

Лабораторна робота №5

Застосування вказівників

Мета

Отримати практичні навички у роботі з вказівниками та адресною арифметикою у мові C, засвоїти застосування вказівників як параметрів функцій.

Завдання 1

Реалізувати функцію згідно варіанту завдання (див. Додаток А). Функція має отримувати число та повертати результати його перетворення за допомогою параметрів-вказівників. Повертане значення функції має містити статус перетворення (1 – успішно, 0 – неуспішно). Продемонструвати застосування функції в основній програмі.

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

Завдання 2

Реалізувати функцію, що виконує операції над одновимірним масивом згідно варіанту завдання (див. Додаток Б). Оброблення масиву необхідно реалізувати за допомогою вказівників, без застосування індексації. Продемонструвати роботу функції в основній програмі.

Завдання 3

Реалізувати функцію (або декілька функцій, залежно від варіанту), що виконує необхідні операції згідно варіанту завдання (див. Додаток В). Продемонструвати роботу функції (функцій) у основній програмі.

Вимоги до звіту

Тексти програм мають починатися з коментаря, в якому зазначається ім'я файлу з вихідним кодом, прізвище та групу студента, дату складання, номер і назву лабораторної роботи, номер завдання:

```
/**
 * @file lab5_1.cpp
 * @author Петренко П.П., гр. 515
 * @date 27 березня 2022
 * @brief Лабораторна робота № 5
 *
 * Застосування вказівників. Завдання 1
 */
```

До прототипу кожної функції має бути записаний коментар, що містить інформацію щодо її призначення, параметрів та повертаного значення:

```
/**
 * Отримання року та місяця випуску з серійного номеру
 *
 * @param serial серійний номер
 * @param year рік випуску
 * @param month місяць випуску
 * @return 1, якщо операція успішна
 */
int get_date (long long serial, int * year, int * month);
```

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;
8. модульні тести та результати їх виконання для завдання 1;
9. висновки (що було зроблено, за допомогою яких засобів, що було вивчено і т.ін.).

Контрольні питання

1. Вказівники. Призначення, відмінності від змінних цілого типу.
2. Області застосування вказівників.
3. Операції розіменування та взяття адреси.
4. Що таке посилальний тип даних?
5. Чи може вказівник вказувати на інший вказівник?
6. Чи можна присвоювати між собою змінні, що мають тип даних `int*` та `float*`?
7. Які арифметичні операції є допустимими над вказівниками?
8. Який зв'язок між іменем масиву та вказівниками?
9. Чи можна застосовувати індексацію для вказівників?
10. Вказівником на що є ім'я двовимірного масиву?
11. Для чого необхідно застосовувати вказівники як параметри функцій?
12. Чи може функція повертати вказівник?

Додаток А. Варіанти завдань (завдання 1)

1. Розробити функцію, що отримує дату як ціле число у форматі YYYYMMDD, та за допомогою параметрів повертає вказівники на рік, місяць, число. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, вхідне число містить недопустимий місяць) – повертає 0.

2. Розробити функцію, що отримує суму як число з плаваючою точкою, та за допомогою параметрів повертає вказівники на суму в гривнях (до десяткової точки) та суму в копійках (після десяткової точки). У випадку, якщо сума в копійках містить більш ніж два знаки після десяткової точки, функція повертає 0, якщо ні – повертає 1.

3. Розробити функцію, що отримує час як ціле число у форматі HHMMSS, та за допомогою параметрів повертає вказівники на години, хвилини, секунди. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, вхідне число містить недопустиму кількість хвилин) – повертає 0.

4. Розробити функцію, що отримує час як кількість секунд, та за допомогою параметрів повертає вказівники на години, хвилини, секунди, що відповідають даній кількості секунд. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, вхідне число від'ємне) – повертає 0.

5. Розробити функцію, що отримує дату як номер дня року (від 1 до 365 або 366) та кількість днів у році, та за допомогою параметрів повертає вказівники на число та місяць, що відповідають даній даті. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, було передано від'ємне число, або передана кількість днів у році не дорівнює 365 або 366) – повертає 0.

6. Розробити функцію, що отримує суму в гривнях, та за допомогою параметрів повертає вказівники на кількість найбільших купюр номіналом 20, 10, 5, 2 та 1 гривня, з яких можна зібрати дану суму. Наприклад, при передаванні 27 гривень функція має повернути 1 купюру номіналом 20 гривень, 1 купюру номіналом 5 гривень та 1 купюру номіналом 2 гривні. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передана від'ємна сума) – повертає 0.

7. Розробити функцію, що отримує маркування шини, що складається з 11 цифр, та за допомогою параметрів повертає вказівники на ширину шини, висоту профіля, діаметр покришки, тиждень та рік виробництва. Перші три цифри – ширина (наприклад, 215), четверта та п'ята – висота профіля (наприклад, 65), шоста та сьома – діаметр покришки (наприклад, 16), восьма та дев'ята – тиждень виробництва (наприклад, 12), десята та одинадцята – дві останні цифри року виробництва (наприклад, 21). У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 11 цифр, тиждень виробництва є некоректним і т.п.) – повертає 0. Як тип даних для зберігання маркування застосовувати long long.

8. Розробити функцію, що отримує дату як ціле число у форматі ДДММРРРР, та за допомогою параметрів повертає вказівники на число, місяць, рік. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, вхідне число містить недопустимий місяць або недопустимий день місяця) – повертає 0.

9. Розробити функцію, що отримує код, який складається з 6 цифр, та за допомогою параметрів повертає вказівники на число, кількість годин та кількість хвилин. Перші дві цифри коду означають число, другі – години, треті – хвилини. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, код містить некоректне число, у коді менше ніж 6 цифр і т.п.) – повертає 0.

10. Розробити функцію, що отримує час як кількість годин, та за допомогою параметрів повертає вказівники на тижні, дні, години, що відповідають даній кількості годин. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передана від'ємна кількість годин) – повертає 0.

11. Розробити функцію, що отримує серійний номер виробу, що складається з 6 цифр, та за допомогою параметрів повертає вказівники на місяць, рік та партію даного виробу. Перші дві цифри серійного номеру означають місяць, другі – рік, треті – партію. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, серійний номер містить некоректний місяць) – повертає 0.

12. Розробити функцію, що отримує суму в копійках, та за допомогою параметрів повертає вказівники на кількість монет номіналом 50, 25, 10, 5, 2 та 1 копійка, з яких можна зібрати дану суму. Наприклад, при передаванні 19 копійок функція має повернути 1 монету номіналом 10 копійок, 1 монету номіналом 5 копійок та 2 монети номіналом 2 копійки. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передана від'ємна сума) – повертає 0.

13. Розробити функцію, що отримує фрагмент метеорологічного коду METAR, що складається з 6 цифр, та за допомогою параметрів повертає вказівники на число поточного місяця, кількість годин та кількість хвилин. Перші дві цифри фрагменту означають число поточного місяця, другі – години, треті – хвилини. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, код містить некоректне число, у коді менше, ніж 6 цифр і т.п.) – повертає 0.

14. Розробити функцію, що отримує реєстраційний номер облікової картки платника податків (як ціле число у форматі NNNNNNNNNN, та за допомогою параметрів повертає вказівники на число, місяць, рік народження людини з даним реєстраційним номером, а також його стать. Для розрахунку необхідно застосовувати перші п'ять цифр реєстраційного номеру, що являють собою номер дня народження, починаючи з 1 січня 1900 року, для визначення статі – 9-у цифру (непарна цифра – чоловіча стать, парна – жіноча). У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні – повертає 0. Як тип даних для зберігання реєстраційного номеру облікової картки платника податків застосовувати long long.

15. Розробити функцію, що отримує серійний номер акумулятора, що складається з 6 цифр, та за допомогою параметрів повертає вказівники на номер конвеєру, число, місяць та рік випуску акумулятору. Перша цифра серійного номеру означає номер конвеєру, на якому був зібраний акумулятор, друга цифра означає останню цифру року виготовлення, третя та четверта цифри є місяцем, п'ята та шоста є днем. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, місяць містить недопустиме значення, к серійному номері менше, ніж 6 цифр і т.п.) – повертає 0.

16. Розробити функцію, що отримує серійний номер, що складається з 6 цифр, та за допомогою параметрів повертає вказівники на рік, місяць та партію. Перша цифра серійного номеру – це остання цифра року, друга та третя – місяць, четверта, п'ята та шоста – номер партії. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, місяць містить недопустиме значення і т.п.) – повертає 0.

17. Розробити функцію, що отримує серійний номер велосипеда, що складається з 8 цифр, та за допомогою параметрів повертає вказівники на розмір рами, рік випуску, місяць випуску та порядковий номер. Перша цифра серійного номеру – це код розміру рами (1 – S, 2 – M, 3 – L, 4 – XL), друга та третя – місяць випуску, четверта та п'ята – рік випуску, шоста, сьома та восьма – порядковий номер. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 8 цифр, перша цифра містить некоректний розмір рами і т.п.) – повертає 0.

18. Розробити функцію, що отримує суму в центрах, та за допомогою параметрів повертає вказівники на кількість монет номіналом 1 долар, 50, 25, 10, 5, 2 та 1 цент, з яких можна зібрати дану суму. Наприклад, при передаванні 126 центів функція має повернути 1 монету номіналом 1 долар, 1 монету номіналом 25 центів та 1 монету номіналом 1 цент. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передана від'ємна сума) – повертає 0.

19. Розробити функцію, що отримує серійний номер телевізора, що складається з 6 цифр, та за допомогою параметрів повертає вказівники на розмір діагоналі екрану, рік випуску, серію та тип тюнеру. Перші дві цифри серійного номеру – це розмір діагоналі, третя – остання цифра року випуску, четверта та п'ята – серія, шоста – тип тюнеру. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 6 цифр і т.п.) – повертає 0.

20. Розробити функцію, що отримує маркування лампи, що складається з 8 цифр, та за допомогою параметрів повертає вказівники на напругу (перші 3 цифри), потужність (четверта, п'ята та шоста цифри) та діаметр цоколю у мм (сьома та восьма цифри). У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 8 цифр і т.п.) – повертає 0.

21. Розробити функцію, що отримує міжнародний стандартний книжний номер (ISBN), що складається з 10 цифр, та за допомогою параметрів повертає вказівники на код групи країн, код видавництва, унікальний номер видання та контрольну цифру. Перші дві цифри – код групи країн, третя, четверта та п'ята – код видавництва, шоста, сьома, восьма та дев'ята – унікальний номер видання, десята – контрольна цифра. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 10 цифр і т.п.) – повертає 0.

22. Розробити функцію, що отримує маркування лампи, що складається з 3 цифр, та за допомогою параметрів повертає вказівники на індекс кольоропередачі та кольорову температуру. Індекс кольоропередачі обчислюється як перша цифра, помножена на 10. Кольорова температура – як друга та третя цифри, помножені на 1000. Наприклад, для переданого маркування 827 функція має повернути вказівники на числа 80 та 2700. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 3 цифри, передане число є від'ємним і т.п.) – повертає 0.

23. Розробити функцію, що отримує серійний номер, який складається з 7 цифр, та за допомогою параметрів повертає вказівники на рік випуску, номер тижня випуску та порядковий номер. Перші дві цифри серійного номеру – це останні дві цифри року, наступні дві – номер тижня року, наступні три – порядковий номер. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, номер тижня містить недопустиме значення, серійний номер складається з меншої кількості цифр і т.п.) – повертає 0.

24. Розробити функцію, що отримує серійний номер, який складається з 7 цифр, та за допомогою параметрів повертає вказівники на рік випуску, номер тижня випуску та порядковий номер. Перша цифра серійного номеру – це остання цифра року, наступні дві – номер тижня року, наступна одна – день тижня (Пн-1, Вт-2, Ср-3, Чт-4, Пт-5), наступні дві – порядковий номер. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, день тижня містить недопустиме значення, серійний номер складається з меншої кількості цифр і т.п.) – повертає 0.

25. Розробити функцію, що отримує суму в гривнях, та за допомогою параметрів повертає вказівники на кількість купюр номіналом 500, 200, 100, 20, 10, 5, 2 та 1 гривня, з яких можна зібрати дану суму. Наприклад, при передаванні суми 524 гривні функція має повернути 1 купюру номіналом 500 гривень, 1 купюру номіналом 20 гривень та 2 купюри номіналом 2 гривні. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передана від'ємна сума) – повертає 0.

26. Розробити функцію, що отримує серійний номер планшету, що складається з 7 цифр, та за допомогою параметрів повертає вказівники на розмір діагоналі екрану, рік випуску, порядковий номер та архітектуру процесору. Перші дві цифри серійного номеру – це розмір діагоналі, третя – остання цифра року випуску, четверта, п'ята та шоста – серійний номер, сьома – архітектура процесору (0 – ARM, 1 – x86, 2 – ARM64). У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 7 цифр і т.п.) – повертає 0.

27. Розробити функцію, що отримує номер патентної заявки, що складається з 8 цифр, та за допомогою параметрів повертає вказівники на рік подачі заявки, місяць подачі заявки та її порядковий номер. Перші дві цифри серійного номеру – це останні цифри року подачі, третя та четверта цифри – номер місяця подачі, останні цифри – порядковий номер. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, місяць містить недопустиме значення, серійний номер складається з меншої кількості цифр і т.п.) – повертає 0.

28. Розробити функцію, що отримує серійний номер вентиля, що складається з 7 цифр, та за допомогою параметрів повертає вказівники на розмір вентиля, тип, рік випуску та порядковий номер. Перша цифра серійного номеру – це код розміру, друга – код типу, третя та четверта – рік випуску, п'ята, шоста та сьома – порядковий номер. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, передане число містить менше, ніж 7 цифр і т.п.) – повертає 0.

29. Розробити функцію, що отримує серійний номер, що складається з 9 цифр, та за допомогою параметрів повертає вказівники на рік, місяць, число та порядковий номер. Перша цифра серійного номеру – це остання цифра року, друга та третя – місяць, четверта та п'ята – число, шоста, сьома, восьма та дев'ята – порядковий номер. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, місяць містить недопустиме значення і т.п.) – повертає 0.

30. Розробити функцію, що отримує серійний номер, що складається з 6 цифр, та за допомогою параметрів повертає вказівники на номер заводу, рік випуску, номер тижня та код кольору. Перші дві цифри серійного номеру – це номер заводу, третя цифра – остання цифра року, четверта та п'ята – номер тижня року, шоста – колір. У випадку, якщо перетворення було виконане успішно, функція повертає 1, якщо ні (наприклад, номер тижня містить недопустиме значення і т.п.) – повертає 0.

Додаток Б. Варіанти завдань (завдання 2)

1. Розробити функцію, що знаходить мінімальний та максимальний непарні елементи масиву.
2. Розробити функцію, що знаходить мінімальний та максимальний парні елементи масиву.
3. Розробити функцію, що знаходить мінімальний елемент масиву та середнє значення усіх елементів.
4. Розробити функцію, що знаходить максимальний елемент масиву та середнє значення усіх елементів.
5. Розробити функцію, що знаходить середні значення усіх парних елементів масиву та усіх непарних.
6. Розробити функцію, що знаходить кількість додатних, від'ємних та таких що дорівнюють нулю елементів масиву.
7. Розробити функцію, що знаходить кількість парних, непарних та таких що дорівнюють нулю елементів масиву.
8. Розробити функцію, що знаходить кількість елементів масиву, які менше 100, більше 100 та дорівнюють 100.
9. Розробити функцію, що знаходить кількість елементів масиву, кратних трьом, кількість елементів масиву, кратних п'яти та кількість елементів масиву, кратних семи.
10. Розробити функцію, що знаходить кількість елементів масиву, що дорівнюють одиниці, кількість елементів масиву, що дорівнюють двом та кількість елементів масиву, що дорівнюють трьом.
11. Розробити функцію, що знаходить два найменших елементи масиву.
12. Розробити функцію, що знаходить кількість елементів масиву, що знаходяться у діапазоні 0..100 включно, та кількість елементів масиву, що знаходяться у діапазоні 101..200 включно.
13. Розробити функцію, що знаходить два найбільших парних елемента масиву.
14. Розробити функцію, що знаходить кількість елементів масиву, кратних дев'яти, та кількість елементів масиву, кратних одинадцяти.
15. Розробити функцію, що знаходить два найменших непарних елемента масиву.
16. Розробити функцію, що знаходить два найбільших парних елемента масиву.
17. Розробити функцію, що знаходить кількість елементів масиву, що знаходяться у діапазоні -100..0 включно, кількість елементів масиву, що знаходяться у діапазоні 1..100 включно, та кількість елементів, що не попадають у вказані діапазони.
18. Розробити функцію, що знаходить три найбільших елемента масиву.
19. Розробити функцію, що знаходить кількість елементів масиву, які менше ніж середнє значення, дорівнюють йому та більше його.

20. Розробити функцію, яка повертає три елемента масиву, найбільш близькі до нуля.
21. Розробити функцію, що знаходить кількість елементів масиву, що знаходяться у діапазоні 0..59, діапазоні 60..74, діапазоні 75..89 та діапазоні 90..100.
22. Розробити функцію, що знаходить кількість елементів масиву, що знаходяться у діапазоні 0..2, кількість елементів масиву, що дорівнюють 3, кількість елементів масиву, що дорівнюють 4 та кількість елементів масиву, що дорівнюють 5.
23. Розробити функцію, що знаходить суму непарних елементів масиву, суму парних елементів масиву та суму усіх елементів.
24. Розробити функцію, що знаходить кількість чотиризначних чисел, тризначних чисел, двозначних чисел серед елементів масиву.
25. Розробити функцію, що знаходить кількість елементів масиву, що знаходяться у діапазоні 1800..1899, діапазоні 1900..1999, діапазоні 2000..2022, а також кількість чисел, що не входять до вказаних діапазонів.
26. Розробити функцію, що знаходить кількість елементів масиву, які знаходяться у діапазоні 0..59, діапазоні 60..67, діапазоні 68..74, діапазоні 75..82, діапазоні 83..89 та діапазоні 90..100.
27. Розробити функцію, яка повертає три елемента масиву, найбільш близькі до десяти.
28. Розробити функцію, що знаходить кількість елементів масиву, які менше 10, більше 10 та дорівнюють 10.
29. Розробити функцію, що знаходить кількість елементів масиву, які кратні 100, а також кількість елементів, які кратні 3.
30. Розробити функцію, що знаходить найменший, найбільший та найбільш близький до середнього значення елементи масиву.

Додаток В. Варіанти завдань (завдання 3)

1. В математиці, інформатиці та інших науках під терміном «вектор» розуміється впорядкований набір чисел. Для представлення векторів у програмі достатньо зручно застосовувати масиви. Операція додавання двох векторів полягає у додаванні кожної з їх складових друг з другом. Наприклад: $(0, 1, 4) + (3, -2, 5) = (0+3, 1-2, 4+5) = (3, -1, 9)$. Необхідно реалізувати функцію `void add_vect(int *v1, int *v2, int *v3, int n);` Дана функція додає вектори `v1` та `v2`, результат поміщує у `v3`. `n` – довжина векторів. Продемонструвати застосування функції в основній програмі.

2. В математиці, інформатиці та інших науках під терміном «вектор» розуміється впорядкований набір чисел. Для представлення векторів у програмі достатньо зручно застосовувати масиви. Необхідно реалізувати функцію `int comp_vect(int *v1, int *v2, int n);` Дана функція має повернути 1, якщо вектори `v1` та `v2` дорівнюють та 0, якщо не дорівнюють один одному. Продемонструвати застосування функції в основній програмі.

3. Представити комплексне число як одновимірний масив типу `float`, що складається з двох комірок (дійсна та уявна частини). Реалізувати функцію додавання двох комплексних чисел `void add_complex(float *c1, float *c2, float *c3);`, а також функцію віднімання двох комплексних чисел `sub_complex(float *c1, float *c2, float *c3);` Обидві функції мають виконувати відповідну операцію над числами `c1` та `c2`, а результат поміщати в `c3`. Продемонструвати застосування функцій в основній програмі.

4. Представити комплексне число як одновимірний масив типу `float`, що складається з двох комірок (дійсна та уявна частини). Реалізувати функцію множення двох комплексних чисел `void mul_complex(float *c1, float *c2, float *c3);` Функція має виконувати операцію множення над комплексними числами `c1` та `c2`, а результат поміщати в `c3`. Реалізувати функцію `void print_complex(float * c);` для виведення комплексного числа. Приклад виведення: $5+6i$, де 5 – дійсна частина, а 6 – уявна. Продемонструвати застосування функцій в основній програмі.

5. Розробити функцію `int is_ident (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – одинична матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

6. Необхідно реалізувати функцію `int comp_square_matrix(int *m1, int *m2, int n);` Дана функція має повернути 1, якщо квадратні матриці розміром $n \times n$ `m1` та `m2` дорівнюють одна одній та 0, якщо не дорівнюють. Реалізувати функцію `void print_matrix(int *m, int n);` для виведення матриці. Продемонструвати застосування функцій в основній програмі.

7. Реалізувати функцію множення матриць розміром $n \times n$ `void mul_square_matrix(int *m1, int *m2, int *m3, int n);` Функція має виконувати операцію множення над матрицями `m1` та `m2`, а результат поміщати в `m3`. Продемонструвати застосування функції в основній програмі.

8. Реалізувати функцію множення матриці розміром $n \times m$ на матрицю розміром $m \times l$ `void mul_matrix(int *m1, int *m2, int *m3, int n, int m, int l);` Функція має виконувати операцію множення над матрицями `m1` та `m2`, а результат поміщати в `m3`. Продемонструвати застосування функції в основній програмі.

9. Необхідно реалізувати функцію додавання матриць `void add_square_matrix(int *m1, int *m2, int *m3, int n);`, а також функцію віднімання `void sub_square_matrix(int *m1, int *m2, int *m3, int n);` Обидві функції мають виконувати відповідну операцію над квадратними матрицями `m1` та `m2` розміром $n \times n$, а результат поміщати в матрицю `m3`. Продемонструвати застосування функцій в основній програмі.

10. Нижній трикутник матриці розміром 5×5 показаний нижче:

```
0 0 0 0 0
0 0 0 0 0
0 0 x 0 0
0 x x x 0
x x x x x
```

Реалізувати функцію `void clear_lower_triangle(int * m, int n);`, яка заповнює нижній трикутник матриці розміром $n \times n$ нулями. Зверніть увагу, що якщо число n парне, у трикутника буде не одне значення на його вершині. Продемонструвати застосування даної функції в основній програмі.

11. Верхній трикутник матриці розміром 5x5 показаний нижче:

```

  x x x x x
0  x x x 0
0 0  x 0 0
0 0 0 0 0
0 0 0 0 0
```

Реалізувати функцію `void rand_upper_triangle(int * m, int n);`, яка заповнює верхній трикутник матриці розміром $n \times n$ випадковими значеннями. Зверніть увагу, що якщо число n парне, у трикутника буде не одне значення на його вершині. Продемонструвати застосування даної функції в основній програмі.

12. Реалізувати функцію `int check_triangles(int * m, int n);`, яка порівнює верхній трикутник матриці розміром $n \times n$ з її нижнім трикутником та повертає 1 у випадку, якщо вони дорівнюють один одному та 0, якщо не дорівнюють.

Приклади:

	1 2 3 4 5	
	6 1 2 3 7	
Для матриці	0 0 1 9 0	функція має повернути 1
	0 1 2 3 0	
	1 2 3 4 5	

	1 2 3 4 5	
	6 1 2 3 7	
Для матриці	0 0 1 9 0	функція має повернути 0
	0 5 6 7 0	
	1 4 8 7 5	

Продемонструвати застосування даної функції в основній програмі.

13. Реалізувати функцію `void vec_to_mat (int *v, int *mat, int n);` v – вектор, що складається з n^2 елементів, упорядкованих по стовпцях, m – матриця розміром $n \times n$. Функція має скопіювати елементи з v в m . Наприклад, якщо v дорівнює (1, 2, 3, 4, 5, 6, 7, 8, 9), то значення m має бути таким:

```

1 4 7
2 5 8
3 6 9
```

Продемонструвати застосування даної функції в основній програмі.

14. Реалізувати функцію `void reverse_vector (int *v, int n);`, яка розміщує елементи вектору у зворотному порядку. Наприклад, якщо значення вектору були (1, 3, 5, 7), то після оброблення даною функцією вони мають стати такими: (7, 5, 3, 1). Продемонструвати застосування даної функції в основній програмі.

15. Реалізувати функцію `void matrix_transpose(int *mat, int n);`, яка транспонує (мінє рядки та стовпці місцями) матрицю `mat` розміром $n \times n$. Продемонструвати застосування даної функції в основній програмі.

16. Реалізувати функцію `void matrix_spiral(int * mat, int n, int k);`, яка заповнює матрицю `mat` розміром $n \times n$ значеннями по спіралі за годинниковою стрілкою з центру починаючи зі значення `k`, кожне наступне значення має бути на 1 більше, ніж попереднє.

Приклад для $n=3, k=5$:

```
11 12 13
10  5  6
 9  8  7
```

Продемонструвати застосування даної функції в основній програмі.

17. Реалізувати функцію `void matrix_shift_right(int *mat, int n, int k);`, яка циклічно зсуває елементи рядків матриці `mat` розміром $n \times n$ вправо на `k` позицій.

Приклад початкової матриці розміром 4×4 :

```
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
```

Приклад матриці після оброблення функцією з $k=2$:

```
3  4  1  2
7  8  5  6
11 12  9 10
15 16 13 14
```

Продемонструвати застосування даної функції в основній програмі.

18. Реалізувати функцію `void matrix_shift_left(int *mat, int n, int k);`, яка циклічно зсуває елементи рядків матриці `mat` розміром $n \times n$ вліво на `k` позицій.

Приклад початкової матриці розміром 4×4 :

```
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
```

Приклад матриці після оброблення функцією з $k=1$:

```
2  3  4  1
6  7  8  5
10 11 12  9
14 15 16 13
```

Продемонструвати застосування даної функції в основній програмі.

19. Реалізувати функцію `void matrix_shift_bottom(int *mat, int n, int k);`, яка циклічно зсуває елементи стовпців матриці `mat` розміром $n \times n$ вниз на k позицій.

Приклад початкової матриці розміром 4×4 :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Приклад матриці після оброблення функцією з $k=2$:

9	10	11	12
13	14	15	16
1	2	3	4
5	6	7	8

Продемонструвати застосування даної функції в основній програмі.

20. Реалізувати функцію `void matrix_shift_top(int *mat, int n, int k);`, яка циклічно зсуває елементи стовпців матриці `mat` розміром $n \times n$ вверх на k позицій.

Приклад початкової матриці розміром 4×4 :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Приклад матриці після оброблення функцією з $k=1$:

5	6	7	8
9	10	11	12
13	14	15	16
1	2	3	4

Продемонструвати застосування даної функції в основній програмі.

21. Реалізувати функцію `void mat_to_vec(int *m, int *vect, int n);` m – матриця розміром $n \times n$, v – вектор, що складається з n^2 елементів, упорядкованих по стовпцях. Функція має скопіювати елементи з m в v . Наприклад, для нижченаведеної `mat`:

1	2	3
4	5	6
7	8	9

значення `v` має бути таким: (1, 4, 7, 2, 5, 8, 3, 6, 9)

Продемонструвати застосування даної функції в основній програмі.

22. Реалізувати функцію `void matrix_spiral(int * mat, int n, int k);`, яка заповнює матриця `mat` розміром $n \times n$ значеннями по спіралі за годинниковою стрілкою з лівого верхнього кута починаючи зі значення `k`, кожне наступне значення має бути на 1 більше, ніж попереднє.

Приклад для $n=3, k=2$:

2	3	4
9	10	5
8	7	6

23. Розробити функцію `int is_zero (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – нульова матриця, та 0 в іншому випадку. Реалізувати функцію `void print_matrix(int *mat, int n);`, призначену для виведення матриці. Продемонструвати застосування даних функцій в основній програмі.

24. Розробити функцію `int is_diagonal (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – діагональна матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

25. Розробити функцію `int is_antidiagonal (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – антидіагональна матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

26. Розробити функцію `int is_persymmetric (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – персиметрична матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

27. Розробити функцію `int is_bisymmetric (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – бісиметрична матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

28. Розробити функцію `int is_tridiagonal (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – тридіагональна матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

29. Розробити функцію `int is_matrix_of_ones (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – матриця одиниць, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.

30. Розробити функцію `int is_unimodular (int *mat, int n);` `mat` – це квадратна матриця розміром $n \times n$. Функція має повернути 1, якщо `mat` – унімодулярна матриця, та 0 в іншому випадку. Продемонструвати застосування даної функції в основній програмі.