

**Министерство цифрового развития, связи и массовых коммуникаций  
Российской Федерации**

**Ордена Трудового Красного Знамени федеральное государственное  
бюджетное образовательное учреждение высшего образования  
«Московский технический университет связи и информатики»**

**Лабораторная работа №1**  
**“Функциональное программирование”**

**Выполнила:**

**Капленко Екатерина Михайловна**

**Студент группы БВТ1901**

**Проверила: Мосева Марина Сергеевна**

**Москва 2021**

1. Переменные `res` – это значения `val` или настоящие переменные `var`?  
**Val**
2. `"crazy" * 3` в REPL
3. Что означает выражение `10 max 2`? В каком классе определен метод `max`?
4. Используя число типа `BigInt`, вычислите  $2^{1024}$

```
C:\Users\Екатерина\.jdk\openjdk-15.0.1\bin\java.exe ...
Welcome to Scala 2.13.7 (OpenJDK 64-Bit Server VM, Java 15.0.1).
Type in expressions for evaluation. Or try :help.

scala> 3+1
val res0: Int = 4

scala> "crazy"*3
val res1: String = crazycrazycrazy

scala> 10 max 2
val res2: Int = 10

scala> BigInt(2) pow 1024
val res3: scala.math.BigInt = 1797693134862315907729305190789024733617976
```

5. Что нужно импортировать, чтобы найти случайное простое число вызовом метода `probablePrime(100, Random)` без использования каких-либо префиксов перед именами `probablePrime` и `Random`

```
scala> import scala.math.BigInt.probablePrime
import scala.math.BigInt.probablePrime

scala> import scala.util.Random

probablePrime(100, Random)
import scala.util.Random

scala>
scala> val res0: scala.math.BigInt = 1139351823905565023199151060727
```

6. Один из способов создать файл или каталог со случайным именем состоит в том, чтобы сгенерировать случайное число типа `BigInt` и преобразовать его в систему счисления по основанию 36, в результате получится строка, такая как `"qsnvbevtomcj38o06kul"`. Отыщите в Scaladoc методы, которые можно было бы использовать для этого.

```
scala> probablePrime(100, Random) toString 36
val res1: String = 1u4etuwfe9xrc5nq748b
```

7. Как получить первый символ строки в языке Scala? А последний символ?

```
scala> "Tarakan".head
val res4: Char = T

scala> "Tarakan".last
val res5: Char = n
```

8. Что делают строковые функции `take`, `drop`, `takeRight` и `dropRight`? Какие преимущества и недостатки они имеют в сравнении с `substring`?

```
scala> "Letter from Russia".take(6)
val res7: String = Letter

scala> "Letter from Russia".drop(6)
val res8: String = " from Russia"

scala> "Letter from Russia".takeRight(6)
val res9: String = Russia

scala> "Letter from Russia".dropRight(6)
val res10: String = "Letter from "
```

9. Сигнум числа равен 1, если число положительное. -1 – если отрицательное, и 0 – если равно нулю. Напишите функцию, вычисляющую это значение.

```
scala> def signum(x: Int) = if (x > 0) 1 else if (x < 0) -1 else 0
def signum(x: Int): Int

scala> signum(43)
val res11: Int = 1

scala> signum(-2)
val res12: Int = -1

scala> signum(0)
val res13: Int = 0
```

10. Какое значение возвращает блок `{}`? Каков его тип?

```
scala> val p = {}
val p: Unit = ()
```

11. Напишите на языке Scala цикл, эквивалентный циклу на языке Java  
`for (int i=10; i>=0; i--) System.out.println(i)`

```
scala> for (i <- 10 to 0 by -1) println(i)
10
9
8
7
6
5
4
3
2
1
0
```

12. Напишите процедуру `countdown (n: Int)`, которая выводит числа от `n` до 0

```
scala> def countdown(n: Int) = for (i <- n to 0 by -1) println(i)
def countdown(n: Int): Unit

scala> countdown(7)
7
6
5
4
3
2
1
0
```

13. Напишите цикл `for` для вычисления кодовых пунктов Юникода всех букв в строке. Например, произведение символов в строке «Hello» равно 9415087488L.
14. Решите предыдущее упражнение без применения цикла. Напишите функцию `product(s: String)`, вычисляющую произведение, как описано в предыдущих упражнениях.

```
scala> (for (i <- "Hello") yield i.toLong).product
val res18: Long = 9415087488

scala> def product(s: String) = s.map(c => c.toLong).product
def product(s: String): Long

scala> product("Hello")
val res19: Long = 9415087488
```

15. Сделайте функцию из предыдущего упражнения рекурсивной.

```
scala> def productr(s:String):Long = { var num = s.head.toLong; if (s.tail.length>0) num*=productr(s.tail); num}
def productr(s: String): Long

scala> productr("Hello")
val res21: Long = 9415087488
```

16. Напишите функцию, вычисляющую  $x^n$ , где  $n$  – целое число.

Используйте следующее рекурсивное определение:

- $x^n = y^2$ , если  $n$  – четное и положительное число, где  $y = x^{n/2}$
- $x^n = x * x^{n-1}$ , если  $n$  – нечетное и положительное число.
- $x^0 = 1$ .
- $x^n = 1/x^{-n}$ , если  $n$  – отрицательное число. Не используйте инструкцию return.

```
scala> def pow(x:Int, n:Int) = n match{
  case n if n > 0 && (n%2==0) => scala.math.pow(x, n/2)
  case n if n > 0 && (n%2==1) => x*scala.math.pow(x, n-1)
  case n if n < 0 => 1/scala.math.pow(x, -n)
  case 0 => 1
}
|      |      |      |      | def pow(x: Int, n: Int): Double

scala> pow(2,5)
val res22: Double = 32.0
```

18.  $f(m,n)$  - сумма всех натуральных чисел от  $m$  до  $n$  включительно, в десятичной записи которых нет одинаковых цифр.

```
scala> def distinctDigits(n: Int): Boolean = {
  val s = n.toString
  s.length == s.distinct.length
}
|      |      | def distinctDigits(n: Int): Boolean

scala> def func(m:Int,n:Int):Int = (m to n).filter(distinctDigits).sum
def func(m: Int, n: Int): Int

scala> func(10,12)
val res0: Int = 22
```

19. Список содержит целые числа, а также другие списки, такие же как и первоначальный. Получить список, содержащий только целые числа из всех вложенных списков.

Пример:  $f(\text{List}(\text{List}(1, 1), 2, \text{List}(3, \text{List}(5, 8)))) = \text{List}(1, 1, 2, 3, 5, 8)$

```
scala> def f(list: List[Any]): List[Int] = {
  var newList = new ListBuffer[Int]()
  list.foreach(x=> {
    if(x.isInstanceOf[Int]) newList +=x.asInstanceOf[Int]
    if(x.isInstanceOf[List[Any]]) newList +=f(x.asInstanceOf[List[Any]])
  })
  newList.toList
}

scala> f(List(List(1,1),2,List(3,List(5,8))))
val res0: List[Int] = List(1, 1, 2, 3, 5, 8)
```

20.  $f(n)$  - сумма цифр наибольшего простого делителя натурального числа  $n$ .

```
scala> def bigNatdel(n:Int): Int={
  var num:Int = n
  var i: Int=2
  while(i != num){
    if (num % i == 0){
      num=num/i
    } else {
      i=i+1
    }
  }
  return num
}

scala> def summa(n:Int):Int={
  var num:Int = bigNatdel(n)
  var sum:Int =0
  while (num!=0){
    sum = sum + num%10
    num = num/10
  }
  return sum
}

scala> summa(22)
val res0: Int = 2
```

21. Список содержит элементы одного, но любого типа. Получить список, содержащий каждый имеющийся элемент старого списка  $k$  раз подряд. Число  $k$  задается при выполнении программы.

```
scala> def frepeat(k:Int, n:List[Any]):List[Any]={
  n.flatMap(x=> List.fill(k)(x))
}

scala> frepeat(3, List('S','u','n'))
val res1: List[Any] = List(S, S, S, u, u, u, n, n, n)
```

24.  $f(m,n)$  - наименьшее общее кратное натуральных чисел  $m$  и  $n$

```
scala> def nod(a:Int, b:Int):Int = {  
  if (a==0) {return b}  
  return nod(b% a,a)  
}  
|      |      | def nod(a: Int, b: Int): Int  
  
scala> def nok(a:Int, b:Int):Int = {  
  return (a/nod(a,b))*b  
}  
|      | def nok(a: Int, b: Int): Int  
  
scala> nok(16,32)  
val res0: Int = 32
```

25. Список содержит элементы одного, но любого типа. Получить список, из элементов исходного, удаляя каждый  $k$ -й элемент. Число  $k$  задается при выполнении программы.

```
scala> def deleteFromList(k:Int, list:List[Any]):List[Any]={  
  list.zipWithIndex.filter(pair => (1+pair._2)%k !=0).map(_._1)  
}  
|      | def deleteFromList(k: Int, list: List[Any]): List[Any]  
  
scala> deleteFromList(3, List('F','l','o','w','e','r'))  
val res1: List[Any] = List(F, l, w, e)
```

26.  $f(n,k)$  - число размещений из  $n$  по  $k$ . Факториал не использовать.

```
scala> def razm(n:Int,k:Int) =(n to (n-k+1) by-1).foldLeft(1:Long)((x,y) => x * y)  
def razm(n: Int, k: Int): Long  
  
scala> razm(3,2)  
val res2: Long = 6
```

27. Список содержит элементы одного, но любого типа. Получить новый список, перемещая циклически каждый элемент на  $k$  позиций влево (при перемещении на одну позицию первый элемент становится последним, второй первым и так далее). Число  $k$  задается при выполнении программы. Если  $k$  отрицательное, то перемещение происходит вправо

```
scala> def sdvig(list:List[Any],k:Int) = {
  (0 to list.size -1).map(i=>list((i+(1-k/list.size)*list.size+k)%list.size)).toList
}

|      | def sdvig(list: List[Any], k: Int): List[Any]

scala> sdvig(List(1,2,3,4,5),2)
val res3: List[Any] = List(3, 4, 5, 1, 2)
```

28.  $f(n)$  - наибольшее совершенное число не превосходящее  $n$ . Совершенным называется натуральное число  $n$  равное сумме своих делителей, меньших  $n$ , например  $6 = 1 + 2 + 3$  ( $f(6) = 6$ ,  $f(7) = 6$ , ... ).

```
scala> def sover(n:Int): Int = (2 to n).filter(x=>(1 to x -1).filter(y =>x%y ==0).sum ==x).max
def sover(n: Int): Int

scala> sover(7)
val res4: Int = 6
```

29. Список содержит элементы одного, но любого типа. Получить два списка из элементов исходного, выбирая в первый элементы с четными индексами, а во второй с нечетными.

```
scala> import scala.collection.mutable.ListBuffer

def oneToTwo(list:List[Any]): List[List[Any]] ={
  var a = new ListBuffer[Any]
  var b = new ListBuffer[Any]
  (0 to list.size -1).foreach(x =>{
    if (x%2==0) { a +=list(x);}
    else { b += list(x);}
  })
  List(b.toList, a.toList)
}

import scala.collection.mutable.ListBuffer

scala>
scala> |      |      |      |      |      |      |      |      | def oneToTw

scala> oneToTwo(List(1,2,3,4,5,6,7,8))
val res5: List[List[Any]] = List(List(2, 4, 6, 8), List(1, 3, 5, 7))
```



30.  $f(n)$  - наибольшее из чисел от 1 до  $n$  включительно, обладающее свойством: сумма цифр  $n$  в некоторой степени  $> 1$  равна самому числу  $n$ .  
Пример:  $512 = 83$

```
scala> def maxWithPow(n: Int): Int = {  
  var max = 1  
  for (i <- 1 to n) {  
    var sum = 0  
    var num = i  
    while (num > 0) {  
      sum = sum + (num % 10)  
      num = num / 10  
    }  
    var power = 2  
    var res = 1  
    var flag = false  
    while (res < i) {  
      res = pow(sum, power).toInt  
      if (res > i) flag = false  
      else if (res == i) flag = true  
      else power = power + 1  
      if (sum == 1 || res == 1) {  
        flag = false  
        res = i  
      }  
    }  
    if (flag && (i > max)) max = i  
  }  
  return max  
}  
  
scala> maxWithPow(512)  
val res16: Int = 512  
  
scala> maxWithPow(511)  
val res17: Int = 81
```

31. Список в качестве элементов содержит кортежи типа: (n, s), где n — целые числа, а s — строки. Получить два списка из элементов исходного, выбирая в первый числа, а во второй строки из кортежей.

```
scala> import scala.collection.mutable.ArrayBuffer

def last(list:List[(Int,String)])={
  var ints = ArrayBuffer.empty[Int]
  var strings = ArrayBuffer.empty[String]
  for (i <- list) {
    ints +=i._1
    strings +=i._2
  }
  println(ints.toList)
  println(strings.toList)
}

import scala.collection.mutable.ArrayBuffer

scala> last(List[(Int,String)]((1,"sun"),(2,"sunshine"),(3,"sunflower")))
List(1, 2, 3)
List(sun, sunshine, sunflower)
```