Taniya Ahmed

21BDS0059

EXPERIMENT 1

BRANCHING

Q1. Alice is an engineer who has been tasked with making an automatic phone operator which lists out different options for a caller to dial on their number pad so that they are able to avail their desired service directly.

PROGRAM:

```
service_choice = "2"


chosen_service = switch(service_choice,
                        "1" = "Queries regarding changing of phone number.",
                        "2" = "Queries regarding roaming services.",
                        "3" = "Report security vulnerability.",
                        "4" = "Report missing number.",
                        "5" = "Queries regarding phone bill",
                        "6" = "Talk to an operator.")
print(chosen_service)
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> print(chosen_service)
[1] "Queries regarding roaming services."
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q2. Alice makes a dataset of people that she has collected in a survey and wants to check the age category the survey takers fall into. A program has been written to basket the people who decided to participate in the survey.

PROGRAM:

```
df = data.frame(name = c("Ram", "Sam", "Cam"), Age = c(34, 35, 56))

average_age = mean(df$Age)


if(average_age > 40){

  df$Category = ifelse(df$Age > 40, "Senior", "Junior")

  print("Average age is above 40 is categorized as senior and if below 40, is categorized as junior")

} else {

  df$Category = "General"

  print("Average age is 40 or below, no special categorization.")

}

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
[1] "Average age is above 40 is categorized as senior and if below 40, is categorized as junior"
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q3. Bob is putting entries into digital records of the customers in his bank and wants to evaluate the mean credit score his new customers have. The program solves this requirement for Bob.

PROGRAM:

```
df_credit = data.frame(name = c("Abra", "Alice", "Perim", "Rahul", "Naina", "Sarita", "Meenakshi"),
                       credit_score = c(890, 450, 233, 850, 566, 390, 684))
mean_credit = mean(df_credit$credit_score)


if(mean_credit > 600){
  print("Higher credit score bracket")
} else {
  print("Lower credit score bracket")
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
[1] "Lower credit score bracket"
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q4. Bob wants to check in the backend of his website if the users signing up are entering their credentials in the correct format he requires them to enter their credentials in upper case only. But in either case he writes a logic to keep a count of all the people who follow this format and in case some one doesn't follow this format he converts it to upper case for them.

PROGRAM:

```
name = "Sam"

upper_name = toupper(name)

count_upper = 0


if(name == upper_name){

 count_upper = count_upper + 1

 print("Already upper case")

} else {

 print("Not upper case")

}


print("Count of users correctly logging in their details")

print(count_upper)

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
[1] "Not upper case"
>
> print("Count of users correctly logging in their details")
[1] "Count of users correctly logging in their details"
> print(count_upper)
[1] 0
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q5. A website has been made which takes feedback from it's users and the maintainers of the website want to evaluate the responses of the users. An engineer has made some categories according to the standards the team wants to maintain and has written some code for it.

PROGRAM:

```
response = 98

if(reponse > 90){

  print("Excellent response")

} else if(response > 80 & response < 90){

  print("Very good response")

} else if(response > 70 & response < 80){

  print("Okay response")

} else {

  print("Bad response")

}

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
[1] "Excellent response"
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q6. Alice has been asked to make charts for visualization of some random data that she has received and has been given some specific conditions. If the length of the data is less 10 then she shouldn't perform any kind of visualization, otherwise if the length of data is equal to 20 she should plot a histogram else she should make a pie chart.

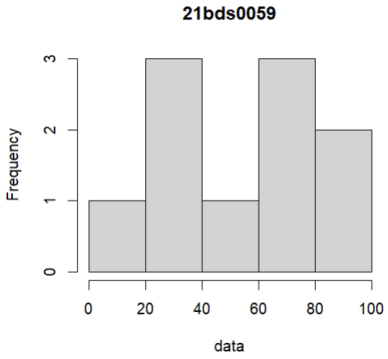
PROGRAM:

```
data = c(13, 43, 24, 68, 34, 90, 77, 24, 95, 76)


if(length(data) < 10){
 print("Will not display charts.")
} else {
 if(length(data) < 20){
  hist(data)
 } else if(length(data) == 20){
  boxplot(data)
 }
 else{
  pie(data)
 }
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

Histogram of Data



LOOPING

Q1. A data scientist has been given the task to calculate some summary statistics on the rent in various cities in India. The scientist faces some obstacles in the form of missing data. He therefore has to write code to impute these missing values and then calculate the statistics in code.

PROGRAM:

```
df = data.frame("City" = c("Mumbai", "Chennai", "Bangalore", "Kolkata"), Average_rent = c(98000, NA, 62000, 38000))

mean_rent = mean(df$Average_rent, na.rm = TRUE)


for(i in 1:nrow(df)){ # check

 if(is.na(df$Average_rent[i])){

   df$Average_rent[i] <- mean_rent

 }

}


print(df)

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> print(df)
       City Average_rent
1    Mumbai        98000
2   Chennai        66000
3 Bangalore        62000
4   Kolkata        38000
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q2. Juan has been assigned by his math teacher to find which is special and has certain properties. These special numbers are unique because when each digit in said number is raised to the number of digits in this special number it results in the number being generated again. To make the calculation even faster Juan has written the following code.

PROGRAM:

```
num = 12345

temp = num

sum = 0

num_digit = nchar(as.character(num))


while(num > 0){

 r = num %% 10

 sum = sum + (r ^ num_digit)

 num = num %/% 10

}


print(sum == temp)

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> print(sum == temp)
[1] FALSE
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q3. Bob learnt in his botany class that there is a series in math that explains the number of petals in flowers in nature very well. The series follows the pattern of a number being a summation of the previous two numbers in that series. He makes a program to calculate the number of petal numbers possible for the ith index in the series.

PROGRAM:

```
fibonacci = function(a){
 if(a <= 1) return(a)

 return(fibonacci(a - 1) + fibonacci(a - 2))
}

print(fibonacci(8))
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> print(fibonacci(8))
[1] 21
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q4. New shop owners have been trying to get ratings on their service but instead of analysing each score they made a program to directly print the response of the customer.

PROGRAM:

```
response = c(45, 50, 94, 57, 87)


for(i in response){
 if(i > 90){
   print(i)
   print("Excellent response")
 } else if(i > 80 & i < 90){
   print(i)
   print("Very good response")
 } else if(i > 70 & i < 80){
   print(i)
   print("Okay response")
 } else {
   print(i)
   print("Bad response")
 }
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
[1] 45
[1] "Bad response"
[1] 50
[1] "Bad response"
[1] 94
[1] "Excellent response"
[1] 57
[1] "Bad response"
[1] 87
[1] "Very good response"
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q5. Roah is playing a game where he has to clap every time a number which is a multiple of 7 comes up. He makes a program using a different looping concept to clap everytime a multiple of 7 comes up.

PROGRAM:

```
iter = 1


repeat{
 if(iter %% 7 == 0){
  print("Clap")
  print(iter)
 }

 iter = iter + 1


 if(iter > 30){
  break
 }
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
[1] "Clap"
[1] 7
[1] "Clap"
[1] 14
[1] "Clap"
[1] 21
[1] "Clap"
[1] 28
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

Q6. A student wants to understand the encryption hat goes into making websites more secure and decides to implement the famous Rivest – Shamir Algorithm which utilizes the uniqueness of large prime numbers and therefore needs to make a module that efficiently generates large prime numbers.

PROGRAM:

```
generate_prime = function(n){
 prime = rep(TRUE, n + 1)
 prime[1] = FALSE
 prime[2] = TRUE

 for(i in 2 : sqrt(n)){
  if(prime[i] == TRUE){
   for(j in seq(i ^ 2, n, i)){
    prime[j] = FALSE
   }
  }
 }

 which(prime)
}

n = 50
prime_number = generate_prime(n)
print(prime_number)
```

OUTPUT:

```
> n = 50
> prime_number = generate_prime(n)
> print(prime_number)
 [1]  2  3  5  7 11 13 17 19 23 29 31 37 41 43 47 51
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```