Лабораторная работа №10.

**Тема работы:** Разработка и отладка алгоритмов и программ с использованием функций

**Цель работы:** Сформировать умения разрабатывать и тестировать программы с использованием функций

**Оснащение рабочего места:** ПК, VS, методические указания для проведения лабораторных работ, индивидуальные задания на ЛР

**Краткие теоретические сведения**

Как и любой объект программы на языке Си, пользовательские функции необходимо декларировать. Объявление функции пользователя, т.е. ее декларация, выполняется в двух формах – в форме описания (объявления) и в форме определения, т.е. любая пользовательская функция должна быть объявлена и определена.

Описанием функции является декларация ее прототипа, который сообщает компилятору о том, что далее будет приведено ее полное определение (текст), т.е. реализация.

Объявление функции (прототип, заголовок) задает ее свойства – идентификатор, тип возвращаемого значения (если такое имеется), количество и типы параметров.

В стандарте языка используется следующий формат декларации (объявления) функций:

*тип\_результата* *ID\_функции* (*список*);

В *списке* перечисляются типы параметров данной функции, причем идентификаторы переменных в круглых скобках прототипа указывать необязательно, т.к. компилятор языка их не обрабатывает.

Описание прототипа дает возможность компилятору проверить соответствие типов и количества параметров при фактическом вызове этой функции.

Пример объявления функции *fun*, которая имеет три параметра типа *int*, один параметр типа *double* и возвращает результат типа *double*:

double fun(int, int, int, double);

Каждая функция, вызываемая в программе, должна быть определена (только один раз). *Определение функции* – это ее полный текст, включающий заголовок и код.

Полное определение (реализация) функции имеет следующий вид:

*тип\_результата ID\_функции(список параметров)*

{

код функции

*return выражение*;

}

Рассмотрим составные части определения пользовательской функции.

*Тип результата* определяет тип *выражения*, значение которого возвращается в точку ее вызова при помощи оператора *return* *выражение*; (возврат). *Выражение* преобразуется к *типу\_результата*, указанному в заголовке функции и передается в точку вызова. Тип возвращаемого функцией значения может быть любым базовым типом, а также указателем на массив или функцию. Если функция не должна возвращать значение, указывается тип *void*. В данном случае оператор *return* можно не ставить. Из функции, которая не описана как *void*, необходимо возвращать значение, используя оператор *return*. Если тип функции не указан, то по умолчанию устанавливается тип *int*.

*Список параметров* состоит из перечня типов и идентификаторов параметров, разделенных запятыми. Список параметров определяет объекты, которые требуется передать в функцию при ее вызове.

В определении и в объявлении одной и той же функции типы и порядок следования параметров должны совпадать. Тип возвращаемого значения и типы параметров совместно определяют тип функции.

Функция может не иметь параметров, но круглые скобки необходимы в любом случае. Если у функции отсутствует список параметров, то при декларации такой функции желательно в круглых скобках указать *void*. Например, *void main*(*void*){ ... }.

В функции может быть несколько операторов *return*, но может и не быть ни одного (тип *void* – это определяется потребностями алгоритма). В последнем случае возврат в вызывающую программу происходит после выполнения последнего оператора кода функции.

Пример функции, определяющей наименьшее значение из двух целочисленных переменных:

int min (int x, int y)

{return (x<y) ? x : y;}

Функции, возвращающие значение, желательно использовать в правой части выражений языка Си, иначе возвращаемый результат будет утерян.

В языке Си каждая функция – это отдельный блок программы, вход в который возможен только через вызов данной функции.

Для вызова функции в простейшем случае нужно указать ее имя, за которым в круглых скобках через запятую перечислить список передаваемых ей *аргументов*. Вызов функции может находиться в любом месте программы, где по синтаксису допустимо выражение того типа, который формирует функция.

Простейший вызов функции имеет следующий формат:

*ID\_функции* (*список аргументов*);

где в качестве аргументов можно использовать константы, переменные, выражения (их значения перед вызовом функции будут определены компилятором).

**Порядок выполнения работы**

1. Ознакомиться с краткими теоретическими сведениями
2. Ознакомиться с поставленной задачей на ЛР

Задания (выполнить все)

1. Написать функцию, которая вычисляет объем цилиндра. Ее параметрами должны быть радиус(R) и высота(H)
2. Написать функцию, которая ищет максимальную цифру в записи данного натурального числа
3. Дано два массива. Вывести как результат тот массив, у которого сумма элементов больше. При решении создать функцию для нахождения суммы массива.
4. Решить программным способом поставленную задачу
5. Составить отчёт по выполненной ЛР

**Содержание отчета:**

1. Титульный лист (см. образец отчёта)
2. Наименование и цель работы.
3. Задание на ЛР
4. Скопированный код из VS.
5. Скриншот выполненной программы
6. Ответы на контрольные вопросы.
7. Выводы по работе.

**Контрольные вопросы**

1. Правила организации функций.
2. Описать в общем виде функцию, правила вызова и списки параметров

**Литература**

Прата С. Язык программирования С++. Лекции и упражнения, 7-е изд.: Пер. с анг.