Лабораторна робота №5. Функції

***Тема.* Функції. Варіативні функції.**

Мета. Придбати практичні навички щодо розроблення програм з використанням функцій.

Загальні відомості.

Функція – це синтаксично виділений іменований програмний модуль, що виконує певну дію або групу дій. Кожна функція має свій інтерфейс і реалізацію (визначення).

Інтерфейс функції – заголовок функції, у якому вказується назва функції, список її параметрів і тип значення, що повертається.

Прототип функції – це оголошення функції, але не її визначення.

Визначення (опис, реалізація) функції – це програмний код, який реалізує розроблений для даної функції алгоритм.

Не можна визначати будь-яку функцію в тілі іншої функції. У мові С є два типи функцій:

- які не повертають значень;
- що повертають значення.

Функції, що не повертають значення по завершенню роботи, мають таке визначення:

```
void ім'я функції(параметри функції)        // заголовок функції
                {
            // тіло функції
                }
```

Наприклад:

```
void printArr(*int, int);
```

Якщо потрібно функції передавати якісь дані, то усередині круглих дужок оголошуються параметри функції, які відділяються один від одного комами.

Функції, що повертають значення по завершенню своєї роботи, визначають у такий спосіб:

```
тип даних, що повертаються ім'я функції (параметри функції)
                {
            // тіло функції
        return  значення, що повертається;
                }
```

Наприклад:

```
int max (int, int);
float inputMassa();
```

return – оператор, який закінчує виконання функції. Він повертає виконання у ту функцію, з якої був виклик; управління передається наступному оператору за оператором виклику. Формат оператора такий;

```
return [<вираз>];
```

При виконанні *return* обчислюється значення виразу, якщо він є, приводиться до типу, який оголошений у функції, і повертається у ту функцію, з якої був виклик. У разі, коли вираз відсутній, значення, що повертається функцією, не визначено.

У С існує можливість помістити оголошення функцій в окремий файл; тоді такий

файл із функціями необхідно буде підключати, як у випадку з підключенням стандартних заголовних файлів. Є два способи розміщення функцій:

– створення файлу типу *.c, у якім оголошуються та визначаються функції;
– створення файлів типу *.c (для визначення) і *.h (для оголошення функцій).

Переважним стилем програмування вважається другий спосіб.

Функції дозволяють зробити програму модульною, тобто розділити її на кілька функцій, які в сукупності виконують поставлене завдання. Ще один величезний плюс функцій у тому, що їх можна багаторазово використовувати.

Функція виконується в момент її виклику. Після оголошення до функції можна звертатися у програмі по імені. Якщо функція не повертає жодного результату, тобто оголошена як void, її виклик не може бути використаний як операнд більш складного виразу (наприклад, значення такої функції не можна привласнити будь-якій змінній). Виклик функції можна використовувати як складову частину більш складного виразу, якщо вона повертає результат.

Прототип указують у тих випадках, коли функція описується пізніше свого використання. Наприклад, можна оголосити функцію до *main*, викликати її з *main*, але описати тільки після *main*.

Формальні й фактичні параметри. Формальні параметри існують у прототипі і тілі визначення функції. Вони задаються деякими унікальними іменами і усередині функції доступні як локальні змінні.

Фактичні параметри існують в основній програмі. Вони вказуються при виклику функції на місці формальних. У момент виклику функції значення фактичних параметрів привласнюються формальним.

Черговість виклику та рекурсія. Одна функція викликається усередині іншої. Зокрема, усередині свого тіла функція може викликати саму себе. Таке явище називається рекурсією, а така функція – рекурсивною.

Способи передачі параметрів у функцію.

1) *Передача параметрів за значенням.* При виклику функції значення фактичного параметра копіюється в локальну змінну, доступну як формальний параметр усередині функції.

Передача параметрів за значенням має такі обмеження:

– з тіла функції не можна звернутися до будь-якого об’єкта, якщо він не є глобальним або якщо його ім’я перекрите однойменною локальною змінною;
– при передачі об’єктів виконується їх копіювання, як наслідок, у випадку великих об’єктів витрачається багато пам’яті.

2) *Передача параметрів по посиланню.* У цих випадках у функцію передається адреса об’єкта і, відповідно, робота усередині функції відбувається не з копією, а з оригіналом об’єкта.

Щоб параметр передавався по посиланню, досить у прототипі функції поставити знак & після типу параметра.

Наприклад, є функція:

```
void func1(int val, int& ref)
{
val++;
ref++;
}
```

В іншій функції є таке:

```
int a = 10, b = 10;
func1(a,b);
```

// a = 10, значення буде збільшено, але усередині функції, яклокальне
// b = 11, буде збільшене значення зовнішньої змінної b

При цьому, навіть якщо імена формального і фактичного параметрів будуть однакові, жодної проблеми не виникне.

Варіативні функції – це функції зі змінною кількістю аргументів. У оголошенні та визначенні такої функції змінне число аргументів задається трьома крапками, обов'язково наприкінці списку формальних параметрів.

При цьому, для коректної роботи функції рекомендується, щоб перший параметр задавав кількість фактично переданих аргументів та/або їх типи (наприклад, як у функції *printf()*).

Для реалізації функцій зі змінною кількістю аргументів у мові програмування С потрібно підключити заголовний файл *stdarg.h*., для С++ – *cstdarg*.

Основне завдання.

Створити функцію яка виконує обчислення у відповідності до індивідуального завдання. Продемонструвати роботу функції на декількох наборах вхідних даних (2..5).

Індивідуальні завдання.

1. Визначити факторіал заданого числа.
2. Підрахувати суму чисел у заданому діапазоні. Наприклад, при вхідних даних 50 та 52 повинно бути 50+51+52 = 153. Обробити ситуацію, коли нижня границя більше, ніж верхня.
3. Підрахувати добуток чисел у заданому діапазоні. Наприклад, при вхідних даних 50 та 52 повинно бути 50*51*52 = 132600. Обробити ситуацію, коли нижня границя більше, ніж верхня.
4. Підвести число ***n*** у ступінь ***m***.
5. Визначити суму розрядів заданого числа.
6. Визначити, чи є задане число простим.
7. Знайти значення максимального числа з заданих, використовуючи функцію з варіативною кількістю аргументів.
8. Знайти значення мінімального числа з заданих, використовуючи функцію з варіативною кількістю аргументів.
9. Визначити, чи є задане число досконалим (таким, що дорівнює сумі своїх дільників). Наприклад, 6 = 1+2+3 –> досконале число.
10. Визначити, скільки разів зустрічається задана цифра в заданому числі. Наприклад, в числі 1234231 цифра 3 зустрічається 2 рази, цифра 4 – 1 раз, цифра 5 – 0 разів. Обробити ситуацію, коли введений символ не є цифрою (не знаходиться в діапазоні 0...9).
11. Визначити об'єм прибутку при вкладі заданої суми грошей ***x*** у банк під встановлений процент річних ***y*** за ***n*** років.
12. Отримати заголовний еквівалент переданого символу, якщо символ – риса літери. Наприклад, 'a' -> 'A', 'X' -> 'X'.
13. Отримати рядковий еквівалент переданого символу, якщо цей символ – заголовна буква. Наприклад, 'A' -> 'a', 'x' -> 'X'.
14. Отримати число, що є дзеркальним відображенням заданого. Наприклад, при вхідному числі 1234 має повернутися 4321, а для числа 12305-> 50321.
15. Визначити, до якого століття належить заданий рік.
16. Знайти значення ***n***-го елементу арифметичної прогресії, при заданих значеннях

початкового значення та кроку прогресії.

17. Знайти значення ***n***-го елементу геометричної прогресії, при заданих значеннях начального значення та шагу прогресії.

18. Визначити подвійний факторіал заданого числа.

19. Визначити, чи є задане число довершеним. (Довершене число таке, що дорівнює сумі своїх дільників).

Контрольні питання.

1. Чому при передачі параметра за значенням усі зміни параметра у функції не відбиваються на значенні аргументу?
2. Скільки операторів *return* може бути в тілі функції?
3. Що таке «прототип функції», яке його призначення?
4. Як визначити список параметрів функції змінної довжини? Наведіть приклад.
5. Як описати функцію, яка не має значень, що повертаються?
6. Як описати, що функція не має аргументів?
7. Наведіть приклад функції, яка не має аргументів та нічого не повертає.
8. Які функції називаються варіативними?
9. Як описати функцію, яка має параметри за замовчуванням?
10. Скільки параметрів за замовчуванням може мати функція?