

1.Красиво вывести информацию о типах данных (целочисленные, строки) и их переменных (int, string) , и других существующих в вашем языке типов данных. Оформить всё аккуратно и красиво.

```
fun main()
{
    println("""
        Integer - целочисленный тип данных, разновидностью 4 байта.
        String - строковый тип данных, динамической размерности.
        Char - символьный тип данных, размером 4 байта.
        Boolean - логический тип данных, размером 1 байт.
        Float ( Double ) - тип данных с плавающей точкой, размером 8
байт ( 8 байт )
        """)
}
```

```
Integer - целочисленный тип данных, разновидностью 4 байта.
String - строковый тип данных, динамической размерности.
Char - символьный тип данных, размером 4 байта.
Boolean - логический тип данных, размером 1 байт.
Float ( Double ) - тип данных с плавающей точкой, размером 8 байт ( 8 байт )
```

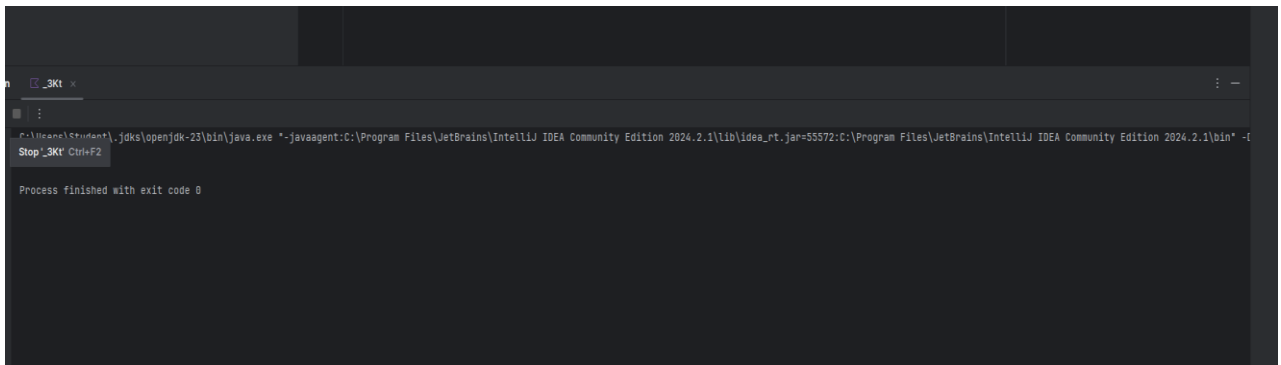
2. Вывести на экран число с точностью до сотых.

```
import kotlin.math.round
fun main()
{
    val number : Float = 5.136727F
    println(number)
    println(round(number*100)*0.01)
}
```

```
C:\Users\Student\jdk\openjdk-23\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=55488:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -I
5.136727
5.14
Process finished with exit code 0
```

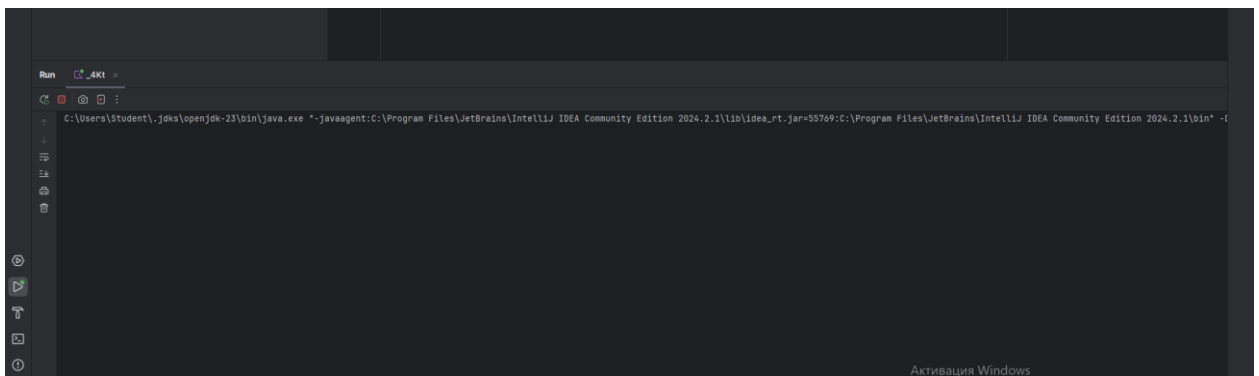
3. Вывести на экран число E (основание натурального логарифма) с точностью до десятых.

```
import kotlin.math.E
import kotlin.math.round
fun main()
{ println(round(E*10)*0.1) }
```



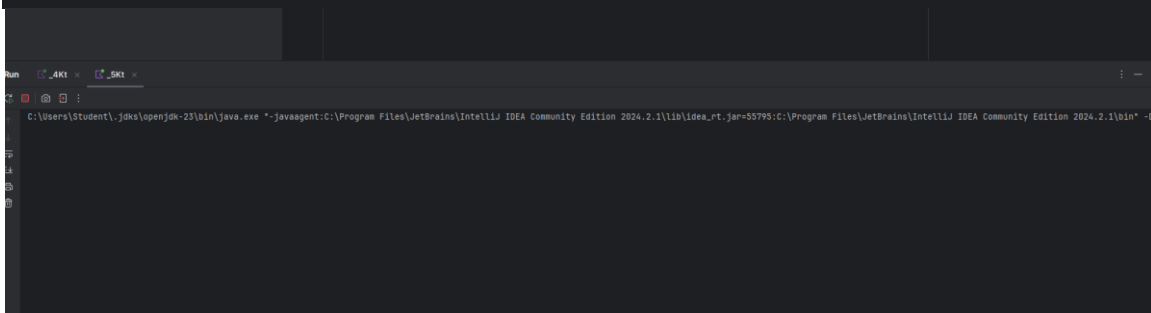
4. Составить программу вывода на экран числа вводимого с клавиатуры. Выводимому числу дано следовать сообщение « Вы ввели число».

```
fun main()
{ println("Вы ввели число: " + readln()) }
```



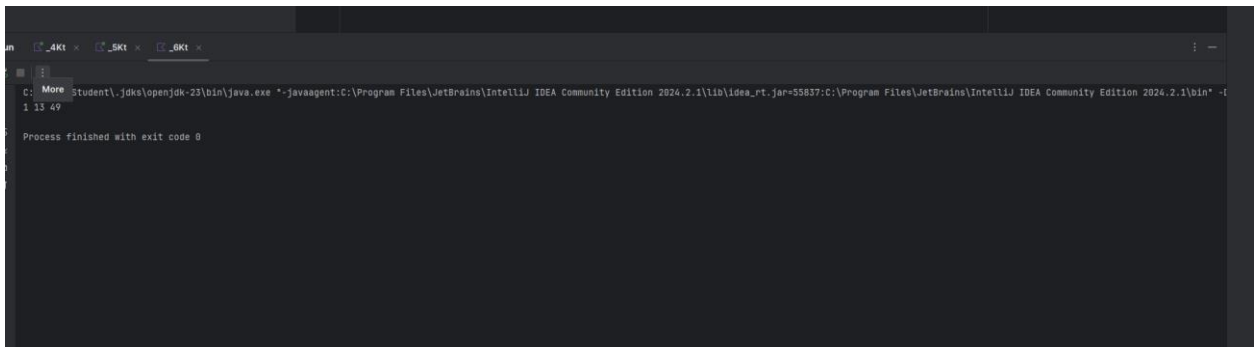
5. Составить программу вывода на экран числа, вводимого с клавиатуры. Выводимому числу дано следовать сообщение « вот такое число Вы ввели».

```
fun main()
{ println(readln() + " - вот такое число вы ввели.") }
```



6. Вывести на одной строке числа 1, 13 и 49 с одним пробелом между ними.

```
fun main()
{ println("1 13 49") }
```

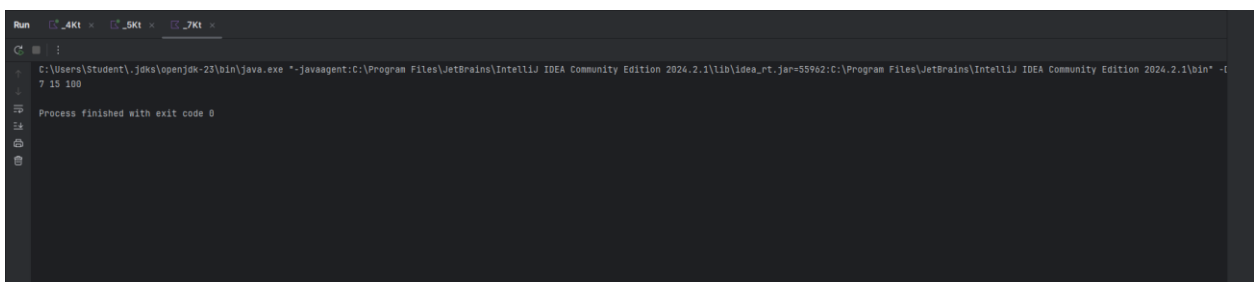


```
C:\Users\Student\jdk\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=55837:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -I
1 13 49

Process finished with exit code 0
```

7. Вывести на одной строке числа 7, 15 и 100 с двумя пробелами между ними.

```
fun main()
{ println("7 15 100") }
```

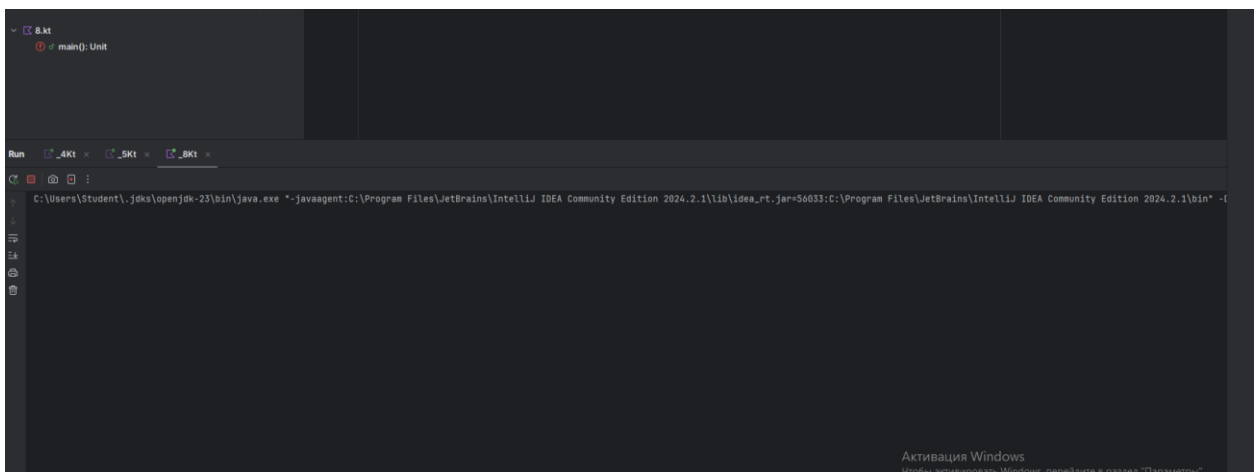


```
C:\Users\Student\jdk\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=55962:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -I
7 15 100

Process finished with exit code 0
```

8. Составить программу вывода на экран в одну строку трёх любых чисел с двумя пробелами между ними.

```
fun main()
{ println(readln() + " " + readln() + " " + readln()) }
```

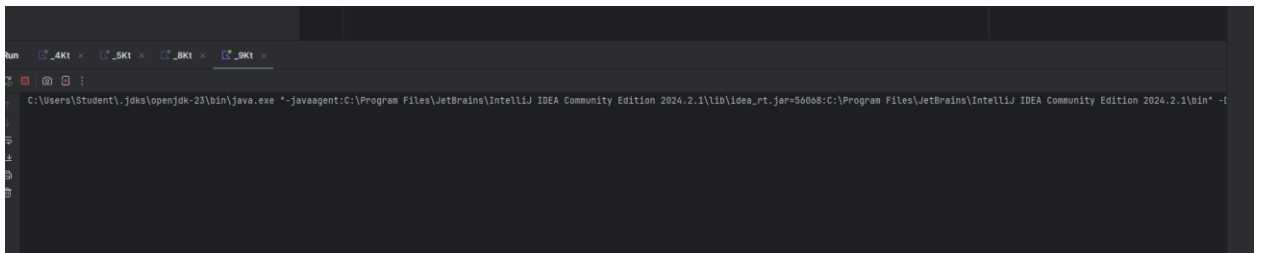


```
C:\Users\Student\jdk\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=56033:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -I
7 15 100

Process finished with exit code 0
```

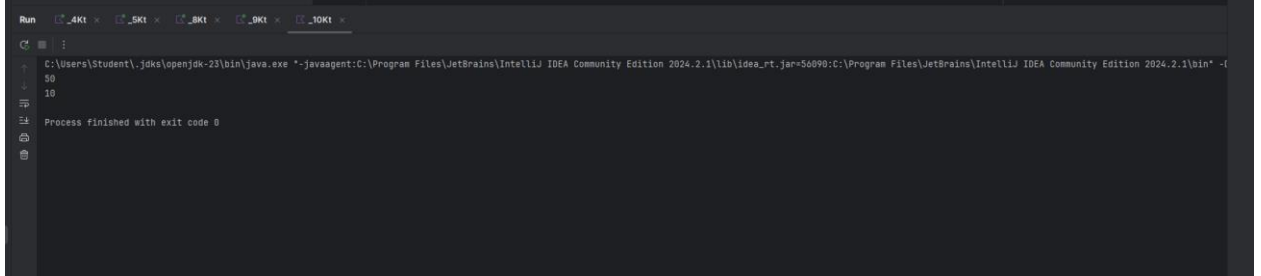
9. Составить программу вывода на экран в одну строку четырёх любых чисел с одним пробелом между ними.

```
fun main()
{ println(readln() + " " + readln() + " " + readln() + " " + readln()) }
```



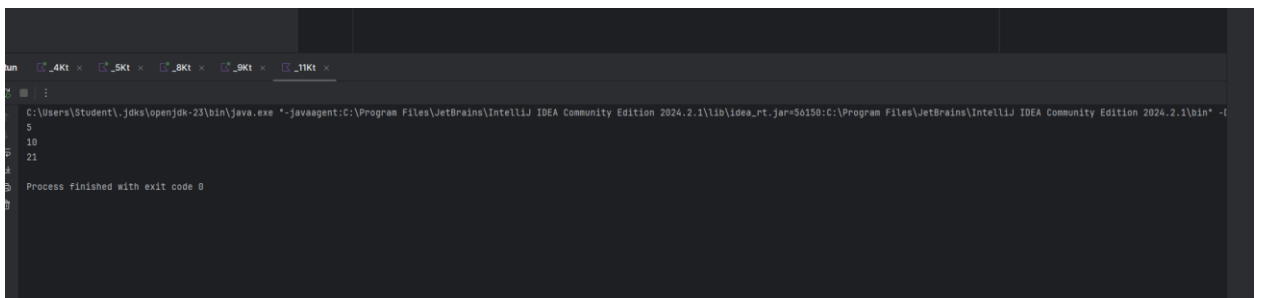
10. Вывести на экран числа 50 и 10 одно под другим.

```
fun main()
{ println("50\n10") }
```



11. Вывести на экран числа 5, 10 и 21 одно под другим.

```
fun main()
{ println("5\n10\n21") }
```



12. Составить программу вывода на экран «столбиком» четырёх любых чисел.

```
fun main()
{ println(readln() + "\n" + readln() + "\n" + readln() + "\n" + readln()) }
```

