Міністерство освіти і науки України

Комунальний заклад освіти

«Українсько-Американський ліцей»

Дніпропетровської міської ради

**Науково-дослідницька робота з інформатики**

**на тему: «Розв’язування задач з масивами на C++»**

Виконала: учениця 10-2 класу

Забрудська Поліна Юріївна

Науковий керівник:

Бабій Микола Володимирович

Дніпро 2022

**ЗМІСТ**

[Вступ 3](#_Toc93862062)

[1. Що таке масив? 5](#_Toc93862063)

[2. Доступ до елементів одновимірного масиву 6](#_Toc93862064)

[3. Особливості обробки масивів компілятором С++. Межі масиву 6](#_Toc93862065)

[4. Ініціалізація символьних масивів 7](#_Toc93862066)

[5. Присвоєння одного масиву іншому 8](#_Toc93862067)

[6. Опис масиву структур. 9](#_Toc93862068)

[Практична частина 12](#_Toc93862069)

[*Задача 1* 12](#_Toc93862070)

[*Задача 2* 19](#_Toc93862071)

[*Задача 3* 21](#_Toc93862072)

[Висновок 23](#_Toc93862073)

[Використані джерела 25](#_Toc93862074)

# Вступ

Ознакою часу, в якому ми живемо, є лавинне нагромадження інформації і бурхливий розвиток мікроелектронної техніки. Наша цивілізація нестримно прямує до комп’ютерної ери. Відбувається перехід до інформаційних технологій, тобто до широкого застосування комп’ютерів і програмного забезпечення у виробництві, управлінні, науці, освіті, медицині, торгівлі, банківській справі тощо. Якщо багато років тому з винайденням парової машини, появою фабрик і заводів відбулася індустріальна революція і було закладено підмурівок індустріального суспільства, то сьогодні завдяки комп’ютеру людство створює суспільство інформаційне.

В інформаційному суспільстві щораз більше людей працює в галузях опрацювання інформації, а не у сфері матеріального виробництва. Сьогодні комп’ютери допомагають нам у різних сферах діяльності. Будь-яка діяльність комп’ютера заснована на використанні певних алгоритмів.

Комп’ютер - це бездушна машина, яка не може робити щось без команди. Сукупність команд і є алгоритмом. Задача програміста у тому, щоб сформувати із алгоритмів, які він розробляє сам або використовує чужі, сукупність інструкцій, тобто програму для комп’ютера. Таке пов’язання команд із логікою і називається програмуванням.

У цій роботі розглянуто задачі, пов’язані з використанням масивів.

Предмет дослідження: Задачі з масивами.

Об’єкт дослідження: мова програмування C++.

Мета роботи: вивчити основні можливості мови C++ та здобути необхідні навички для роботи в цьому середовищі програмування; описати теоретичні аспекти роботи з масивами та описати в практичній частині коди програм задач.

**Теоретична частина**

C++ (Сі-плюс-плюс) — мова програмування загального призначення з підтримкою кілька парадигм програмування: об'єктно-орієнтованої, узагальненої, процедурної та ін.  Б'ярн Страуструп (англ. *Bjarne Stroustrup*) почав створювати C++ в AT&T Bell Laboratories ([Мюррей-Хілл](https://uk.wikipedia.org/w/index.php?title=%D0%9C%D1%8E%D1%80%D1%80%D0%B5%D0%B9-%D0%A5%D1%96%D0%BB%D0%BB&action=edit&redlink=1), Нью-Джерсі) у 1979 року. На етапі зародження мова мала назву «Сі з класами». Згодом Страуструп перейменував мову на *C++* у 1984р. Бере витоки з мови програмування С. Вперше описана міжнародним стандартом ISO/IEC 14882:1998 (C++98), найбільш актуальним же є стандарт ISO/IEC 14882:2020 (C++20).

У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки прикладного програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. C++ суттєво вплинула на інші популярні сьогодні мови програмування: C# та Java.

Назва «Сі++» була вигадана Ріком Масситті (Rick Mascitti) і вперше було використана в грудні 1983 року. Раніше, на етапі розробки, нова мова називалася «Сі з класами». Ім'я, що вийшло у результаті, походить від оператора Сі «++» (збільшення значення змінної на одиницю) і поширеному способу присвоєння нових імен комп'ютерним програмам, що полягає в додаванні до імені символу «+» для позначення поліпшень. Згідно зі Страуструпом, «ця назва указує на еволюційну природу змін Ci». Виразом «С+» називали ранішню, не пов'язану з Сі++, мову програмування.

Деякі програмісти на Сі можуть відмітити, що якщо виконуються вирази x=3; y=x++; то в результаті вийде x=4 і y=3, тому що x збільшується тільки після присвоєння його у. Проте якщо другий вираз буде y=++x; то вийде x=4 і y=4. Виходячи з цього, можна зробити висновок, що логічніше було б назвати мову не Сі++, а ++Сі. Проте обидва вирази c++ і ++c збільшують с, а крім того вираз c++ поширеніший.

Педанти також можуть відмітити, що введення мови Сі++ не змінює самого Сі, тому найточнішим ім'ям було б «С+1».

# Що таке масив?

Масив – набір змінних однакового типу. Доступ до цих змінних здійснюється одним ім'ям. Це ім'я називається ім'ям масиву.

Масиви використовуються для групування пов'язаних змінних між собою.

Масиви можуть бути одновимірними та багатовимірними. В одновимірних масивах для доступу до елементу масиву використовується один індекс. У багатовимірних масивах для доступу до елементу масиву використовується кілька індексів.

Загальна форма опису одновимірного масиву:

**тип ім’я\_масива[**розмір**];**

У наведеному вище описі:

* + тип – це тип елементів масиву. Він ще називається базовим типом. Базовий тип визначає кількість даних кожного елемента, що становить масив. Тип елементів масиву може бути як базовим типом, так і складовим (наприклад, структура).
  + розмір – кількість елементів у масиві.
  + ім’я\_масива – безпосередньо ім'я масиву, яким здійснюється доступ до елементів масиву.

Після опису масиву значення елементів може бути нульовим або невизначеним.

# Доступ до елементів одновимірного масиву

Доступ до окремого елемента масиву здійснюється за допомогою індексу. Індекс визначає позицію елемента у масиві. Перший елемент масиву має нульовий індекс. Щоб отримати доступ до окремого елемента масиву за його індексом, потрібно після імені масиву квадратних дужках вказати номер цього елемента.

# Особливості обробки масивів компілятором С++. Межі масиву

В C++ не ведеться строгого контролю за доступом до елементів за межами масиву. Якщо описати масив зі 100 елементів, то можна прочитати або змінити 101, 102 і наступні елементи. На цих позиціях можуть бути комірки пам'яті, виділені для інших змінних або навіть для вашої програми. Це може призвести до знищення програми за відсутності будь-яких зауважень компілятора C++.

Вся відповідальність за дотримання меж масивів належить суворо програмісту. Програміст повинен писати такий код, що гарантує коректну роботу з масивами. Це реалізується за допомогою включення до програми спеціальних перевірок.

В C++ підтримується два види ініціалізації масивів:

* ініціалізація із завданням розміру масиву;
* "безрозмірна" ініціалізація.

Загальний вид ініціалізації із завданням розміру масиву:

тип ім'я\_масиву[розмір] = {список\_значень};

де:

* тип – тип елементів масиву;
* розмір – кількість елементів масиву вказаного типу;
* список значень – список значень ініціалізації елементів масиву. Елементи масиву поділяються символом ',' (кома).

Загальний вигляд «безрозмірної» ініціалізації:

тип ім'я\_масиву[] = {список\_значень};

У цьому випадку розмір масиву визначається кількістю елементів, які описані в списку значень.

Приклад 1. Масив B ініціалізований із завданням розміру.

// ініціалізація масиву B

int B[10] = { 5, 6, 9, -8, 3, 2, 4, -90, -103, 0};

Приклад 2. Масив C ініціалізований на основі списку значень (безрозмірна ініціалізація).

// ініціалізація масиву C без завдання розміру

float C[] = {-3.9, 2.8, -1.6, 2.2};

# Ініціалізація символьних масивів

Для символьних масивів можна використати скорочений варіант ініціалізації:

char ім'я\_масиву [розмір] = "рядок";

У цьому випадку кожному елементу масиву надається один символ рядка. Значення розмір визначає розмір пам'яті, який виділяється для масиву. Значення розміру має бути не менше довжини рядка, інакше компілятор видасть повідомлення про помилку.

Допускається також варіант ініціалізації масиву без зазначення розміру

char имя\_масива[] = "рядок";

У цьому випадку розмір масиву дорівнює довжині рядка.

***Приклад:***

Ініціалізація символьного масиву під назвою str.

// символьний масив – скорочений варіант ініціалізації

char str[] = "Hello!";

// Інший варіант ініціалізації символьного масиву

char str2[] ​​= { 'H', 'e', ​​'l', 'l', 'o', '!', '\0'};

У наведеному вище прикладі масиви str і str2 містять однакові значення.

# Присвоєння одного масиву іншому

***Приклад:***

У мові програмування C++ (на відміну інших мов) не можна привласнювати безпосередньо один масив іншому. Надавати можна тільки поелементно з використанням оператора циклу. При цьому обидва масиви повинні мати однаковий тип елементів.

Нехай задані два масиви цілих чисел. Фрагмент коду, який надає один масив іншому:

// опис масивів A та B

int A[10], B[10];

int i;

//A = B; // помилка!

for (i=0; i<10; i++)

B[i] = 0;

for (i=0; i<10; i++)

A[i] = B[i];

# Опис масиву структур.

***Приклад***

Нехай дано опис структури, що містить інформацію про книгу:

struct BOOK

{

char title[50];

char author[50];

int year;

float price;

};

Цей опис визначає новий тип struct BOOK. За такого опису пам'ять не виділяється. Це лише інформація про новий тип даних. Структура має бути описана поза визначення будь-якого класу на початку визначення простору імен.

У програмі масив із 5 книг (структур) можна описати так:

struct BOOK Books[5];

***Доступ до елементів структури у програмі:***

strcpy(Books[0].title, "Title-1");

strcpy(Books[0].author, "Author-1");

Books[0].year = 1970;

Books[0].price = 28.85;

У наведеному вище коді використана функція роботи з рядками

strcpy(str1, str2);

Ця функція копіює рядок str2 у рядок str1. Щоб використовувати цю та інші функції роботи з рядками на початку програми (перед описом простору імен), потрібно вказати рядок

#include <cstring>

або

#include <string.h>

На практиці часто виникає необхідність в обробці даних у вигляді довільного набору значень, тобто масивів. Масив являє собою кінцеву іменовану послідовність величин од­ного типу, які розрізняються за порядковим номером. Опис ма­сивiв у програмі відрізняється від опису простої змінної наявніс­тю після імені квадратних дужок «[ ]», в яких задається кількість елементів масиву (розмірність). Слід нагадати, що *у мові C++ нумерація елементів масиву починається з 0.*

Розглянемо одновимірні масиви, оголошення яких допус­кае одну з таких форм запису:

<тип> <ім’я> [n];

<тип> <ім’я> [n] = {значення};

<тип> <ім’я> [  ] = {значення};

При оголошенні одновимірного масиву, коли масив відразу iніціюється, можна не вказувати його розмір. Якщо ж ініціюван­ня не здійснюється під час оголошення масиву, то кількість ін­дексів слід задати обов’язково константним виразом. Наприклад:

float m [6];  
float m [6] = {3.4, 4.5, 5.6, 6.7, 8.9, 10.3};  
float m [  ] = {3.45, 4.56, 5.67, 6.78);

Зрозуміло, що надалі кількість елементів змінити немож­ливо. Для того щоб обнулити елементи оголошеного масиву, достатньо ініціювати його перший елемент: int mas[0]={0};

# Практична частина

# *Задача 1*

Умова

Програма має меню з п’яти варіантів. Якщо ввести число один, програма додає елемент в масив за заданим індексом. Якщо ввести число два, програма додає елемент в кінець масиву. Якщо ввести число три, програма видаляє з масиву елемент за заданим індексом. Якщо ввести чотири, програма відсортує всі елементи масиву від найменшого до найбільшого. Якщо ввести число п’ять, програма відсортує всі елементи масиву від найбільшого до найменшого значення.

Вхідні данні

Натуральне число з проміжку від 1 до 5 та масив.

Вихідні дані

Результати різних операцій з масивом.

Код

#define \_CRT\_SECURE\_NO\_WARNINGS

# include <iostream>

#include <iomanip>//setw()

#include <windows.h>//setcolor

#include <conio.h>//getch()

#include <ctime>//rand()time

#include<string>

using namespace std;

template <class A>

void add\_arr(A\*& ar, int& n, A val)

{

A\* temp = new A[++n];

for (int i = 0; i < n; i++)

temp[i] = i < n - 1 ? ar[i] : val;

delete[]ar;

ar = temp;

}

template <class S>

void add\_arr2(S\*& ar, int& n, S val, int index)

{

S\* temp = new S[++n];

for (int i = 0; i < n; i++)

{

if (i < index)

{

temp[i] = ar[i];

}

if (i == index)

{

temp[i] = val;

}

else if (i > index)

{

temp[i] = ar[i - 1];

}

}

delete[]ar;

ar = temp;

}

template <class C>

void del\_arr(C\*& ar, int& n, int index)

{

C\* temp = new C[--n];

for (int i = 0; i <= n; i++)

{

if (i < index)

{

temp[i] = ar[i];

}

if (i > index)

{

temp[i - 1] = ar[i];

}

}

delete[]ar;

ar = temp;

}

void sort\_minmax(int\* mas, int num)

{

for (int i = 0; i < num - 1; i++)

{

int min = i;

for (int j = i + 1; j < num; j++)

if (mas[j] < mas[min])

min = j;

swap(mas[i], mas[min]);

}

for (int i = 0; i < num; i++)

{

cout << mas[i] << ' ';

}

cout << endl;

}

void sort\_maxmin(int\* mas, int num)

{

for (int i = 0; i < num - 1; i++)

{

int max = i;

for (int j = i + 1; j < num; j++)

if (mas[j] > mas[max])

max = j;

swap(mas[i], mas[max]);

}

for (int i = 0; i < num; i++)

{

cout << mas[i] << ' ';

}

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "");

while (1)

{

cout << "Якщо ви бажаєте додати елемент по заданому iндексу, то введiть 1" << endl;

cout << "Якщо ви бажаєте додати елемент в кiнець, то введiть 2" << endl;

cout << "Якщо ви бажаєте видалити елемент, то введiть 3" << endl;

cout << "Якщо ви бажаєте вiдсортувати значення масиву вiд меншого до бiльшого, то введiть 4" << endl;

cout << "Якщо ви бажаєте вiдсортувати значення масиву вiд бiльшого до меншого, то введiть 5" << endl;

int n;

cin >> n;

int num = 6;

int\* mas = new int[num];

cout << "Добре, спочатку потрiбно створити масив i ввести значення" << endl;

cout << "Введiть" << ' ' << num << ' ' << "елементiв за допомогою ентеру" << endl;

for (int i = 0; i < num; i++)

{

int zn;

cin >> zn;

mas[i] = zn;

}

cout << "Тепер виведемо отриманий масив" << endl;

cout << "Масив: " << endl;

for (int i = 0; i < num; i++)

{

cout << mas[i] << ' ';

}

cout << endl;

switch (n)

{

case 1:

cout << "Ви бажаєте додати елемент за заданим iндексом" << endl;

int index;

int num\_in;

cout << "Введiть iндекс" << endl;

cin >> index;

cout << "Введiть елемент" << endl;

cin >> num\_in;

add\_arr2(mas, num, num\_in, index);

cout << "Тепер виведемо масив з доданим елементом" << endl;

for (int i = 0; i < num; i++)

{

cout << mas[i] << ' ';

}

cout << endl;

break;

case 2:

cout << "Ви бажаєте додати елемент в кiнець масива" << endl;

int num\_end;

cout << "Введiть елемент" << endl;

cin >> num\_end;

add\_arr(mas, num, num\_end);

cout << "Тепер виведемо масив з доданим елементом" << endl;

for (int i = 0; i < num; i++)

{

cout << mas[i] << ' ';

}

cout << endl;

break;

case 3:

cout << "Ви бажаєте видалити елемент" << endl;

int ind;

cout << "Введiть iндекс, який бажаєте видалити" << endl;

cin >> ind;

cout << "" << endl;

del\_arr(mas, num, ind);

for (int i = 0; i < num; i++)

{

cout << mas[i] << ' ';

}

cout << endl;

break;

case 4:

cout << "Ви бажаєте вiдсортувати значення масиву вiд меншого до бiльшого" << endl;

cout << "Виведемо вiдсортований масив" << endl;

sort\_minmax(mas, num);

break;

case 5:

cout << "Ви бажаєте вiдсортувати значення масиву вiд бiльшого до меншого" << endl;

cout << "Виведемо вiдсортований масив" << endl;

sort\_maxmin(mas, num);

break;

}

cout << endl;

}

system("pause");

}

# *Задача 2*

Умова

Знайти значення максимального елементу в масиві.

Вхідні дані

Масив з заданими елементами matrix[].

Вихідні дані

Значення максимального елементу в масиві max.

Код

#define \_CRT\_SECURE\_NO\_WARNINGS

# include <iostream>

#include <iomanip>//setw()

#include <windows.h>//setcolor

#include <conio.h>//getch()

#include <ctime>//rand()time

#include<string>

using namespace std;

int main()

{

int matrix[5][5];

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

matrix[i][j] = rand();

}

}

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

cout<< matrix[i][j]<< " ";

}

cout << endl;

}

int max = matrix[0][0];

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

if (j == i)

{

if (max < matrix[i][j])

{

max = matrix[i][j];

}

else

{

max = max;

}

}

}

}

cout << max;

}

# *Задача 3*

Умова

Замінити парні рядки у масиві на непарні.

Вхідні дані

Масив із заданими елементами matrix[].

Вихідні дані

Масив, у якому парні рядки були замінені на непарні matrix[].

Код

#define \_CRT\_SECURE\_NO\_WARNINGS

# include <iostream>

#include <iomanip>//setw()

#include <windows.h>//setcolor

#include <conio.h>//getch()

#include <ctime>//rand()time

#include<string>

using namespace std;

int main()

{

int matrix[4][4];

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

matrix[i][j] = rand();

}

}

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

cout<< matrix[i][j]<< " ";

}

cout << endl;

}

for (int i = 0; i < 4; i+=2)

{

for (int j = 0; j < 4; j++)

{

swap(matrix[i][j], matrix[i + 1][j]);

}

}

cout << endl;

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

# Висновок

Навчально-дослідницька робота дала мені можливість на належному рівні

засвоїти роботу середовища C++. Особлива увага була приділена працюванню з масивами.

За час виконання роботи мною були використані наступні методи дослідження: аналіз літератури та джерел із Інтернету, практичні вправи. Також я приймала участь в олімпіаді з інформатики, де теж використовувала свої знання у С++. Тобто, я доклала зусилля, щоб покращити свої навички.  Запропонована тема є актуальною, тому що у мови С++ є багато плюсів(швидка мова, універсальна, вона активно підтримується, одна з найпопулярніших та найвикористовуваніших мов програмування, а також корисна в якості фундаменту для вивчання інших мов) і вона може застаріти і стати неактуальною лише тоді, якщо відбудеться якась неймовірна технологічна революція.

Тож вважаю, що засвоєні мною знання, вміння, навички можуть бути застосовані при вивченні інших споріднених програмних продуктів

Надалі буду намагатися удосконалювати свої вміння і навички при роботі з програмою C++. В моїй роботі було поставлено питання: вивчити основи програмування мови С ++ і реалізувати завдання даною мовою. Після закінчення роботи можна зробити наступні висновки: в даний час мова програмування С ++ є одною з основних в практичному програмуванні. Переваги мови С ++ це перш за все простота і лаконічність. Мова С ++ легко вчиться. Структура даних в С ++ дозволяє обробляти змінні такі як цілочисельні , рядкові , з плаваючою комою і масиви. За допомогою даної мови можна вирішувати широкий спектр завдань, такі як написання прикладних програм, призначених для користувача, додатків, розробку ігр та програм вузького призначення. Програмування на С ++ зараз стає все більш популярним. На С ++ пишуться ігри, додатки, прикладні програми, інтерпретатори і CAD системи. А так же виконується широкий спектр завдань обчислювальної математики, методів оптимізації і теорії ланцюгів і мереж.

# Використані джерела

1. Книга четырех авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров, В. В. Васильев. СПб.: Издательство, 2010. 000 с.
2. Иванов И. И. Книга одного-трех авторов. М.: Издательство, 2010. 000 с.
3. Книга пяти и более авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров и др.. СПб.: Издательство, 2010. 000 с.
4. Описание книги под редакцией / под ред. И.И. Иванова СПб., Издательство, 2010. 000 с.
5. Описание методических указаний / сост.: И.И. Иванов, П.П. Петров. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.
6. Иванов И.И. Описание учебного пособия и текста лекций: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.
7. Описание статьи с четырьмя и более авторами из журнала / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название журнала. 2010, вып. (№) 00. С. 000–000.
8. Иванов И.И. Описание статьи с одним-тремя авторами из журнала // Название журнала. 2010, вып. (№) 00. С. 000–000.
9. Описание тезисов доклада с четырьмя и более авторами / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название конференции: тез. докл. III международной науч.-техн. конф., СПб,  00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПБ, 2010, С. 000–000.
10. Иванов И.И. Описание тезисов доклада с одним-тремя авторами / Название конференции: тез. докл. III международной науч.-техн. конф., СПб,  00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПБ, 2010, С. 000–000.
11. Описание электронного ресурса // Наименование сайта. URL: http://east-front.narod.ru/memo/latchford.htm (дата обращения: 00.00.2010).
12. ГОСТ 0.0–00. Описание стандартов. М.: Изд-во стандартов, 2010.
13. Пат. RU 00000000. Описание патентных документов / И. И. Иванов, П. П. Петров, С. С. Сидоров. Опубл. 00.00.2010. Бюл. № 00.
14. Иванов И.И. Описание авторефератов диссертаций: автореф. дисс. канд. техн. наук / СПбГЭТУ «ЛЭТИ», СПБ, 2010.
15. Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 383 с.
16. Гришин, В.Н. Информационные технологии в профессиональной деятельности: Учебник / В.Н. Гришин, Е.Е. Панфилова. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 416 c.
17. Демин, А. Ю. Информатика. Лабораторный практикум : учебное пособие для среднего профессионального образования / А. Ю. Демин, В. А. Дорофеев. — Москва : Издательство Юрайт, 2019. — 133 с.
18. Зимин, В. П. Информатика. Лабораторный практикум в 2 ч. Часть 1 : учебное пособие для среднего профессионального образования / В. П. Зимин. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 126 с.
19. Кедрова Г. Е.Информатика для гуманитариев : учебник и практикум для среднего профессионального образования / Г. Е. Кедрова [и др.] ; под редакцией Г. Е. Кедровой. — Москва : Издательство Юрайт, 2019. — 439 с.