## Лабораторная работа № 5.

## Основные типы данных и операторы языка *C*#. Методы

При решении задач из этой лабораторной работы необходимо использовать основные типы данных, операции и операторы, изученные на лекциях. При необходимости во всех задачах должна проверяться корректность входных данных и возможные исключительные ситуации.

При решении задач нужно использовать наиболее подходящий вариант оператора цикла. Обратите внимание на то, что если в задании требуется определить, например, содержит ли набор данных какое-ли­бо значение, то при нахождении первого подходящего значения дальнейший поиск продолжать не следует.

* 1. Разработайте консольное приложение, которое при запуске выдает «меню» со списком возможных заданий, а также вариант («8») – закрытие приложения. Пользователь вводит номер задания.

1. В данном натуральном числе переставьте цифры таким образом, чтобы образовалось наибольшее из записанных этими же цифрами число.
2. Найдите все пары дружественных чисел, не превышающих A. Два числа называются дружественными, если каждое из них равно сумме всех делителей другого (само число в качестве делителя не рассматривается). Целое число *A* вводится с клавиатуры. Числа в каждой паре должны стоять по возрастанию.

Например, для *А*= 5000 необходимо вывести пары чисел:

220 284

1184 1210

2620 2924

1. Найдите все автоморфные числа, меньшие *A* (*A* вводится с клавиатуры). Автоморфным называется число, равное последним цифрам своего квадрата (например, 252 = 625). Учитывайте, что автоморфное число может заканчиваться цифрами 1, 5 или 6, а не проверяйте все числа подряд.
2. Определите, можно ли сложить несколько подряд идущих натуральных чисел (не обязательно начиная с единицы), чтобы получить число *A* (*A* вводится с клавиатуры). Если это возможно, то выведите, из каких чисел получается такая сумма (один вариант).
3. Пользователь вводит натуральное число *A*. Выясните, имеются ли среди чисел от *A* до 2 × *A* близнецы, т. е. простые числа, разность между которыми равна двум.
4. Используя рекурсию, выведите латинский алфавит следующим образом:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| А | B | C | D | E | … | V | W | X | Y | Z |
|  | B | C | D | E | … | V | W | X | Y |  |
|  |  | C | D | E | … | V | W | X |  |  |
|  |  |  |  |  | ... |  |  |  |  |  |
|  |  | C | D | E | … | V | W | X |  |  |
|  | B | C | D | E | … | V | W | X | Y |  |
| А | B | C | D | E | … | V | W | X | Y | Z |

В самом среднем ряду должно выводиться *N* букв (0 < *N ≤* 26, определяется случайным образом), следовательно, количество строк варьируется в зависимости от *N*.

После выполнения каждого задания меню отображается заново.

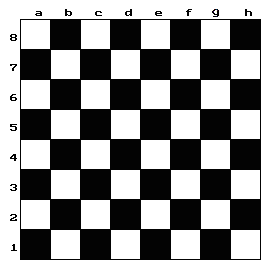
* 1. При решении задачи нужно использовать наиболее подходящий вариант оператора цикла. Создавайте приложение *Windows Forms*. Используйте элементы управления TextBox, Label, Button и др.

На главной форме приложения имеются кнопки «1» – «8», а также кнопка закрытия приложения. При нажатии на номера заданий, загружаются дополнительные формы, на которых выполняется следующее.

1. По форме вверх-вниз должно вставать и садиться солнце, нарисованное из одного круга и лучиков-отрезков. Используйте компонент Timer. При движении вверх солнце должно увеличиваться, а при движении вниз – уменьшаться. При этом в произвольное время справа налево или слева направо «пролетают» овальные облака на случайной скорости случайного размера (от 0 до 5 одновременно).
2. Создайте следующее изображение, расположите его в центре формы.

Входные данные – длина стороны первого квадрата и количество вложенных фигур – вводятся пользователем. Проверьте корректность входных данных.

1. Изобразите на форме шахматную доску из черных и белых квадратов. Не забудьте указать подписи.



1. Нарисуйте график функции *y*= *x* × *sin*(*x*) на плоскости в декартовой системе координат. Должны быть изображены оси со стрелками и подписями, деления (по 5 пунктов), значения по обеим осям от –30 до 30.
2. Составьте программу рисования снежинки. Снежинка состоит из 8 лучиков, каждый из которых в свою очередь оканчивается снежинкой меньшего размера. Уровень вложенности и размер снежинки вводятся пользователем.

Используя метод рисования снежинки с параметрами: уровень вложенности 3, размер – случайное корректное число, реализуйте приложение, в котором при щелчке мыши на форме отображается снежинка. После этого она начинает «падать на землю» (вертикально вниз), пока не достигнет нижнего края формы. Когда снежинка исчезает, пользователь может щелчком мыши создать в любом месте формы новую снежинку, которая так же начинает «падать на землю».

1. Напишите рекурсивную процедуру, с помощью которой рисуется следующая картинка (уровень вложенности и размер первой фигуры определяются константами, цвета на каждом уровне совпадают):

После выполнения каждого задания при закрытии дополнительных форм отображается главная форма. Пока выполняются задания, главная форма недоступна. Программа закрывается при нажатии на кнопку «Закрыть» на главной форме.

***Баллы***: Каждая из двух задач оценивается в 4 балла. Задачи принимаются только целиком.