

Лабораторна робота №3

Використання масивів

Мета

Вивчити можливості мови програмування C для організації упорядкованих структур даних, оволодіти основними принципами оброблення одновимірних та двовимірних масивів, вивчити переваги використання масивів для вирішення практичних задач.

Завдання 1

Доопрацювати програму, яка перевіряє, чи є у числі цифри, що повторюються (див. лекції), щоб вона виводила, які саме цифри повторюються та скільки разів. Приклад роботи програми:

Введіть число: 272767

Є цифри, що повторюються: 2 (2 рази) 7 (3 рази)

Підготовлення інформації про кількість цифр, що повторюються, має бути реалізоване у вигляді окремої функції мови C. Виведення результатів базуючись на підготовленій інформації також має бути реалізоване у вигляді окремої функції мови C.

Розробити модульні тести з використанням Google Test Framework для перевірки коректності роботи функції.

Завдання 2

Розробити функцію з оброблення одновимірних масивів згідно з варіантом завдання (див. додаток А). Кількість елементів масиву n задається користувачем (максимальна кількість – 100). Передбачити можливість введення елементів масиву як з клавіатури, так і з використанням функції `rand()`. Розробити програму, що демонструє використання розроблених функцій.

Завдання 3

Розробити функцію з оброблення двовимірних масивів (матриць) згідно з варіантом завдання (див. додаток Б). Передбачити можливість введення елементів матриці як з клавіатури, так і з використанням функції `rand()`. Обмежити діапазон випадкових чисел згідно з варіантом завдання. Обчислення визначника реалізувати у вигляді окремої функції. Розробити програму, що демонструє використання розроблених функцій.

Завдання 4

Розробити програму згідно з варіантом завдання (див. додаток В). У колоді карт має бути 52 карти. Масті та ранги карт необхідно задати з використанням константних масивів. Здача карт має виконуватися

випадковим чином. Передбачити неможливість повторної здачі тієї ж самої карти.

Примітка. Для виведення мастей карт можна використовувати такі коди символів:

♥ '\x03'

♦ '\x04'

♣ '\x05'

♠ '\x06'

Вимоги до звіту

Тексти програм мають починатися з коментаря, в якому зазначається номер і назва лабораторної роботи, номер завдання, а також прізвище студента. Наприклад, для першого завдання коментар має бути таким:

```
/**
 * @file   lab3_1.c
 * @author Петренко П.П., гр. 515
 * @date   03 березня 2022
 * @brief  Лабораторна робота № 3
 *
 * Використання масивів. Завдання 1
 */
```

Звіт з лабораторної роботи має включати:

1. титульний аркуш із зазначенням номеру та назви лабораторної роботи;
2. мету роботи;
3. варіант і тексти завдань;
4. схеми алгоритмів;
5. вихідні тексти програм;
6. результати роботи програм (скріншоти або текст, який виводять програми);
7. тестові набори + посилання на онлайн проєкти;
8. модульні тести та результати їх виконання;
9. висновки (що було зроблено, за допомогою яких засобів, що було вивчено і т.ін.).

Контрольні питання

1. Оголошення масивів.
2. Що таке індексація масивів?
3. Елементи яких типів даних можуть бути організовану у масиви?
4. Яким має бути тип індексу масиву?
5. Оголошено масив `int p[]={2, 4, 6, 10, 1};`
Чому дорівнює значення елемента `p[3]`?
6. Оголошено масив `int a[6]={5, 3, 2};`
Чому дорівнює елемент `a[4]`?
7. Оголошено двовимірний масив `m[i][j]`. Який індекс позначає рядки, а який стовпці?

8. Оголошено масив `int m[3][3]={2, 4, 6, 10, 1, 12, 4, 0, 0};` Чому дорівнює значення елемента `m[2][2]`?
9. Оголошено масив `int m[2][2]={ {5, 3}, {0, 2} };` Чому дорівнює елемент `m[1][1]`?
10. Оголошено матрицю `m[4][5]`. Чому дорівнюють максимальні значення індексів для рядків та для стовпців?
11. Як розміщуються в пам'яті багатовимірні масиви?
12. Константні масиви. Відмінності від «звичайних» масивів.

Додаток А. Варіанти завдань (Завдання 2)

1. Обчислити добуток елементів масиву, що знаходяться між максимальним та мінімальним елементами.
2. Обчислити суму елементів масиву, що знаходяться між першим та останнім нульовими елементами.
3. Обчислити суму елементів масиву, що знаходяться до останнього додатного елемента.
4. Обчислити суму елементів масиву, що знаходяться між першим та останнім додатними елементами.
5. Вичислить добуток елементів масиву, що знаходяться між першим та вторим нульовими елементами.
6. Обчислити суму елементів масиву, що знаходяться між першим та вторим від'ємними елементами.
7. Обчислити суму елементів масиву, що знаходяться до мінімального елемента.
8. Обчислити суму модулів елементів масиву, що знаходяться після останнього від'ємного елемента.
9. Обчислити суму елементів масиву, що знаходяться після останнього елемента, що дорівнює нулю.
10. Обчислити суму модулів елементів масиву, що знаходяться після мінімального по модулю елемента.
11. Обчислити суму елементів масиву, що знаходяться після мінімального елемента.
12. Обчислити суму елементів масиву, що знаходяться після першого додатного елемента.
13. Обчислити суму модулів елементів масиву, що знаходяться після першого від'ємного елемента.
14. Обчислити суму модулів елементів масиву, що знаходяться після першого елемента, що дорівнює нулю.
15. Обчислити суму додатних елементів масиву, що знаходяться до максимального елемента.

16. Обчислити суму від'ємних елементів масиву, що знаходяться до мінімального елемента.

17. Обчислити суму модулів елементів масиву, що знаходяться після першого від'ємного елемента.

18. Обчислити суму елементів масиву, що знаходяться після першого елемента, що дорівнює нулю.

19. Обчислити суму модулів елементів масиву, що знаходяться після останнього елемента, який є непарним числом.

20. Обчислити суму елементів масиву, що знаходяться після останнього елемента, який є парним числом.

21. Обчислити суму елементів масиву с парними індексами, що знаходяться між першим та вторим від'ємними елементами.

22. Обчислити суму елементів масиву с непарними індексами, що знаходяться до мінімального елемента.

23. Обчислити суму модулів елементів масиву с парними індексами, що знаходяться після останнього від'ємного елемента.

24. Обчислити суму елементів масиву с непарними індексами, що знаходяться після останнього елемента, що дорівнює нулю.

25. Обчислити суму модулів елементів масиву с парними індексами, що знаходяться після мінімального по модулю елемента.

26. Обчислити суму додатних елементів масиву, кратних п'яти.

27. Обчислити суму елементів масиву, які по модулю більше середнього значення елементів масиву.

28. Обчислити суму елементів масиву, які більше максимального непарного числа з масиву.

29. Обчислити суму елементів масиву, кратних десяти.

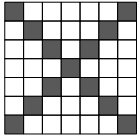
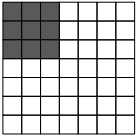
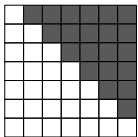
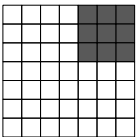
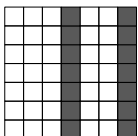
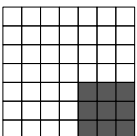
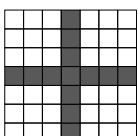
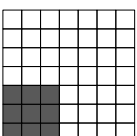
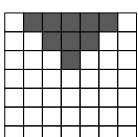
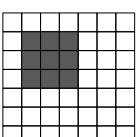
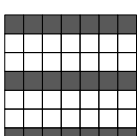
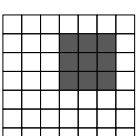
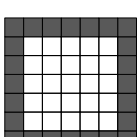
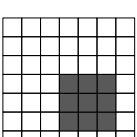
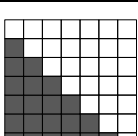
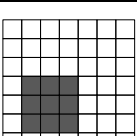
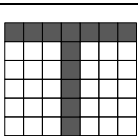
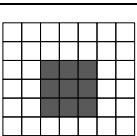
30. Обчислити суму непарних елементів масиву, значення яких знаходяться у діапазоні від 1 до 9 включно.

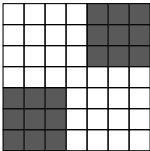
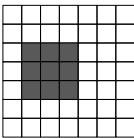
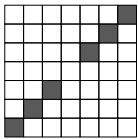
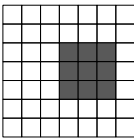
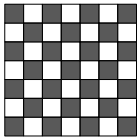
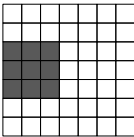
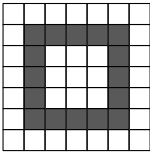
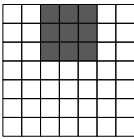
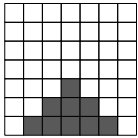
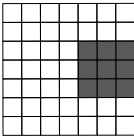
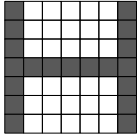
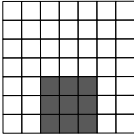
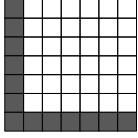
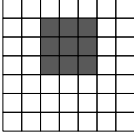
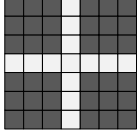
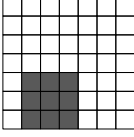
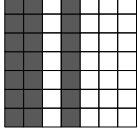
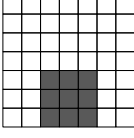
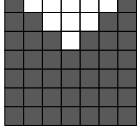
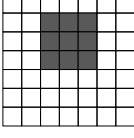
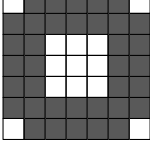
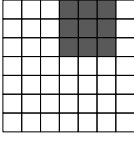
Додаток Б. Варіанти завдань (Завдання 3)

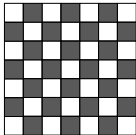
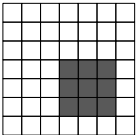
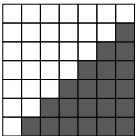
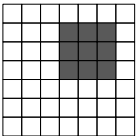
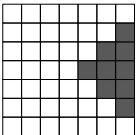
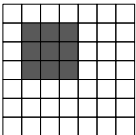
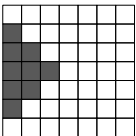
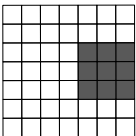
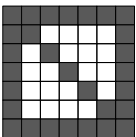
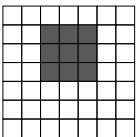
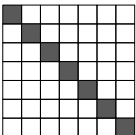
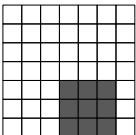
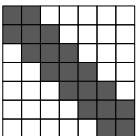
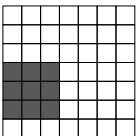
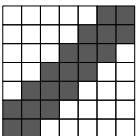
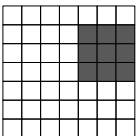
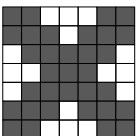
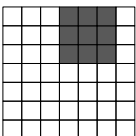
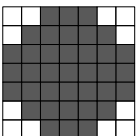
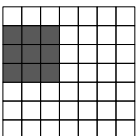
Задача 3.1 – знайти суму виділених елементів матриці.

Задача 3.2 – обчислити визначник виділеної підматриці розміром 3x3.

При заповненні матриці випадковими числами вони мають знаходитися у межах діапазону, заданого для даного варіанту.

| № | Задача 3.1 | Задача 3.2 | Діапазон випадкових чисел |
|----|---|---|---------------------------|
| 1. |  |  | -100..100 |
| 2. |  |  | -50..50 |
| 3. |  |  | 0..100 |
| 4. |  |  | -100..0 |
| 5. |  |  | 10..150 |
| 6. |  |  | -10..80 |
| 7. |  |  | -70..70 |
| 8. |  |  | -45..45 |
| 9. |  |  | -20..150 |

| | | | |
|-----|---|---|-----------|
| 10. |  |  | -45..85 |
| 11. |  |  | -10..10 |
| 12. |  |  | -250..250 |
| 13. |  |  | 0..75 |
| 14. |  |  | 0..120 |
| 15. |  |  | -30..30 |
| 16. |  |  | -90..90 |
| 17. |  |  | -70..70 |
| 18. |  |  | 0..300 |
| 19. |  |  | -40..120 |
| 20. |  |  | 0..50 |

| | | | |
|-----|---|---|-----------|
| 21. |  |  | -120..120 |
| 22. |  |  | -75..75 |
| 23. |  |  | 0..800 |
| 24. |  |  | 0..1000 |
| 25. |  |  | -50..50 |
| 26. |  |  | -150..150 |
| 27. |  |  | -100..100 |
| 28. |  |  | -50..110 |
| 29. |  |  | -36..36 |
| 30. |  |  | -220..220 |

Додаток В. Варіанти завдань (Завдання 4)

1. Програма здає по 6 карт чотирьом гравцям, випадковим чином призначає козирну масть та підраховує кількість козирних карт у кожного гравця.
2. Програма здає по 6 карт чотирьом гравцям та підраховує кількість «великих» карт (В, Д, К, Т) у кожного гравця.
3. Програма здає по 6 карт чотирьом гравцям та підраховує кількість «дрібних» карт (2, 3, 4, 5, 6, 7, 8, 9, 10) у кожного гравця.
4. Програма здає по 9 карт трьом гравцям та підраховує кількість послідовностей з трьох карт по порядку (наприклад, **2♦ 3♠ 4♣**, **10♥ В♠** Д♠ і т.п.) у кожного гравця.
5. Програма здає по 9 карт трьом гравцям та підраховує кількість послідовностей з трьох карт одного ранга у кожного гравця.
6. Програма здає по 6 карт чотирьом гравцям та підраховує кількість послідовностей з двох карт одного ранга у кожного гравця.
7. Програма здає по 9 карт трьом гравцям та підраховує кількість послідовностей з трьох карт однієї масті по порядку (наприклад, **7♣ 8♣ 9♣**, **В♠ Д♠ К♠** і т.п.) у кожного гравця.
8. Програма здає по 9 карт трьом гравцям та підраховує кількість послідовностей з трьох та більше карт однієї масті у кожного гравця.
9. Програма здає по 6 карт трьом гравцям, випадковим чином призначає козирну масть та підраховує кількість «великих» козирних карт (В, Д, К, Т) у кожного гравця.
10. Програма здає по 6 карт трьом гравцям, випадковим чином призначає козирну масть та підраховує кількість «дрібних» козирних карт (2, 3, 4, 5, 6, 7, 8, 9, 10) у кожного гравця.
11. Програма здає по 6 карт трьом гравцям та перевіряє наявність комбінації «фул-хаус» (три карти одного ранга та дві карти другого ранга, наприклад **10♥ 10♦ 10♠ 8♣ 8♥**) у кожного гравця.
12. Програма здає по 6 карт трьом гравцям та перевіряє наявність комбінації «флеш» (п'ять карт однієї масті, наприклад **К♠ В♠ 8♠ 4♠ 3♠**) у кожного гравця.
13. Програма здає по 9 карт чотирьом гравцям та перевіряє наявність комбінації «каре» (чотири карти одного ранга, наприклад **3♥ 3♦ 3♣ 3♠**) у кожного гравця.
14. Програма здає по 6 карт чотирьом гравцям, випадковим чином призначає козирну масть та перевіряє наявність комбінації «белла» (козирний король та козирна дама) у кожного гравця.
15. Програма здає по 6 карт чотирьом гравцям та перевіряє наявність комбінації «терц» (три карти однієї масті в ряд, наприклад **9♠ 10♠ В♠**) у кожного гравця.
16. Програма здає по 6 карт чотирьом гравцям та перевіряє наявність комбінації «полтина» (чотири карти однієї масті в ряд, наприклад **10♥ В♥ Д♥ К♥**) у кожного гравця.

17. Програма здає по 6 карт чотирьом гравцям, випадковим чином призначає козирну масть та підраховує кількість некозирних карт у кожного гравця.

18. Програма здає по 6 карт чотирьом гравцям, випадковим чином призначає козирну масть та перевіряє наявність козирного валета та козирної дев'ятки у кожного гравця.

19. Програма здає по 9 карт трьом гравцям, випадковим чином призначає козирну масть та підраховує кількість козирних карт у кожного гравця.

20. Програма здає по 6 карт чотирьом гравцям та підраховує кількість послідовностей з трьох або більше карт одного ранга у кожного гравця.

21. Програма здає по 6 карт чотирьом гравцям та перевіряє наявність комбінації «пара» (дві карти одного ранга, наприклад 9♠ 9♥) у кожного гравця.

22. Програма здає по 6 карт трьом гравцям та перевіряє наявність комбінації «тройка» (три карти одного ранга 8♠ 8♣ 8♥) у кожного гравця.

23. Програма здає по 6 карт чотирьом гравцям та перевіряє наявність комбінації «дві пари» (дві комбінації по дві карти одного ранга, наприклад 10♠ 10♥ 2♣ 2♥) у кожного гравця.

24. Програма здає по 6 карт чотирьом гравцям, випадковим чином призначає козирну масть та перевіряє наявність комбінації «козирний терц» (три козирних карти однієї масті в ряд, наприклад В♠ Д♠ К♠) у кожного гравця.

25. Програма здає по 6 карт трьом гравцям, перевіряє наявність комбінації «терц» (три карти однієї масті в ряд, наприклад 7♠ 8♠ 9♠) у кожного гравця, виводить, у кого з гравців «терц» має найбільший «зріст» (найбільший ранг у останньої карти комбінації).

26. Програма здає по 9 карт трьом гравцям та перевіряє наявність комбінації «стрит» (п'ять послідовних по рангу карт різних мастей, наприклад 6♠ 5♠ 4♥ 3♦ 2♦) у кожного гравця.

27. Програма здає по 9 карт трьом гравцям та перевіряє наявність комбінації «стрит-флеш» (п'ять послідовних по рангу карт однієї масті, наприклад 9♣ 8♣ 7♣ 6♣ 5♣) у кожного гравця.

28. Програма здає по 6 карт трьом гравцям, випадковим чином призначає козирну масть та підраховує кількість «великих» некозирних карт (В, Д, К, Т) у кожного гравця.

29. Програма здає по 6 карт трьом гравцям, випадковим чином призначає козирну масть та підраховує кількість «дрібних» некозирних карт (2, 3, 4, 5, 6, 7, 8, 9, 10) у кожного гравця.

30. Програма здає по 9 карт трьом гравцям та перевіряє наявність комбінації «роял-флеш» (п'ять послідовних по рангу карт однієї масті починаючи з 10, наприклад Т♥ К♥ Д♥ В♥ 10♥) у кожного гравця.