.

Заголовний файл: <math.h>

cos, sin, tan

cosl, sinl, tanl

Синтаксис: **double** *cos* (**double** x) ;
double *sin* (**double** x) ;
double *tan* (**double** x) ;
long double *cosl* (**long double** x) ;
long double *sinl* (**long double** x) ;
long double *tanl* (**long double** x) ;

Повертає синус, косинус або тангенс кута. Величина кута повинна бути задана в радіанах.

Заголовний файл: <math.h>

exp, expl

Синтаксис: **double** *exp*(**double** x);
long double *expl*(**long double** (x));

Повертає значення, що дорівнює експоненті аргументу (тобто e^x , де e – основа натурального логарифма).

Заголовний файл: <math.h>

pow, powl

Синтаксис: **double** *pow* (**double** x, **double** y);
long double *powl*(**long double** (x), **long double** (y));

Повертає значення, що дорівнює x^y .

Заголовний файл: <math.h>

sqrt

Синтаксис: **double** sqrt(**double** x);

Повертає значення, що дорівнює квадратному кореню з аргументу.

Заголовний файл: <math.h>

rand

Синтаксис: **int** rand(**void**);

Повертає випадкове ціле число в діапазоні від 0 до RAND_MAX. Перед першим зверненням до функції rand необхідно ініціалізувати генератор випадкових чисел. Для цього треба викликати функцію srand.

Заголовний файл: <stdlib.h>

srand

Синтаксис: **void** srand (**unsigned** x) ;

Ініціалізує генератор випадкових чисел. Звичайно як параметр функції використовують змінну, значення якої передбачити наперед не можна, наприклад, це може бути поточний час.

Заголовний файл: <stdlib.h>

Б.2 Функції переведення

Наведені нижче функції виконують переведення рядків у числове значення і чисел у рядкове зображення.

atof

Синтаксис: **double** atof (**const char*** s) ;

Повертає дробове число, значення якого передане функції як аргумент. Функція обробляє рядок доти, поки символи рядка є допустимими. Рядок може бути значенням числа як в експоненціальному так і у форматі з плаваючою точкою.

Заголовний файл: <stdlib.h>

atoi, atol

Синтаксис: **int** atoi (**const char*** s) ;

long atol (**const char*** s) ;

Повертає ціле число відповідного типу, зображення якого віддане функції як аргумент. Функція обробляє символи рядка доти, поки не зустрине символ, що не є десятковою цифрою.

Заголовний файл: <stdlib.h>

gcvt

Синтаксис: **char** *gcvt(**double** *Значення*, **int** *Цифра* **char*** *Рядок*) ;

Переводить дробове число в рядок. При переведенні робиться спроба одержати вказану кількість значущих цифр якщо це зробити неможливо, то число зображується у формі плаваючої точки.

Заголовний файл: <stdlib.h>

itoa, Itoa, ultoa

Синтаксис: **char*** itoa (**int** *Значення* **char*** *Рядок*, **int** *Основа*) ;

char* Itoa (**long** *Значення* **char*** *Рядок*, **int** *Основа*);

char* ultoa(**unsigned long** *Значення* **char*** *Рядок*, **int** *Основа*);

Відповідно переводять ціле, довге ціле і довге беззнакове ціле число в рядок. Число зображається у вказаному рядку при виклику функції в певній системі числення.

Рядок – покажчик на рядок, куди буде поміщене зображення числа.

Основа – задає основу системи числення (від 2 до 36).

Максимальна довжина рядка, формованого функцією *itoa*, 17 байт, а функціями *ltoa* і *ultoa* – 33 байти.

Заголовний файл: <stdlib.h>

sprintf

Синтаксис: **int** *sprintf* (**char** **Рядок*, **const char*** *Формат*, *Список Змінних*)

Виконує форматований вивід у рядок.

Список Змінних – розділені комами імена змінних, задає змінні, значення яких повинні бути виведені. Параметр *Формат* задає спосіб відображення значень змінних.

Дія функції *sprintf* аналогічна дії функції *printf*, але вивід виконується в рядок-буфер, а не на екран.

Заголовний файл: <stdio.h>

Б.3 Функції введення-виводу

printf

Синтаксис: **int** *printf*(**const char** **format* [, *argument*]...);

Виконує форматований вивід на екран.

*const char *format* – рядок, який буде виведений на екран, *argument* – розділені комами імена змінних, значення яких підставляються в рядок.

Значення першої змінної виводиться відповідно до першого специфікатора формату, другий – до другого і т.д.

Рядок форматування, окрім тексту, може містити специфікатори формату і керуючі символи.

Специфікатор формату – набір символів, який визначає формат виводу значення змінної, вказаної як аргумент. Виглядає так:

%[*flags*] [*width*] [*.precision*] *type*

flags – символ, параметри вирівнювання, заповнення або формат виводу значення змінної. Він може набувати значення, які наведені в таблиці Б.1;

Таблиця Б.1 - Можливі значення символу *flags*

Символ <i>flags</i>	Значення	Значення за умовчанням
-	Значення вирівнюється зліва по ширині поля виводу	Вирівнювання справа
+	При виведенні числа додається префікс (+ або -), якщо число знакове	Знак виводиться тільки для від'ємних чисел
0	Незайнята область ширини поля виводу значення заповнюється нулями	–

Символ flags	Значення	Значення за умовчанням
#	Коли використовується разом з нулем x або X, значення змінної супроводжується відповідним префіксом 0, 0x або 0X	–
#	Коли використовується разом з e, E, f, a або A, значення змінної супроводжується десятковою точкою у будь-якому випадку	Десяткова точка виводиться тільки тоді, коли число дійсно містить дробову частину
#	Коли використовується разом з g або G, значення змінної супроводжується десятковою точкою у будь-якому випадку і нулями	–

width – десяткове число, задає ширину поля, що виділяється для виводу значення змінної;

precision – точність, з якою виводиться значення. Залежить від типу значення. Точність може набувати значення, які наведені в таблиці Б.2.

Таблиця Б.2 - Можливі значення точності

Тип значення	Значення	Значення за умовчанням
a, A	Кількість знаків після точки	Точність за умовчанням – 6. Коли точність дорівнює 0, точка не виводиться, якщо не використаний знак #
c, C	Точність не враховується	–
d, i, u, o, x, X	Задає мінімальне число знаків. Якщо в числі кількість знаків менша ніж задана точність, до числа зліва додаються нулі	Точність за умовчанням – 1.
e, E, f, g, G	Кількість знаків після коми. Останній знак округляється	Точність за умовчанням – 6. Коли точність дорівнює 0, точка не виводиться, якщо не використаний знак #
s, S	Максимальна кількість символів, що виводяться	Виводиться весь рядок

Type – тип значення, що виводиться

Type може набувати значення, які вказані в таблиці Б.3.

Таблиця Б.3 - Можливі значення type

Символ	Тип	Формат виводу
c	int	Символ
C	int	Символ
d	int	Знакове ціле в десятковій системі числення
i	int	Знакове ціле в десятковій системі числення
o	int	Беззнакове ціле у вісімковій системі числення
u	int	Беззнакове ціле в десятковій системі числення
x	int	Беззнакове ціле у шістнадцятковій системі числення. Використовується символ «abcdef»

Символ	Тип	Формат виводу
X	int	Беззнакове ціле у шістнадцятковій системі числення. Використовується символ «ABCDEF»
e	double	Знакове дійсне число у формі [-]d.dddd e [sign]dd[d]. d – одна десяткова цифра dddd – одна або декілька десяткових цифр dd[d] – 2 або 3 цифри, залежить від формату виводу, розміру експоненти і знаку + або -
E	double	Аналогічно e
f	double	Знакове дійсне число у формі [-]dddd.dddd. dddd – одна або декілька десяткових цифр
g	double	Знакове дійсне число. Для виводу використовується формат e або f залежно від довжини числа.
G	double	Аналогічно g
a	double	Знакове дійсне число у шістнадцятковій системі числення у формі [X]0xh.hhhh p±dd, де h.hhhh – шістнадцяткові цифри або символи «abcdef» – мантиса dd – 1 або 2 цифри експоненти
A	double	Знакове дійсне число у шістнадцятковій системі числення у формі [X]0Xh.hhhh p±dd, де h.hhhh – шістнадцяткові цифри або символи «ABCDEF» – мантиса dd – 1 або 2 цифри експоненти
n	Показчик на ціле	Виводиться показчик на ціле
p	Показчик на void	Виводиться адреса у шістнадцятковому зображенні
s	String	Виводиться рядок символів
S	String	Виводиться рядок символів

Керуючі символи можуть набувати значення, які наведені в таблиці Б.4.

Таблиця Б.4 - Можливі значення керуючих символів

Символ	Дія
\n	Переводить курсор на початок наступного рядка
\t	Переводить курсор у чергову позицію табуляції
\\	Символ '\'
\'	Лапка

Заголовний файл: <stdio.h>

scanf

Синтаксис: **int scanf(const char *format [, argument]...);**

Вводить з клавіатури значення змінних відповідно до вказаних специфікатором формату. Перша змінна набуває значення за першим специфікатором формату, друга – за другим і т.д. Параметри:

const char *format – рядок, який містить специфікатори формату (аналогічно printf)

argument – розділені комами адреси змінних, в які будуть записані введені з клавіатури значення.

Спеціальні символи для введення даних можуть набувати значення, які вказані в таблиці Б.5.

Таблиця Б.5 - Спеціальні символи для введення даних

Специфікатор	Дія
%i %d	Десяткове число із знаком
%u	Беззнакове ціле десяткове число
%f %e	Дробове число
%s	Рядок символів
%c	Символ

Заголовний файл: <stdio.h>

puts

Синтаксис: **int puts(const char *str);**

Виводить на екран рядок символів і переводить курсор на початок наступного рядка екрану. Як параметр функції можна використовувати рядкову константу або рядкову змінну.

Заголовний файл: <stdio.h>

gets

Синтаксис: **char *gets(char* s);**

Вводить з клавіатури рядок символів. Уведений рядок може містити пропуски.

Заголовний файл: <stdio.h>

putch

Синтаксис: **int putch(int c);**

Виводить на екран символ.

Заголовний файл: <conio.h>

getch

Синтаксис: **int getch (void);**

Повертає код символу натиснутої клавіші. Якщо натиснута службова клавіша, то функція getch повертає 0. У цьому випадку, для того щоб визначити, яка службова клавіша натиснута, потрібно звернутися до функції getch ще раз.

Зауваження

Функція getch не виводить на екран символ, відповідний натиснутій клавіші.

Заголовний файл: <conio.h>

Б.4 Функції роботи з файлами

fopen

Синтаксис: **FILE *fopen(const char *filename, const char *mode);**

Відкриває файл із вказаним ім'ям (filename) для дії, яка задається параметром режиму (mode).

Параметр режиму може набувати значення, які наведені в таблиці Б.6.

Таблиця Б.6 - Режими відкриття файлу

Режим	Дія
R	Тільки запис. Файл відкривається тільки для читання
W	Читання. Файл відкривається для запису. Якщо файл із вказаним ім'ям як перший параметр функції fopen вже існує, то нові дані записуються поверх старих, тобто старий файл фактично знищується
A	Додавання. Файл відкривається для запису даних у кінці існуючого файлу. Якщо файл із вказаним ім'ям як перший параметр функції fopen не існує, то він буде створений

Якщо файл відкривається як текстовий, то після символної константи, що визначає режим відкриття файлу, потрібно додати символ t. Наприклад, рядок rt означає, що для читання відкривається текстовий файл.

У разі успішного відкриття файлу функція fopen повертає покажчик на потік, з якого можна читати або в який можна записувати. Якщо з якої-небудь причини операція відкриття файлу не була виконана, fopen повертає NULL. У цьому випадку, щоб одержати інформацію про причину помилки, слід звернутися до функції perror.

Заголовний файл: <stdio.h>

fprintf

Синтаксис: **int fprintf(FILE *stream, const char *format [, argument]...);**

Виконує форматований вивід (див. printf) у файл, пов'язаний з потоком, вказаним як перший параметр.

Файл, пов'язаний з потоком, повинен бути відкритий як текстовий у режимі, що допускає запис (див. fopen).

Заголовний файл: <stdio.h>

fscanf

Синтаксис: **int fscanf(FILE *stream, const char *format [, argument]...);**

Виконує форматоване (див. scanf) читання значень змінних з файлу, пов'язаного з потоком, вказаним як перший параметр.

Файл, пов'язаний з потоком, повинен бути відкритий як текстовий у режимі, що допускає читання (див. fopen).

Заголовний файл: <stdio.h>

fgets

Синтаксис: **char *fgets (char *str, int n, FILE *stream);**

Параметри:

char *str – рядок, в який записується результат

int n – кількість прочитуваних символів

FILE *stream – файл, з якого прочитується рядок

Читає із вказаного потоку символи і записує їх у рядок. Читання закінчується, якщо прочитаний символ з номером $n-1$ або якщо черговий символ є символом нового рядка.

Прочитаний з файлу символ нового рядка замінюється нульовим символом.

Файл, пов'язаний з потоком, повинен бути відкритий як текстовий у режимі, що допускає читання (див. `fopen`).

Заголовний файл: `<stdio.h>`

fgetc

Синтаксис: **`int fgetc(FILE *stream);`**

Читає із вказаного потоку символ.

Файл, пов'язаний з потоком, повинен бути відкритий як текстовий у режимі, що допускає читання (див. `fopen`).

Заголовний файл: `<stdio.h>`

fputs

Синтаксис: **`int fputs (const char *str, FILE *stream);`**

Записує у вказаний потік рядок символів. Символ кінця рядка, нуль-символ у потік не записується.

Параметри:

`char *str` – рядок, який записується у файл

`FILE *stream` – файл, у який записується рядок

Файл, пов'язаний з потоком, повинен бути відкритий як текстовий у режимі, що допускає запис (див. `fopen`).

Заголовний файл: `<stdio.h>`

fputc

Синтаксис: **`int fputc(int c, FILE *stream);`**

Записує у вказаний потік *stream* символ *c*.

Файл, пов'язаний з потоком, повинен бути відкритий як текстовий у режимі, що допускає запис (див. `fopen`).

Заголовний файл: `<stdio.h>`

fseek

Синтаксис: **`int fseek(FILE *stream, long offset, int origin);`**

Переміщає файловий покажчик на задану позицію.

Параметри:

`FILE *stream` – початковий файл

`long offset` – кількість байт, які необхідно відступити від початкової позиції

`int origin` – початкова позиція файлового покажчика. Набуває одне з таких значень:

`SEEK_CUR` – поточна позиція файлового покажчика.

`SEEK_END` – кінець файлу.

`SEEK_SET` – початок файлу.

Заголовний файл: `<stdio.h>`

fsetpos

Синтаксис: **`int fsetpos(FILE *stream, const fpos_t *pos);`**

Встановлює індикатор позиції у потоці на задане значення *pos*.

Заголовний файл: <stdio.h>

fgetpos

Синтаксис: **int fgetpos(FILE **stream*, const fpos_t **pos*);**

Записує в змінну *pos* значення індикатора позиції у потоці.

Заголовний файл: <stdio.h>

ferror

Синтаксис: **int ferror (FILE **stream*);**

Повертає ненульове значення, якщо остання операція із вказаним потоком завершилася помилкою.

Заголовний файл: <stdio.h>

feof

Синтаксис: **int feof (FILE **stream*);**

Повертає ненульове значення, якщо в результаті виконання останньої операції читання з потоку досягнутий кінець файлу.

Заголовний файл: <stdio.h>

fclose

Синтаксис: **int fclose (FILE **stream*);**

Закриває вказаний потік.

Заголовний файл: <stdio.h>

Б.5 Функції роботи з рядками

strcat

Синтаксис: **char *strcat (char **strDestination*, const char **strSource*);**

Об'єднує рядки *strDestination* і *strSource* і записує результат у рядок *strDestination*.

Заголовний файл: <string.h>

strncat

Синтаксис: **char *strncat(char **strDest*, const char **strSource*, size_t *count*);**

Об'єднує задане число символів *count* рядка *strDestination* і *strSource* і записує результат у рядок *strDestination*.

Заголовний файл: <string.h>

strcpy

Синтаксис: **char *strcpy (char **strDestination*, const char **strSource*);**

Копіює рядок *strSource* в рядок *strDestination*.

Заголовний файл: <string.h>

strncpy

Синтаксис: **char *strncpy(char **strDest*, const char **strSource*, size_t *count*);**

Копіює задане число символів *count* рядка *strSource* в рядок *strDestination*.

Заголовний файл: <string.h>

strlen

Синтаксис: **int strlen (const char* string);**

Повертає довжину рядка. Нульовий символ не враховується.

Заголовний файл: <string.h>

strcmp

Синтаксис: **int strcmp(const char *string1, const char * string2);**

Порівнює рядки *string1* і *string2*. Повертає 0, якщо рядки рівні між собою, число менше нуля, якщо *string1* < *string2*, і число більше нуля, якщо *string1* > *string2*.

Заголовний файл: <string.h>

strncmp

Синтаксис: **int strncmp(const char *string1, const char *string2, size_t count);**

Порівнює задане число символів *count* рядка *string1* і *string2*. Повертає 0, якщо рядки рівні між собою, число менше нуля, якщо *string1* < *string2*, і число більше нуля, якщо *string1* > *string2*.

Заголовний файл: <string.h>

strlwr

Синтаксис: **char* strlwr(char* string);**

Перетворює рядкові символи на великі (обробляє тільки букви латинського алфавіту).

Заголовний файл: <string.h>

strupr

Синтаксис: **char*strupr(char* string);**

Перетворює великі символи на рядкові (обробляє тільки букви латинського алфавіту).

Заголовний файл: <string.h>

strset

Синтаксис:

char * strset(char *str, int sym);

Заповнює рядок *str* вказаним при виклику функції символом *sym*.

Заголовний файл: <string.h>

strchr

Синтаксис: **char* strchr(const char* string, int sym);**

Виконує пошук символу *sym* в рядку *string* і повертає покажчик на перший знайдений символ або, якщо символ не знайдений, то на null.

Заголовний файл: <string.h>

strstr

Синтаксис: **char *strstr(const char *str, const char *strSearch);**

Параметри:

const char **str* – початковий рядок

const char **strSearch* – рядок, який необхідно знайти

Виконує пошук рядка в рядку і повертає покажчик на перший знайдений рядок або, якщо рядок не знайдений, то на null.

Заголовний файл: <string.h>

strtok

Синтаксис: **char *strtok(char *strToken, const char *strDelimit);**

Ділить рядок на слова.

Параметри:

char *strToken – початковий рядок

const char *strDelimit – рядок, що містить символи, які можуть розділяти слова в рядку.

Функція знаходить один із розділювальних символів, замість нього в рядок записує нульовий символ і повертає адресу решти рядка. Якщо весь рядок оброблений, функція повертає NULL. Тобто, щоб розділити рядок на слова, необхідно викликати цю функцію кілька разів, поки повертаним значенням не буде NULL.

Заголовний файл: <string.h>

strrev

Синтаксис: **char *strrev(char *str);**

Функція проводить інвертування рядка.

Заголовний файл: <string.h>

ДОДАТОК С - ОРГАНІЗАЦІЯ ПОШУКУ І СОРТУВАННЯ

Навчальною метою розділу є ознайомлення студентів з організацією сортування і пошуку у мові програмування C++.

У результаті вивчення даного розділу студенти повинні знати:

- визначення лінійного пошуку;
- визначення двійкового пошуку;

уміти:

- застосовувати різні алгоритми сортування даних.

С.1 Пошук

Пошук, вставка і видалення – основні операції при роботі з даними. Розглянемо, як ці операції використовуються для найвідоміших об'єктів – масивів.

Існують різні способи пошуку елементів. Розглянемо лінійний і двійковий пошук на прикладі одновимірного масиву із семи елементів з числовими значеннями.

С.1.1 Лінійний пошук

Для того, щоб знайти в масиві потрібне число, можна використовувати *лінійний пошук*. Лінійний пошук полягає у такому: розглядаються по черзі всі елементи масиву, і визначається позиція потрібного нам елементу.

Тому кількість порівнянь буде дорівнювати кількості елементів у масиві, що не раціонально, оскільки для пошуку в масиві з великою кількістю елементів необхідно багато часу.

С.1.2 Двійковий пошук

Якщо відомо, що дані відсортовані, можна застосувати *двійковий пошук* (рис. С.1).

0	4	←	LB
1	7		
2	16		
3	20	←	M
4	37		
5	38		
6	43	←	UB

Рисунок С.1 - Двійковий пошук

Змінні Lb і Ub містять відповідно верхню і нижню межі відрізка масиву, де знаходиться потрібний елемент. Починаємо завжди з дослідження середнього елемента відрізка.

Якщо шукане значення менше середнього елемента, переходимо до пошуку у верхній половині відрізка, де всі елементи менше тільки що перевіреного. Іншими словами, значенням Ub стає $(M - 1)$ і на наступній ітерації працюємо з половиною масиву. Таким чином, у результаті кожної перевірки вдвоє звужуємо область пошуку.

Так, у нашому прикладі після першої ітерації область пошуку – всього лише три елементи, після другої залишається тільки один елемент. Таким чином, якщо довжина масиву дорівнює 6, нам достатньо трьох ітерацій, щоб знайти потрібне число.

Двійковий пошук – дуже могутній метод. Якщо, наприклад, довжина масиву дорівнює 1023, після першого порівняння область звужується до 511-ти елементів, а після другого – до 255. Легко порахувати, що для пошуку в масиві з 1023-х елементів достатньо 10 порівнянь.

С.2 Сортування

С.2.1 Сортування вставками

Один з простих способів відсортувати масив – *сортування вставками*.

Розглянемо сортування вставками на прикладі одновимірного масиву з 4 елементів. Перебір елементів починається з другого елемента (рис. Б.2, а). Другий елемент виймається (число 3), потім елементи, розташовані вище, зсуваються вниз доти, поки не знайдемо місце, куди потрібно вставити 3. Тобто вниз зсуваються всі елементи, більші за елемент вставки (число 3). Потім переходимо до наступного (третього) елемента масиву (рис. Б.2, б) і шукаємо позицію для числа 1 за принципом, викладеним вище. Нарешті, (рис. Б.2, в) завершуємо сортування, помістивши четвертий елемент масиву (число 2) на потрібне місце.

Якщо довжина нашого масиву дорівнює n , потрібно пройти по $n - 1$ елементу. Кожного разу може виникнути потреба зсунути $n - 1$ інших елементів. От чому цей метод вимагає дуже багато часу.

Сортування вставками належить до методів сортування *за місцем*. Іншими словами, йому не потрібна допоміжна пам'ять, отже, сортуємо елементи масиву, використовуючи тільки пам'ять, займану лише масивом. Крім того, цей метод є стійким, і якщо серед сортованих ключів є однакові, після сортування вони залишаються в початковому положенні.

С.2.2 Сортування методом Шелла

Метод, запропонований Дональдом Л. Шеллом, є нестійким сортуванням за місцем. Ефективність методу Шелла пояснюється тим, що зсуванні елементи швидко потрапляють на потрібні місця.

На рис. С.3, а наведений приклад сортування вставками. Спочатку виймаємо 1, потім зсуваємо 3 і 5 на одну позицію вниз, після чого вставляємо 1. Таким чином, було потрібно два зсуви. Наступного разу використаємо знову два зсуви, щоб вставити на потрібне місце 2. На весь процес витратимо $2+2+1=5$ зсувів.

На рис. С.3, б показано сортування методом Шелла. Починаємо сортування вставками з кроком 2. Спочатку розглядаємо числа 3 та 1: витягуємо 1, зсуваємо 3 на одну позицію з кроком 2, вставляємо 1.

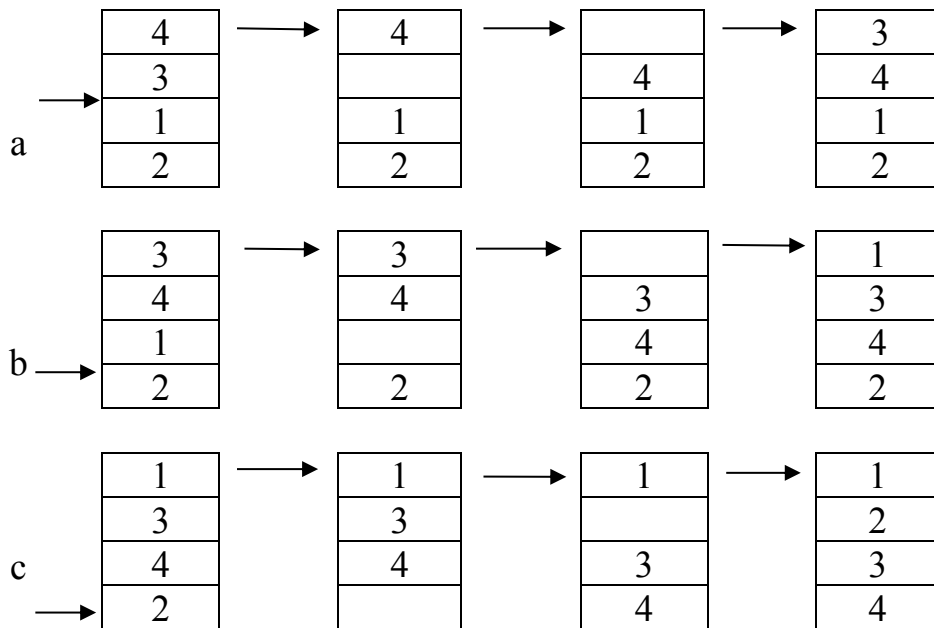


Рисунок С.2 - Сортування вставками: а, b, с – перший, другий і третій кроки відповідно

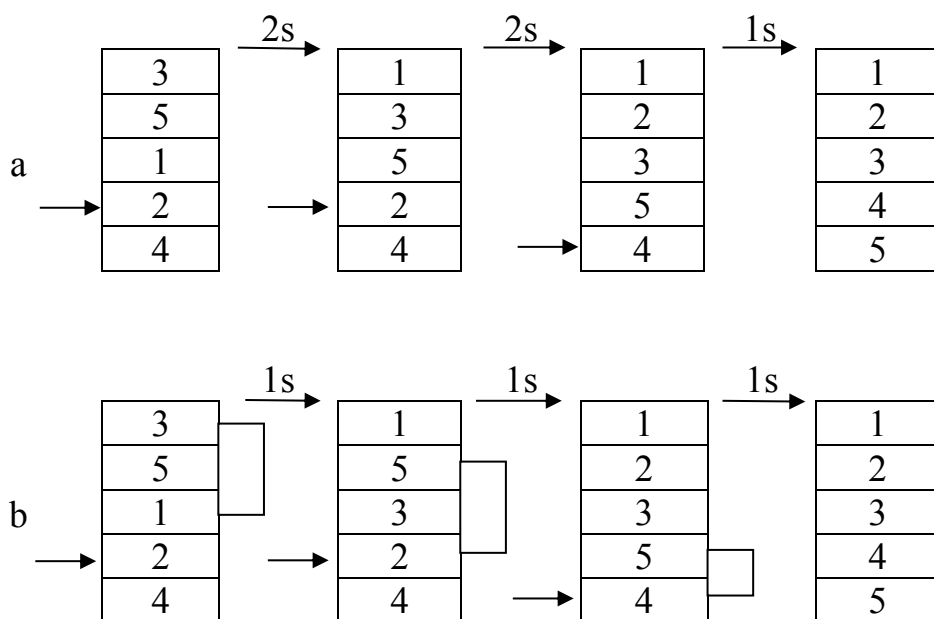


Рисунок С.3 - Сортування методом Шелла: а, b – перший і другий кроки відповідно

Потім повторюємо те саме для чисел 5 і 2: витягуємо 2, зсуваємо вниз 5, вставляємо 2 і т.д. Закінчивши сортування з кроком 2, проводимо його з кроком 1, тобто виконуємо звичайне сортування вставками.

Всього при цьому знадобиться $1 + 1 + 1 = 3$ зсуви. Таким чином, використавши спочатку крок, більший за 1, добиваємося меншої кількості зсувів. Можна використовувати найрізноманітніші схеми вибору кроків.

Як правило, спочатку сортуємо масив з великим кроком, потім зменшуємо крок і повторюємо сортування. У самому кінці сортуємо з кроком 1. Хоча цей метод легко пояснити, його формальний аналіз досить важкий. Зокрема, теоретикам не вдалося знайти оптимальну схему вибору кроків.

Формула вибору кроків h для масиву довжини N : у послідовності $h_1 = 1, h_{s+1} = 3h_s + 1$ узяти h_t , якщо $h_{t+2} \geq N$.

Ось декілька перших значень h :

$$h_1 = 1;$$

$$h_2(3 \cdot 1) + 1 = 4;$$

$$h_3(3 \cdot 4) + 1 = 13;$$

$$h_4(3 \cdot 13) + 1 = 40;$$

$$h_5(3 \cdot 40) + 1 = 121.$$

Щоб відсортувати масив завдовжки 100, перш за все знайдемо номер s , для якого $h_s \geq 100$. Згідно з наведеними формулами $s = 5$. Потрібне значення знаходиться двома рядками вище. Таким чином, послідовність кроків при сортуванні буде такою: 13-4-1. Звичайно, не потрібно зберігати цю послідовність: чергове значення h знаходиться з попереднього за формулою

$$h_{s-1} = (h_s - 1) / 3.$$

С.2.3 Швидке сортування

Цей алгоритм був розроблений Е. Хоаром. В алгоритмі швидкого сортування використовуються три ідеї:

- розділення сортованого масиву на 2 частини, ліву і праву;
- взаємне впорядкування двох частин (підмасивів) так, щоб всі елементи лівої частини не перевищували елементи правої;
- рекурсія, при якій підмасив упорядковується таким самим способом, як і весь масив.

Для розділення масиву на дві частини (підмасиви) потрібно вибрати деяке «бар'єрне» значення ключа. Це значення повинне задовольняти єдину умову: лежати в діапазоні значень для даного масиву (тобто між мінімальною і максимальною величиною).

За «бар'єр» можна вибрати значення ключа будь-якого елемента масиву, наприклад, першого або останнього, або того, що знаходиться усередині.

Далі потрібно зробити так, щоб у лівому підмасиві опинилися всі елементи з ключем, меншим за бар'єр, а в правому – більшим.

Потім, проглядаючи масив зліва направо, необхідно знайти позицію першого елемента з ключем, більшим за бар'єр, а переглядаючи справа наліво – знайти перший елемент з ключем, меншим за бар'єр. Слід поміняти ці значення, потім продовжити зустрічний рух до наступної пари елементів, призначених для обміну.

Необхідно повторювати цю процедуру, поки між індексами лівого і правого переглядів не відбудеться збіг. Місце збігу стане межею між двома взаємно впорядкованими підмасивами.

Далі алгоритм рекурсивно застосовується до кожного з підмасивів (лівого і правого). Кінець кінцем приходимо до сукупності з n взаємно впорядкованих одноелементних масивів, які ділити далі неможливо. Ця сукупність утворює один повністю впорядкований масив. Сортування завершено.

С.3 Висновки

У даному додатку розглянуті наведені нижче основні питання:

- лінійний пошук;
- сортування вставками;
- сортування методом Шелла;
- швидке сортування.

С.4 Контрольні питання

1. Які основні операції застосовуються при роботі з даними?
2. У чому полягає лінійний пошук?
3. У чому полягає двійковий пошук?
4. Коли застосовують двійковий пошук?
5. Доведіть, що для двійкового пошуку в масиві з 1023-х елементів достатньо 10 порівнянь.
6. Які відомі способи сортування?
7. Який спосіб сортування є найпростішим?
8. У чому полягає спосіб швидкого сортування?
9. Розкажіть про спосіб сортування Шелла?
10. У чому полягає сортування вставками?
11. Який спосіб сортування розробив Е. Хоар?

12. Розкажіть про термін „бар’єр”, що вводиться при сортуванні?
13. При якому способі сортування вводиться термін „бар’єр”?
14. Який спосіб сортування належить до методів сортування за місцем?
15. Які методи сортування є стійкими за місцем?

ТКАЧОВ Віктор Васильович
НАДТОЧИЙ Володимир Валентинович

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ
ЕЛЕКТРОТЕХНІЧНИЙ ФАКУЛЬТЕТ
Кафедра кіберфізичних та інформаційно-вимірювальних систем

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ТА ПРОГРАМУВАННЯ

Методичні рекомендації до виконання навчальної практики
з дисципліни для студентів спеціальності
151 «Автоматизація та комп'ютерно-інтегровані технології»

Видано в авторській редакції

Національний технічний університет «Дніпровська політехніка»
49005, м. Дніпро, пр. Дмитра Яворницького, 19.