

#####

```
# write a function that has two inputs:
# vec is a character vector
# x is a character
# your function should report the number of x
# that appears in vec.
```

```
# For example,
# if vec is "a" "a" "a" "b"
# x is "a"
# the result will be 3
```

```
report_character_no <- function(vec,x){
  return(result)
}
```

#####

```
# Write a function that finds the nth Fibonacci number for
# a given n.
# Fibonacci sequence is 1 1 2 3 5 8 13 21 (next number is sum of previous two)
# For example for n=3 function should return 2 and 13 for n=7
# Hint: Iterate until n by summing last two numbers. First two numbers are 1 and 1
```

```
fibon<-function(n){
  fibonnaci<-c(1,1)

  return(number)
}
```

#####

```
## right-triangle out of stars:
# draw a triangle with right angle (90 degrees) * character
# where length of the sides is given as argument
# dik ucgen
# examples:
# > printRightTriangle(2)
# [1] "*"
# [1] "***"
# > printRightTriangle(4)
# [1] "*"
# [1] "***"
# [1] "****"
# [1] "*****"
#
```

```
# Hints:
# use paste(..,sep="") and print(..) functions
# use nested loop
```

```
printRightTriangle <- function(x)
{
}
```

#####ANSWER KEY#####

#####

#Q1:

```
report_character_no <- function(vec,x){
  c_no <- 0

  for(c in vec){
    if(c == x){
      c_no <- c_no + 1
    }
  }

  return(c_no)
}
```

#####

#Q2:

```
fibon<-function(n){
  fibonnaci<-c(1,1)
  for(i in 3:n){fibonnaci[i]=fibonnaci[i-1]+fibonnaci[i-2]}
  number<-fibonnaci[n]
  return(number)
}
```

#####

#Q3:

```
printRightTriangle <- function(x)
{
  for (i in c(1:x))
  {
    str <- "";
    for (j in c(1:i))
    {
      str <- paste(str,"*",sep="");
    }
    print(str)
  }
}
```