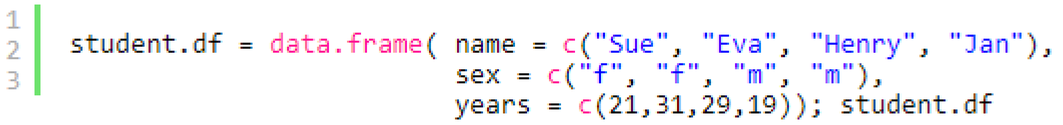
NAME: Hariprasad K K

REG NO : 19BCE7079

COURSE: DATA_ANALYTICS (LAB-2)

DATE: 24-10-2021

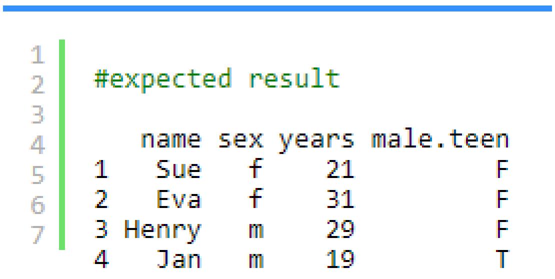
1) Create the data frame ‘student.df’ with the data provided below:



```
student.df = data.frame( name = c("Sue", "Eva", "Henry", "Jan"),
                         sex = c("f", "f", "m", "m"),
                         years = c(21,31,29,19)); student.df
```

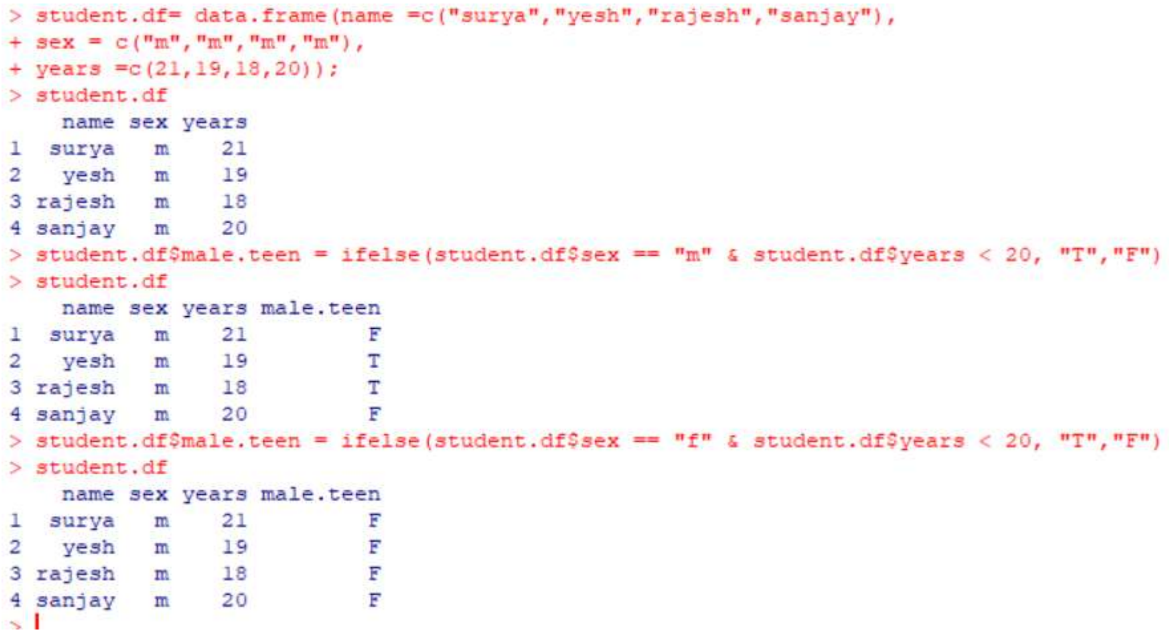
Use a simple ‘ifelse’ statement to add a new column ‘male.teen’ to the data frame. This is a boolean column, indicating T if the observation is a male younger than 20 years.

Expected result:



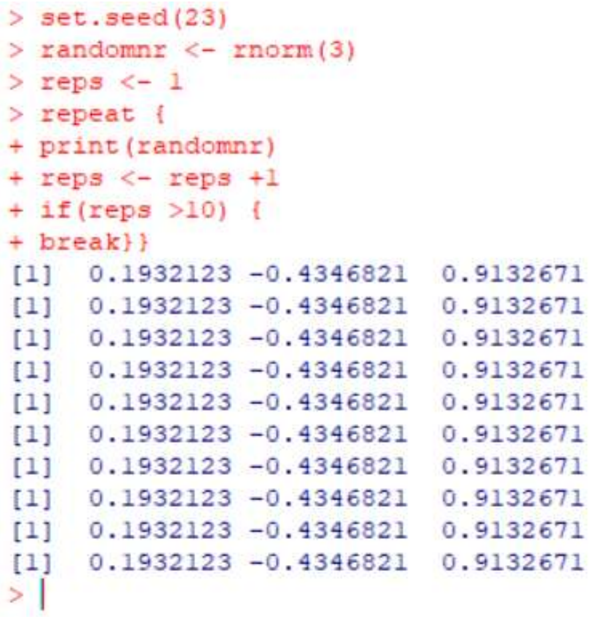
```
#expected result

   name sex years male.teen
1   Sue   f    21         F
2   Eva   f    31         F
3 Henry   m    29         F
4   Jan   m    19         T
```



```
> student.df= data.frame(name =c("surya","yesh","rajesh","sanjay"),
+ sex = c("m","m","m","m"),
+ years =c(21,19,18,20));
> student.df
    name sex years
1  surya   m    21
2   yesh   m    19
3 rajesh   m    18
4 sanjay   m    20
> student.df$male.teen = ifelse(student.df$sex == "m" & student.df$years < 20, "T","F")
> student.df
    name sex years male.teen
1  surya   m    21         F
2   yesh   m    19         T
3 rajesh   m    18         T
4 sanjay   m    20         F
> student.df$male.teen = ifelse(student.df$sex == "f" & student.df$years < 20, "T","F")
> student.df
    name sex years male.teen
1  surya   m    21         F
2   yesh   m    19         F
3 rajesh   m    18         F
4 sanjay   m    20         F
> 
```

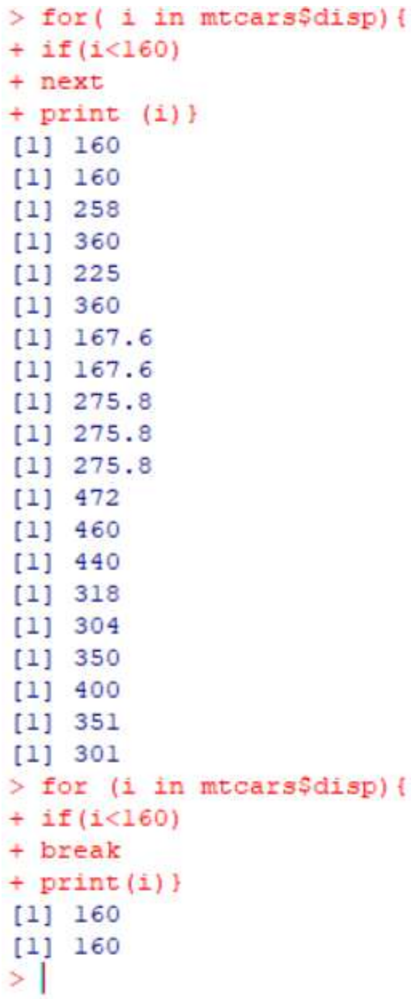
1) Write a repeat loop containing three random numbers. The loop repeats itself exactly ten times before it stops.



```
> set.seed(23)
> randomnr <- rnorm(3)
> reps <- 1
> repeat {
+ print(randomnr)
+ reps <- reps +1
+ if(reps >10) {
+ break}}
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
[1]  0.1932123 -0.4346821  0.9132671
> |
```

2) For loop combined with if statement

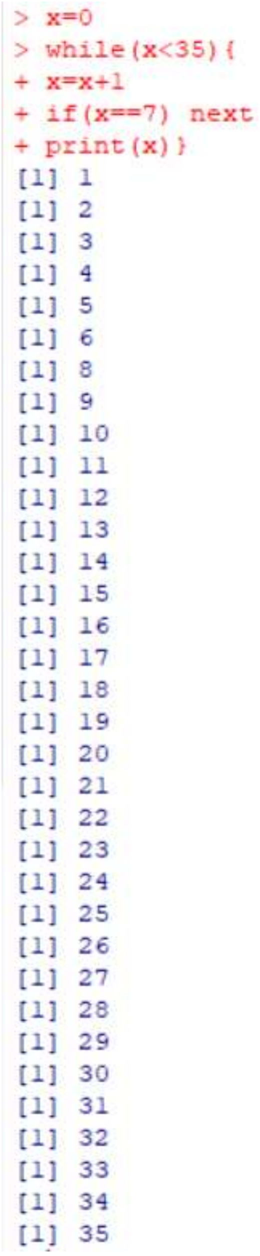
Write a for loop that prints the D*isplacement* ('disp') of the 'mtcars' dataset.
a. This loop will only print observations of 160 or higher in 'disp'.
b. This loop will stop as soon as an observation is smaller than 160 in 'disp'.



```
> for( i in mtcars$disp){
+ if(i<160)
+ next
+ print (i)}
[1] 160
[1] 160
[1] 258
[1] 360
[1] 225
[1] 360
[1] 167.6
[1] 167.6
[1] 275.8
[1] 275.8
[1] 275.8
[1] 472
[1] 460
[1] 440
[1] 318
[1] 304
[1] 350
[1] 400
[1] 351
[1] 301
> for (i in mtcars$disp){
+ if(i<160)
+ break
+ print(i)}
[1] 160
[1] 160
> 
```

3) Simple while loop

Write a while loop starting with x = 0. The loop prints all numbers up to 35 but it skips number 7.

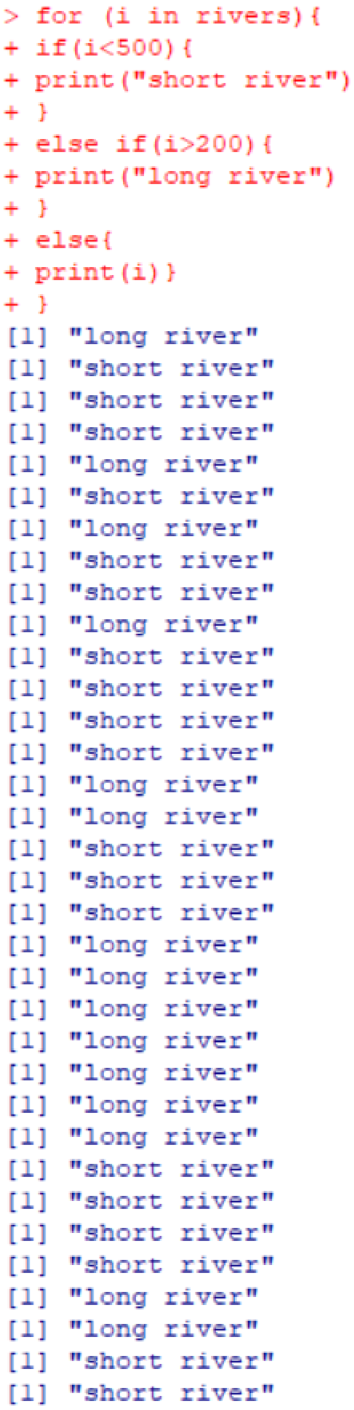


```
> x=0
> while(x<35){
+ x=x+1
+ if(x==7) next
+ print(x)}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
[1] 20
[1] 21
[1] 22
[1] 23
[1] 24
[1] 25
[1] 26
[1] 27
[1] 28
[1] 29
[1] 30
[1] 31
[1] 32
[1] 33
[1] 34
[1] 35
```

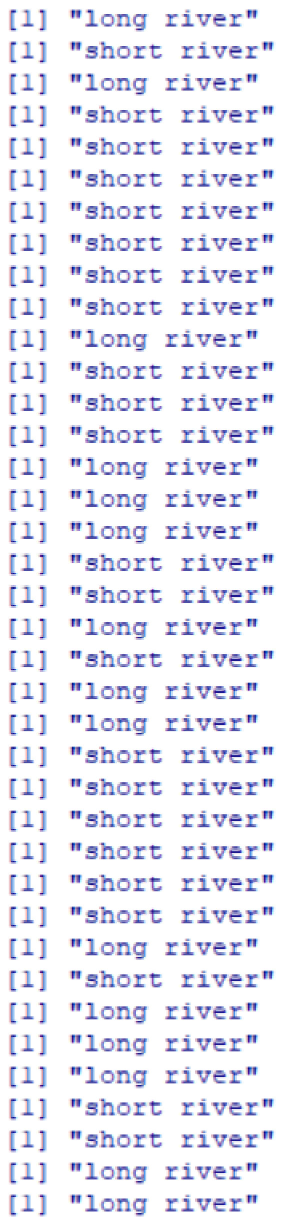
4) River classifications

Use the *‘rivers’* dataset to write a for loop. The loop prints the dataset:

- rivers shorter than 500 are a *‘short river’*;
- rivers longer than 2000 are a *‘long river’*;
- and rivers in the middle range are printed in their *original numbers*.



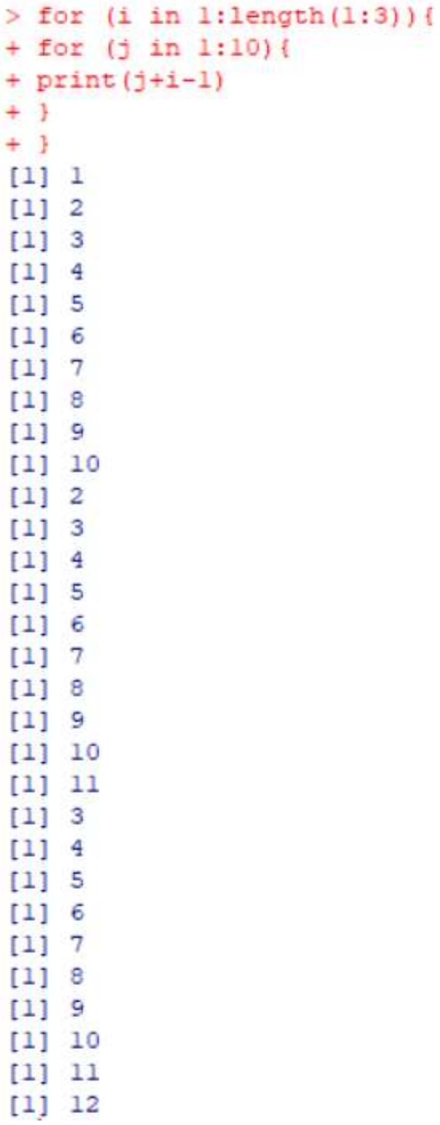
```
> for (i in rivers){
+ if(i<500){
+ print("short river")
+ }
+ else if(i>200){
+ print("long river")
+ }
+ else{
+ print(i)}
+ }
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "long river"
[1] "short river"
[1] "short river"
```



```
[1] "long river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "long river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "short river"
[1] "long river"
[1] "long river"
[1] "long river"
[1] "short river"
[1] "short river"
[1] "long river"
[1] "long river"
```

5) Double for loop

Write a double for loop which prints 30 numbers (1:10, 2:11, 3:12). Those are three clusters of ten numbers each. The first loop determines the number of clusters (3) via its length; the second loop the numbers to be printed (1 to 10 at the beginning). Each cluster starts one number higher than the previous one.



```
> for (i in 1:length(1:3)){
+ for (j in 1:10){
+ print(j+i-1)
+ }
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
```