**Лабораторная Работа № 28-30**

**Тема:**  Программирование задач с использованием функций пользователя.

**Цель:** Овладение навыками алгоритмизации и программирования задач с использованием подпрограмм пользователя. Овладение навыками написания подпрограмм и обращения к ним, выбор параметров подпрограмм.

**Теоретические сведения:**

Приступая к решению задач на предложенную тему, следует вспомнить, что:

* Функцией называется выделенная последовательность инструкций, предназначенных для решения определенной задачи. Вы можете написать свои функции и затем использовать их так же, как функции из библиотек языка С или С++.
* Собственные функции помещаются после закрывающей фигурной скобки main(). Каждая функция имеет структуру, аналогичную структуре функции main().
* Функцию можно вызвать из любого места программы, в том числе, из другой функции. Инструкция return(0); в теле функции всегда передает управление инструкции, следующей за вызывом функции.
* Если компилятор С следует стандарту ANSI, вы должны указать тип функции даже в том случае, если функция не возвращает никаких значений. При этом тип определяется как void. Эта информация говорит компилятору, что функция не будет возвращать вызывающей функции никакого значения. В С++ введение ключевого слова void является не обязательным, но желательным.
* Для передачи данных в функцию надо использовать только параметры. Глобальные переменные, то есть переменные, объявленные вне функции, использовать не рекомендуется.
* Тип каждого фактического параметра (константы или переменной) в инструкции вызова функции должен совпадать с типом соответствующего формального параметра, указанного в объявлении функции.
* Если параметр функции используется для возврата результата, то в объявлении функции этот параметр должен быть ссылкой, а в инструкции вызова функции в качестве фактического параметра должен быть указан адрес переменной.
* При работе с компиляторами языка С++(и некоторыми С - компиляторами) желательно начинать программу с прототипов функций. Прототип – это строка определения функции, повторенная в начале программы перед main(). Прототип сообщает компилятору типы и количество аргументов тех функций, которые будут использованы в программе.

**Методические рекомендации по разработке программ, содержащих подпрограммы пользователя:**

1.При разработке программ с большим количеством подпрограмм всегда стремитесь уменьшить число глобальных переменных до минимума, давайте им осмысленные имена и держите каждую из них под постоянным контролем.

2. Все рабочие переменные, которые используются в подпрограмме для реализации её алгоритма, описывайте как локальные; при этом имена переменных могут быть любые, но желательно не совпадающие с именами глобальных переменных.

3. Если планируется использовать подпрограмму в нескольких разных программах, то в ней вообще не должны использоваться глобальные переменные, то есть подпрограмма должна быть полностью автономной и должна обмениваться данными с основной программой только через параметры.

4. Избегайте вложенных подпрограмм, так как при их использовании вероятность случайно изменить значение переменной резко возрастает.

**Задание:**

Написать программу, осуществляющую опрос школьника по простейшим вычислениям (сложение, вычитание в рамках однозначных чисел). Требования к программе следующие:

1. Форма предложения по прохождению теста выбирается студентом.

2. Школьнику предлагается пройти тест из десяти примеров: выдается на экран задание-пример на который надо дать ответ. Если ответ неправильный, то школьнику дается три попытки на введение верного ответа. Если правильный ответ так и не дан, то предлагается следующий пример. Все три попытки (успешные или нет) засчитываются как один ответ из десяти.

3. По окончанию проверки знаний по «всем» тестам школьнику выставляется оценка.

**Примечание:**

Данное задание является объединяющим по таким темам дисциплины как:

1. Циклические конструкции.
2. Оператор выбора.
3. Функции пользователя.

l.

**По данной лабораторной работе отчет не требуется!**