Problem- Create a list of tuples, where the 1st element of the tuple is an int and the second element is a string.

Here we have created a tuple named as tuple1 which a list of tuples which contains an integer and a string in tuples as shown below-

```
scala> val tuple1 : List[(Int,String)] = List((1,"alpha"),(2,"beta"),(3,"gamma"),(4,"zeta"),(5,"omega"))
tuple1: List[(Int, String)] = List((1,alpha), (2,beta), (3,gamma), (4,zeta), (5,omega))

scala>
```

1. **For the above list, print the numbers where the corresponding string length is 4**

   Solution- We will run below command to find the numbers where the corresponding string length is 4-

   Tuple1.filter(_._2.length == 4).foreach(x=> println(x._1))

   It gives the result as 2,4 which is for **beta** and **zeta**

```
scala> tuple1.filter(_._2.length == 4).foreach(x=> println(x._1))
2
4
```

2. **Find the average of all numbers, where the corresponding string contains alphabet 'm' or alphabet 'z.**

We are again creating another instance of same list of tuples named as tup-

var tup = List((1,"alpla"),(2,"beta"),(3,"gamma"),(4,"zeta"),(5,"omega"))

Below command will first filter out those elements which contains the alphabet **m** or **z-**

var tup1 = tup.filter(a =>(a._2.count(_ == 'm') != 0 || a._2.count(_ == 'z') != 0))

Below command will calculate the average of the integers associated with those strings which were filtered out containing **m** or **z-**

tup1.map(_._1).sum/tup1.size

```
scala> tup.filter(a =>( a._2.count(_ == 'm') != 0 || a._2.count(_ == 'z') != 0)).map(_._1).sum
res15: Int = 12

scala> var tup1 = tup.filter(a =>( a._2.count(_ == 'm') != 0 || a._2.count(_ == 'z') != 0))
tup1: List[(Int, String)] = List((3,gamma), (4,zeta), (5,omega))

scala> tup1.map(_._1).sum/tup1.size
res16: Int = 4
```