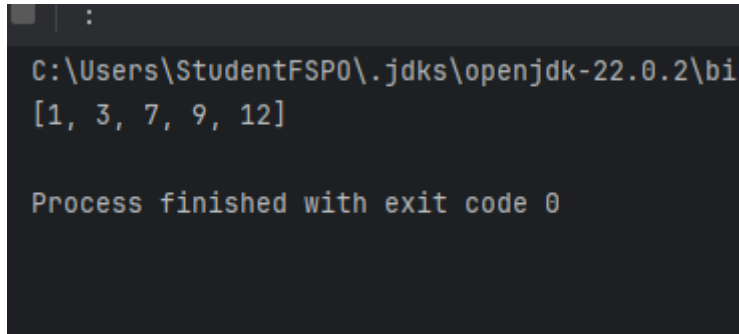


Практическая работа 6

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

```
fun main(){  
    val numbers = arrayOf(1,3,7,9,12)  
    println(numbers.contentToString())  
}
```



```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin  
[1, 3, 7, 9, 12]  
  
Process finished with exit code 0
```

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

```
fun main(){  
    print("Введите количество элементов массива (n): ")  
    val n = readLine()?.toIntOrNull()?:0  
    if (n<=0){  
        println("Некоректное значение n.")  
        return  
    }  
    val numbers = IntArray(n){(0..n).random()}  
    val sum = numbers.sum()  
    println("Массив: ${numbers.contentToString()}")  
    println("Сумма элементов: $sum")  
}
```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe
Введите количество элементов массива (n): 10
Массив: [9, 0, 8, 9, 3, 4, 3, 3, 2, 3]
Сумма элементов: 44

Process finished with exit code 0
```

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

```
fun main(){
    val numbers = IntArray(10){(1..10).random()}
    val max= numbers.maxOrNull():0
    val min= numbers.minOrNull():0
    println("Массив: ${numbers.contentToString()}")
    println("Максимальное значение: $max")
    println("Минимальное значение: $min")
}
```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe "-j
Массив: [8, 3, 4, 6, 8, 9, 6, 1, 7, 3]
Максимальное значение: 9
Минимальное значение: 1

Process finished with exit code 0
```

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

```
fun main(){
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull():0
```

```

if (n<=0){
    println("Некоректное значение n.")
    return
}

val numbers= IntArray(n){(0..n).random()}

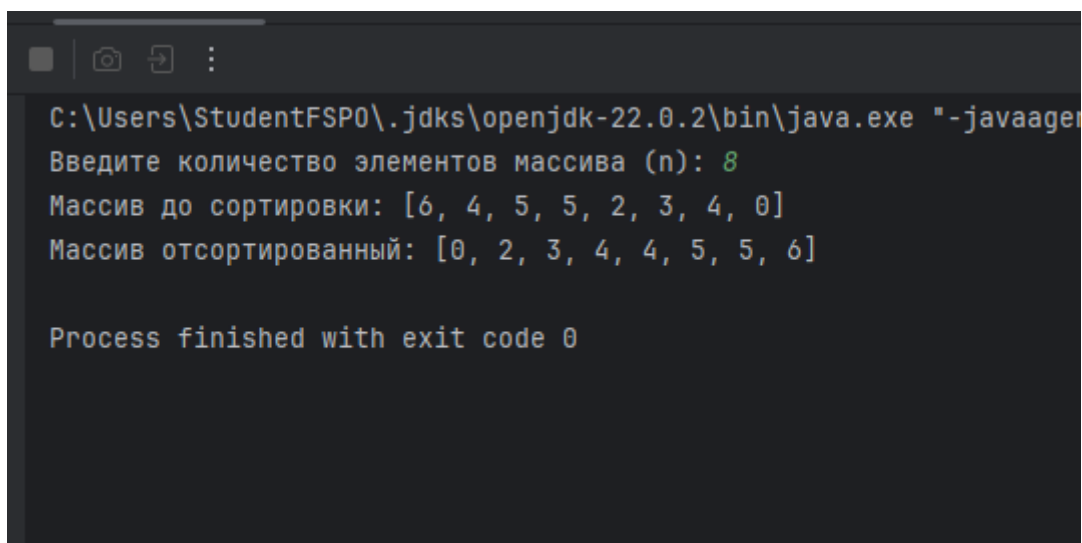
println("Массив до сортировки: ${numbers.contentToString()}")

numbers.sort()

println("Массив отсортированный: ${numbers.contentToString()}")

}

```



```

C:\Users\StudentFSP0\jdk\openjdk-22.0.2\bin\java.exe -javaagent...
Введите количество элементов массива (n): 8
Массив до сортировки: [6, 4, 5, 5, 2, 3, 4, 0]
Массив отсортированный: [0, 2, 3, 4, 4, 5, 5, 6]

Process finished with exit code 0

```

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

```

fun main(){
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull()?:0
    if (n<=0){
        println("Некоректное значение n.")
        return
    }

    val numbers= IntArray(n){(0..n).random()}

    println("Оригинальный массив: ${numbers.contentToString()}")

    val uniqueNumbers = numbers.toSet().toIntArray()

    println("Уникальные элементы: ${uniqueNumbers.contentToString()}")
}

```

```
}
```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe "-javaage
Введите количество элементов массива (n): 10
Оригинальный массив: [10, 6, 1, 2, 4, 7, 8, 7, 5, 3]
Уникальные элементы: [10, 6, 1, 2, 4, 7, 8, 5, 3]

Process finished with exit code 0
```

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

```
fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }

    val numbers = IntArray(n) { (0..n).random() }
    val evenNumbers = numbers.filter { it % 2 == 0 }.toIntArray()
    val oddNumbers = numbers.filter { it % 2 != 0 }.toIntArray()
    println("Ваш массив: ${numbers.contentToString()}")
    println("Четные числа: ${evenNumbers.contentToString()}")
    println("Нечетные числа: ${oddNumbers.contentToString()}")
}
```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe "-javaa
Введите количество элементов массива (n): 10
Ваш массив: [5, 4, 4, 1, 5, 4, 10, 0, 2, 3]
Четные числа: [4, 4, 4, 10, 0, 2]
Нечетные числа: [5, 1, 5, 3]

Process finished with exit code 0
```

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

```
fun main() {  
    print("Введите количество элементов массива (n): ")  
    val n = readLine()?.toIntOrNull() ?: 0  
    if (n <= 0) {  
        println("Некоректное значение n.")  
        return  
    }  
    val numbers = IntArray(n) { (0..n).random() }  
    val reversedNumbers = numbers.reversedArray()  
    println("Массив: ${numbers.contentToString()}")  
    println("Реверсивный массив: ${reversedNumbers.contentToString()}")  
}
```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe "-j  
Введите количество элементов массива (n): 5  
Массив: [5, 0, 4, 1, 5]  
Реверсивный массив: [5, 1, 4, 0, 5]  
  
Process finished with exit code 0
```

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

```
fun main() {  
    print("Введите количество элементов массива (n): ")  
    val n = readLine()?.toIntOrNull() ?: 0  
    if (n <= 0) {  
        println("Некоректное значение n.")  
        return  
    }  
    val numbers = IntArray(n) { (0..n).random() }
```

```

print("Введите элемент для поиска: ")
val elementToFind = readLine()?.toIntOrNull()?:0
val index = numbers.indexOf(elementToFind)
println("Массив: ${numbers.contentToString()}")
if (index != -1){
    println("Индекс элемента $elementToFind: $index")
} else {
    println("Элемент $elementToFind не найден")
}
}

```

```

C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe
Введите количество элементов массива (n): 7
Введите элемент для поиска: 4
Массив: [7, 3, 2, 6, 3, 2, 4]
Индекс элемента 4: 6

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некоректное значение n.")
        return
    }
    val originalArray = IntArray(n){(0..n).random()}
    val copiedArray = originalArray.copyOf()
    println("Оригинал: ${originalArray.contentToString()}")
    println("Копия: ${copiedArray.contentToString()}")
}

```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe
Введите количество элементов массива (n): 5
Оригинал: [4, 2, 4, 4, 2]
Копия: [4, 2, 4, 4, 2]

Process finished with exit code 0
```

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

```
fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n){(0..n).random()}
    val sumOfEven = numbers.filter { it % 2 == 0 }.sum()
    println("Массив: ${numbers.contentToString()}")
    println("Сумма четных чисел: $sumOfEven")
}
```

```
C:\Users\StudentFSP0\.jdk\openjdk-22.0.2\bin\java.exe
Введите количество элементов массива (n): 6
Массив: [3, 6, 4, 0, 3, 1]
Сумма четных чисел: 10

Process finished with exit code 0
```

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

```

fun main() {
    print("Введите количество элементов массива №1 (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некоректное значение n.")
        return
    }
    val numbers = IntArray(n){(0..n).random()}
    print("Введите количество элементов массива №2 (m): ")
    val m = readLine()?.toIntOrNull() ?: 0
    if (m <= 0) {
        println("Некоректное значение m.")
        return
    }
    val numbers2 = IntArray(m){(0..m).random()}
    val intersection = numbers.intersect(numbers2.toList())
    println("Массив №1: ${numbers.contentToString()}")
    println("Массив №2: ${numbers2.contentToString()}")
    println("Пересечение массивов: ${intersection.joinToString()}")
}

```

```

MainKt x
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:
Введите количество элементов массива №1 (n): 7
Введите количество элементов массива №2 (m): 6
Массив №1: [1, 0, 2, 0, 1, 6, 1]
Массив №2: [2, 4, 4, 6, 1, 6]
Пересечение массивов: 1, 2, 6

Process finished with exit code 0
|

```

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

```

fun main() {

```



```

print("Введите количество элементов массива (n): ")

val n = readLine()?.toIntOrNull() ?: run {
    println("Некорректное значение n. n должно быть целым числом больше 0.")
    return
}

if (n <= 0) {
    println("Некорректное значение n. n должно быть больше 0.")
    return
}

val numbers = IntArray(n) { (0 until 1000).random() }
println("Массив до перестановки: ${numbers.contentToString()}")

fun readIntIndex(prompt: String): Int? {
    print(prompt)
    val input = readLine()
    return input?.toIntOrNull()?.takeIf { it in 0 until n }
}

val index1 = readIntIndex("Введите индекс первого элемента для перестановки (от 0 до ${n - 1}): ")
val index2 = readIntIndex("Введите индекс второго элемента для перестановки (от 0 до ${n - 1}): ")

if (index1 == null || index2 == null) {
    println("Некорректные индексы. Индексы должны быть целыми числами от 0 до ${n - 1}.")
    return
}

if (index1 == index2) {
    println("Индексы должны быть различны.")
    return
}

```

```

swapElements(numbers, index1, index2)

println("Массив после перестановки: ${numbers.contentToString()}")
}

```

```

fun swapElements(array: IntArray, index1: Int, index2: Int) {

    val temp = array[index1]

    array[index1] = array[index2]

    array[index2] = temp

}

```

```

MainKt x
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3\lib\idea_rt.jar=5729
Введите количество элементов массива (n): 10
Массив до перестановки: [375, 304, 442, 803, 107, 901, 798, 728, 343, 16]
Введите индекс первого элемента для перестановки (от 0 до 9): 4
Введите индекс второго элемента для перестановки (от 0 до 9): 2
Массив после перестановки: [375, 304, 107, 803, 442, 901, 798, 728, 343, 16]

Process finished with exit code 0

```

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

```

fun main() {
    val numbers = IntArray(20) { (0 until 100).random() }
    println("Массив: ${numbers.contentToString()}")
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3\lib\idea_r
Массив: [64, 95, 33, 0, 98, 28, 95, 48, 96, 35, 96, 47, 26, 82, 75, 36, 75, 78, 16, 5]

Process finished with exit code 0
|

```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
}

```

```

if (n <= 0) {
    println("Некоректное значение n.")
    return
}
val numbers = IntArray(n) { (0..n).random() }
val div3 = numbers.filter{it % 3 == 0}
println("Массив: ${numbers.contentToString()}")
println("Числа, делящиеся на 3: ${div3.joinToString()}")
}

```

A screenshot of a terminal window showing the execution of a Java program. The command prompt is "C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagen". The program prompts the user to enter the number of array elements (n), with the input "8" shown in green. It then displays the generated array: "Массив: [0, 0, 3, 1, 8, 3, 6, 8]". Next, it shows the numbers divisible by 3: "Числа, делящиеся на 3: 0, 0, 3, 3, 6". Finally, it states "Process finished with exit code 0".

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некоректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    val Palindromes = numbers.contentEquals(numbers.reversedArray())
    println("Массив: ${numbers.contentToString()}")
    println("Является ли палиндромом: ${Palindromes}")
}

```

A screenshot of a terminal window showing the execution of a Java program. The command prompt is "C:\Program Files\Java\jdk-11\bin\java.exe" "-javaag". The program prompts the user to enter the number of array elements (n), with the input "5" shown in green. It then displays the generated array: "Массив: [5, 2, 1, 5, 0]". Next, it checks if it is a palindrome and outputs "Является ли палиндромом: false". Finally, it states "Process finished with exit code 0".

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "  
Массив: [4, 5, 6, 5, 4]  
Является ли палиндромом: true  
  
Process finished with exit code 0
```

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

```
fun main() {  
    print("Введите количество элементов массива №1 (n): ")  
    val n = readLine()?.toIntOrNull() ?: 0  
    if (n <= 0) {  
        println("Некоректное значение n.")  
        return  
    }  
    val numbers = IntArray(n){(0..n).random()}  
    print("Введите количество элементов массива №2 (m): ")  
    val m = readLine()?.toIntOrNull() ?: 0  
    if (m <= 0) {  
        println("Некоректное значение m.")  
        return  
    }  
    val numbers2 = IntArray(m){(0..m).random()}  
    val combo = numbers + numbers2  
    println("Массив №1: ${numbers.contentToString()}")  
    println("Массив №2: ${numbers2.contentToString()}")  
    println("Объединенный массив: ${combo.contentToString()}")  
}
```

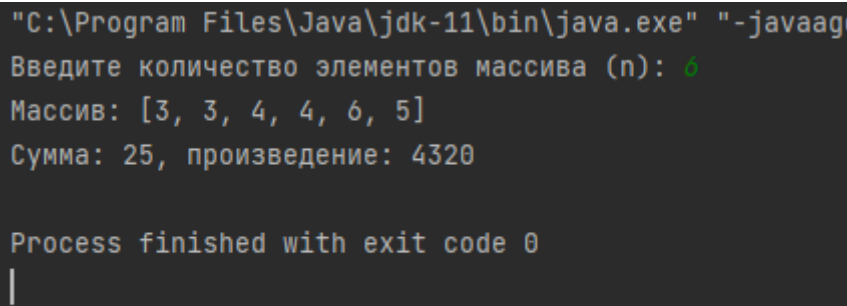
```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C  
Введите количество элементов массива №1 (n): 5  
Введите количество элементов массива №2 (m): 4  
Массив №1: [4, 2, 3, 2, 5]  
Массив №2: [3, 1, 4, 4]  
Объединенный массив: [4, 2, 3, 2, 5, 3, 1, 4, 4]  
  
Process finished with exit code 0
```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    val sum = numbers.sum()
    val product = numbers.fold(1){ acc, i -> acc * i }
    println("Массив: ${numbers.contentToString()}")
    println("Сумма: $sum, произведение: $product")
}

```



```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaag
Введите количество элементов массива (n): 6
Массив: [3, 3, 4, 4, 6, 5]
Сумма: 25, произведение: 4320

Process finished with exit code 0
|

```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    println("Массив: ${numbers.contentToString()}")
    groupNumbers(numbers, 5)
}

fun groupNumbers(array: IntArray, groupSize: Int) {
    for (i in array.indices step groupSize) {
        val group = array.sliceArray(i until minOf(i + groupSize, array.size))
        println(group.contentToString())
    }
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3
Введите количество элементов массива (n): 15
Массив: [0, 7, 11, 6, 9, 3, 14, 13, 10, 6, 3, 4, 8, 12, 7]
[0, 7, 11, 6, 9]
[3, 14, 13, 10, 6]
[3, 4, 8, 12, 7]

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите количество элементов массива №1 (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некоректное значение n.")
        return
    }
    val numbers = IntArray(n){(0..n).random()}
    print("Введите количество элементов массива №2 (m): ")
    val m = readLine()?.toIntOrNull() ?: 0
    if (m <= 0) {
        println("Некоректное значение m.")
        return
    }
    val numbers2 = IntArray(m){(0..m).random()}
    numbers.sort()
    numbers2.sort()
    val combo = numbers + numbers2
    println("Отсортированный массив №1: ${numbers.contentToString()}")
    println("Отсортированный массив №2: ${numbers2.contentToString()}")
    println("Объединенный массив: ${combo.contentToString()}")
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\In
Введите количество элементов массива №1 (n): 5
Введите количество элементов массива №2 (m): 6
Отсортированный массив №1: [1, 2, 2, 3, 5]
Отсортированный массив №2: [0, 1, 3, 4, 5, 6]
Объединенный массив: [1, 2, 2, 3, 5, 0, 1, 3, 4, 5, 6]

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите количество элементов массива: ")
    val size = readLine()!!.toInt()
    val array = IntArray(size) { (0..99).random() }
    println("Исходный массив: ${array.joinToString()}")
    array.sort()
    println("Отсортированный массив: ${array.joinToString()}")
    print("Введите шаг арифметической прогрессии: ")
    val step = readLine()!!.toInt()
    val progressionArray = IntArray(size) { array[it] + it * step }
    println("Массив в арифметической прогрессии: ${progressionArray.joinToString()}")
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3\lib\ide
Введите количество элементов массива: 10
Исходный массив: 70, 23, 81, 79, 51, 20, 67, 24, 71, 70
Отсортированный массив: 20, 23, 24, 51, 67, 70, 70, 71, 79, 81
Введите шаг арифметической прогрессии: 6
Массив в арифметической прогрессии: 20, 29, 36, 69, 91, 100, 106, 113, 127, 135

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
}

```

```

val numbers = IntArray(n) { (0..n).random() }
println("Исходный массив: ${numbers.contentToString()}")
print("Введите элемент для удаления: ")
val elementToRemove = readLine()?.toIntOrNull() ?: 0

val newArray = removeElement(numbers.copyOf(), elementToRemove) //Use copy to not modify
original
println("Массив без элемента $elementToRemove: ${newArray.contentToString()}")
}

fun removeElement(array: IntArray, element: Int): IntArray {
    return array.filter { it != element }.toIntArray()
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Inte
Введите количество элементов массива (n): 10
Исходный массив: [9, 3, 4, 5, 4, 5, 2, 6, 10, 3]
Введите элемент для удаления: 5
Массив без элемента 5: [9, 3, 4, 4, 2, 6, 10, 3]

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    val secondMax = findSecondMax(numbers)
    println("Массив: ${numbers.contentToString()}")
    println("Второй максимальный элемент: $secondMax")
}

fun findSecondMax(array: IntArray): Int? {
    if (array.size < 2) return null
    val sorted = array.sortedArrayDescending()
    return sorted[1]
}

```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaage
Введите количество элементов массива (n): 10
Массив: [0, 2, 5, 6, 7, 8, 7, 6, 0, 7]
Второй максимальный элемент: 7

Process finished with exit code 0
```

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

По сути повторяется с 16 и 19 заданиями.

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

```
fun main() {
    val rows = readMatrixDimension("Введите количество строк матрицы:")
    val cols = readMatrixDimension("Введите количество столбцов матрицы:")

    if (rows <= 0 || cols <= 0) {
        println("Размеры матрицы должны быть больше 0.")
        return
    }

    val matrix = createRandomMatrix(rows, cols)
    println("Исходная матрица:")
    printMatrix(matrix)
    println("\nТранспонированная матрица:")
    printMatrix(transposeMatrix(matrix))
}

fun readMatrixDimension(message: String): Int {
    print("$message ")
    while (true) {
        val input = readLine()?.toIntOrNull()
        if (input != null && input > 0) {
            return input
        } else {
            println("Некорректный ввод. Пожалуйста, введите целое число больше 0.")
        }
    }
}

fun createRandomMatrix(rows: Int, cols: Int): Array<IntArray> {
    return Array(rows) { IntArray(cols) { (0..100).random() } }
}
```

```

}

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

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

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:
Введите количество строк матрицы: 4
Введите количество столбцов матрицы: 3
Исходная матрица:
[4, 23, 4]
[14, 75, 46]
[42, 67, 12]
[25, 84, 8]

Транспонированная матрица:
[4, 14, 42, 25]
[23, 75, 67, 84]
[4, 46, 12, 8]

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
}

```

```

val numbers = IntArray(n) { (0..n).random() }
println("Исходный массив: ${numbers.contentToString()}")
print("Введите элемент для поиска: ")
val elementToFind = readLine()?.toIntOrNull() ?: 0

val found = linearSearch(numbers, elementToFind)
println("Элемент $elementToFind ${if (found) "найден" else "не найден"}")
}

fun linearSearch(array: IntArray, element: Int): Boolean {
    return array.contains(element)
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\
Введите количество элементов массива (n): 10
Исходный массив: [1, 6, 6, 10, 4, 3, 6, 3, 2, 4]
Введите элемент для поиска: 6
Элемент 6 найден

Process finished with exit code 0

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C
Введите количество элементов массива (n): 10
Исходный массив: [9, 7, 7, 5, 1, 4, 2, 0, 5, 3]
Введите элемент для поиска: 8
Элемент 8 не найден

Process finished with exit code 0

```

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

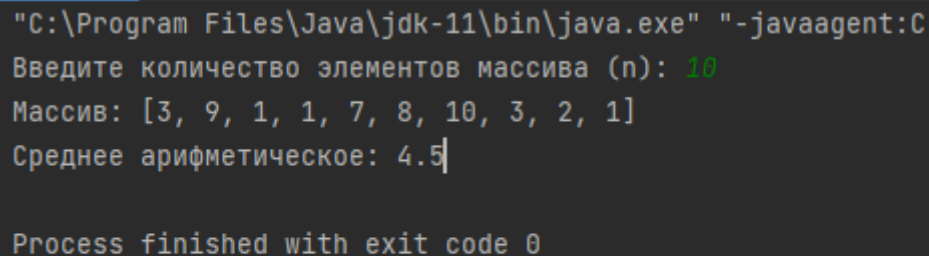
```

fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    val average = calculateAverage(numbers)
    println("Массив: ${numbers.contentToString()}")
}

```

```
println("Среднее арифметическое: $average")
}
```

```
fun calculateAverage(array: IntArray): Double {
    return array.average()
}
```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-11\bin\javaagent.jar -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9090 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false"
Введите количество элементов массива (n): 10
Массив: [3, 9, 1, 1, 7, 8, 10, 3, 2, 1]
Среднее арифметическое: 4.5
Process finished with exit code 0
```

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

```
fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    val maxSequence = findMaxSequence(numbers)
    println("Массив: ${numbers.contentToString()}")
    println("Максимальная последовательность: ${maxSequence.contentToString()}")
}
```

```
fun findMaxSequence(array: IntArray): IntArray {
    if (array.isEmpty()) return intArrayOf() // Return empty array for empty input

    var maxCount = 1
    var currentCount = 1
    var maxStartIndex = 0
    var currentStartIndex = 0

    for (i in 1 until array.size) {
        if (array[i] == array[i - 1]) {
            currentCount++
        } else {
            if (currentCount > maxCount) {
                maxCount = currentCount
                maxStartIndex = currentStartIndex
            }
            currentCount = 1
            currentStartIndex = i
        }
    }

    return array[maxStartIndex..maxStartIndex + maxCount - 1]
}
```

```

    }
    currentStartIndex = i
    currentCount = 1
  }
}

// Check if the last sequence is the longest
if (currentCount > maxCount) {
    maxStartIndex = currentStartIndex
    maxCount = currentCount
}

return array.sliceArray(maxStartIndex until maxStartIndex + maxCount)
}

```

```

main.kt x
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\
Введите количество элементов массива (n): 6
Массив: [5, 3, 5, 5, 0, 3]
Максимальная последовательность: [5, 5]

Process finished with exit code 0

```

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

```

fun main() {
    print("Введите элементы массива через пробел: ")
    val input = readLine() ?: ""
    val array = input.split(" ").mapNotNull { it.toIntOrNull() }.toIntArray()

    if (array.isEmpty()) {
        println("Массив пуст.")
        return
    }

    println("Введенный массив: ${array.contentToString()}")
}

```

```
MainKt x
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 20
Введите элементы массива через пробел: 3 56 3 2 8 23 10 9
Введенный массив: [3, 56, 3, 2, 8, 23, 10, 9]

Process finished with exit code 0
```

29.Нахождение медианы: Напишите программу, которая находит медиану

в массиве.

```
fun main() {
    print("Введите количество элементов массива (n): ")
    val n = readLine()?.toIntOrNull() ?: 0
    if (n <= 0) {
        println("Некорректное значение n.")
        return
    }
    val numbers = IntArray(n) { (0..n).random() }
    val median = findMedian(numbers)
    println("Массив: ${numbers.contentToString()}")
    println("Медиана: $median")
}
```

```
fun findMedian(array: IntArray): Double {
    val sorted = array.sortedArray()
    val mid = sorted.size / 2
    return if (sorted.size % 2 == 0) {
        (sorted[mid - 1] + sorted[mid]).toDouble() / 2
    } else {
        sorted[mid].toDouble()
    }
}
```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-11\bin\java.exe"
Введите количество элементов массива (n): 10
Массив: [1, 1, 0, 5, 0, 6, 7, 3, 7, 9]
Медиана: 4.0

Process finished with exit code 0
```

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

Повторяется с 18, просто здесь чуть больше конкретики

```
fun main() {
    val numbers = IntArray(100) { (0..100).random() }
    println("Массив: ${numbers.contentToString()}")
    groupNumbers(numbers, 10)
}

fun groupNumbers(array: IntArray, groupSize: Int) {
    for (i in array.indices step groupSize) {
        val group = array.sliceArray(i until minOf(i + groupSize, array.size))
        println(group.contentToString())
    }
}
```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3\lib\idea_rt.jar=59488:C:\IntelliJ IDEA 2023.1.3\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\user\IdeaProjects\untitled\out\production\untitled
Массив: [55, 19, 22, 58, 18, 5, 41, 50, 100, 85, 83, 3, 22, 48, 12, 41, 82, 15, 54, 73, 91, 83, 30, 7, 36, 71, 70, 10, 82, 90, 77, 26, 37, 79, 75, 26, 49, 66, 5, 7, 51, 80, 84, 60, 61, 94, 45, 65, 0, 47, 61, 48, 25, 54, 74, 13]
[55, 19, 22, 58, 18, 5, 41, 50, 100, 85]
[83, 3, 22, 48, 12, 41, 82, 15, 54, 73]
[91, 83, 30, 7, 36, 71, 70, 10, 82, 90]
[77, 26, 37, 79, 75, 24, 49, 66, 5, 7]
[51, 80, 84, 60, 61, 94, 45, 65, 0, 47]
[61, 48, 25, 54, 74, 13, 59, 44, 28, 88]
[49, 91, 37, 28, 98, 81, 9, 62, 87, 74]
[25, 70, 5, 17, 49, 11, 75, 52, 80, 23]
[18, 10, 98, 62, 76, 27, 1, 78, 8, 84]
[16, 29, 46, 28, 10, 13, 31, 26, 36, 59]

Process finished with exit code 0
```