**Практическая работа 2**

**Цель**: выполнить задания, изучить построение циклов и функций.

**Основные сведения о функциях**

Функция – подпрограмма, которая организовывает тот или иной алгоритмический процесс и возвращает некоторое значение. Программа на языке Си имеет по крайней мере одну функцию main(), которая автоматически вызывается при запуске программы. Сама функция main() может вызывать другие функции, а те, в свою очередь, сами могут вызывать те или иные функции.

Когда имя функции встречается в программе, то управление передается к телу данной функции, то есть осуществляется вызов функции. При вызове функции ей при помощи аргументов (формальных параметров) передаются некоторые значения (фактические параметры), которые используются функцией во время ее работы. Любой аргумент функции может быть константой, переменной или выражением. Общий формат определения функции имеет следующий вид:

*тип\_возвращаемого\_значения имя\_функции (список\_параметров)*

*{*

*объявления*

*операторы*

*}*

Тип возвращаемого значения объявляет тип значения, возвращаемого функцией. Если же функция не возвращает значения, тогда в качестве типа возвращаемого значения выступает пустой тип – void. Если функция не получает аргументы, список параметров также объявляется как void. Функции могут возвращать значения. После обращения к функции она выполняет некоторые действия и в качестве результата своей работы может возвратить некоторое значение. С помощью инструкции return происходит возврат результата от вызываемой функции к вызывающей. После слова return может следовать любое выражение:

*return выражение;*

Выражение, если это необходимо, приводится к возвращаемому типу функции.

Операторов return в функции может быть несколько. Тогда данные операторы будут фиксировать соответствующие точки выхода.

Прототип функции сообщает компилятору тип возвращаемого функцией значения, количество параметров, получаемых функцией, а также типы параметров и порядок их следования. Компилятор осуществляет проверку соответствия типов передаваемых фактических параметров типам формальных параметров, то есть компилятор использует прототип функции для проверки правильности вызовов данной функции. Прототип функции имеет такой же вид, что и определение функции, с той лишь разницей, что тело функции отсутствует, причем имена формальных параметров также могут быть опущены. Если прототип функции не задан, а встретился вызов функции, то строится неявный прототип из анализа формы вызова функции.

Например, в нижеприведенной программе, которая запрашивает два числа a и b, после чего выводит на экран середину отрезка [a,b], используется функция Middle(). Прототип данной функции имеет следующий вид:

*double Middle(double, double);*

Данный прототип указывает, что функция Middle() имеет два аргумента типа double и возвращает результат типа double.

|  |
| --- |
| double Middle(double, double); /\* прототип функции \*/  int main()  {  double a, b;  scanf("%lf%lf", &a, &b);  printf("middle = %f\n", Middle(a,b));  return 0; }  /\* описание функции Middle() \*/  double Middle(double a, double b)  {  return (a + b) / 2;  } |

**Задание**

Пользователь вводит число N. Программа должна выписать данные о числе N по этим задачам:

1. Положительное или отрицательное число (если равно 0, то выписать что число является ни отрицательным, ни положительным);
2. Все кратные числа;
3. Количество цифр;
4. Среднее значение цифр;
5. Поменять порядок цифр на обратный;

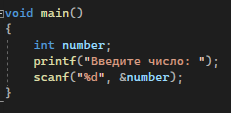
**Для каждой задачи создайте отдельную функцию.**

Пример:

|  |
| --- |
| Введите число: 123  1) Число положительное  2) Кратные числа: 1, 3, 41, 123  3) В этом числе 3 цифры  4) Среднее значение цифр – 2  5) Обратный порядок - 321 |

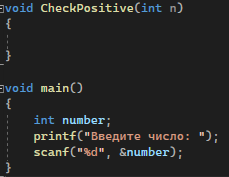
**Создание функций**

Создайте в функции main() переменную и ввод значения в нее:

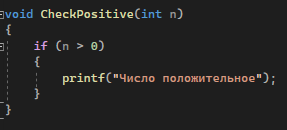


Далее создадим функцию для вычисления положительного или отрицательного числа и передадим нашу переменную в нее, а после нужно вызвать данную функцию.

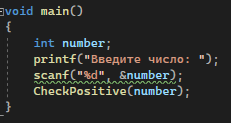
Создаем функцию над main():



Пишем проверку на условие если число положительное:



Вызываем данную функцию в main() и передаем в нее переменную number:



Допишите в функции CheckPositive() условия для нахождения отрицательного числа и 0.