

```

// 01_ForeverCycleWithBreak1.pas
{ Тема. Обработка последовательностей в цикле. Оператор break
  Задание. Вводятся целые, конец ввода - 0. Сколько целых было введено?
  Решите задачу Series12, используя БЕСКОНЕЧНЫЙ цикл
  repeat
    ...
  until False;
  с выходом из середины, используя оператор break
}
uses PT4;
begin
  Task('Series12');
  // Инициализация
  repeat
    var x := ReadInteger;
    if x=0 then
      break;
    // Обработка x
  until False;
end.

// 01_ForeverCycleWithBreak2.pas
{ Тема. Обработка последовательностей в цикле. Оператор break
  Задание. Вводятся целые, конец ввода - 0. Сколько целых было введено?
  Решите задачу Series12, используя БЕСКОНЕЧНЫЙ цикл
  while True do
  begin
    ...
  end;
  с выходом из середины, используя оператор break
}
uses PT4;
begin
  Task('Series12');
  // Инициализация
  while True do
  begin
    var x := ReadInteger;
    if x=0 then
      break;
    // Обработка x
  end;
end.

// 02_IsPrime.pas
{ Тема. Оператор break. Является ли число простым
  Задание. Дано целое  $x \geq 2$ . Является ли оно простым
  Решение. Вначале считаем, что число простое:
    IsPrime := True
  Проверим делимость x на i=2,3, ... x-1
    Если x делится на i, то
      IsPrime := False
      break
  Замечание. Достаточно проверять делимость на i=2,3, ... Round(Sqrt(x))
  Прислушаться и записать объяснения преподавателя, почему
}
uses PT4;
begin
  Task('While22');
end.

```

```

// 03_Inhabitants1.pas
{ Тема. Обработка последовательностей Соседние значения
  Задание. Дано N целых. Вывести те элементы, которые меньше левого соседа,
    и их количество C
  Решение.
    C := 0
    x := прочитать первый элемент
    Цикл N-1 раз
      Предыдущий := x
      x := прочитать элемент
      Если x < Предыдущий то
        Вывести x
        C += 1
    }
uses PT4;
begin
  Task('Series20');
end.

// 03_Inhabitants2.pas
{ Тема. Обработка последовательностей Соседние значения
  Задание. Дано N целых. Проверить, образуют ли они возрастающую последовательность

  Решение.
    Возрастающая := True
    x := прочитать первый элемент
    Цикл N-1 раз
      Предыдущий := x
      x := прочитать элемент
      Если x < Предыдущий то
        Возрастающая := False
        break
    }
uses PT4;
begin
  Task('Series21');
end.

// 04_Random_Marks.pas
{ Тема. Случайные числа
  Указание. Функция Random(a,b) возвращает случайное целое число от a до b
  Задание 1. Промоделируем оценки ученика, получаемые им в течение четверти
по информатике
  Для этого выведем в цикл 10 случайных оценок от 2 до 5: Random(2,5)
  Указание. Позапускайте программу несколько раз и убедитесь, что каждый раз

    получаются разные последовательности оценок
  }
begin
end.

// 05_Random_Орлы_Решки.pas
{ Тема. Случайные числа
  Задание. Промоделируем 100 бросаний монеты: сколько раз выпадет орёл, а сколько
раз - решка
  Договоримся:
    0 - выпадение орла
    1 - выпадение решки
  Тогда бросание монеты задаётся случайным числом
    Random(0,1)
  Решение. В цикле 100 раз

```

```

        Если Random(0,1)=0, то
            Орлы += 1
        Решки := 100 - Орлы
    Указание. Позапускайте программу много раз.
    Добейтесь того чтобы выпало ровно 50 орлов и 50 решек
}
begin
end.

// 06_Орлы_Решки_50_50.pas
{ Тема. Понятие математического эксперимента
  Задание. В каком проценте случаев при 100 бросаниях выпадет ровно
    50 орлов и 50 решек?
  Указание. Решение предыдущей задачи
  Цикл 100 раз
    Если Random(0,1)=0, то
        Орлы += 1
    Решки := 100 - Орлы
  оаким циклом m раз, где m - большое (например, m=100000):
  Счетчик := 0
  Цикл m раз
    Цикл 100 раз
      Если Random(0,1)=0, то
          Орлы += 1
      Решки := 100 - Орлы
    Если Орлы = 50 то
      Счетчик += 1
  Вывести (Счетчик / m * 100,'%');
}
begin
end.

// 07_Кости.pas
{ Тема. Понятие математического эксперимента. Игра в 101
  Задание. Про моделируем бросание двух игральных костей:
    var (a,b) := Random2(1,6)
  Будем бросать до тех пор, пока сумма очков не станет равной 101
  Если сумма очков станет больше 101, то все очки сгорают
}
begin
end.

```