

Практическая работа 6

Массивы

- 1. Создание и вывод элементов: Создайте массив из 5 целых чисел и выведите их на экран.**

```
fun main() {  
    println(arrayOf(1,2,3,4,5).joinToString())  
}
```

1, 2, 3, 4, 5

- 2. Сумма элементов массива: Напишите программу, которая находит сумму всех элементов массива чисел.**

```
fun main() {  
    print("Введите числа через пробел: ")  
    val nums = readln().split(" ").map { it.toInt() }  
    println("Сумма чисел: ${nums.sum()}")  
}
```

Введите числа через пробел: 56 78 12 1 2 3 4 45 -234
Сумма чисел: -33

- 3. Максимальное и минимальное значение: Создайте массив из 10 чисел, найдите и выведите максимальное и минимальное значение.**

```
fun main() {  
    val nums = Array(10) { (-100..100).random() }  
    println("Массив: ${nums.joinToString()}")  
    println("Максимальное значение: ${nums.max()}")  
    println("Минимальное значение: ${nums.min()}")  
}
```

Массив: -35, 40, 86, 18, 65, -98, -60, 55, -100, 16
Максимальное значение: 86
Минимальное значение: -100

- 4. Сортировка массива: Реализуйте алгоритм сортировки для массива чисел и выведите отсортированный массив.**

```
fun main() {  
    print("Введите размер массива: ")  
    val nums = Array(readln().toInt()) { (-100..100).random() }  
    println("Исходный массив: ${nums.joinToString()}")  
    println("Отсортированный массив: ${nums.sorted().joinToString()}")  
}
```

Введите размер массива: 10
Исходный массив: 17, 89, -17, 9, -20, -83, -80, -31, 82, -77
Отсортированный массив: -83, -80, -77, -31, -20, -17, 9, 17, 82, 89

5. Уникальные элементы: Напишите программу, которая выводит уникальные элементы из массива.

```
fun main() {  
    val base = Array(15) {if (it % 2 == 0) ('A'..'Z').random() else  
(1..100).random()}  
    val arr = base + base  
    println("Исходный массив: ${arr.joinToString()}")  
    println("Уникальные элементы: ${arr.distinct().joinToString()}")  
}
```

Исходный массив: X, 35, Y, 95, W, 68, R, 33, T, 69, X, 24, K, 62, C, X, 35, Y, 95, W, 68, R, 33, T, 69, X, 24, K, 62, C
Уникальные элементы: X, 35, Y, 95, W, 68, R, 33, T, 69, 24, K, 62, C

6. Четные и нечетные числа: Создайте массив и разделите его на четные и нечетные числа, сохранив их в разные массивы.

```
fun main() {  
    val n = Array(15) {(-100..100).random()}  
    val (even, odd) = n.partition { it % 2 == 0 }  
    println("Исходный массив: ${n.joinToString()}")  
    println("Четные элементы: ${even.joinToString()}")  
    println("Нечетные элементы: ${odd.joinToString()}")  
}
```

Исходный массив: 32, -94, 90, -18, -79, -9, -71, -77, -84, 12, 94, 40, 20, 31, 67
Четные элементы: 32, -94, 90, -18, -84, 12, 94, 40, 20
Нечетные элементы: -79, -9, -71, -77, 31, 67

7. Реверс массива: Напишите программу, которая реверсирует массив чисел.

```
fun main() {  
    print("Размер массива: ")  
    val n = Array(readln().toInt()) {(-100..100).random()}  
    println("Исходный массив: ${n.joinToString()}")  
    println("Перевернутый массив: ${n.reversed().joinToString()}")  
}
```

Размер массива: 10
Исходный массив: -68, 67, -96, -37, 86, -45, 66, 44, 25, 78
Перевернутый массив: 78, 25, 44, 66, -45, 86, -37, -96, 67, -68

8. Поиск элемента: Реализуйте поиск элемента в массиве и выводите его индекс.

```
fun main() {  
    print("Введите размер массива: ")  
    val arr = Array(readln().toInt()) {(-100..100).random()}  
    println("Массив: ${arr.joinToString()}")  
    print("Введите элемент: ")  
    println("Индекс элемента: ${arr.indexOf(readln().toInt())}")  
}
```

```
Введите размер массива: 10
Массив: -72, -20, -10, -58, -97, -63, -42, -5, 51, -21
Введите элемент: 51
Индекс элемента: 8
```

9. Копирование массива: Создайте новый массив, скопировав в него элементы из другого массива.

```
fun main() {
    print("Размер массива: ")
    val orig = Array(readln().toInt()) { (-100..100).random() }
    val copy = orig.clone()
    println("Оригинал: ${orig.joinToString()}")
    println("Копия: ${copy.joinToString()}")
}
```

```
Размер массива: 10
Оригинал: 64, -67, -99, -19, 95, -23, 70, 31, -98, 94
Копия: 64, -67, -99, -19, 95, -23, 70, 31, -98, 94
```

10. Сумма четных чисел: Напишите программу, которая находит сумму всех четных чисел в массиве.

```
fun main() {
    print("Введите размер массива: ")
    val n = Array(readln().toInt()) { (-100..100).random() }
    println("Исходный массив: ${n.contentToString()}")
    println("Сумма четных чисел: ${n.filter { it % 2 == 0 }.sum()}")
}
```

```
Введите размер массива: 10
Исходный массив: [10, 66, -3, -91, 96, 37, 38, 19, -46, -81]
Сумма четных чисел: 164
```

11. Пересечение массивов: Напишите программу, которая находит пересечение двух массивов и выводит результат.

```
fun main() {
    print("Размер массива: ")
    val size = readln().toInt()
    val ar1 = Array(size) { it*2 }
    val ar2 = Array(size) { it*3 }
    println("Первый массив: ${ar1.joinToString()}")
    println("Второй массив: ${ar2.joinToString()}")
    println("Пересечение: ${ar1.intersect(ar2.toList()).joinToString()}")
}
```

```
Размер массива: 10
Первый массив: 0, 2, 4, 6, 8, 10, 12, 14, 16, 18
Второй массив: 0, 3, 6, 9, 12, 15, 18, 21, 24, 27
Пересечение: 0, 6, 12, 18
```

12. Перестановка элементов: Реализуйте функцию, которая меняет местами два элемента в массиве.

```
fun main() {
    print("Введите размер массива: ")
    val size = readln().toInt()
    val array = List(size) { (-100..100).random() }
    println("Исходный массив: $array")

    print("Введите индекс первого элемента (от 0 до ${size-1}): ")
    val i = readln().toInt()

    print("Введите индекс второго элемента (от 0 до ${size-1}): ")
    val j = readln().toInt()

    val swapped = array.toMutableList().apply { this[i] = array[j].also { this[j]
= array[i] } }
    println("Измененный массив: $swapped (поменяли элементы с индексами $i и
$j)")
}
```

```
Введите размер массива: 5
Исходный массив: [-86, 24, 75, -79, 4]
Введите индекс первого элемента (от 0 до 4): 1
Введите индекс второго элемента (от 0 до 4): 4
Измененный массив: [-86, 4, 75, -79, 24] (поменяли элементы с индексами 1 и 4)
```

13. Заполнение случайными числами: Создайте массив из 20 случайных чисел от 1 до 100 и выведите его на экран.

```
fun main() {
    val ar = List(20) { (1..100).random() }
    println(ar)
}
[31, 51, 47, 56, 93, 27, 25, 48, 84, 35, 32, 73, 7, 75, 81, 19, 47, 30, 59, 28]
```

14. Числа Прокопенко: Напишите программу, которая выводит все числа в массиве, делящиеся на 3.

```
fun main() {
    println("Введите размер массива: ")
    val s = readln().toInt()
    val ar = List(s) { (-100..100).random() }
    println("Массив: $ar")
    println("Числа, делящиеся на 3: ${ar.filter { it % 3 == 0 }}")
}
```

Введите размер массива:

10

Массив: [-61, -82, -67, 0, -27, 11, 28, 85, -85, -7]

Числа, делящиеся на 3: [0, -27]

15.Проверка на палиндром: Напишите программу, которая проверяет, является ли массив палиндромом.

```
fun main() {  
    println("Введите массив через пробел: ")  
    val ar = readln().split(" ").map{it.toInt()}.toIntArray()  
    println("Массив: ${ar.joinToString()}")  
    println("Является палиндромом: ${ar.toList() == ar.toList().reversed()}")  
}
```

```
Введите массив через пробел:  
1 2 3 2 1  
Массив: 1, 2, 3, 2, 1  
Является палиндромом: true
```

16.Конкатенация двух массивов: Создайте два массива и соедините их в один.

```
fun main() {  
    println("Введите первый массив через пробел: ")  
    val ar1 = readln().trim().split(" ").filter { it.isNotEmpty() }.map {  
it.toInt() }.toIntArray()  
    println("Введите второй массив через пробел: ")  
    val ar2 = readln().trim().split(" ").filter { it.isNotEmpty() }.map {  
it.toInt() }.toIntArray()  
    val resar = ar1 + ar2  
    println("Сумма массивов: ${resar.joinToString(" ")}")  
}
```

```
Введите первый массив через пробел:  
1 2 3  
Введите второй массив через пробел:  
2 3  
Сумма массивов: 1 2 3 2 3
```

17.Сумма и произведение: Напишите программу, которая находит и выводит сумму и произведение всех элементов массива.

```
fun main() {  
    println("Введите массив через пробел: ")  
    val ar = readln().trim().split(" ").filter { it.isNotEmpty() }.map {  
it.toInt() }.toIntArray()  
    val sum = ar.sum()  
    val prod = ar.fold(1L){acc, i -> acc * i}  
    println("Сумма: $sum")  
    println("Произведение: $prod")  
}
```

```
Введите массив через пробел:  
1 2 3 4 5  
Сумма: 15  
Произведение: 120
```

18.Группировка чисел: Разделите массив на группы по 5 элементов и выведите их.

```
fun main() {  
    println("Введите размер массива (должен быть кратен 5): ")  
    val s = readln().toInt()  
    if (s % 5 != 0) {  
        println("Ошибка: размер массива должен быть кратен 5")  
        return  
    }  
    val ar = IntArray(s) { (-100..100).random() }  
    println("Массив: ${ar.joinToString()}")  
    println("\nГруппы по 5 элементов: ")  
    ar.toList().chunked(5).forEachIndexed {index, group ->  
        println("Группа номер ${index + 1}: ${group.joinToString()}")  
    }  
}
```

Введите размер массива (должен быть кратен 5):

15

Массив: 16, -11, -60, 24, -88, -35, -88, -65, -7, 7, 93, -46, 66, 25, 13

Группы по 5 элементов:

Группа номер 1: 16, -11, -60, 24, -88

Группа номер 2: -35, -88, -65, -7, 7

Группа номер 3: 93, -46, 66, 25, 13

19.Слияние двух массивов: Напишите программу, которая сливает два отсортированных массива в один отсортированный массив.

```
import kotlin.random.Random
fun main() {
    println("Введите размер первого массива:")
    val size1 = readln().toInt()
    println("Введите размер второго массива:")
    val size2 = readln().toInt()
    val originalArray1 = IntArray(size1) { Random.nextInt(-100, 101) }
    println("\nПервый исходный массив:")
    println(originalArray1.joinToString())
    val originalArray2 = IntArray(size2) { Random.nextInt(-100, 101) }
    println("\nВторой исходный массив:")
    println(originalArray2.joinToString())
    val array1 = originalArray1.sorted().toIntArray()
    val array2 = originalArray2.sorted().toIntArray()
    val mergedArray = array1 + array2
    mergedArray.sort()
    println("\nПервый отсортированный массив:")
    println(array1.joinToString())
    println("\nВторой отсортированный массив:")
    println(array2.joinToString())
    println("\nСлитый отсортированный массив:")
    println(mergedArray.joinToString())
}
```

```
Введите размер первого массива:
5
Введите размер второго массива:
4

Первый исходный массив:
-75, 82, 59, 25, -42

Второй исходный массив:
-63, -83, 43, -13

Первый отсортированный массив:
-75, -42, 25, 59, 82

Второй отсортированный массив:
-83, -63, -13, 43

Слитый отсортированный массив:
-83, -75, -63, -42, -13, 25, 43, 59, 82
```

20.Числовая последовательность: Создайте массив целых чисел, представляющий арифметическую прогрессию, и выведите его.

```
fun main() {
    println("Введите начальное число: ")
    val start = readln().toInt()
    println("Введите операцию (+, -): ")
    val oper = readln().trim()
    println("Введите шаг: ")
    val step = readln().toInt()
    println("Введите количество элементов: ")
    val count = readln().toInt()
    val prog = IntArray(count)
    for (i in 0 until count) {
        prog[i] = if(oper == "+") start + i * step else start - i * step
    }
    println("Арифметическая прогрессия: ${prog.joinToString()}")
}
```

```
Введите начальное число:
1
Введите операцию (+, -):
+
Введите шаг:
2
Введите количество элементов:
10
Арифметическая прогрессия: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19
```

21.Удаление элемента: Реализуйте функцию, которая удаляет заданный элемент из массива.

```
fun main() {
    println("Введите размер массива: ")
    val s = readln().toInt()
    val ar = IntArray(s) { (-100..100).random() }
    println("Массив: ${ar.joinToString()}")
    println("Введите элемент, который хотите удалить: ")
    val el = readln().toInt()
    val newAr = ar.filter { it != el }.toIntArray()
    println("Новый массив: ${newAr.joinToString()}")
}
```

```
Введите размер массива:
5
Массив: 100, -37, 96, -78, 65
Введите элемент, который хотите удалить:
-78
Новый массив: 100, -37, 96, 65
```


22. Поиск второго максимального: Напишите программу, которая находит второй по величине элемент в массиве.

```
fun main() {
    println("Введите размер массива: ")
    val s = readln().toInt()
    val ar = IntArray(s) { (-100..100).random() }
    println("Массив: ${ar.joinToString()}")
    val sortedAr = ar.sortedDescending()
    var secMax = sortedAr[0]
    for (el in sortedAr) {
        if (el < secMax) {
            secMax = el
            break
        }
    }
    println("Второй максимальный элемент: $secMax")
}
```

Введите размер массива:

20

Массив: 98, 51, -14, 74, -82, 76, -67, -11, 4, 41, -34, 42, -63, 32, 65, -75, 72, 52, 67, -32

Второй максимальный элемент: 76

23. Объединение массивов: Напишите функцию, которая объединяет несколько массивов чисел и выводит результирующий массив.

```
fun main() {
    val allArrays = mutableListOf<IntArray>()
    while (true) {
        println("\nВыберите действие:")
        println("1 - Добавить новый массив")
        println("2 - Показать все введенные массивы")
        println("3 - Объединить массивы и показать результат")
        println("0 - Выход")
        val choice = readln().trim()
        when (choice) {
            "1" -> {
                val array = readArray()
                if (array.isNotEmpty()) {
                    allArrays.add(array)
                    println("Массив добавлен. Всего массивов: ${allArrays.size}")
                }
            }
            "2" -> {
                displayArrays(allArrays)
            }
            "3" -> {
                if (allArrays.isEmpty()) {
                    println("Не введено ни одного массива")
                } else {
                    val resultArray = mergeArrays(allArrays)
                    println("\nРезультат объединения (${resultArray.size} элементов):")
                    println(resultArray.joinToString())
                }
            }
            "0" -> {
            }
        }
    }
}
```

```

        println("Программа завершена")
        return
    }
    else -> {
        println("Неверный выбор. Попробуйте снова.")
    }
}
}
}
fun readArray(): IntArray {
    println("Введите элементы массива через пробел:")
    val input = readln().trim()
    if (input.isEmpty()) {
        println("Введена пустая строка. Массив не добавлен.")
        return IntArray(0)
    }
    return input.split(" ")
        .filter { it.isNotEmpty() }
        .map { it.toInt() }
        .toIntArray()
}
fun displayArrays(arrays: List<IntArray>) {
    if (arrays.isEmpty()) {
        println("Не введено ни одного массива")
        return
    }
    println("\nВведенные массивы:")
    for (i in arrays.indices) {
        println("Массив ${i + 1}: ${arrays[i].joinToString() }")
    }
}
fun mergeArrays(arrays: List<IntArray>): IntArray {
    return arrays.flatMap { it.toList() }.toIntArray()
}
}

```

Выберите действие:

- 1 - Добавить новый массив
- 2 - Показать все введенные массивы
- 3 - Объединить массивы и показать результат
- 0 - Выход

1

Введите элементы массива через пробел:

1 2 3

Массив добавлен. Всего массивов: 1

Выберите действие:

- 1 - Добавить новый массив
- 2 - Показать все введенные массивы
- 3 - Объединить массивы и показать результат
- 0 - Выход

1

Введите элементы массива через пробел:

2 3 4

Массив добавлен. Всего массивов: 2

```

Выберите действие:
1 - Добавить новый массив
2 - Показать все введенные массивы
3 - Объединить массивы и показать результат
0 - Выход
3

Результат объединения (6 элементов):
1, 2, 3, 2, 3, 4

Выберите действие:
1 - Добавить новый массив
2 - Показать все введенные массивы
3 - Объединить массивы и показать результат
0 - Выход
0

Программа завершена

```

24. Транспонирование матрицы: Создайте матрицу (двумерный массив) и напишите программу, которая транспонирует её.

```

fun main() {
    println("Введите количество строк:")
    val rows = readln().toInt()
    println("Введите количество столбцов:")
    val cols = readln().toInt()
    val matrix = Array(rows) { IntArray(cols) }
    println("Введите элементы матрицы:")
    for (i in 0 until rows) {
        println("Строка ${i + 1}:")
        val rowValues = readln().split(" ").map { it.toInt() }
        for (j in 0 until cols) {
            matrix[i][j] = rowValues[j]
        }
    }
    println("\nИсходная матрица:")
    printMatrix(matrix)
    val transposed = transposeMatrix(matrix)
    println("\nТранспонированная матрица:")
    printMatrix(transposed)
}

fun transposeMatrix(matrix: Array<IntArray>): Array<IntArray> {
    val rows = matrix.size
    val cols = matrix[0].size
    val result = Array(cols) { IntArray(rows) }
    for (i in 0 until rows) {
        for (j in 0 until cols) {
            result[j][i] = matrix[i][j]
        }
    }
    return result
}

fun printMatrix(matrix: Array<IntArray>) {
    for (row in matrix) {
        println(row.joinToString(" "))
    }
}

```

```
}  
}
```

```
Введите количество строк:  
4  
Введите количество столбцов:  
4  
Введите элементы матрицы:  
Строка 1:  
1 2 3 4  
Строка 2:  
5 6 7 8  
Строка 3:  
9 10 11 12  
Строка 4:  
13 14 15 16  
  
Исходная матрица:  
1 2 3 4  
5 6 7 8  
9 10 11 12  
13 14 15 16  
  
Транспонированная матрица:  
1 5 9 13  
2 6 10 14  
3 7 11 15  
4 8 12 16
```

25.Линейный поиск: Реализуйте линейный поиск элемента в массиве с возвратомBool-значения (найден или нет).

```
fun main() {  
    println("Введите размер массива:")  
    val size = readln().toInt()  
    val array = IntArray(size) { (-100..100).random() }  
    println("Сгенерированный массив: ${array.joinToString()}")  
    println("Введите элемент для поиска:")  
    val target = readln().toInt()  
    val found = linearSearch(array, target)  
    if (found) {  
        println("Элемент $target найден в массиве")  
    } else {  
        println("Элемент $target не найден в массиве")  
    }  
}  
  
fun linearSearch(array: IntArray, target: Int): Boolean {  
    for (element in array) {  
        if (element == target) {  
            return true  
        }  
    }  
    return false  
}
```

Введите размер массива:

5

Сгенерированный массив: 10, -82, 8, 86, -53

Введите элемент для поиска:

9

Элемент 9 не найден в массиве

Введите размер массива:

5

Сгенерированный массив: -34, 3, -97, 83, -55

Введите элемент для поиска:

3

Элемент 3 найден в массиве

26.Среднее арифметическое: Напишите программу, которая находит среднее арифметическое всех чисел в массиве.

```
fun main() {  
    println("Введите размер массива:")  
    val size = readln().toInt()  
    val array = IntArray(size) { (-100..100).random() }  
    println("Сгенерированный массив: ${array.joinToString()}")  
    val average = array.average()  
    println("Среднее арифметическое: $average")  
}
```

```
Введите размер массива:  
2  
Сгенерированный массив: -31, -62  
Среднее арифметическое: -46.5
```

27.Максимальная последовательность: Найдите максимальную последовательность одинаковых элементов в массиве.

```
fun main() {  
    println("Введите размер массива:")  
    val size = readln().toInt()  
    val array = IntArray(size) { (-10..10).random() }  
    println("Сгенерированный массив: ${array.joinToString()}")  
    var maxCount = 1  
    var currentCount = 1  
    var maxElement = array[0]  
    for (i in 1 until array.size) {  
        if (array[i] == array[i - 1]) {  
            currentCount++  
            if (currentCount > maxCount) {  
                maxCount = currentCount  
                maxElement = array[i]  
            }  
        } else {  
            currentCount = 1  
        }  
    }  
    println("Максимальная последовательность: $maxCount одинаковых элементов ($maxElement)")  
}
```

```
Введите размер массива:  
30  
Сгенерированный массив: 4, 8, -10, 8, -4, 2, 9, 2, 8, -7, 0, -9, 3, 8, -10, 1, -10, -9, 8, -8, -8, 1, 2, 8, -3, -2, 7, -6, 2, 7  
Максимальная последовательность: 2 одинаковых элементов (-8)
```

28.Ввод и вывод массива: Напишите программу, которая запрашивает у пользователя ввод массива чисел и затем выводит его.

```
fun main() {  
    println("Введите элементы массива через пробел:")  
    val array = readln().split(" ").filter { it.isNotEmpty() }.map {  
it.toInt() }.toIntArray()  
    println("Введенный массив: ${array.joinToString()}")  
}
```

```
Введите элементы массива через пробел:  
1 2 3 4 5  
Введенный массив: 1, 2, 3, 4, 5
```

29.Нахождение медианы: Напишите программу, которая находит медиану в массиве.

```
fun main() {  
    println("Введите размер массива: ")  
    val s = readln().toInt()  
    val ar = IntArray(s) { (-100..100).random() }  
    println("Массив: ${ar.joinToString()}")  
    val sortedAr = ar.sorted()  
    println("Отсортированный массив: ${sortedAr.joinToString()}")  
    val median = if (s % 2 == 0) {  
        (sortedAr[s / 2 - 1] + sortedAr[s / 2]) / 2.0  
    } else {  
        sortedAr[s / 2].toDouble()  
    }  
    println("Медиана: $median")  
}
```

```
Введите размер массива:  
4  
Массив: -62, 33, 49, -21  
Отсортированный массив: -62, -21, 33, 49  
Медиана: 6.0
```

30.Распределение по группам: Создайте массив из 100 целых чисел и разделите их на 10 групп по 10 элементов, затем выведите результаты.

```
fun main() {  
    val array = IntArray(100) { (-100..100).random() }  
    println("Массив:")  
    array.forEachIndexed { index, value ->  
        print("$value")  
        if (index < array.size - 1) print(", ")  
        if ((index + 1) % 15 == 0) println()  
    }  
    println()  
  
    println("\nГруппы по 10 элементов:")  
    for (i in 0 until 10) {  
        val startIndex = i * 10  
        val group = array.slice(startIndex until startIndex + 10)  
        println("Группа ${i + 1}: ${group.joinToString()}")  
    }  
}
```

Массив:

61, 5, -55, 70, 66, 57, -87, -45, -81, -31, 40, 96, -31, 41, 40,
79, 88, -23, 22, 63, 88, -92, -39, 57, 66, 84, 57, -29, -94, -75,
-11, 58, -80, -58, -87, -93, 9, 69, -93, -73, -22, -8, 45, -90, -32,
70, -70, 55, 16, 4, 78, 9, -46, -63, -29, -11, 72, -74, -22, -3,
-92, 41, -97, -17, -86, 30, -5, 87, -57, -96, -24, -99, -91, 91, -8,
-35, 36, -70, -75, 25, 14, 56, 38, 3, -84, -42, -9, -63, 37, 67,
43, -34, 86, -80, 94, 80, 71, -97, 73, -93

Группы по 10 элементов:

Группа 1: 61, 5, -55, 70, 66, 57, -87, -45, -81, -31

Группа 2: 40, 96, -31, 41, 40, 79, 88, -23, 22, 63

Группа 3: 88, -92, -39, 57, 66, 84, 57, -29, -94, -75

Группа 4: -11, 58, -80, -58, -87, -93, 9, 69, -93, -73

Группа 5: -22, -8, 45, -90, -32, 70, -70, 55, 16, 4

Группа 6: 78, 9, -46, -63, -29, -11, 72, -74, -22, -3

Группа 7: -92, 41, -97, -17, -86, 30, -5, 87, -57, -96

Группа 8: -24, -99, -91, 91, -8, -35, 36, -70, -75, 25

Группа 9: 14, 56, 38, 3, -84, -42, -9, -63, 37, 67

Группа 10: 43, -34, 86, -80, 94, 80, 71, -97, 73, -93