

Лабораторна робота №16

Функції користувача, передача аргументів

Мета роботи: отримання практичних навичок написання та використання функцій користувача, вивчення особливостей передачі параметрів.

Зміст:

Короткі теоретичні відомості.....	1
Функції в мові C.....	1
Робоче завдання.....	8
Контрольні питання	10

Короткі теоретичні відомості

Функції в мові C

Функція – це фундаментальна модульна одиниця в мові C. Зазвичай функції створюють для виконання певного завдання.

Функція складається з оголошень і операторів.

З використанням функцій в мові C пов'язані три поняття:

- визначення функції (опис дій, які виконуються функцією),
- оголошення функції (опис форми звертання до функції)
- виклик функції.

Визначення (опис) містить тіло функції – код, який виконується при звертанні до функції. Визначення функції складається із:

- ✓ тип значення, що повертається,
- ✓ ім'я функції,
- ✓ типи, кількість формальних параметрів і їх імена,
- ✓ тіло функції, яке містить оголошення змінних і оператори, які визначають дію функції.

У визначенні функції також може бути заданий клас пам'яті.

Відповідно до синтаксису мови C визначення функції має наступну форму:

```
[специфікатор-класу-пам'яті] [специфікатор-типу] ім'я-функції  
(  
    [список-формальних-параметрів]  
)  
{  
    тіло-функції  
}
```

Необов'язковий *специфікатор-класу-пам'яті* задає клас пам'яті функції, який може бути **static** або **extern**.

Специфікатор-типу функції задає тип значення, що повертається і може бути будь-якого типу. Якщо специфікатор-типу не заданий, то вважається, що функція повертає значення типу **int**.

Наприклад:

```
int rus (unsigned char r)  
{  
    if (r>='A' && c<=' ')  
        return 1;  
    else  
        return 0;  
}
```

У мові C немає вимоги, щодо визначення функції обов'язково до її виклику. Визначення функцій що використовуються можуть слідувати за визначенням функції *main*, перед ним, або знаходитися в іншому файлі.

Однак для того, щоб компілятор міг здійснити перевірку відповідності типів переданих фактичних параметрів типам формальних параметрів, до виклику функції потрібно помістити оголошення (прототип) функції.

Функція повертає значення, якщо її виконання закінчується оператором **return**, що містить деякий вираз. Зазначений вираз обчислюється, перетворюється, якщо необхідно, до типу значення, що повертається і повертається в точку виклику функції в якості результату. Якщо оператор **return** не містить виразу або виконання функції завершується після виконання останнього її оператора (без виконання оператора **return**), то значення що повертається не визначено. Для функцій, що не використовують значення, що повертається, повинен бути використаний тип **void**, який вказує на відсутність значення, що повертається. Якщо функція визначена як функція, що повертає деяке значення, а в операторі **return** при виході з неї відсутній вираз, то поведінка функції що викликала після передачі їй управління може бути непередбачуваною.

Функція не може повертати масив або функцію, але може повертати вказівник на будь-який тип, в тому числі і на масив і на функцію. Тип значення, що повертається, що задається у визначенні функції, повинен відповідати типу в оголошенні цієї функції.

Список-формальних-параметрів – це послідовність оголошень формальних параметрів, розділена комами. Формальні параметри – це змінні, що використовуються всередині тіла функції і отримують значення при виконанні функції шляхом копіювання в них значень відповідних фактичних параметрів.

Якщо функція не використовує параметрів, то наявність круглих дужок обов'язкова, а замість списку параметрів рекомендується вказати слово **void**.

Порядок і типи формальних параметрів повинні бути однаковими у визначенні функції і у всіх її оголошеннях. Типи фактичних параметрів при виклику функції повинні бути сумісні з типами відповідних формальних параметрів. Тип формального параметра може бути будь-яким основним типом, структурою, об'єднанням, перерахуванням, вказівником або масивом.

Ідентифікатори формальних параметрів використовуються в тілі функції в якості посилань на передані значення. Ці ідентифікатори не можуть бути перевизначені в блоці, утворюючому тіло функції, але можуть бути перевизначені у внутрішньому блоці всередині тіла функції.

При передачі параметрів у функцію, якщо необхідно, виконуються звичайні арифметичні перетворення для кожного формального параметра і кожного фактичного параметра незалежно. Після перетворення формальний параметр не може бути коротше ніж **int**, тобто оголошення формального параметра з типом **char** рівнозначно його оголошенню з типом **int**. А параметри, які є дійсними числами, мають тип **double**.

Тіло функції – це складений оператор, що містить оператори, що визначають дію функції.

Всі змінні, оголошені в тілі функції без вказівки класу пам'яті, є локальними. При виконанні функції локальним змінним відводиться пам'ять в стеку і проводиться їх ініціалізація. Управління передається першому оператору тіла функції і починається виконання функції, яке триває до тих пір, поки не зустрінеться оператор **return** або останній оператор тіла функції. Управління при цьому повертається в точку, наступну за точкою виклику, а локальні змінні стають недоступними. При новому виклику функції для локальних змінних пам'ять виділяється знову, і тому старі значення локальних змінних втрачаються.

Параметри функції передаються за значенням і можуть розглядатися як локальні змінні, для яких виділяється пам'ять при виконанні функції і проводиться ініціалізація значеннями фактичних параметрів. При виході з функції значення цих змінних втрачаються. Оскільки передача параметрів відбувається за значенням, в тілі функції не можна змінити значення зовнішніх змінних у функції, що викликала. Однак, якщо в якості параметра передати вказівник на деяку змінну, то використовуючи операцію непрямого доступу можна змінити значення цієї змінної.

Оголошення функції вказує її назву, тип параметра що повертається та атрибути функції, описаної в іншому місці програми. Оголошення функції повинно передувати зверненню до неї в тексті програми. Оголошення, яке також містить інформацію про

кількість і тип переданих аргументів, називається також прототипом функції. Компілятор використовує прототип для звірки типів аргументів і необхідних перетворень типів в наступних викликах функції.

Для функції з попереднього прикладу прототип буде мати вигляд:

```
int rus (unsigned char r);
```

або

```
rus (unsigned char);
```

У програмах на мові C широко використовуються, так звані, бібліотечні функції, тобто функції попередньо розроблені і записані в бібліотеки. Прототипи бібліотечних функцій знаходяться в спеціальних файлах із заголовками, що поставляються разом з бібліотеками в складі систем програмування, і включаються в програму за допомогою директиви **#include**.

Якщо оголошення функції не задано, то за замовчуванням будується прототип функції на основі аналізу першого посилання на функцію. Однак такий прототип не завжди узгоджується з наступним визначенням чи викликом функції. Рекомендується завжди задавати прототип функції. Це дозволить компілятору або видавати діагностичні повідомлення, при неправильному використанні функції, або коректним чином регулювати невідповідність аргументів при виконанні програми.

Виклик функції передає управління з функції, що викликає, до функції, що викликається. Виконання виклику функції відбувається наступним чином:

1. Обчислюються значення в списку виразів і, якщо потрібно, піддаються звичайним арифметичним перетворенням. Потім, якщо відомий прототип функції, тип отриманого фактичного аргументу порівнюється з типом відповідного формального параметра. Якщо вони не співпадають, то або виконується перетворення типів, або формується повідомлення про помилку. Число виразів в списку виразів має збігатися з числом формальних параметрів. Якщо в прототипі функції зазначено, що їй не потрібні параметри, а при виклику вони вказані, формується повідомлення про помилку.

2. Відбувається присвоювання значень фактичних параметрів відповідним формальним параметрам.

3. Управління передається першому оператор функції. Далі виконання тіла функції відбувається оператор за оператором, згідно правилам мови C.

4. Виконання оператора **return** в тілі функції повертає управління і можливо, значення в функцію що викликає. При відсутності оператора **return** управління повертається після виконання останнього оператора тіла функції, а значення що повертається не визначено.

Адресний вираз, що стоїть перед дужками визначає адресу функції що викликається. Це означає, що функція може бути викликана через вказівник на функцію.

Приклад:

```
int (*fun)(int x, int *y);
```

Тут оголошена змінна *fun* як вказівник на функцію з двома параметрами: типу **int** та вказівником на **int**. Сама функція повинна повертати значення типу **int**. Круглі дужки, що містять ім'я вказівника *fun* і ознака вказівника *, обов'язкові, інакше запис

```
int *fun (intx,int *y);
```

буде інтерпретуватися як оголошення функції *fun* яка повертає вказівник на **int**.

Виклик такої функції можливий тільки після ініціалізації значення вказівника *fun* і має вигляд:

```
(*fun)(i, &j);
```

У цьому виразі для отримання адреси функції, на яку посилається вказівник *fun* використовується операція непрямого доступу *.

Вказівник на функцію може бути переданий в якості параметра функції. При цьому непрямий доступ відбувається під час виклику функції, на яку посилається

вказівник на функцію. Присвоїти значення вказівника на функцію можна в операторі присвоювання, вживши ім'я функції без списку параметрів.

Будь-яка функція в програмі на мові C може бути викликана рекурсивно, тобто вона може викликати саму себе. Компілятор допускає будь-яке число рекурсивних викликів. При кожному виклику для формальних параметрів і локальних змінних виділяється нова область пам'яті в стеку, так що їх значення з попередніх викликів не втрачаються, але в кожен момент часу доступні тільки значення поточного виклику.

Змінні, оголошені з класом пам'яті **static**, не вимагають виділення нової області пам'яті в стеку при кожному рекурсивному виклику функції та їх значення доступні протягом усього часу виконання програми.

Класичний приклад рекурсії - це математичне визначення факторіала $n!$:

$$n! = 1 \text{ при } n=0;$$
$$n*(n-1)! \text{ при } n>1 .$$

Функція, що обчислює факторіал, матиме наступний вигляд:

```
long fact(int n)
{
    return ( (n==1) ? 1 : n*fact(n-1) );
}
```

Хоча компілятор мови C не обмежує число рекурсивних викликів функцій, це число обмежується ресурсом пам'яті комп'ютера і при дуже великому числі рекурсивних викликів може статися переповнення стека.

Робоче завдання

- Скласти програму обчислення значень функції $\alpha(x, y)$ при заданих x, y . Для обчислення $\alpha(x, y)$ та $f(\xi, \zeta)$ створити **окремі універсальні функції** користувача. Кожна із функцій повинна повертати одне значення і приймати в якості параметрів два значення.
- В функції *main()* викликати функцію $\alpha(x, y)$ із параметрами, зазначеними в останніх двох колонках таблиці із варіантами. Результат обчислення вивести на екран.

Номер варіанта	Функція $\alpha(x, y)$	Функція $f(\xi, \zeta)$	Значення x	Значення y
1	$\frac{f(x^2, \sqrt{y}) - f(\frac{x+y}{3+x}, \sqrt{x})}{f(x, \sqrt{y+x}) - f(\frac{y-x}{3+y}, \frac{y}{3})}$	$\begin{cases} \sin \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 5}{\zeta^2 - 2}, & \xi \leq 0 \end{cases}$	1	2
2	$\frac{f(y, \sqrt{x-y}) - f(\frac{5y-x}{3y+1}, \frac{y}{4})}{f(\sqrt[3]{x}, \sqrt{y}) - f(\frac{x+4}{3+y}, \sqrt{x})}$	$\begin{cases} \cos \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 2\zeta}{\zeta^3 - 2\xi}, & \xi \leq 0 \end{cases}$	2	1
3	$\frac{f(x, y^2) - f(\frac{\sqrt{ x+y }}{2+y}, \sqrt{x})}{f(x, \frac{x+5}{5y-4}) - f(\frac{x-y}{2+x}, \frac{x}{4})}$	$\begin{cases} \frac{2\xi + 5\sqrt{ \zeta }}{\zeta^2 - \xi^3}, & \xi > 0 \\ \frac{\xi + 1}{\xi - \zeta^2}, & \xi \leq 0 \end{cases}$	1	1.5
4	$\frac{f(y+1, 4x+3y) - f(\frac{5x-2}{3y+4x}, \frac{x}{4})}{f(\sqrt[3]{x} + \sqrt{y}, \frac{x}{y}) - f(\sqrt{x+y}, \frac{2x+3}{3y+1})}$	$\begin{cases} \frac{\ln \sin \xi + 5\sqrt{ \xi + \zeta }}{1 - \xi^4}, & \xi > 0 \\ \frac{4\xi + 3\zeta}{\xi^2 - \zeta}, & \xi \leq 0 \end{cases}$	1	1
5	$\frac{f(\frac{x+1}{ y }, y^3) - f(\frac{\ln x+y }{2x+4y}, \sqrt{x+2y})}{f(\frac{2x+6y}{y-4}, \frac{y+2x}{2y-3}) - f(\frac{x+y}{2+y}, \frac{2x}{4-3y})}$	$\begin{cases} \frac{\sqrt[3]{\xi} + \sqrt{ \zeta }}{2\zeta - 3\xi^2}, & \xi > 0 \\ \frac{\sin \xi + \zeta}{\xi^3 - \cos \zeta^2}, & \xi \leq 0 \end{cases}$	2	2
6	$\frac{f(x^2, \sqrt{y}) - f(\frac{x+y}{3+x}, \sqrt{x})}{f(x, \sqrt{y+x}) - f(\frac{y-x}{3+y}, \frac{y}{3})}$	$\begin{cases} \sin \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 5}{\zeta^2 - 2}, & \xi \leq 0 \end{cases}$	1	2
7	$\frac{f(y, \sqrt{x-y}) - f(\frac{5y-x}{3y+1}, \frac{y}{4})}{f(\sqrt[3]{x}, \sqrt{y}) - f(\frac{x+4}{3+y}, \sqrt{x})}$	$\begin{cases} \cos \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 2\zeta}{\zeta^3 - 2\xi}, & \xi \leq 0 \end{cases}$	2	1
8	$\frac{f(x, y^2) - f(\frac{\sqrt{ x+y }}{2+y}, \sqrt{x})}{f(x, \frac{x+5}{5y-4}) - f(\frac{x-y}{2+x}, \frac{x}{4})}$	$\begin{cases} \frac{2\xi + 5\sqrt{ \zeta }}{\zeta^2 - \xi^3}, & \xi > 0 \\ \frac{\xi + 1}{\xi - \zeta^2}, & \xi \leq 0 \end{cases}$	1	1.5

Номер варіанта	Функція $\alpha(x, y)$	Функція $f(\xi, \zeta)$	Значення x	Значення y
9	$\frac{f(y, 4x+3y) - f(\frac{5x-2}{3y+4x}, \frac{x}{4})}{f(\sqrt[3]{x} + \sqrt{y}, \frac{x}{y}) - f(\sqrt{x}, \frac{2x+3}{3y+1})}$	$\begin{cases} \frac{\ln \sin \xi + 5\sqrt{ \xi + \zeta }}{1 - \xi^4}, & \xi > 0 \\ \frac{4\xi + 3\zeta}{\xi^2 - \zeta}, & \xi \leq 0 \end{cases}$	1	1
10	$\frac{f(\frac{x+1}{ y }, y^3) - f(\frac{\ln x+y }{2x+4y}, \sqrt{x+2y})}{f(\frac{2x+6y}{y-4}, \frac{y+2x}{2y-3}) - f(\frac{x+y}{2+y}, \frac{2x}{4-3y})}$	$\begin{cases} \sqrt[3]{\xi} + \sqrt{ \zeta }, & \xi > 0 \\ \frac{2\zeta - 3\xi^2}{\xi^3 - \cos\zeta^2}, & \xi \leq 0 \end{cases}$	2	2
11	$\frac{f(x^2, \sqrt{y}) - f(\frac{x+y}{3+x}, \sqrt{x})}{f(x, \sqrt{y+x}) - f(\frac{y-x}{3+y}, \frac{y}{3})}$	$\begin{cases} \sin \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 5}{\zeta^2 - 2}, & \xi \leq 0 \end{cases}$	1	2
12	$\frac{f(y, \sqrt{x-y}) - f(\frac{5y-x}{3y+1}, \frac{y}{4})}{f(\sqrt[3]{x}, \sqrt{y}) - f(\frac{x+4}{3+y}, \sqrt{x})}$	$\begin{cases} \cos \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 2\zeta}{\zeta^3 - 2\xi}, & \xi \leq 0 \end{cases}$	2	1
13	$\frac{f(x, y^2) - f(\frac{\sqrt{ x+y }}{2+y}, \sqrt{x})}{f(x, \frac{x+5}{5y-4}) - f(\frac{x-y}{2+x}, \frac{x}{4})}$	$\begin{cases} \frac{2\xi + 5\sqrt{ \zeta }}{\zeta^2 - \xi^3}, & \xi > 0 \\ \frac{\xi + 1}{\xi - \zeta^2}, & \xi \leq 0 \end{cases}$	1	1.5
14	$\frac{f(y, 4x+3y) - f(\frac{5x-2}{3y+4x}, \frac{x}{4})}{f(\sqrt[3]{x} + \sqrt{y}, \frac{x}{y}) - f(\sqrt{x}, \frac{2x+3}{3y+1})}$	$\begin{cases} \frac{\ln \sin \frac{\xi}{2} + 5\sqrt{ \xi + \zeta }}{2 - \xi^4}, & \xi > 0 \\ \frac{4\xi + 3\zeta}{\xi^2 - \zeta}, & \xi \leq 0 \end{cases}$	1	1
15	$\frac{f(\frac{x+1}{ y }, y^3) - f(\frac{\ln x+y }{2x+4y}, \sqrt{x+2y})}{f(\frac{2x+6y}{y-4}, \frac{y+2x}{2y-3}) - f(\frac{x+y}{2+y}, \frac{2x}{4-3y})}$	$\begin{cases} \sqrt[3]{\xi} + \sqrt{ \zeta }, & \xi > 0 \\ \frac{\sin \xi + \zeta}{\xi^3 - \cos\zeta^2}, & \xi \leq 0 \end{cases}$	2	2
16	$\frac{f(x^2, \sqrt{y}) - f(\frac{x+y}{3+x}, \sqrt{x})}{f(x, \sqrt{y+x}) - f(\frac{y-x}{3+y}, \frac{y}{3})}$	$\begin{cases} \sin \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 5}{\zeta^2 - 2}, & \xi \leq 0 \end{cases}$	1	2
17	$\frac{f(y, \sqrt{x-y}) - f(\frac{5y-x}{3y+1}, \frac{y}{4})}{f(\sqrt[3]{x}, \sqrt{y}) - f(\frac{x+4}{3+y}, \sqrt{x})}$	$\begin{cases} \cos \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 2\zeta}{\zeta^3 - 2\xi}, & \xi \leq 0 \end{cases}$	2	1
18	$\frac{f(x, y^2) - f(\frac{\sqrt{ x+y }}{2+y}, \sqrt{x})}{f(x, \frac{x+5}{5y-4}) - f(\frac{x-y}{2+x}, \frac{x}{4})}$	$\begin{cases} \frac{2\xi + 5\sqrt{ \zeta }}{\zeta^2 - \xi^3}, & \xi > 0 \\ \frac{\xi + 1}{\xi - \zeta^2}, & \xi \leq 0 \end{cases}$	1	1.5
19	$\frac{f(\frac{y}{2}, 4x+3y) - f(\frac{5x-2}{3y+4x}, \frac{x}{4})}{f(\sqrt[3]{x} + \sqrt{y}, \frac{x}{y}) - f(\sqrt{x^2 + y^2}, \frac{2x+3}{3y+1})}$	$\begin{cases} \frac{\ln \sin \xi + 5\sqrt{ \xi + \zeta }}{1 - \xi^4}, & \xi > 0 \\ \frac{4\xi + 3\zeta}{\xi^2 - \zeta}, & \xi \leq 0 \end{cases}$	1	1

Номер варіанта	Функція $\alpha(x, y)$	Функція $f(\xi, \zeta)$	Значення x	Значення y
20	$\frac{f\left(\frac{x+1}{ y }, y^3\right) - f\left(\frac{\ln x+y }{2x+4y}, \sqrt{x+2y}\right)}{f\left(\frac{2x+6y}{y-4}, \frac{y+2x}{2y-3}\right) - f\left(\frac{x+y}{2+y}, \frac{2x}{4-3y}\right)}$	$\begin{cases} \sqrt[3]{\xi} + \sqrt{ \zeta }, & \xi > 0 \\ \frac{\sin \xi + \zeta}{\xi^3 - \cos \zeta^2}, & \xi \leq 0 \end{cases}$	2	2
21	$\frac{f(x^2, \sqrt{y}) - f\left(\frac{x+y}{3+x}, \sqrt{x}\right)}{f(x, \sqrt{y+x}) - f\left(\frac{y-x}{3+y}, \frac{y}{3}\right)}$	$\begin{cases} \sin \xi + \ln \zeta , & \xi > 0 \\ \frac{\xi + 5}{\zeta^2 - 2}, & \xi \leq 0 \end{cases}$	1	2

Контрольні питання

1. Дайте визначення функції в мові C?
2. Чи можуть функції мови C бути вкладеними?
3. Для чого служать оголошення, визначення і виклик функції?
4. Змінні, оголошені в тілі функції, є локальними або глобальними?
5. Чи доступний рекурсивний виклик для функцій мови C?
6. Чи є обмеження на кількість рекурсивних викликів в мові C?