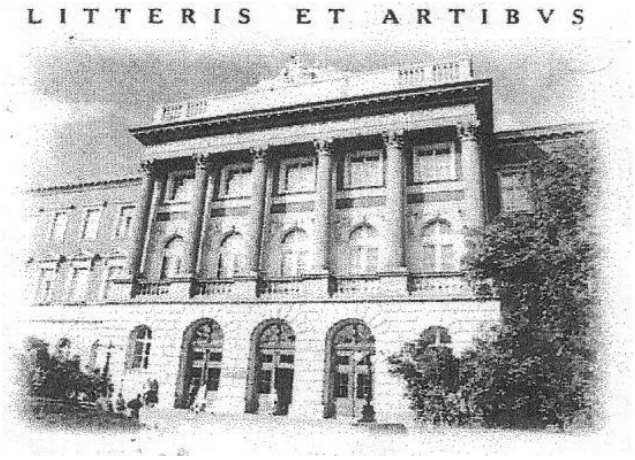
Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Використання вказівників для обробки масивів в С. Пошук в одновимірному масиві. Робота з двовимірними масивами. Двовимірні масиви та вказівники.

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт №№ 7, 8
з дисципліни «Основи інформатики та програмування»
для студентів спеціальності 105 – "Прикладна фізика та наноматеріали"
для першого (бакалаврського) рівня освіти.

Затверджено
на засіданні кафедри
обчислювальної математики
та програмування
Протокол № 10 від 07.06.2018р.

Львів – 2018

Використання вказівників для обробки масивів в С. Пошук в одновимірному масиві. Робота з двовимірними масивами. Двовимірні масиви та вказівники. Методичні вказівки до виконання лабораторної роботи № 3 для студентів спеціальності 105 – "Прикладна фізика та наноматеріали" для першого (бакалаврського) рівня освіти / Укл.: Гнатів Л.Б., Ментинський С.М., Пелех Я.М., Угрин С.З., 2018. - 16 с.

Реєстр. № 8185 від 19.06.2018

Укладачі: Гнатів Л.Б., к. ф-м. н., доцент,
Ментинський С.М., ст. викл,
Пелех Я.М., к. ф-м. н., доцент,
Угрин С.З., асистент.

Відповідальний за випуск: Угрин С.З.

Рецензенти: Будз І.С., к.ф-м.н., доцент кафедри ОМП,
Філь Б.М., к.ф-м.н., доцент кафедри ОМП.

Передмова

У методичних вказівках розглянуто способи організації індексованих сукупностей однотипних даних в мові C/C++ за допомогою одно- та двовимірних масивів. Приділено також увагу механізмові покажчиків та його застосуванню до зберігання та доступу до елементів масиву.

Методичні вказівки містять коротку довідкову інформацію, варіанти індивідуальних завдання до виконання роботи та приклад виконання одного із варіантів з коментарями.

Методичні вказівки призначені для студентів спеціальності 105 – "Прикладна фізика та наноматеріали" для першого (бакалаврського) рівня освіти і укладені відповідно до робочої навчальної програми з дисципліни «Основи інформатики та програмування».

Лабораторна робота № 7

Використання вказівників для обробки масивів в С. Пошук в одновимірному масиві.

Мета роботи: вивчення основних прийомів роботи з одновимірними масивами.

Короткі теоретичні відомості.

Масив - це впорядкований набір однотипних елементів для звертання до яких використовують спільне ім'я. В С можна створювати масиви будь-якого типу, вони можуть бути як одновимірні так і багатовимірні. До окремого елемента масиву звертаються за його індексом, індекс записують в квадратних дужках.

Загальна форма оголошення одновимірного масиву:

```
<тип елементів> <назва масиву>[<розмір масиву>];
```

наприклад:

```
double x[10];
int a[7];
```

Для роботи з елементами масиву,використовують цикл for, на кшталт:

```
for(int i=0;i<20;i++)
      cout<<p[i]<<'\n';
```

Ввід елементів масиву можна здійснити за допомогою циклу, але для зручності дозволяється ініціалізувати масив елементами відразу при його описі, наприклад:

```
int p[] = {1, 3, 4, 5, 2, 6, 5, 9};
```

В такому випадку розмір масиву можна не вказувати, - компілятор автоматично визначає його за кількістю елементів, записаних в фігурних дужках.

Поняття вказівника, оголошення та використання вказівників у програмі

Вказівники (покажчики) - це змінні спеціального типу, значеннями яких є адреси різних об'єктів програми. Якщо ми використовуємо ім'я того чи іншого об'єкта для вилучення його значення або для зміни його значення, то прийнято говорити про ***безпосередній*** (прямий) доступ до об'єкта. У тому випадку, коли адреса об'єкта поміщений в вказівник, то мова йде про ***непрямий доступі до об'єкта, на який "дивиться" вказівник.***

Для оголошення одного вказівника з ім'ям p1 або кількох вказівників з іменами p1, p2, ..., які повинні будуть "дивитися" на об'єкти типу type1, використовується одна з наступних синтаксичних конструкцій:

```
type1 *p1;
type1 *p1,*p2,...;
```

Оголошені вказівники ще ***нікуди*** конкретно ***не дивляться***, одна з типових помилок програмістів-початківців - спроба записати що-небудь за вказівниками, яким ще не присвоєно значення. Ініціалізацію вказівника можна поєднати з його оголошенням:

```
int x=2,y;
int *p1=&x;  //ініціалізація адресою змінної x
int *p2(&x); //ініціалізація адресою змінної x
int *p3=p1;  //ініціалізація значенням іншого вказівника
```

Вказівник є змінною і його значення можна задати або змінити за допомогою оператора присвоєння:

```
p1 = &y; / / тепер значенням p1 є адреса змінної y
```

У програмах на мові С можна зустріти нагромадження символів "*". Лякатися не треба - це просто багатоступенева адресація:

```
int x, y;
int * p1 = &y;
int ** p2 = &p1;
x = ** p2; / / те ж, що x = * (* p2) = * (p1) = y
```

Зв’язок масивів і вказівників

У простому випадку для так званих одновимірних масивів (векторів) індекс масиву і його порядковий номер збігаються. Так як в мовах С, С + + прийнято відраховувати індекси від 0, то позначення xy [2] відповідає третьому елементу масиву з ім'ям xy. Якщо елементи цього масиву мають тип double і початковий елемент xy [0] розташований в оперативній пам'яті з адресою 0x02000000, то для обчислення адреси елемента xy [2] достатньо виконати кілька операцій - 0x02000000 +2 * 8. Природно, що такого роду операції перекладаються на компілятор, а програміст може записувати алгоритм обробки елементів масиву, маніпулюючи з індексами його елементів.

Позначення елементів масиву, більш звичні для програміста, компілятор перетворює в вирази з вказівниками за правилами приведення індексу:

```
a[6]     еквівалентно  *(a+6)
```

Ці перетворення засновані на наступних угодах мови С. Ім'я одновимірного масиву a одночасно є вказівником на його перший елемент, тобто значенням, доступним за адресою * a, є елемент масиву a [0]. Ім'я двовимірного масиву b одночасно є вказівником на вказівник його першого рядка, тобто значенням, доступним за адресою ** b, є елемент масиву b [0] [0].

Вказівник b +1 "дивиться" на вказівник, що визначає адреса першого елемента другого рядка масиву b. Сенс подібного роду перетворень полягає в підвищенні ефективності програми, тому операції з вказівниками виконуються набагато швидше. Іноді до такого роду перетворенням вдаються і програмісти, наприклад, зводячи обробку двовимірного масиву до одновимірним наведеним індексам.

```
int a[10];
  int *p1=a;            //p1 вказує на початок масиву a
  int *p2=&a[0];        //p2 також вказує на початок масиву a
  int *p3=(int *)&a;    //p3 також вказує на початок масиву a
```

Коли вказівник p2 "дивиться" на початок масиву q, то доступ до елементів цього масиву можна організувати одним із таких способів:

```
int q[20];
  int *p2=q;
...........
  y=*(p2+5);    //тепер y=q[5]
  x=p2[3];      //тепер x=q[3]
  *(p2+1)=7;    //тепер q[1]=7
```

Важливим аспектом використання покажчиків при роботі з масивами є доступ до так званої ***динамічної пам'яті***. Кількість елементів масиву оголошеного у звичний спосіб повинна бути константною величиною і не змінюватися у під час виконання програми. Оскільки пам'ять для зберігання змінних виділяється до початку виконання функції, то їх розмір повинен бути фіксованим і наперед відомим. Проте в С та С++ є засоби для виділяння пам'яті під зберігання даних: в С – це функція ***malloc***, що виділяє вказану кількість байт та повертає безтипний покажчик (void*) на неї, в С++ – оператор ***new***, який виділяє кількість пам'яті потрібну для зберігання даних вказаного типу і повертає вказівник цього ж типу.

Згаданою можливістю можна скористатися для створення масиву, кількість елементів якого користувач задає у ході виконання програми. Наприклад:

```
Int *mas, size;
cout<<"Enter size:\n";
cin>>size;
mas = new int [size];
for(int i=0;i<size;i++){
      mas[i] = i * i * i;
      cout<<mas[i]<<"\t";
}
delete [] mas;// звільнення виділеної пам'яті
```

При роботі з динамічно виділеною пам'яттю слід не забувати про її звільнення по завершенні роботи. Як видно з прикладу в мові С++ для цього використовують оператор ***delete*** (а в мові С є аналогічна функція ***free***)

Завдання. Скласти програму розв'язання задачі відповідно до варіанта. Задано три масиви дійсних чисел A[10], B[10] та C[10], кожен містить по 10 елементів. Масив A заповнити довільно в коді програми при його ініціалізації. Масив B заповнити за вказаним правилом. Масив C утворити з елементів масивів A та B згідно варіанта. Знайти в кожному з масивів вказану величину, вивести на консоль елементи кожного масиву у порядку зростання.

1. Масив B утворити за правилом $B_k = \dfrac{(-1)^k\left(2k^2+1\right)}{k}$ $k = 0,1,\ldots,9$. Масив C утворити з масиву B, замінивши в ньому всі від'ємні елементи відповідними елементами масиву A (з тими ж порядковими номерами). В кожному масиві знайти середнє арифметичне додатних елементів.

2. Масив B заповнити випадковими числами з відрізка [-2; 3] (скористатися методом random() класу Math). Масив C утворити з масиву B, замінивши в ньому всі додатні елементи максимальним елементом масиву A. В кожному масиві знайти кількість елементів, більших за їх середнє арифметичне.

3. Масив B утворити за правилом $B_k = \dfrac{99\sin k}{(k+1)^2}$, $k = 0,1,\ldots,9$. Масив C утворити з масиву B, замінивши в ньому всі від'ємні елементи мінімальним елементом масиву A. В кожному масиві знайти кількість елементів, абсолютна величина яких більша за 5.

4. Масив B утворити за правилом $B_k = (-1)^k\sqrt{k!}$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = 2A_k - 3B_k$. В кожному масиві знайти суму додатних елементів.

5. Масив B утворити за правилом $B_k = 11\sin\left(\sqrt{(k+3)^3}\right)$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = \max\left\{\dfrac{A_k}{B_k};\dfrac{B_k}{A_k}\right\}$. В кожному масиві знайти суму елементів, абсолютна величина яких менша за 1.

6. Масив B заповнити випадковими числами з відрізка [-4; 2] (скористатися методом random() класу Math). Масив C утворити з масиву B, замінивши в ньому всі від'ємні елементи середнім значенням елементів масиву A. В кожному масиві знайти елемент, значення якого найближче до числа 1.

7. Масив B утворити за правилом $B_k = k\sin(2k) + 2k\sin k$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = \max\{A_k; B_k\}$. В кожному масиві знайти різницю максимального та мінімального елемента (розмах значень елементів масиву).

8. Масив B утворити за правилом $B_k = 20\cos k - k$, $k = 0,1,\ldots,9$. Масив C утворити з масиву B, віднявши від кожного його елемента відповідний елемент масиву A. В кожному масиві знайти суму квадратів мінімального та максимального елементів.

9. Масив B утворити за правилом $B_k = (-1)^k \sqrt{(k+2)(k+1)}$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = \min\{A_{9-k}; B_k\}$. В кожному масиві знайти різницю між максимальним елементом та середнім арифметичним усіх елементів.

10. Масив B утворити заповнити випадковими числами з відрізка [-5; 5] (скористатися методом random() класу Math). Масив C утворити з елементів масиву A, додавши до кожного середнє арифметичне від'ємних елементів масиву B. В кожному масиві знайти відношення максимального елемента до мінімального.

11. Масив B утворити за правилом $B_k = 15\cos k - 12\sin(10-k)$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = \min\{3A_k; B_k\}$. В кожному масиві знайти відношення абсолютної величини суми від'ємних елементів до суми додатних елементів.

12. Масив B утворити за правилом $B_k = \dfrac{k\sin(2k)}{(k+1)!}$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = 2A_k + B_k$. В кожному масиві знайти суму від'ємних елементів.

13. Масив B утворити за правилом $B_k = 12\cos\left(\sqrt[k+1]{k^2}\right)$, $k = 0,1,\ldots,9$. Масив C утворити за правилом $C_k = \max\{2A_k; 5B_k\}$. В кожному масиві знайти номер елемента, найближчого за величиною до числа 5.

14. Масив B утворити заповнити випадковими числами з відрізка [-2.5; 1.5] (скористатися методом random() класу Math). Масив C утворити за правилом $C_k = \min\{A_k; B_{9-k}\}$. В кожному масиві знайти найбільший серед від'ємних елементів.

15. Масив B утворити за правилом $B_k = \dfrac{\cos(12k)}{k!}$, $k = 0,1,\ldots,9$. C утворити з масиву B, замінивши в ньому всі додатні елементи мінімальним елементом масиву A. В кожному масиві знайти номер елемента, найближчого за величиною до середнього арифметичного усіх елементів масиву.

Приклад виконання

16. Масив B утворити за правилом $B_k = 2\sin k + \cos k$, $k = 0,1,\ldots,9$. Масив C утворити додаванням оберненого масиву A до масиву B, тобто $C_0 = A_9 + B_0$, $C_1 = A_8 + B_1$, $C_2 = A_7 + B_2$, і т.д. В кожному масиві знайти кількість елементів, значення яких належать інтервалу (-1; 1).

```
#include <iostream>
#include <math.h>
using namespace std;

void main()
{
	//ініціалізація масиву a при оголошенні
	double a[] = {2.1,0.4,0.1,-0.5,0.6,7.1,-2.4,-1.3,-0.7,-9};
	double b[10], c[10];
	int n = 0; //змінна для підрахунку кількостей
	//функції sin та cos приймають тип double
	//використовуємо явне приведення типу
	for(int k = 0; k < 10; k++)
		b[k] = 2 * sin((double)k) + cos((double)k);
	//доступ до елементів масиву через покажчики
	for(int k = 0; k < 10; k++)
			*(c + k) = *(a + 9 - k) + *(b + k);
	setlocale(0,"ukr");
	cout<<"Утворені масиви:\nA:\n";
	for(int k=0; k < 10; k++)
	{
		cout << a[k] << '\t';
		if(a[k] > -1 && a[k] < 1) n++;
	}
	cout << "\nКількість елементів з (-1; 1) рівна "<< n <<"\nB:\n";
	n=0;
	for(int k=0; k<10; k++)
	{
		cout << b[k] << '\t';
		if(b[k] > -1 && b[k] < 1)n++;
	}
	cout<<"\nКількість елементів з (-1; 1) рівна "<< n <<"\nC:\n";
	n=0;
	for(int k=0; k<10; k++)
	{
		cout << c[k] << '\t';
		if(c[k] > -1 && c[k] < 1)n++;
	}
	cout << "\nКількість елементів з (-1; 1) рівна " << n;
}
```

Лабораторна робота № 8

Робота з двовимірними масивами. Двовимірні масиви та вказівники.

Мета роботи: Оволодіння практичними навичками роботи з двовимірними масивами.

Короткі теоретичні відомості.

Двовимірний масив можна вважати масивом одновимірних масивів. Елементи двовимірних масивів, відповідно, індексуються парою індексів – кожен записують в окремих квадратних дужках. Загальна форма оголошення двовимірного масиву:

```
<тип елементів> <назва
масиву>[<розмірність1>][<розмірність2>];
```

Для опрацювання двовимірних масивів використовують вкладені цикли for, наприклад:

```
int max=A[0][0];
for(int i=0;i<3;i++)
     for(int j=0;j<4;j++)
          if (max < A[i][j]) max = A[i][j];
```

Дозволяється також ініціалізація двовимірного масиву елементами під час опису, наприклад:

```
int A[3][4] ={ {5, 6, 1, 3}, {3, 4, 2, 1}, {1, 2, 2, 2} };
```

З точки зору виділення пам’яті для зберігання даних спосіб організації двовимірних масивів C/C++ має певну специфіку. Елементи масиву зберігаються в пам’яті по рядках підряд, фактично у вигляді одновимірного масиву, а ім’я масиву є покажчиком на перший елемент першого рядка масиву. При такій організації пам’яті для індексації елементів важливою є інформація про кількість елементів у рядку. Відповідно, ім’я оголошеного вище масиву A буде покажчиком типу ***int (*)[4]*** і не співпадає ні з типом покажчика на одновимірний масив ***int ****, ні з типом покажчика на покажчик ***int ***, (двовимірний динамічний масив).

В C/C++ можна використовувати також тривимірні масиви, чотиривимірні масиви і т. д. Наприклад:

```
double M[3][3][4], Q[4][4][4][3];
```

кількість розмірностей масивів в C, взагалі кажучи необмежена, проте зі збільшенням розмірності масиву різко зростає кількість його елементів (наприклад масив Q в прикладі має 4*4*4*3 = 192 елемети), що може призвести до проблем з вільним місцем в оперативній пам’яті ПК.

Завдання. Задано цілочисельний масив (матриця) А розмірності 5х5. Масив А ініцалізувати елементами в тексті програми при його описі (заповнити довільно – 5 рядків по 5 чисел). В масиві А знайти вказані у варіанті завдання величини та вивести результат у вигляді повідомлення на консоль, вивівши попередньо масив у вигляді таблиці. Утворити новий масив В, згідно з вказівками відповідного варіанту, утворений масив вивести на консоль.

Варіанти завдань

1. В масиві А знайти кількість, суму та середнє значення додатних елементів. Масив В утворити з масиву А, замінивши в ньому всі додатні елементи числом 5, а елементи, менші за -5 – середнім значенням додатних елементів.

2. В масиві А знайти різницю середніх значень окремо взятих додатних та від'ємних елементів. Масив В утворити з масиву А, замінивши в ньому всі елементи менші за -5 на протилежні.

3. В масиві А знайти кількості елементів, значення яких лежать в діапазонах від 1 до 10, від 11 до 25 та від 1 до 20. Масив В утворити з масиву А, замінивши в ньому всі елементи на протилежні.

4. В матриці А знайти добуток мінімальних елементів кожного рядка. Матрицю В утворити транспонуванням матриці А.

5. В матриці А знайти суму середніх значень елементів рядків. Матрицю В утворити з матриці А, замінивши всі елементи під головною діагоналлю знайденою сумою.

6. В матриці А знайти відношення кількості додатних елементів до кількості від'ємних. Матрицю В утворити з матриці А, відобразивши її симетрично відносно середнього рядка.

7. В матриці А знайти кількості додатних елементів у кожному рядку. Матрицю В утворити з матриці А, видаливши з неї перший стовпчик, змістивши решту її стовпців на один вліво та додавши останнім стовпцем знайдені кількості додатних елементів рядка.

8. В матриці А знайти суму елементів, розташованих над головною діагоналлю та суму елементів, розташованих під нею. Матрицю В утворити з матриці А, замінивши елементи головної діагоналі різницею знайдених сум.

9. В матриці А знайти відношення мінімального з додатних елементів до максимального з від'ємних. Матрицю В утворити з матриці А перестановкою її стовпців в зворотному порядку.

10. В масиві А знайти відношення значень максимального та мінімального елементів. Масив В утворити з масиву А, замінивши в ньому всі елементи більші за 10 на максимальний елемент, а від'ємні елементи на 0.

11. В матриці А знайти відношення середнього значення елементів, розташованих над головною діагоналлю до середнього значення елементів цієї діагоналі. Матрицю В утворити з матриці А перестановкою її рядків в зворотному порядку.

12. В матриці А знайти середнє арифметичне елементів, розташованих над головною діагоналлю та середнє елементів, розташованих під нею. Матрицю В утворити з матриці А, відобразивши її симетрично відносно другорядної діагоналі.

13. В матриці А знайти кількості додатних елементів у кожному стовпці. Матрицю В утворити з матриці А, видаливши з неї перший рядок, змістивши решту її рядків на один вгору та додавши останнім рядком знайдені кількості додатних елементів стовпця.

14. В матриці А знайти відношення кількості нульових елементів до кількості ненульових. Матрицю В утворити з матриці А, відобразивши її симетрично відносно середнього стовпця.

15. В матриці А знайти суму середніх значень елементів стовпців. Матрицю В утворити з матриці А, замінивши всі елементи над головною діагоналлю знайденою сумою.

16. В масиві А знайти розмах значень його елементів (різницю значень максимального та мінімального елементів). Масив В утворити з масиву А, замінивши в ньому всі елементи більші за 5 на мінімальний елемент, а елементи, менші за -5 – на максимальний.

Приклад виконання лабораторної роботи.

Варіант 16. В масиві А знайти розмах значень його елементів (різницю значень максимального та мінімального елементів). Масив В утворити з масиву А, замінивши в ньому всі елементи більші за 5 на мінімальний елемент, а елементи, менші за -5 – на максимальний.

```
#include<iostream>
using namespace std;

void main()
{
	int i,j,max,min,d;
	int B[5][5], A[5][5]={{1,2,3,7,9},{-5,-7,-8,4,5},
			{-3,-4,-5,-6,-7},{0,7,-6,8,-9},{4,11,-22,21,4}};
	//int *PA = A; помилка - типи покажчиків не співпадають
	//int **PA = A; типи покажчиків також не співпадають

	//відшукання мінімального та максимального елементів
	max = min = A[0][0];
	for(i=0; i<5; i++)
		for(j=0; j<5; j++)
		{
			if(A[i][j]>max) max = A[i][j];
			if(A[i][j]<min) min = A[i][j];
		}
	system("cls");//очистка консольного вікна від попереднього тексту
	cout<<"\t\tA\n";
	for(i=0; i<5; i++)
	{
		int *row = A[i];//покажчик на перший елемент і-го рядка
		// приклад звертання до елементів двовимірного
		for(j=0; j<5; j++)
		{
			cout<<*(row + j)<<'\t';
		}
		cout<<'\n';
	}

	d=max-min;
	cout<<"\nMax - Min = "<<d;
	for(i=0; i<5; i++)
		for(j=0; j<5; j++)
		{
			if(A[i][j]>5) B[i][j] = min;
			else
			{
				if(A[i][j]<-5) B[i][j] = max;
				else B[i][j] = A[i][j];
			}
		}
```

```
	cout<<"\n\n\t\tB\n";
	for(i=0; i<5; i++)
	{
		for(j=0; j<5; j++)
		{
			cout<<B[i][j]<<'\t';
		}
		cout<<'\n';
	}
}
```

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для чого використовують масиви в програмуванні?
2. Який синтаксис опису одновимірного масиву в С?
3. Що таке ініціалізація масиву?
4. Як отримати доступ до окремого елемента одновимірного масиву?
5. Чим є ім’я одновимірного масиву в С?
6. Який синтаксис опису двовимірного масиву в С?
7. Яка максимальна розмірність масиву в мові С?
8. Яка максимальна розмірність масиву в мові С++?
9. Як послідовно опрацювати усі елементи двовимірного масиву?
10. Чим є ім’я двовимірного масиву в С?
11. Що таке покажчик (вказівник) в С?
12. Для чого в програмах на С можна використовувати покажчики?

Рекомендована література:

1. Програмування на С(С++). Парадигма процедурного програмування. Навчальний посібник. /Гнатів Б.В., Гнатів Л.Б. . – Львів: Видавництво «Растр-7», 2017. – 264 с.
2. Основи програмування на С/С++. Конспект лекцій з курсу «Основи інформатики і програмування, частина 2» спеціальності 105 – «Прикладна фізика та наноматеріали» для першого (бакалаврського) рівня освіти/ Укл.: Ментинський С.М., 2016. – 140 с
3. П. Каленюк, І. Клюйник, І. Кравець, Л. Гнатів, Л.Демків, І. Подун Основи програмування в середовищі С. Лекції та завдання до лабораторних робіт. Львів 2007 р., 98 с.
4. Петрович Р. Й., Тумашова О. В. “Основи програмування мовою Сі.”. Навчальний посібник. Видавництво НУ”ЛП”, Львів, 2005р. 100с.
5. Програмування мовою С: навчальний посібник для вузів / Зореслава Ярославівна Шпак . — Львів: Оріяна-Нова, 2006 . — 431 с.
6. Креневич, А.П. С у задачах і прикладах : навчальний посібник із дисципліни "Інформатика та програмування" / А.П. Креневич, О.В. Обвінцев. – К. : Видавничо-поліграфічний центр "Київський університет", 2011. – 208 с.
7. Шилдт, Герберт. Полный справочник по С++, 4-е издание. . Пер. с англ. — М. : Издательский дом “Вильямс”, 2006. — 800 с. : ил.
8. Стивен Пратта Язык программирования С. Лекции и упражнения - К.: Диасофт, 2002.

НАВЧАЛЬНЕ ВИДАННЯ

Використання вказівників для обробки масивів в С. Пошук в одновимірному масиві. Робота з двовимірними масивами. Двовимірні масиви та вказівники.

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт № № 7,8
з дисципліни «Основи інформатики та програмування»

для студентів спеціальності 105 – "Прикладна фізика та наноматеріали" для першого (бакалаврського) рівня освіти.

Укладачі: Гнатів Л.Б., к. ф-м. н., доцент,
Ментинський С.М., ст. викл,
Пелех Я.М., к. ф-м. н., доцент,
Угрин С.З., асистент.

Комп’ютерне складання: Ментинський С.М., ст. викл.