**Data Science with R Training**

**Day 2 Assignment Question:**

1. Make Scientific Calculator using Switch statement

Code:

cat("enter first number")

a=scan()

cat("enter second number")

b=scan()

cat("n")

n=scan()

for(i in 1:n){

cat(i)

c[i]=scan(what="character")

switch(c[i],"+"=print(a[i]+b[i]),"-"=print(a[i]-b[i]),"\*"=print(a[i]\*b[i]),"%"=print(a[i]%%b[i]))}

Output:

enter first number1: 12: 23: 34: Read 3 itemsenter second number1: 22: 33: 44: Read 3 itemsn1: 32: Read 1 item11: +2: Read 1 item[1] 3

21: -2: Read 1 item[1] -1

31: \*2: Read 1 item[1] 12

2. Use Loop statement to make following patterns

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Code:for(i in 1:5)

{for(j in 1:i){

cat("\*")

}

cat("\n")

}

Output:

|  |
| --- |
| > source('~/rzeal/pattern1.r')\*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\* |
|  |
| |  | | --- | | > | |

  1

1 2 1

1 2 3 2 1

Code:

for(i in 1:5)

{

for(j in 5:i)

{ cat(" ")}

for(k in 1:i)

{cat(k)

}

for(q in i-1:i)

{

if(q==0)

cat(" ")

else cat(q)}

cat("\n")

}

Output:

> source('~/rzeal/pattern2.r') 1

121

12321

1234321

123454321

3. Find Prime Numbers from 1 to number entered by user

n=scan()

cat(2)

cat(" ")

for(i in 2:n)

{ prime=1

for(j in 2:(i/2))

{

if((i%%j)==0)

{

prime=0

break

}

}

if(prime==1)

{ cat(i)

cat(" ")

}

}

**OUTPUT**

source('~/prime.R')

1: 100

2:

Read 1 item

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

4. Find Factorial of a Number

Code:

a=scan()

b=1

if(a<=0)

{ cat(1)}

for(i in 1:a)

{b=(b\*i)}

cat(b)

Output:

> source('~/rzeal/factorial.r')1: 5: Read 1 item120

1. Generate Fibonacci Series

Code:

n=scan()

a=0

b=1

cat(a)

cat(" ")

cat(b)

cat(" ")

for(i in 1:n)

{c=a+b

cat(c)

cat(" ")

a=b

b=c

}

Output:

|  |
| --- |
| > source('~/rzeal/fibonacci.r')1: 102: Read 1 item0 1 1 2 3 5 8 13 21 34 55 89 |
|  |
| |  | | --- | |  | |

6. Calculate the Prime Number of 1 to 100 and break once first 5 prime numbers are found.

**CODE**

count=1

cat(2)

cat(" ")

for(i in 2:100)

{ prime=1

for(j in 2:(i/2))

{

if((i%%j)==0)

{

prime=0

break

}

}

if(prime==1)

{ cat(i)

cat(" ")

count=count+1

if(count==5)

break } }

**OUTPUT**

> source('~/prime6.R')

2 3 5 7 11

1. Calculate the Prime Number from 1 to 100 but skip for 6th Number is the series.

count=1

cat(2)

cat(" ")

for(i in 2:100)

{ prime=1

for(j in 2:(i/2))

{

if((i%%j)==0)

{

prime=0

break

}

}

if(prime==1)

{ count=count+1

if(count==6)

next

else{cat(i)

cat(" ")}

} }

**OUTPUT**

|  |
| --- |
| > source('~/prime7.R')  2 3 5 7 11 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 |

8. Write a Program to perform multiple operations in Switch

cat("enter first number")

a=scan()

cat("enter second number")

b=scan()

cat("n")

n=scan()

for(i in 1:n){

cat(i)

c[i]=scan(what="character")

switch(c[i],"+"=print(a[i]+b[i]),"-"=print(a[i]-b[i]),"\*"=print(a[i]\*b[i]),"%"=print(a[i]%%b[i]))}

Output:

enter first number1: 12: 23: 34: Read 3 itemsenter second number1: 22: 33: 44: Read 3 itemsn1: 32: Read 1 item11: +2: Read 1 item[1] 3

21: -2: Read 1 item[1] -1

31: \*2: Read 1 item[1] 12

**Data Science with R Training**

**Day 3 Assignment Question:**

1. Consider a vector: x <- c(4,6,5,7,10,9,4,15), What is the value of: c(4,6,5,7,10,9,4,15) < 7

> x<-c(4,6,5,7,10,9,4,15)> x<7[1] TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE

Consider two vectors: p <- c (3, 5, 6, 8) and q <- c (3, 3, 3), What is the value of: p+q

> p<-c(3,5,6,8)> q<-c(3,3,3)> p+q[1] 6 8 9 11

3. If: Age <- c(22, 25, 18, 20)

Name <- c("James", "Mathew", "Olivia", "Stella")

Gender <- c("M", "M", "F", "F")

then what is the R-code for getting the following output;

## Age Name Gender

## 1 22 James M

## 2 25 Mathew M

> age<-c(22,23,24,25)> name<-c("saswat","manu","priyanshu","aayushi")> gender<-c> gender<-c("m","f","m","f")> table<-data.frame(name,age,gender)> table name age gender

1 saswat 22 m

2 manu 23 f

3 priyanshu 24 m

4 aayushi 25 f

> table[table$gender %in% "m",] name age gender

1 saswat 22 m

3 priyanshu 24 m

4. If z <- 0:9, then what is the output from the following R-statements:

digits <- as.character(z)

as.integer(digits)

> z<-0.9> digits<-as.character(z)> digits[1] "0.9"> as.integer(digits)[1] 0

5. Consider the vector: x <- c(1,2,3,4), What is the value of k for:

(x+2)[(!is.na(x)) & x > 0] -> k.

> x<-c(1,2,3,4)> k<-(x+2)[!is.na(x) & x>0]> k[1] 3 4 5 6

1. If x <- c(2, 4, 6, 8) and y <- c(TRUE, TRUE, FALSE, TRUE), What is the value of: sum(x[y])

> x<-c(2,4,6,8)

> y<-c(TRUE,TRUE,FALSE,TRUE)

> sum(x[y])

[1] 14

7. Consider the vector: x <- c(34, 56, 55, 87, NA, 4, 77, NA, 21, NA, 39)

Write a R-statement that will count the number of NA values in x?

> x<-c(34,56,55,87,NA,4,77,NA,21,NA,39)

> x

[1] 34 56 55 87 NA 4 77 NA 21 NA 39

> k<-c(is.na(x))

> k

[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE

> table(is.na(x))

FALSE TRUE

8 3

8. Consider two vectors, x, y

x=c(4,6,5,7,10,9,4,15)

y=c(0,10,1,8,2,3,4,1)

What is the value of: x\*y

> x<-c(4,6,5,7,10,9,4,15)

> y<-c(0,10,1,8,2,3,4,1)

> x\*y

[1] 0 60 5 56 20 27 16 15

9. Consider two vectors, a, b

a=c(1,2,4,5,6)

b=c(3,2,4,1,9)

What is the value of: cbind(a,b)

10. Consider two vectors, a, b

a=c(1,5,4,3,6)

b=c(3,5,2,1,9)

What is the value of: a<=b

11. Consider two vectors, a, b

a=c(10,2,4,15)

b=c(3,12,4,11)

What is the value of: rbind(a,b)

QUESTION.9,10,11

> a=c(1,2,4,5,6)

> b=c(3,2,4,1,9)

> cbind(a,b)

a b

[1,] 1 3

[2,] 2 2

[3,] 4 4

[4,] 5 1

[5,] 6 9

> a<=b

[1] TRUE TRUE TRUE FALSE TRUE

> rbind(a,b)

[,1] [,2] [,3] [,4] [,5]

a 1 2 4 5 6

b 3 2 4 1 9

12. If x=c(1:12)

What is the value of: length(x)

> x=c(1:12)

> d<-length(x)

> d

[1] 12

13. If a=c(12:5)

What is the value of: is.numeric(a)

QUESTION.13

> a=c(12:5)

> is.numeric(a)

[1] TRUE

14. Consider two vectors, x, y

x=letters[1:10]

y=letters[15:24]

What is the value of: x<y

> x=letters[1:10]> y=letters[15:24]> x<y [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

15. If x=c('blue','red','green','yellow')

What is the value of: is.character(x)

> x=c("blue","red","green","yellow")

> is.character(x)

[1] TRUE

16. If x=c('blue',10,'green',20)

What is the value of: is.character(x)

> x=c('blue',10,'green',20)> is.character(x)[1] TRUE

17. The numbers below are the 1rst ten days of rainfall amounts. Read them into a vector using the c() function - 0.1 0.6 33.8 1.9 9.6 4.3 33.7 0.3 0.0 0.1(Answer the following questions)

* + Mean and Standard Deviation of RainFall

> rain<-c(0.1,0.6,33.8,1.9,9.6,4.3,33.7,0.3,0.0,0.1)

> mean(rain)

[1] 8.44

> sd(rain)

[1] 13.66473

* + Calculate the cumulative rainfall (’running total’) over these ten days

> sum(rain)

[1] 84.4

* + Which day saw the highest rainfall (write code to get the answer)?

> max(rain)

[1] 33.8

**Data Science with R Training**

**Day 4 Assignment Question:**

1. Write a function that adds two numbers, and divides the result by 2

add\_num2<-function(x,y){

return((x+y)/2)

}

Version:1.0 StartHTML:0000000107 EndHTML:0000001133 StartFragment:0000000127 EndFragment:0000001115

> source('~/rzeal/function.r')> add\_num2(2,8)[1] 5

1. write a function called firstTwoChars that extracts the 1rst two characters of any bit of text.

firstTwoChars<-function(){

c=scan(what = "character")

return(substr(c,1,2))

}

Version:1.0 StartHTML:0000000107 EndHTML:0000001392 StartFragment:0000000127 EndFragment:0000001374

> firstTwoChars()1: aayushyamaan2: Read 1 item[1] "aa"

1. Write a function that will print missing values in any data

a=scan()

d=c(which(is.na(a)))

cat("\n Values of NA are at positions", d)

cat("\nLength of NA is ",length(d))

> source('~/missingdata.R')

1: 1

2: 2

3: 3

4: NA

5: 4

6: NA

7: 5

8: 7

9: 2

10: NA

11:

Read 10 items

Values of NA are at positions 4 6 10

Length of NA is 3

1. Write a function called getAge that asks the user to type his/her age.(If age is less than 18 print “Not eligible to Vote”)

getAge<-function(){

a=scan()

if(a<18){

cat("not eligible to vote")

}else{

cat("eligible to vote")

}

}

Version:1.0 StartHTML:0000000107 EndHTML:0000001383 StartFragment:0000000127 EndFragment:0000001365

> getAge()1: 232: Read 1 itemeligible to vote

1. Write a function that returns the con1dence interval for a vector. The function should have two inputs: the vector, and the desired ’alpha’.
2. Make Scientific Calculator using Switch statement using functions

calculate<-function(){

cat("enter first number")

a=scan()

cat("enter second number")

b=scan()

cat("n")

n=scan()

for(i in 1:n){

cat(i)

c[i]=scan(what="character")

switch(c[i],"+"=print(a[i]+b[i]),"-"=print(a[i]-b[i]),"\*"=print(a[i]\*b[i]),"%"=print(a[i]%%b[i]))}

}

Version:1.0 StartHTML:0000000107 EndHTML:0000004466 StartFragment:0000000127 EndFragment:0000004448

> calculate()enter first number1: 22: 33: 44: Read 3 itemsenter second number1: 42: 53: 64: Read 3 itemsn1: 32: Read 1 item11: +2: Read 1 item[1] 6

21: \_2: Read 1 item31: \*2: Read 1 item[1] 24

1. Write a function to find Prime Numbers from 1 to number entered by user

prime<-function()

{ n=scan()

cat(2)

cat(" ")

for(i in 2:n)

{ prime=1

for(j in 2:(i/2))

{

if((i%%j)==0)

{

prime=0

break

}

}

if(prime==1)

{ cat(i)

cat(" ")

}

}

}

|  |
| --- |
| > source('~/prime1.R')  > prime()  1: 100  2:  Read 1 item  2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67  71 73 79 83 89 97 |

8. Write a recursive function Find Factorial of a Number

recursive.fact<-function(x) {

if (x == 0) return (1)

else return (x \* recursive.fact(x-1))

}

Version:1.0 StartHTML:0000000107 EndHTML:0000000928 StartFragment:0000000127 EndFragment:0000000910

> recursive.fact(5)[1] 120

9Write a function Generate Fibonacci Series upto n numbers entered by user

fibonacci<-function(n){

a=0

b=1

cat(a)

cat(" ")

cat(b)

cat(" ")

for(i in 1:n)

{c=a+b

cat(c)

cat(" ")

a=b

b=c

}

}

Version:1.0 StartHTML:0000000107 EndHTML:0000000984 StartFragment:0000000127 EndFragment:0000000966

> fibonacci(10)0 1 1 2 3 5 8 13 21 34 55 89

1. Write a function to check whether given number is Armstrong’s Number or not.

count=0

sum=0

armstrong<-function(x)

{ num=x

num1=x

while(x!=0)

{

y=x%%10

x=x%/%10

count=count+1

}

cat ("\ncount=",count," ")

while(num!=0)

{

y=num%%10

num=num%/%10

sum=sum+(y^count)

}

cat ("\n sum=",sum," ")

if (num1==sum)

cat("\n\n armstrong")

else print("\n\n not armstrong")

}

> source('~/armstrong.R')

> armstrong(1634)

count= 4

sum= 1634

armstrong

11. Write a function to find Sum of Natural Numbers using Recursion

sum <- function() {

cat("enter number")

num<-scan()

sum=0

while(num>=1)

{

sum=sum+num

num=num-1

}

return(sum)

}

Version:1.0 StartHTML:0000000107 EndHTML:0000001383 StartFragment:0000000127 EndFragment:0000001365

> sum()enter number1: 102: Read 1 item[1] 55

12. Write a Function to find L.C.M

a=scan()

b=scan()

if(a>b){

c=a}else c=b

while(c>=2)

{

if(c%%a==0&&c%%b==0)

{lcm=c

cat("lcm",lcm)

break}

else {c=c+1}

}

|  |
| --- |
| > source('~/.active-rstudio-document')  1: 4  2: Read 1 item  1: 2  2: Read 1 item  lcm 4  > source('~/.active-rstudio-document')  1: 7  2: Read 1 item  1: 4  2: Read 1 item  lcm 28 |
|  |
| |  | | --- | | > | |

1. Write a Function to Calculate H.C.F or G.C.D

hcf<-function(x,y)

{

if(x<y)

smaller=x

else

smaller=y

for(i in 1:smaller)

{

if((x%%i==0)&&(y%%i==0))

hcf=i

}

return (hcf)

}

> source('~/hcf.R')

> hcf(16,12)

[1] 4

14. Write a Function to Find Factors of a Number

Code:

n=scan()

for(i in 1:n){

if(n%%i==0){

cat(i," ")

}

}

Output:

|  |
| --- |
| > source('~/.active-rstudio-document')  1: 60  2: Read 1 item  1 2 3 4 5 6 10 12 15 20 30 60 |
|  |
| |  | | --- | |  | |

15. Write a Function to Convert Binary to Decimal

dec=0

i=0

bi2dec<-function(a)

{

while(a!=0)

{

rem=a%%10

a=a%/%10

dec<-dec+rem\*(2^i)

i=i+1

}

return (dec)

}

|  |
| --- |
| > source('~/bi2dec.R')  > bi2dec(00111)  [1] 7 |

15**. Write a Function to find whether the year entered by user is**

**Leap year or not**

leap<-function(year)

{

if(year%%4==0)

{

if(year%%100==0)

{

if(year%%400==0)

print("leap year")

else

print("not leap year")

}

else print("leap year")

}

else print("not leap year")

}

> source('~/leap.R')

> leap(1900)

[1] "not leap year"

> leap(2016)

[1] "leap year"