

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

ФУНКЦІЇ КОРИСТУВАЧА В C++

**Методичні рекомендації та варіанти завдань
до виконання лабораторних робіт
з дисципліни «Програмування та алгоритмічні мови»**

бакалаврами напрямку підготовки 17 Електроніка та телекомунікації

Дніпро
2021

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



ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ФУНКЦІЇ КОРИСТУВАЧА В C++

**Методичні рекомендації та варіанти завдань
до виконання лабораторних робіт
з дисципліни «Програмування та алгоритмічні мови»**
бакалаврами напрямку підготовки 17 Електроніка та телекомунікації

Дніпро
НТУ «ДП»
2021

Функції користувача в С++. Методичні рекомендації та варіанти завдань до виконання лабораторних робіт з дисципліни «Програмування та алгоритмічні мови» бакалаврами напряму підготовки 17 Електроніка та телекомунікації / Упоряд.: О.А. Жукова, Ю.А. Мілінчук ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2021. – 20 с.

Упорядники:

О.А. Жукова, ст. викл.;

Ю.А. Мілінчук, асист.

Затверджено методичною комісією за напрямом «Електроніка та телекомунікації» (протокол № 2 від 09.03.2021) за поданням кафедри безпеки інформації та телекомунікацій (протокол № 10 від 09.03.2021).

Подано методичні рекомендації до виконання лабораторних робіт з дисципліни «Програмування та алгоритмічні мови» для бакалаврів напряму 17 Електроніка та телекомунікації.

Відповідальний за випуск зав. кафедри БІТ В.І. Корнієнко, д-р техн. наук, проф.

ЗМІСТ

ПЕРЕДМОВА.....	4
1 ЛАБОРАТОРНА РОБОТА № 1.....	5
1.1 Мета роботи.....	5
1.2 Постановка задачі.....	5
1.3 Теоретичні відомості.....	5
1.4 Оформлення звіту.....	8
1.5 Варіанти завдань	8
1.6 Приклад.....	11
1.7 Контрольні питання	11
2 ЛАБОРАТОРНА РОБОТА № 2.....	12
2.1 Мета роботи.....	12
2.2 Постановка задачі.....	12
2.3 Теоретичні відомості.....	12
2.4 Оформлення звіту.....	14
2.5 Варіанти завдань	14
2.6 Приклад А.....	17
2.7 Приклад В.....	17
2.8 Контрольні питання	19
ЛІТЕРАТУРА	19

ПЕРЕДМОВА

При написанні програм середнього і високого рівня складності виникає потреба в їх розбитті на частини. Розбиття великої програми на менші частини дозволяє зменшити ризик виникнення помилок, підвищує читабельність програмного кода завдяки його структуризації.

Крім того, якщо деякий програмний код повторюється кілька разів (або є близьким по сенсу), доцільно організувати його у вигляді функції, яку потім можна викликати багато разів, використовуючи її ім'я. Таким чином, відбувається економія пам'яті, зменшення початкового коду програми і тому подібне.

Функція – це частина програми, яка має наступні властивості або ознаки:

- є логічно самостійною частиною програми;
- має ім'я, на підставі якого здійснюється виклик функції (виконання функції).
Ім'я функції повинне відповідати правилам створення ідентифікаторів мови C++;
- може містити список параметрів, які передаються їй для обробки або використання. Якщо функція не містить списку параметрів, то така функція називається функцією без параметрів;
- може повертати (не обов'язково) деякі значення. У випадку, якщо функція не повертає ніякого значення, тоді вказується ключове слово *void*;
- має власний програмний код, який розташований між фігурним дужками `{}`. Програмний код вирішує задачу, поставлену для цієї функції. Програмний код функції, реалізований у фігурних дужках, називається «Тіло функції».

Використання функцій в програмах дає наступні переваги:

- компактна організація програми шляхом зручного виклику програмного коду по його імені, який в програмі може повторюватися ;
- економія пам'яті, розміру початкового і виконавчого коду і т.д.;
- зменшення ризиків виникнення помилок для великих наборів кодів;
- підвищення читабельності програмного коду.

1. ЛАБОРАТОРНА РОБОТА № 1

ФУНКЦІЇ

1.1. Мета роботи

Дослідити принципи побудови функцій користувача, основні складові функцій, прототипи, оголошення та опис функцій.

Навчитися розробляти і використовувати на практиці власні функції для вирішення завдань, які стоять перед розробником програм.

1.2. Постановка задачі

Скласти схему алгоритму, таблицю імен та програму для реалізації завдання з використанням функцій відповідно до заданого варіанта.

1.3. Теоретичні відомості

Поняття функції та її складові

Функція – поіменована послідовність описів і операторів, яка виконує деяку закінчену послідовність дій. Будь-яка функція складається із заголовка (оголошення функції) і тіла (визначення функції).

Оголошення функції описує її **прототип** (іноді кажуть «сигнатура»).

Прототип функції оголошується наступним чином:

ТипПовернення *Ім'яФункції* (*СписокОголошенихПараметрів*);

Тут *ТипПовернення* – тип даних, що повертається функцією. Якщо він не зазначений, то за замовчуванням вважається, що повертається тип *int*.

СписокОголошенихПараметрів задає тип і ім'я кожного з параметрів функції, розділених комами (ім'я параметра можна опускати). Список параметрів функції може бути порожнім. Приклади прототипів функцій:

```
double max (double par1, double par2);
```

```
int swap(int, int) ;
```

```
void func() ;
```

Визначення функції складається з її **заголовка** і власне **тіла**, вкладеного у фігурні дужки і такого, що має смислове навантаження. Якщо функція повертає будь-яке значення, в тілі функції обов'язково повинен бути присутнім оператор повернення з параметром того ж типу.

Тип Поверн Ім'я Функції (Список Оголошених Параметрів)

```
{ < тіло функції > }
```

Виклик функції – вказівка ідентифікатора функції (її імені), за яким в круглих дужках слідує список *аргументів*, розділених комами.

```
double maxVal = max (2.5, 1235.2);  
int j = swap(10, 2);  
func();
```

Функції, що не повертають значення

Це функції типу *void* – ті, що не повертають значення – можуть розглядатися, як деякий різновид команд, реалізований особливими програмними операторами. Оператор *func()*; виконує функцію *void func()* , тобто передає керування функції, доки не виконаються усі її оператори. Коли функція поверне керування в основну програму, тобто завершить свою роботу, програма продовжить своє виконання з того місця, де розташовується наступний оператор за оператором *func()*.

Якщо функція повертає значення типу *void*, то її виклик слід організувати так, щоб значення, яке повертається, не використовувалося. Тобто, таку функцію не використовують у правій частині виразу.

Рекурсивні функції

Рекурсія – це спосіб організації обчислювального процесу, при якому функція в ході виконання операторів звертається сама до себе.

Функція називається **рекурсивною**, якщо під час її роботи можливий повторний її виклик безпосередньо (прямий виклик) або шляхом виклику іншої функції, в якій міститься звернення до неї (непрямий виклик).

Прямою (безпосередньою) **рекурсією** називається рекурсія, при якій всередині тіла деякої функції міститься виклик тієї ж функції.

```
void fn(int i)  
{ ... fn(i); ... }
```


Непрямою рекурсією називається рекурсія, що здійснює рекурсивний виклик функції шляхом ланцюга викликів інших функцій. При цьому всі функції ланцюга, що здійснюють рекурсію, вважаються рекурсивними.

```
void fnA(int i)
    { ... fnB(i); ... }
void fnB(int i)
    { ... fnC(i); ... }
void fnC(int i)
    { ... fnA(i); ... }
```

Наприклад, функція обчислення факторіала $n!$ будет виглядат так:

```
double fact(int n)
{
    if (n<=1) return 1;
    return (fact(n-1)*n);
}
void main()
{
    int n; double value;
    cout << "N="; cin >>n;
    value=fact(n); // Виклик функції fact
    cout <<"d! = "<<value;
    getch();
}
```

Аргументи функції за замовчуванням

C++ допускає при виклику функцій опускати деякі її параметри. Досягається це зазначенням в прототипі функції значень аргументів за замовчуванням. Наприклад, функція, прототип якої наведено нижче, може при виклику мати різний вигляд в залежності від ситуації.

// Прототип функції:

```
void ShowInt(int i,bool Flag=true,char symbol='\n');
// Виклик функції ShowInt:
ShowInt(A, false, 'a');
ShowInt(B, false);
ShowInt(C);
```

1.4. Оформлення звіту

Звіт повинен містити:

- номер і назву лабораторної роботи;
- умову задачі та дані свого варіанту;
- схеми алгоритмів, включаючи детальні схеми функцій;
- таблицю символічних імен;
- C-програму з коментарями;
- результати розрахунку програми.

1.5. Варіанти завдань

1. Написати програму з функцією для обчислення $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$. Обчислити факторіали чисел від 1 до 7.

2. Написати програму з функцією для обчислення суми n членів арифметичної прогресії за формулою $S = \frac{(a_1 + a_n) \cdot n}{2}$. Обчислити суму арифметичної прогресії: 1, 2, 3, ..., 100.

3. Написати програму з функцією для обчислення суми n членів геометричної прогресії за формулою $S = \frac{a_1 \cdot (1 - q^n)}{1 - q}$. Обчислити суму геометричної прогресії: 1, 2, 4, 8, ..., 100.

4. Написати програму з функцією для піднесення числа x у степінь m (m – натуральне число). Обчислити x^2, x^4, x^5 .

5. Написати програму з функцією для обчислення арксинуса. Обчислити $\arcsin(0,9), \arcsin(0,1), \arcsin(-0,9), \arcsin(0,99)$. При обчисленні скористатися формулою $y = \arctg\left(\frac{x}{\sqrt{1-x^2}}\right)$.

6. Написати програму з функцією для обчислення арккосинуса. Обчислити $\arccos(0,5), \arccos(0,6), \arccos(-0,5), \arccos(0,99)$. Для обчислень скористатися формулою $y = \arctg\left(\frac{\sqrt{1-x^2}}{x}\right)$.

7. Дан масив розміру $A(7)$ з елементами дійсного типу. Написати програму для обчислення суми додатних елементів масиву. Обчислення суми додатних елементів оформити у вигляді функції.

8. Дан масив розміром $B(5)$ з елементами дійсного типу. Написати програму, для обчислення суми від'ємних елементів масиву. Обчислення суми в масиві від'ємних елементів оформити у вигляді функції.

9. Дан масив розміром $C(4)$ з елементами дійсного типу. Написати програму, що визначає індекс елемента, якій дорівнює 0. Визначення індексу елемента в масиві оформити у вигляді функції.

10. Даний масив розміром $X(6)$ з елементами цілого типу. Написати програму для обчислення добутку ненульових елементів масиву. Обчислення добутку ненульових елементів оформити у вигляді функції.

11. Даний масив розміром $A(6)$ з елементами дійсного типу. Написати програму для обчислення індексу мінімального елемента масиву. Обчислення індексу мінімального елемента оформити у вигляді функції.

12. Даний масив розміром 3×5 з елементами цілого типу. Написати програму для обчислення номера стовпчика, у якому знаходиться мінімальний елемент масиву. Обчислення номера стовпчика, у якому знаходиться мінімальний елемент оформити у вигляді функції.

13. Даний масив розміру 6×2 з елементами дійсного типу. Написати програму для обчислення $M = \max - \min$, де \max – максимальний за модулем елемент масиву, а \min – мінімальний за модулем елемент масиву. Обчислення значення M оформити у вигляді функції.

14. Написати функцію для обчислення тангенса. Обчислити $ctg(0,9)$, $ctg(0,1)$, $ctg(-0,9)$. При обчисленні скористатися формулою $Y = \frac{\cos(x)}{\sin(x)}$.

15. Даний масив розміром 3×3 з елементами символьного типу. Написати програму для обчислення кількості елементів масиву, що не є цифрами. Обчислення кількості елементів масиву, що не є цифрами, оформити у вигляді функції.

16. Даний масив розміром 4×4 з елементами цілого типу. Написати програму для обчислення добутку додатних елементів масиву. Обчислення добутку додатних елементів оформити у вигляді функції.

17. Написати функцію для обчислення тангенса. Обчислити $tg(0,9)$, $tg(0,1)$, $tg(-0,9)$. При обчисленні скористатися формулою $Y = \frac{\sin(x)}{\cos(x)}$.

18. Написати функцію для піднесення числа x у степінь 3, тобто обчислити x^3 .

19. Даний масив розміром 5×5 з елементами цілого типу. Написати програму, що визначає чи є в масиві додатні елементи. Визначення наявності в масиві додатних елементів оформити у вигляді функції.

20. Даний масив розміром 3×4 з елементами дійсного типу. Написати програму, що визначає кількість негативних елементів. Визначення кількості негативних елементів оформити у вигляді функції.

21. Написати функцію для обчислення площі трикутника за формулою Герона: $S = \sqrt{p(p-a)(p-b)(p-c)}$, де a, b, c – сторони трикутника, а p – напівпериметр. Обчислити площу трикутників за такими вихідними даними:

Сторона a , см	Сторона b , см	Сторона c , см
3.1	4.5	1.2
4.0	6.7	2.9

22. Написати функцію для обчислення третьої сторони трикутника за відомими двома сторонами і куту між ними. Обчислити сторони трикутників за табличними даними. Для обчислень скористатися формулою $a^2 = b^2 + c^2 - 2 \cdot a \cdot c \cdot \cos \alpha$.

Сторона a , см	Сторона b , см	Кут α , рад
3	4	$\pi/2$
2	3	$\pi/4$

23. Написати функцію для обчислення об'єму кульового сектора $V = \frac{2}{3} \pi R^2 H$, де R – радіус сфери, а H – висота сектора. Обчислити об'єм кульових секторів.

24. Дан масив розміром $B(5)$ з елементами цілого типу. Написати програму, для обчислення додатка додатніх елементів масиву. Обчислення суми в масиві від'ємних елементів оформити у вигляді функції.

25. Дан масив розміром $F(4)$ з елементами дійсного типу. Написати програму, що визначає індекс максимального негативного елемента.

1.6. Приклад

Написати функцію для обчислення значення x^m .

```
long int Step( long int a, long int b )
{
    long int r;
    r = 1;
    if ( b > 0 )
        for( int i = 0; i < b; i++ )
        {
            r = r * a;
        }
    else r = 0;
    return r;
}
int main()
{
    long int x, s;
    cout << "Уведить число: ";
    cin >> x;
    cout << "Уведить степінь: ";
    cin >> s;
    cout << x << " у степеню " << s << " = " << Step( x, s ) << endl;
    return 0;
}
```

1.7. Контрольні питання

1. Дайте визначення функції.
2. Наведіть відомі Вам ознаки функції.
3. Як оголошується прототип функції? Наведіть приклад.
4. З чого складається визначення функції? Наведіть приклад.
5. Які функції не повертають значення?
6. Що таке рекурсія?
7. Яка рекурсія називається прямою (безпосередньою)?
8. Яка рекурсія називається непрямою?
9. Коли C++ допускає при виклику функцій опускати деякі її параметри?

2. ЛАБОРАТОРНА РОБОТА № 2

ФУНКЦІЇ З ПАРАМЕТРАМИ

2.1. Мета роботи

Дослідити принципи побудови функцій користувача, основні складові функцій, прототипи, оголошення та опис функцій.

Навчитися розробляти і використовувати на практиці власні функції для вирішення завдань, які стоять перед розробником програм.

2.2. Постановка задачі

Скласти схему алгоритму, таблицю імен та програму для реалізації необхідної задачі з використанням функцій з параметрами, відповідно заданого варіанта.

2.3. Теоретичні відомості

Передавання параметрів у функцію

Механізм передавання параметрів є основним способом обміну інформацією між функціями. Параметри, перераховані в заголовку опису функції, називаються **формальними** параметрами (або просто параметрами), а записані в операторі виклику функції – **фактичними** параметрами (або аргументами).

Існує два способи передачі параметрів у функцію: за значенням і за адресою.

При **передаванні за значенням** в стек заносяться копії значень аргументів, і оператори функції працюють з цими копіями. Доступу до початкових значень параметрів у функції немає, а, отже, немає і можливості їх змінити.

При **передаванні за адресою** в стек заносяться копії адрес аргументів, а функція здійснює доступ до комірок пам'яті за цими адресами і може змінити значення аргументів:

```
void f(int i, int* j, int& k);
int main()
{
    int i=1,j=2,k=3;
    cout <<"i j k\n";
```

```

        cout<<i<<" "<<j<<" "<<k<<'\n'; // Результат: 1 2 3
        f(i,&j,k);
        cout<<i<<" "<<j<<" "<<k; // Результат: 1 3 4
        return 0;
    }
void f(int i, int* j, int& k)
{
    i++; (*j)++; k++;
}

```

Перший параметр (*i*) передається за значенням. Його зміна в функції не впливає на вихідне значення. Другий параметр (*j*) передається за адресою за допомогою покажчика, при цьому для передавання у функцію адреси фактичного параметра використовується операція взяття адреси, а для отримання його значення в функції потрібна операція розіменування. Третій параметр (*k*) передається адресою за допомогою посилання.

При передаванні за посиланням у функцію передається адреса зазначеного при виклику параметра, а всередині функції всі звернення до параметру неявно розіменовуються. Використання посилань замість покажчиків, по-перше, покращує читабельність програми, позбавляючи від необхідності застосовувати операції одержання адреси та розіменування; і, по-друге, не вимагає копіювання параметрів, що важливо при передаванні структур даних великого обсягу.

Якщо потрібно заборонити зміну параметра усередині функції, використовується модифікатор *const*:

```

int f(const char*);
char* t(char* a, const int* b);

```

Масиви як параметри функції

Усі параметри, за винятком параметрів типу покажчик та масив, передаються за значенням. Це означає, що під час виклику функції їй передаються тільки значення змінних. Сама функція не в змозі змінити цих значень у функції, що їх викликає.

Одновимірний масив як параметр

Ім'я масиву є покажчиком на його нульовий елемент, тому у функцію масиви передаються за покажчиком. Кількість елементів масиву може передаватися окремим параметром.

2.4. Оформлення звіту

Звіт повинен містити:

- номер і назву лабораторної роботи;
- умову задачі та дані свого варіанту;
- схеми алгоритма, включаючи детальні схеми функцій;
- таблицю символічних імен;
- C-програму з коментарями;
- результати розрахунку програми.

2.5. Варіанти завдань

1. Дано матриці $A(4,3)$ і $U(4,3)$. Для кожної матриці обчислити суму елементів над головною діагоналлю, під головною діагоналлю і на головній діагоналі. Уведення, виведення матриці й обчислення сум оформити у вигляді функцій.

2. Дано матриці $A(4,3)$ і $U(4,3)$. Для кожної матриці знайти найбільший і найменший елементи головної діагоналі. Уведення, виведення матриці, пошук мінімального і максимального елементів оформити у вигляді функцій.

3. Дано матриці $A(3,3)$, $U(3,3)$, $C(3,3)$. Для кожної матриці визначити максимальні елементи в рядках і перемістити їх на головну діагональ. Уведення, виведення матриці, перетворення матриці оформити у вигляді функцій.

4. Дано матриці $A(4,3)$ і $U(4,3)$. Знайти середні геометричні позитивних елементів кожного рядка матриць. Уведення, виведення матриці та пошук середніх геометричних оформити у вигляді функцій.

5. Дано матриці $A(6,3)$ і $U(6,3)$. Знайти найбільший елемент кожної матриці і записати 1 у той рядок і стовпчик, у яких він знаходиться. Уведення, виведення матриці і перетворення матриці оформити у вигляді функцій.

6. Дано матриці $A(3,4)$ і $U(3,4)$. Транспонувати матриці. Уведення, виведення матриці і транспонування оформити у вигляді функцій.

7. Дано матриці $A(6,2)$ і $U(6,2)$. Знайти найбільший елемент кожної матриці і номери рядків і стовпчиків, у яких він знаходиться. Уведення, виведення матриці і пошук максимального елемента оформити у вигляді функцій.

8. Дано матриці $A(4,4)$ і $U(4,4)$. Знайти найменший елемент кожної матриці і записати в рядок, де знаходиться цей елемент. Уведення, виведення матриці і перетворення матриці оформити у вигляді функцій.

9. Дано матриці $A(5,5)$ і $U(5,5)$. Для кожної матриці обчислити суму елементів рядків і визначити найменше значення цих сум і номер відповідного рядка. Уведення, виведення матриці і пошук номера рядка оформити у вигляді функцій.

10. Дано матриці $A(3,3)$ і $U(3,3)$. Визначити кількість додатних і від'ємних елементів матриць. Уведення, виведення матриці і пошук кількості додатних і від'ємних елементів оформити у вигляді функцій.

11. Дано матриці $A(5,2)$ і $U(5,2)$. Для кожної матриці знайти найбільші елементи кожного рядка матриці і записати їх в одномірний масив. Уведення, виведення матриці і формування одномірного масиву оформити у вигляді функцій.

12. Дано три матриці $A(4,3)$, $B(4,3)$, $C(4,3)$ і два дійсних числа x і y . Знайти $x*A$, $y*B$, $x*C+y*B$. Уведення, виведення матриці, додавання матриць, множення матриці на число оформити у вигляді функцій.

13. Дано три матриці $A(4,3)$, $B(4,3)$, $C(4,3)$. Знайти AT , BT , CT , де AT , BT , CT – транспоновані матриці. Уведення, виведення, транспонування матриці оформити у вигляді функцій.

14. Дано матриці $A(4,3)$, $B(4,3)$, $C(4,3)$. Для кожної матриці обчислити суму елементів над головною діагоналлю. Уведення, виведення матриці й обчислення суми оформити у вигляді функцій.

15. Дано матриці $A(5,5)$, $U(5,5)$, $C(5,5)$. Для кожної матриці обчислити суму елементів під головною діагоналлю. Уведення, виведення матриці й обчислення суми оформити у вигляді функцій.

16. Дано матриці $A(3,5)$, $U(3,5)$, $C(3,5)$. У кожній матриці поміняти місцями перший та останій рядки. Уведення, виведення матриці, обмін рядків оформити у вигляді функцій.

17. Дано матриці $A(3,5)$, $U(3,5)$, $C(3,5)$. У кожній матриці поміняти місцями перший і останній стовпчики. Уведення, виведення матриці, обмін стовпчиків оформити у вигляді функцій.

18. Дано матриці $A(3,5)$, $U(3,5)$, $C(3,5)$. У кожній матриці визначити місце розташування максимального елемента. Уведення, виведення матриці, пошук місця розташування максимального елемента оформити у вигляді функцій.

19. Дано матриці $A(3,5)$, $U(3,5)$, $C(3,5)$. Утворити три нові матриці діленням кожного елемента вихідної матриці на середнє арифметичне її елементів. Уведення, виведення матриці, змінення матриць оформити у вигляді функцій.

20. Дано матриці $A(4,4)$, $U(4,4)$, $C(4,4)$. Для кожної матриці обчислити добуток елементів на головній діагоналі. Уведення, виведення матриці й обчислення добутка оформити у вигляді функцій.

21. Дано матриці $A(4,4)$, $U(4,4)$, $C(4,4)$. Для кожної матриці обчислити суму від'ємних елементів головної діагоналі. Уведення, виведення матриці, обчислення суми елементів головної діагоналі оформити у вигляді функцій.

22. Дано матриці $A(3,5)$, $U(3,5)$, $C(3,5)$. Для кожної матриці обчислити суми рядків. Уведення, виведення матриці, обчислення сум рядків, виведення вектора оформити у вигляді функцій.

23. Дано матриці $A(4,5)$, $U(4,5)$, $C(4,5)$. Для кожної матриці обчислити суми стовпчиків. Уведення, виведення матриці, обчислення сум стовпчиків, виведення вектора оформити у вигляді функцій.

24. Дано матриці $A(5,5)$, $U(5,5)$, $C(5,5)$. Для кожної матриці обчислити суму елементів над головною діагоналлю. Уведення, виведення матриці й обчислення суми оформити у вигляді функцій.

25. Дано матриці $A(3,5)$, $U(3,5)$, $C(3,5)$. У кожній матриці поміняти місцями другий та останій рядки. Уведення, виведення матриці, обмін рядків оформити у вигляді функцій.

2.6. Приклад А

Написати програму, яка підраховує, в якому з двох масивів кількість додатних елементів більша.

```
int n_posit(int* mas, int n);
int main()
{
    int n,i;
    cout<<"Input count of symbols: ";
    cin>>n;
    int *A = new int[n];
    int *B = new int[n];
    for (i=0; i<n; i++) //генеруємо елементи масивів
    {
        A[i] = rand()%100 - 50;
        B[i] = rand()%200 - 100;
    }
    cout << "\n\nA: "; //виводимо масиви на екран
    for (i=0; i<n; i++) cout << " " << A[i];
    cout << "\n\nB: ";
    for (i=0; i<n; i++) cout << " " << B[i]; //порівнюємо кількість додатних
елементів
    if (n_posit(A,n) > n_posit(B,n))
        cout << "\n\nFirst array A is winner!";
    else if (n_posit(A,n) < n_posit(B,n))
        cout << "\n\nSecond array B is winner!";
    else cout << "\n\nEqual count!";
    _getch();
    return 0;
}
int n_posit(int* mas, int n)
{
    int count = 0;
    for (int i=0; i<n; i++)
        if (mas[i]>0) count++;
    return count;
}
```

2.7. Приклад В

Дано матрицю $A(4,5)$. Знайти максимальний і мінімальний елементи. При цьому необхідно створити функції для пошуку максимального і мінімального елементів. Уведення та виведення матриці виконати за допомогою функцій.

```

const int N = 2, M = 3;
void Input( int x[N][M], int n, int m, char *name ) // функція введення матриці
{
    for ( int i = 0; i < n; i++ )
    {
        for ( int j = 0; j < m; j++ )
        {
            cout << "Input" << name << "[" << i << "]"[" << j << "]" << ": ";
            cin >> x[ i ][ j ];
        }
    }
}
void Output( int x[N][M], int n, int m, char *name )// функція виводу матриці
{
    for ( int i = 0; i < n ; i++ )
    {
        for ( int j = 0; j < m; j++ )
        {
            cout << x[ i ][ j ] << '\t';
        }
        cout << endl;
    }
}
void PoiskMaxMin( int x[N][M], int n, int m, int *max, int *min )
// функція пошука мінімального, максимального елементів та їх індексів
{
    *max = x[0][0];
    *min = x[0][0];
    for ( int i = 0; i < n; i++ )
    {
        for ( int j = 0; j < m; j++ )
        {
            if ( *max < x[ i ][ j ] ) *max = x[ i ][ j ];
            if ( *min > x[ i ][ j ] ) *min = x[ i ][ j ];
        }
    }
}
int main()
{

    int mat[ N ][ M ];
    int mx,mn;
    Input(mat,N,M,"mat"); //Виклик функції Input
    cout << endl << endl;
    Output(mat,N,M,"mat"); //Виклик функції Output
}

```

```

cout << endl << endl;
PoiskMaxMin( mat,N, M, &mx, &mn ); //Виклик функції PoiskMaxMin
cout << "Максимальний елемент = " << mx << endl;
cout << "Мінімальний елемент = " << mn << endl;
return 0;
}

```

2.7. Контрольні питання

1. Який механізм є основним способом обміну інформацією між функціями?
2. Які параметри називаються формальними?
3. Які параметри називаються фактичними?
4. Назвіть способи передачі параметрів у функцію.
5. Що заноситься в стек при передаванні за значенням?
6. Що заноситься в стек при передаванні за адресою?

ЛІТЕРАТУРА

1. Пахомов Б. С/C++ и MS Visual C++ 2010 для начинающих. – СПб.: БХВ-Петербург, 2011. – 736 с. ISBN: 978-5-9775-0599-4.
2. Дейтел Х. М. Как программировать на С: 3-е издание / Х.М. Дейтел, П. Дж. Дейтел ; пер. с англ. В.В. Тимофеев. – М: Бином-Пресс, 2002. – 1168 с. – ISBN 5-9518-0002-1 (в пер.).
3. Дейтел Х.М. Как программировать на С++: 5-е издание / Х.М. Дейтел, П. Дж. Дейтел ; пер. с англ. В.В. Тимофеев. – М: Бином-Пресс, 2008. – 1456 с. – ISBN 978-5-9518-0224-8 (в пер.).
4. Шилдт Г., Полный справочник по С / Г. Шилдт. – СПб.: Вильямс, 2003. – 800 с. – ISBN 5-8459-0226-6 (в пер.).
5. Алгоритмы: построение и анализ, 2-е издание / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн ; пер. с англ. И.В. Красикова и др. – М. : Издательский дом «Вильямс», 2005. – 1296 с. – ISBN 5-8459-0857-4 (в пер.).
6. Лаптев В.В. С++. Объектно-ориентированное программирование: Учебное пособие / В.В. Лаптев. – СПб.: Питер, 2008. – 464 с. – ISBN 978-5-91180-200-4.
7. Керниган Б. У. Язык программирования С, 2-е издание / Б.У. Керниган, Д.М. Ритчи ; пер. с англ. В.Л. Бродового. – М. : Издательский дом «Вильямс», 2009. – 304 с. – ISBN 978-5-8459-0891-9 (в пер.).
8. Страуструп Б. Язык программирования С++. Специальное издание / Б. Страуструп ; пер. с англ. Н. Н. Мартынова. – М.: Бином, 2011. – 1136 с. – ISBN 978-57989-0425-9 (в пер.).

Упорядники:

Жукова Олена Андріївна

Мілінчук Юлія Анатоліївна

ФУНКЦІЇ КОРИСТУВАЧА В C++

**Методичні рекомендації та варіанти завдань
до виконання лабораторних робіт**

з дисципліни «Програмування та алгоритмічні мови»

бакалаврами напряму підготовки 17 Електроніка та телекомунікації

Видано в редакції упорядників

Комп'ютерний дизайн, верстка та обробка – **О.А. Жукова**

Підписано до друку 27.04.2021. Формат 30x42/4.

Папір офсет. Ризографія. Ум. друк. арк. 1,1.

Обл.-вид. арк. 1,1. Тираж 6 пр. Зам. №

Національний технічний університет «Дніпровська політехніка»
49005, м. Дніпро, просп. Д. Яворницького, 19.