Question 3: Copy the introductory code. Vector name stores the extracted names

```
library(stringr)

 raw.data <- "555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555-
6542Rev. Timothy Lovejoy555 8904Ned Flanders636-555-3226Simpson,
Homer5553642Dr. Julius Hibbert"

 names <- unlist(str_extract_all(raw.data, "[[:alpha:]., ]{2,}"))

 ## extracting firstname from vector name.

 firstname <-  unlist(str_extract_all(names,"[.,] [[A-z]]{2,}$|[[A-z]]{2,}
"))
 firstname  <- unlist(str_extract_all(firstname,"[[A-z]]{2,}"))
 firstname
## [1] "Moe"        "Montgomery" "Timothy"    "Ned"        "Homer"
## [6] "Julius"
 ##extracting lastname from vector name.

 lastname <-   unlist(str_extract_all(names,"[^[.,]] [[A-z]]{2,}$|[[A-
z]]{2,}, "))
 lastname <-   unlist(str_extract_all(lastname,"[[A-z]]{2,}"))
 lastname
## [1] "Szyslak"  "Burns"    "Lovejoy"  "Flanders" "Simpson"  "Hibbert"
 ## displaying dataframe Names with colnames.

Names <- data.frame(FirstName = c(firstname), LastName = c(lastname))
Names
##    FirstName LastName
## 1        Moe  Szyslak
## 2 Montgomery    Burns
## 3    Timothy  Lovejoy
## 4        Ned Flanders
## 5      Homer  Simpson
## 6     Julius  Hibbert
## Extracting titles from the name vector.

Person_Title <- unlist(str_extract_all(names,"[[A-z]]{2,}\\."))
Person_Title
## [1] "Rev." "Dr."
## checking names if contain titles.

Title_Exits <- data.frame(FullName = c(names), Title_Exists =
c(str_detect(names,Person_Title)))
Title_Exits <- data.frame(FullName = c(names), Title_Exists =
c(str_detect(names,Person_Title)))
Title_Exits
##               FullName Title_Exists
## 1          Moe Szyslak        FALSE
## 2 Burns, C. Montgomery        FALSE
## 3 Rev. Timothy Lovejoy         TRUE
## 4         Ned Flanders        FALSE
## 5       Simpson, Homer        FALSE
```

```
## 6   Dr. Julius Hibbert         TRUE
```

Question 4 : Describe the types of strings that conform to the following regular expressions and construct ans exmaple that is matched by the regular expression.

```
## [0-9]\\$
'Any digit between 0-9 ending with $ symbol'
## [1] "Any digit between 0-9 ending with $ symbol"
Regexpression_1 <- c("0", "123", "4444", "1234567", "11223$", "0$")
result_1 <- str_detect(Regexpression_1,"[0-9]\\$")
result_1
## [1] FALSE FALSE FALSE FALSE  TRUE  TRUE
##  \b[a-z]{1,4}\b
'Any 4 letters of lower case letters from a- z in a whole word'
## [1] "Any 4 letters of lower case letters from a- z in a whole word"
 Regexpression_2 <- c("a","2", "a2c", "ccc", "123567")
 result_2 <- str_detect(Regexpression_2,"\b[a-z]{1,4}\b")
 result_2
## [1] FALSE FALSE FALSE FALSE FALSE
 ## \d{2}/\d{2}/\d{4}
 'its a date format string with two digitsfollowed by forwared slash, again 2
digits followed by slash and 4 digits'
## [1] "its a date format string with two digitsfollowed by forwared slash,
again 2 digits followed by slash and 4 digits"
  Regexpression_3 <- c("1", "01/01/2018", "01/" , "1/01/0121", "11/2012")
  result_3 <- str_detect(Regexpression_3,"\\d{2}/\\d{2}/\\d{4}")
  result_3
## [1] FALSE  TRUE FALSE FALSE FALSE
  ##.*?\.txt$
  'any string followed by .txt at the end'
## [1] "any string followed by .txt at the end"
  Regexpression_4 <- c("x", "x.text", "x.txt", ".txt")
  result_4 <- str_detect(Regexpression_4, ".*?\\.txt$")
  result_4
## [1] FALSE FALSE  TRUE  TRUE
  ## <(.+?)>.+?</\1>
  'any html opening tag follwed tag text followed  by closing tag'
## [1] "any html opening tag follwed tag text followed  by closing tag"
   Regexpression_5 <- c("<div>element</div>", "<>none</>", "<head></head>",
"<title><title>")
   result_5 <- str_detect(Regexpression_5,"<(.+?)>.+?</\\1>")
   result_5
## [1]  TRUE FALSE FALSE FALSE
```

Question 9: Break the code.

```
code_Expression <-
"clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0Tanwo
Uwisdij7Lj8kpf03AT5Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigO
d6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5
fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr"
```

```
code_Expression
## [1]
"clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0Tanwo
Uwisdij7Lj8kpf03AT5Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigO
d6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5
fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr"
'using paste function to check if we can match a pattern'
## [1] "using paste function to check if we can match a pattern"
  'using all-lower string function'
## [1] "using all-lower string function"
 all_Lower <- str_extract_all(code_Expression,"[[:lower:].!]")
 all_Lower
## [[1]]
##   [1] "c" "l" "c" "o" "p" "o" "w" "z" "m" "s" "t" "c" "d" "w" "n" "k" "i"
##  [18] "g" "v" "d" "i" "c" "p" "u" "g" "g" "v" "h" "r" "y" "n" "j" "u" "w"
##  [35] "c" "z" "i" "h" "q" "r" "f" "p" "x" "s" "j" "d" "w" "p" "n" "a" "n"
##  [52] "w" "o" "w" "i" "s" "d" "i" "j" "j" "k" "p" "f" "d" "r" "c" "o" "c"
##  [69] "b" "t" "y" "c" "z" "j" "a" "t" "a" "o" "o" "t" "j" "t" "j" "n" "e"
##  [86] "c" "f" "e" "k" "." "r" "w" "w" "w" "o" "j" "i" "g" "d" "v" "r" "f"
## [103] "r" "b" "z" "." "b" "k" "n" "b" "h" "z" "g" "v" "i" "z" "c" "r" "o"
## [120] "p" "." "w" "g" "n" "b" "." "q" "o" "f" "a" "o" "t" "f" "b" "w" "m"
## [137] "k" "t" "s" "z" "q" "e" "f" "y" "n" "d" "t" "k" "c" "f" "g" "m" "c"
## [154] "g" "x" "o" "n" "h" "k" "!" "g" "r"
 all_upper <- unlist(str_extract_all(code_Expression, "[[:upper:].! ]"))
 all_upper
##  [1] "C" "O" "N" "G" "R" "A" "T" " " "U" "L" "A" "T" "I" "O" "N" "S" "."
## [18] "Y" "O" " " "U" "." "A" "R" "E" "." "A" "." "S" "U" "P" "E" "R" " "
## [35] "N" "E" "R" "D" "!"
code_word <- paste(all_upper, collapse = "")
code_word
## [1] "CONGRAT ULATIONS.YO U.ARE.A.SUPER NERD!"
 code_word <- str_replace_all(code_word, "[\\.]", " ")
 code_word
## [1] "CONGRAT ULATIONS YO U ARE A SUPER NERD!"
```