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»

Кафедра Математической кибернетики и информационных технологий

Отчет по Таски

по дисциплине «Введение в ИТ»

Выполнил: студент группы БВТ903

Ндайисенга Жерар

Руководитель:

Марина Мосева Сергейвна

Москва 2021

1. Переменные res – это значения val или настоящие переменные var?
   Это настоящие переменные val.
2. Язык Scala позволяет умножать строки на числа – попробуйте выполнить выражение "crazy" * 3 в REPL. Что получилось в результате? Где в Scaladoc можно найти ее описание?

```
sbt:p01> console
[info] Starting scala interpreter...
Welcome to Scala 2.12.14 (Java HotSpot(TM) Client VM, Java 1.8.0_241).
Type in expressions for evaluation. Or try :help.

scala> "cryzy" * 3
res0: String = cryzycryzycryzy

scala>
```

3. Что означает выражение 10 max 2? В каком классе определен метод max?

```
scala> import scala.math.{abs, max}
import scala.math.{abs, max}

scala> max(10,2)
res2: Int = 10

scala>
```

   что этот метод возвращает большее из двух чисел. Этот метод не существует в Java, поэтому в RichInt

4. Используя число типа BigInt, вычислите 2^{1024} ?

```
scala> BigInt(2).pow(1024)
res3: scala.math.BigInt = 179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137216

scala>
```

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

   Нужно импортировать *scala.util.Random, scala.math.BigInt.probableInt* scala.math.BigInt

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

scala> import util.Random
import util.Random

scala> probablePrime(100,Random)
res4: scala.math.BigInt = 672300758500870286920561909003

scala>
```

6. Один из способов создать файл или каталог со случайным именемпреобразовать его в систему счисления по основанию 36, в результатеполучится строка, такая как "qsnvbevtomcj38o06kul". Отыщите в Scaladoc методы, которые можно было бы использовать для этого.

```
scala> probablePrime(100, Random).toString(36)
res5: String = 28y10nov70bfdxk8odft

scala>
```

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

```
scala> val x = "Moseva"
x: String = Moseva

scala> x.head
res6: Char = M

scala> x(0)
res7: Char = M

scala> x.last
res8: Char = a

scala>
```

или

```
scala> "Pgirard".head
res0: Char = P

scala> "Pgirard".last
res1: Char = d

scala>
```

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

   take: **Выбирает первые n элементов** // Selects the first n elements
   drop: **Выбирает все элементы, кроме первых n** // Selects all elements except first n ones
   takeRight: **Выбирает последние n элементов** // Selects the last n elements
   dropRight:**Выбирает все элементы, кроме последних** n // Selects all elements except last n ones

```
scala>  val z = "Marina"
z: String = Marina

scala> z.take(3)
res14: String = Mar

scala> z.drop(3)
res15: String = ina

scala> z.takeRight(1)
res16: String = a

scala> z.dropRight(1)
res17: String = Marin

scala>
```

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

```
scala> def fnum(N:Int) = if (N>0) 1 else if (N<0) -1 else 0
fnum: (N: Int)Int

scala> fnum(2020)
res3: Int = 1

scala> fnum(-2020)
res4: Int = -1
```

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

Unit = ()

```
scala> var fnum={}
fnum: Unit = ()

scala>
```

11. Напишите на языке Scala цикл, эквивалентный циклу на языке Java for (inti=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

scala>
```

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)
countdown: (n: Int)Unit

scala> countdown(12)
12
11
10
9
8
7
6
5
4
3
2
1
0

scala>
```

13. Напишите цикл for для вычисления кодовых пунктов Юникода всехбукв в строке. Например, произведение символов в строке «Hello» равно9415087488L.

```
scala> def fnum(str:String):Long={
     | var res:Long=1
     | for(i<-str) res = res * i.toLong
     | res
     | }
fnum: (str: String)Long

scala> fnum("Hello")
res7: Long = 9415087488

scala>
```

14. Решите предыдущее упражнение без применения цикла

```
scala> def fnum(s:String):Long={
     | if (s.length==1) return s.charAt(0).toLong
     | else s.take(1).charAt(0).toLong *fnum(s.drop(1))}
fnum: (s: String)Long

scala> fnum("Hello")
res8: Long = 9415087488

scala>
```

15. Напишите функцию product(s: String), вычисляющую произведение, как описано в предыдущих упражнениях.

```
scala> def fnum(s:String):Long={
     | if (s.length==1) return s.charAt(0).toLong
     | else s.take(1).charAt(0).toLong *fnum(s.drop(1))}
fnum: (s: String)Long

scala> fnum("Hello")
res8: Long = 9415087488

scala>
```

16. Сделайте функцию из предыдущего упражнения рекурсивной.

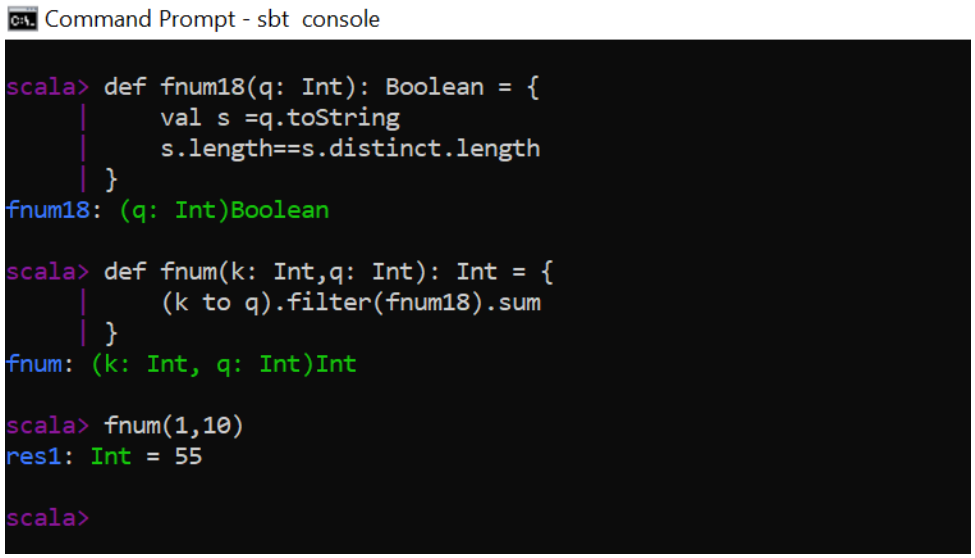
```
scala> def fnum16(string: String) :Long = {
     | if (string.size > 0)
     |    fnum16(string.tail)* string.head.toLong
     | else
     | 1
     | }
fnum16: (string: String)Long

scala> fnum16("Hello")
res8: Long = 9415087488

scala>
```

17. Напишите функцию, вычисляющую x^n, где n – целое число. Используйте следующее рекурсивное определение:

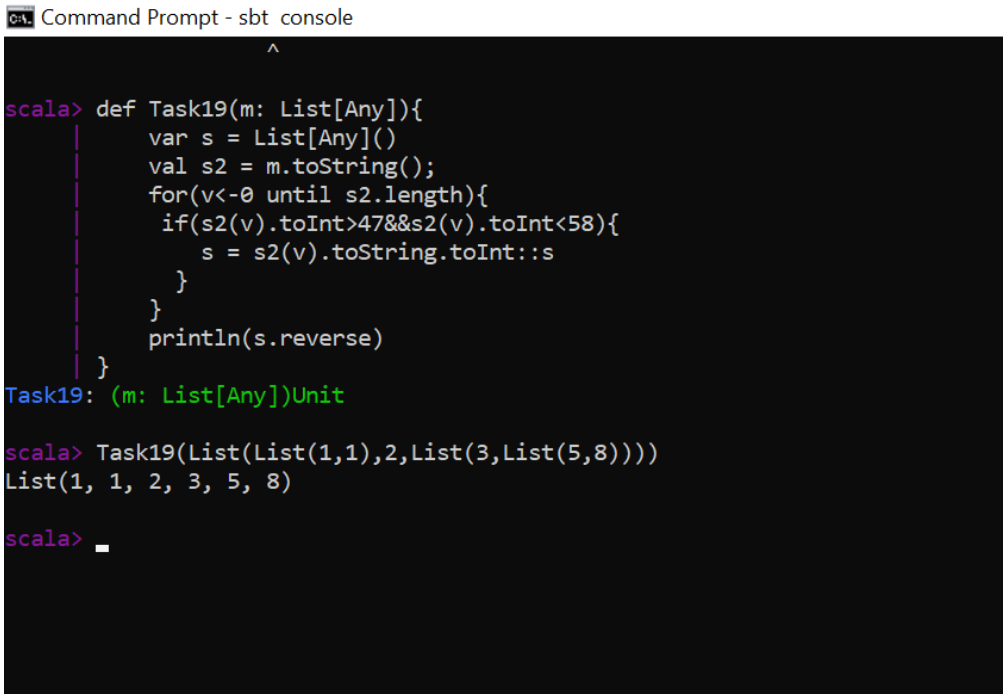
- $x^n = y^2$, если n – четное и положительное число, где $y=x^{n/2}$
- $x^n = x*x^{n-1}$, если n – нечетное и положительное число.
- $x^0=1$.
- $x^n=1/x^{-n}$, если n – отрицательное число. Не используйте инструкцию return.

```
scala> def fnum17(x:Double,n:Int):Double = {
     | if (n==0) 1
     | else {
     | if (n>0) {
     | if (n % 2 ==0 && n>2) {
     | fnum17(fnum17(x,n/2),2)
     | }else {
     | x *fnum17(x,n-1)
     | }
     | }else (1/fnum17(x,-n))
     |  }
     |  }
fnum17: (x: Double, n: Int)Double

scala> fnum17(2,2)
res9: Double = 4.0

scala> fnum17(4,2)
res10: Double = 16.0

scala>
```

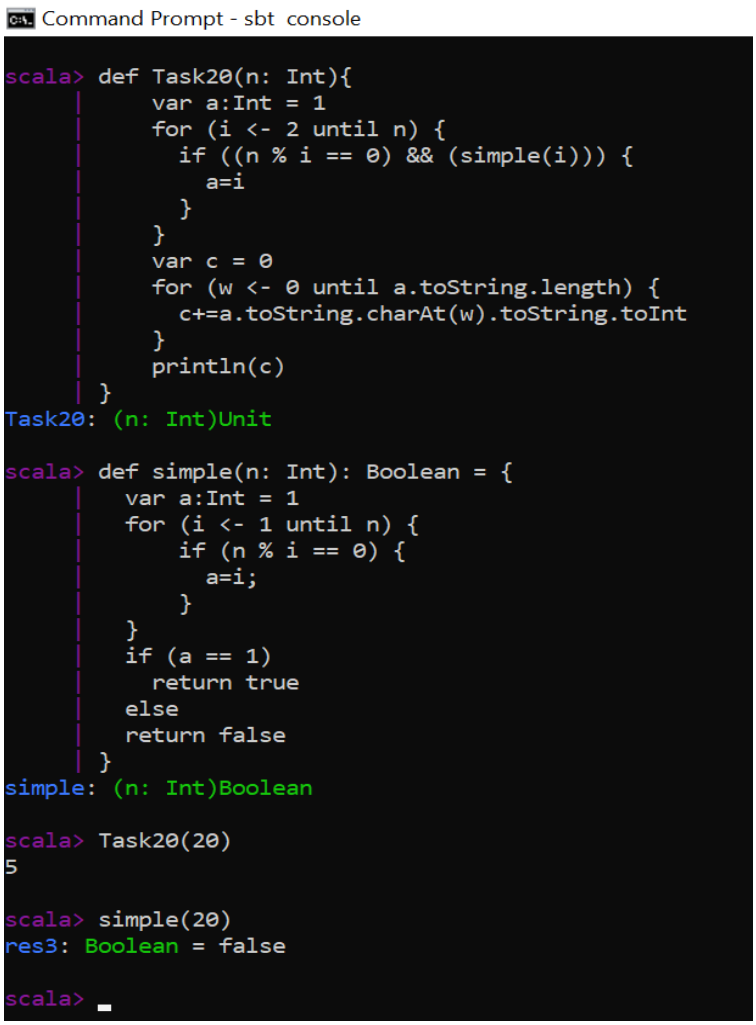
18. f(m,n) - сумма всех натуральных чисел от m до n включительно, в десятичной записи которых нет одинаковых цифр.



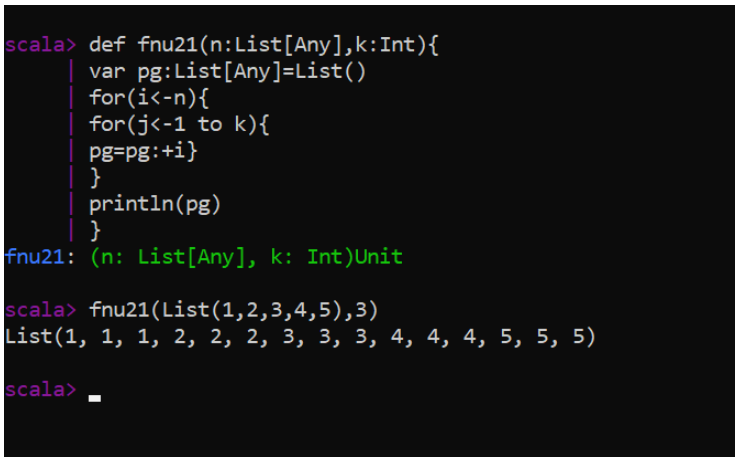
19.
Список содержит целые числа, а также другие списки, такие же как и первоначальный. Получить список, содержащий только целые числа из
всех вложенных списков. Пример:
f(List(List(1, 1), 2, List(3, List(5, 8)))) = List(1, 1, 2, 3, 5, 8)


Command Prompt - sbt console
^
scala> def Task19(m: List[Any]){
var s = List[Any]()
val s2 = m.toString();
for(v<-0 until s2.length){
if(s2(v).toInt>47&&s2(v).toInt<58){
s = s2(v).toString.toInt::s
}
}
println(s.reverse)
}
Task19: (m: List[Any])Unit
scala> Task19(List(List(1,1),2,List(3,List(5,8))))
List(1, 1, 2, 3, 5, 8)
scala>

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



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 fnu21(n:List[Any],k:Int){
     | var pg:List[Any]=List()
     | for(i<-n){
     | for(j<-1 to k){
     | pg=pg:+i}
     | }
     | println(pg)
     | }
fnu21: (n: List[Any], k: Int)Unit

scala> fnu21(List(1,2,3,4,5),3)
List(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5)

scala>
```

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

// 22 - смотри 20

23.Список содержит элементы одного, но любого типа. Получить список, содержащий каждый имеющийся элемент старого списка k раз

// 23 - смотри 21

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

```
scala> def gcd(a: Int, b: Int): Int = {
     |     if(b == 0){
     |       a
     |     }
     |     else {
     |       gcd(b,a% b)
     |     }
     | }
gcd: (a: Int, b: Int)Int

scala> // рекурсивный алгоритм вычисляющий общий делитель чисел

scala>

scala>

scala> def Task24(n: Int, m: Int){
     |     println(n / gcd(n, m) * m)
     | }
Task24: (n: Int, m: Int)Unit

scala> Task24(145,45)
1305

scala> Task24(205,35)
1435

scala> ~
```

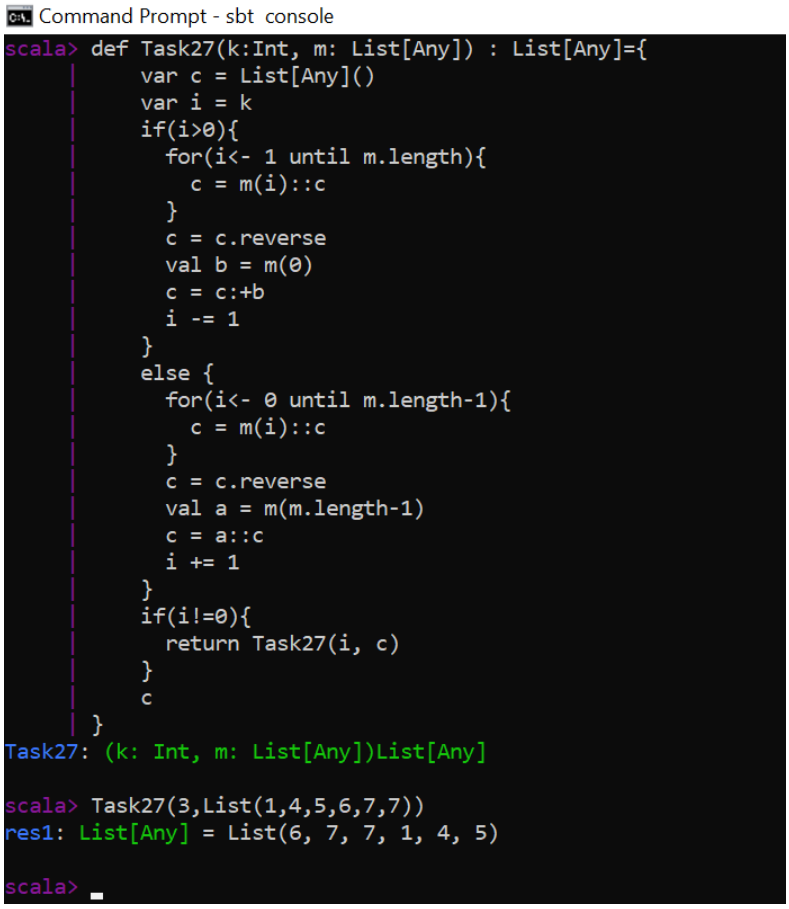
25.Список содержит элементы одного, но любого типа. Получить список, из элементов исходного, удаляя каждый k-й элемент. Число k задается при выполнении программы.

```
scala> def fnum25(n:List[Any] ,del:Int){
     | var res:List[Any]=List()
     | var count:Int=0;
     | for(i<-n){
     | count+=1
     | if(count<del) res=res:+i
     | else count=0
     | }
     | println(res)}
fnum25: (n: List[Any], del: Int)Unit

scala> fnum25(List(1,2,3,4,5,6,7,8,9),2)
List(1, 3, 5, 7, 9)

scala>
```

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

```
scala> def faktorial(n:Int):Int={
     |     if(n<2){
     |       1
     |     }
     |     else {
     |       n*faktorial(n-1)
     |     }
     | }
faktorial: (n: Int)Int

scala>

scala> def Task26(n: Int, k: Int){
     |     println(faktorial(n)/faktorial(n-k))
     | }
Task26: (n: Int, k: Int)Unit

scala> Task26(20,8)
-4

scala> Task26(10,4)
5040
```

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.



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, ...).

Command Prompt - sbt console

```
Task28: (n: Int)Int

scala> def fnum(n: Int): Int = {
     |     var c = 1
     |     for(v <- 2 until n){
     |       if(n%v==0)
     |         c+=v
     |     }
     |     c
     | }
fnum: (n: Int)Int

scala>

scala>

scala> def Task28(n: Int): Int ={
     |     for(v <- 0 to n-1){
     |       if(n-v==fnum(n-v))
     |         return n-v
     |     }
     |     0
     | }
Task28: (n: Int)Int

scala> Task28(141539)
res6: Int = 8128

scala>
```

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> def Task29(m: List[Any]){
     |     var c1 = List[Any]()
     |     var c2 = List[Any]()
     |     for(i<- 0 until m.length){
     |       if(i%2 == 0){
     |         c1 = m(i)::c1
     |       }
     |       else {
     |         c2 = m(i)::c2
     |       }
     |     }
     |     println(c1.reverse)
     |     println(c2.reverse)
     | }
Task29: (m: List[Any])Unit

scala> Task29(List(1,2,3,4,5,6,7,8,9))
List(1, 3, 5, 7, 9)
List(2, 4, 6, 8)

scala>
```

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 = 8^3

```
Command Prompt - sbt  console
             ^

scala> def Task30(n: Int): Int = {
     |     var count = 1
     |     var sum = 0
     |     for(i<-n to 1 by -1){
     |       var a = i
     |       while (a>0){
     |         sum+=a%10
     |         a/=10
     |       }
     |       var b = sum
     |       if(sum>1){
     |         while (b<i){
     |           b*= sum
     |           count+=1
     |         }
     |       }
     |       if(b == i && count !=1 ){
     |         print(sum, '^', count)
     |         return count
     |       }
     |       sum=0
     |     }
     |     0
     | }
Task30: (n: Int)Int

scala> Task30(512)
(8,^,3)res4: Int = 3

scala>
```

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> def Task31(m: List[(Int, String)]){
     |     var c1 = List[Int]()
     |     var c2 = List[String]()
     |     for(i<- 0 until m.length){
     |       c1 = m(i)._1::c1
     |       c2 = m(i)._2::c2
     |     }
     |     println(c1.reverse)
     |     println(c2.reverse)
     | }
Task31: (m: List[(Int, String)])Unit

scala> Task31(List((1, "a"), (2, "bb"), (3, "ccc")))
List(1, 2, 3)
List(a, bb, ccc)

scala>
```