

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, размерность 4 байта.
        String - тип данных, значениями которого является произвольная
        последовательность (строка) символов алфавита.
        Float - Тип данных float служит для хранения чисел с плавающей
        запятой.
        Char - тип данных, предназначенный для хранения одного символа
        (управляющего или печатного) в определённой кодировке.
        Double - компьютерный формат представления числа с плавающей запятой.
        Boolean - примитивный тип данных в информатике, которые могут
        принимать два возможных значения, иногда называемых истиной ( true ) и ложью
        ( false ).
        -----
        -----
        -----
        """).trimIndent()
}
```

```
C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=54408:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Dfile.encoding=UTF-8
integer - Целочисленный тип данных, размерность 4 байта.
String - тип данных, значениями которого является произвольная последовательность (строка) символов алфавита.
Float - Тип данных float служит для хранения чисел с плавающей запятой.
Char - тип данных, предназначенный для хранения одного символа (управляющего или печатного) в определённой кодировке.
Double - компьютерный формат представления числа с плавающей запятой.
Boolean - примитивный тип данных в информатике, которые могут принимать два возможных значения, иногда называемых истиной ( true ) и ложью ( false ).
-----
-----
-----
Process finished with exit code 0
nov_Tkachuk > src > Main.kt > main 12:5 LF UTF-8 4 spaces
```

2) Вывести на экран число с точностью до сотых.

```
fun main() {
    println(5.111)
}
```

```
C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=54435:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Dfile.encoding=UTF-8
5.111
Process finished with exit code 0
nov_Tkachuk > src > Main.kt 4:1 LF UTF-8 4 spaces
```

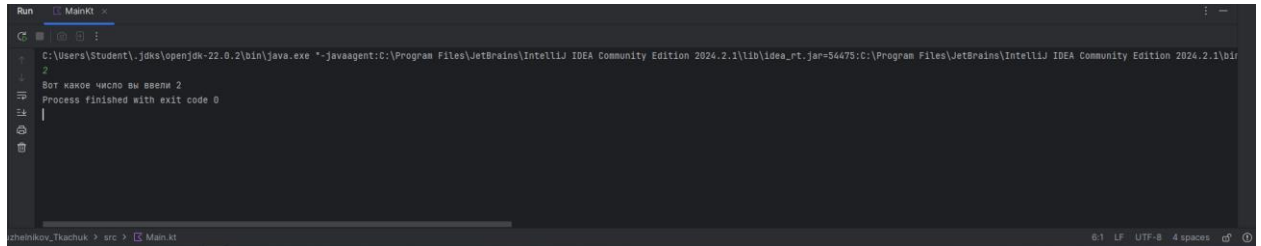
3) Вывести на экран число e (основание натурального логарифма) с точностью до десятых.

```
fun main() {
    println(2.7)
}
```

```
C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=54403:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Dfile.encoding=UTF-8
2.7
Process finished with exit code 0
nov_Tkachuk > src > Main.kt 4:1 LF UTF-8 4 spaces
```

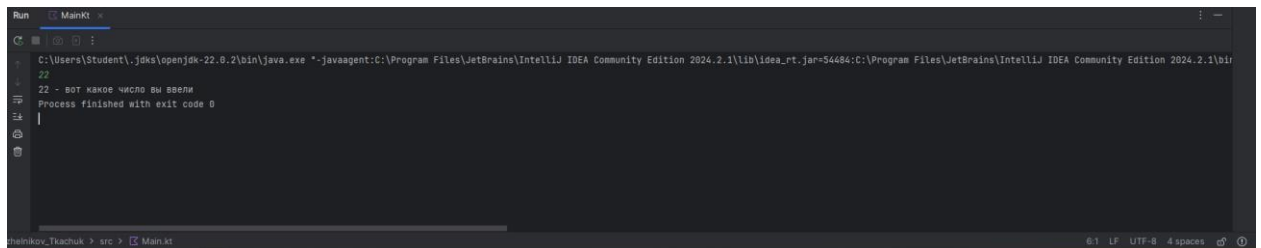
4) Составить программу вывода на экран числа, вводимого с клавиатуры. Выводимому числу должно предшествовать сообщение "Вы ввели число".

```
fun main() {  
    val bar = readln()  
    print("Вот какое число вы ввели ")  
    print(bar)  
}
```



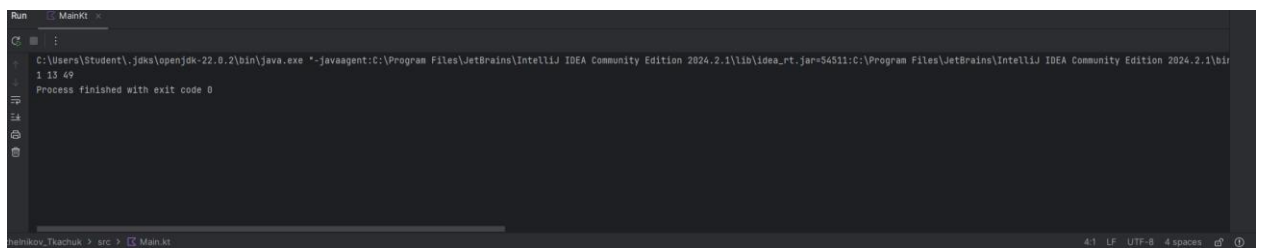
5) Составить программу вывода на экран числа, вводимого с клавиатуры. После выводимого числа должно следовать сообщение " — вот какое число Вы ввели".

```
fun main() {  
    val bar = readln()  
    print(bar)  
    print(" - вот какое число вы ввели")  
}
```



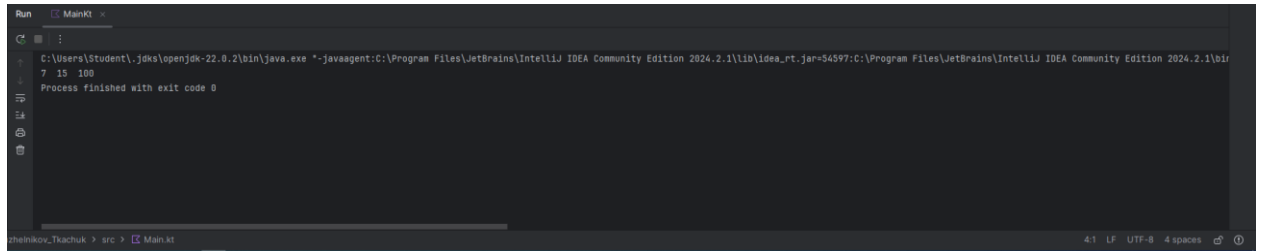
6) Вывести на одной строке числа 1, 13 и 49 с одним пробелом между ними.

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



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


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



The screenshot shows the Run console of IntelliJ IDEA. The command prompt path is visible at the top. The output of the program is "7 15 100", where the numbers are separated by two spaces. Below the output, it says "Process finished with exit code 0".

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

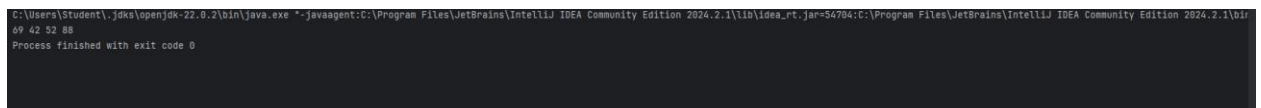
```
fun main() {  
    print("9  88  10")  
}
```



The screenshot shows the Run console of IntelliJ IDEA. The output of the program is "9 88 10", where the numbers are separated by two spaces. Below the output, it says "Process finished with exit code 0".

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

```
fun main() {  
    print("69 42 52 88")  
}
```



The screenshot shows the Run console of IntelliJ IDEA. The output of the program is "69 42 52 88", where the numbers are separated by one space. Below the output, it says "Process finished with exit code 0".

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

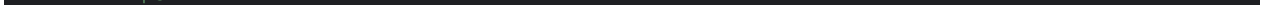
```
fun main() {  
    print("""  
        |50  
        |10  
    """).trimMargin()  
}
```



The screenshot shows the Run console of IntelliJ IDEA. The output of the program is "50" on the first line and "10" on the second line. Below the output, it says "Process finished with exit code 0".

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

```
fun main() {  
    print("""  
        |5
```



The screenshot shows the beginning of the Run console of IntelliJ IDEA. The output of the program is "5" on the first line. Below the output, it says "Process finished with exit code 0".

```
    |10
    |21
    """.trimMargin())
```

```
C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=54847:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin\
5
10
21
Process finished with exit code 0
```

12) . Составить программу вывода на экран "столбиком" четырех любых чисел

```
fun main() {
    print("""
        |12
        |16
        |42
        |52
        """.trimMargin())
}
```

```
C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=54806:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin\
12
16
42
52
Process finished with exit code 0
```