

## **Тема 2: «Использование подпрограмм для обработки простых типов данных»**

*Задание 1 (5 баллов), задание 1 и 2 (10 баллов)*

Требуется разработать программу, которая демонстрирует создание и использование собственной функции. Функция должна получать исходные данные через параметры, вычислять по заданию результат и передавать его в место вызова.

Функция с результатом передает одно вычисленное значение (результат) через оператор `return`. Функция может передавать и несколько результатов через параметры-переменные.

В языке C# для обмена информацией между вызываемой и вызывающей функцией предусмотрено 4 вида параметров:

- ✓ параметры-значения – подставляются при вызове функции, но не изменяются. Обычно используются в качестве входных параметров функции;
- ✓ параметры-переменные – подставляются в виде ссылки со служебным словом `ref`, переданное значение переменной может быть изменено внутри функции;
- ✓ выходные параметры, описываются ключевым словом `out`, используются для передачи результатов;
- ✓ параметры-массивы, описываются с помощью ключевого слова `params`.

### **Вариант 1.**

1. Разработать подпрограммы, позволяющие распознавать и складывать двоичные числа.
2. Используя созданные подпрограммы, определить сумму двух двоичных чисел, если они правильно введены пользователем.

### **Вариант 2.**

1. Разработать подпрограмму, позволяющую распознавать и увеличивать на 1 восьмеричные числа.
2. Используя созданную подпрограмму, ввести с клавиатуры несколько чисел, и те из них, которые являются восьмеричными, увеличить на 1.

### **Вариант 3.**

1. Разработать подпрограмму, для перевода натурального числа из десятичной системы счисления в двоичную.
2. Используя созданную подпрограмму, перевести в двоичную систему числа, вводимые пользователем с клавиатуры, результаты работы сохранить в текстовый файл.

#### **Вариант 4.**

1. Разработать подпрограмму для перевода натурального десятичного числа в восьмеричную систему счисления.
2. Используя созданную подпрограмму, перевести в восьмеричную систему наименьшее из трех введенных пользователем чисел.

#### **Вариант 5.**

1. Разработать подпрограмму для нахождения наибольшего общего делителя двух натуральных чисел.
2. Используя созданную подпрограмму, сократить дробь, заданную в виде числителя  $a$  и знаменателя  $b$ .

#### **Вариант 6.**

1. Разработать функцию, позволяющую распознавать числа-палиндромы («перевертыши»), то есть такие числа, десятичная запись которых одинаково читается слева направо и справа налево.
2. Даны два натуральных многозначных числа. Выяснить, является ли хоть одно из них палиндромом.

#### **Вариант 7.**

1. Разработать подпрограмму для вычисления расстояния между точками, заданными своими координатами. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Задать с клавиатуры координаты 3 точек. Используя подпрограмму, найти пару точек, расстояние между которыми наименьшее.

#### **Вариант 8.**

1. Разработать подпрограмму для вычисления расстояния между точками, заданными своими координатами. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Вычислить периметр многоугольника, заданного координатами своих вершин, используя созданную подпрограмму. Координаты вершин прочитать из текстового файла.

#### **Вариант 9.**

1. Разработать подпрограмму для определения площади треугольника по трем сторонам. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, вычислить площади всех треугольников, которые возможно построить, выбирая значения сторон из четырех заданных отрезков.

**Вариант 10.**

1. Разработать подпрограмму для вычисления корней квадратного уравнения в действительной области. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, вычислить корни всех квадратных уравнений, коэффициенты которых прочитаны из текстового файла.

**Вариант 11.**

1. Разработать подпрограмму, позволяющую распознавать простые числа. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, найти и вывести на экран все простые числа из заданного интервала.

**Вариант 12.**

1. Разработать подпрограмму, вычисляющую сумму цифр натурального многозначного числа (с любым количеством разрядов). Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Даны несколько натуральных многозначных чисел. Используя созданную подпрограмму, определить, в каком из них больше сумма цифр. Числа задать с помощью таблицы.

**Вариант 13.**

1. Разработать подпрограмму, вычисляющую цифровой корень натурального числа. Для получения цифрового корня многозначного числа нужно сложить все цифры этого числа, затем все цифры найденной суммы и так далее, пока не получится однозначное число. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Запросить у пользователя день, месяц и год рождения, и записав все эти данные в виде единого многозначного числа, вычислить его цифровой корень с помощью созданной подпрограммы, и выдать сообщение: «Ваша счастливая цифра ...»

**Вариант 14.**

1. Разработать подпрограмму для перевода натурального числа в двоичную систему счисления. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, перевести в двоичную систему счисления все числа, прочитанные из текстового файла.

**Вариант 15.**

1. Разработать подпрограмму для перевода натурального числа в восьмеричную систему счисления. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, перевести в восьмеричную систему счисления все числа, прочитанные из текстового файла.

**Вариант 16.**

1. Разработать подпрограмму для получения «перевернутого» натурального числа, то есть числа, цифры которого идут в обратном порядке. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, получить для каждого из трех заданных натуральных чисел сумму самого числа и «перевернутого» числа, а также найти минимальную из этих сумм.

**Вариант 17.**

1. Разработать подпрограммы для расчета периметра и площади параллелограмма и равнобедренной трапеции. Проверить работу подпрограмм, организовав их вызов с фактическими параметрами.
2. Выбрать вид фигуры, запросить её измерения и, используя созданные подпрограммы, вычислить периметр и площадь фигуры.

**Вариант 18.**

1. Разработать подпрограмму определения, является ли заданный год високосным. Високосным считается год, значение которого делится на 4, кроме тех годов, которые делятся на 100, но не делятся на 400. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, вывести все високосные года из указанного промежутка.

**Вариант 19.**

1. Разработать подпрограмму, определяющую количество дней в месяце по заданному номеру месяца. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Используя созданную подпрограмму, вычислить дату (день и месяц текущего года), если известно, сколько дней прошло с 1 января до искомой даты.

**Вариант 20.**

1. Разработать подпрограмму, определяющую количество дней в месяце по заданному номеру месяца.
2. Используя созданную подпрограмму, найти дату (день и месяц текущего года), если известно, сколько дней должно пройти от сегодняшней даты до искомой даты.