ITA0448 - R PROGRAM
Vishwa
192124161

1. Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.

code:

```
name = readline(prompt="Input your name: ")
age =  readline(prompt="Input your age: ")
print(paste("My name is",name, "and I am",age ,"years old."))
print(R.version.string)
```

input:

```
Name:naresh
age:20
```

output:

```
[1] "My name is naresh and I am 20 years old."
[1] "R version 4.2.2 (2022-10-31 ucrt)"
>
```

2. Write a R program to get the details of the objects in memory.

code:

```
name = "python";
n1 =   10;
n2 =   0.5
nums = c(10, 20, 30, 40, 50, 60)
print(ls())
print("Details of the objects in memory:")
print(ls.str())
```

output:

```
rec : function (s, e, i, n)
s :   num 55
v :   NULL
w :   num [1:10] 63 81 56 91 47 57 76 72 62 48
```

3. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

code:

```
rint("Sequence of numbers from 20 to 50:")
```

```
print(seq(20,50))
print("Mean of numbers from 20 to
60:")
print(mean(20:60))
print("Sum of numbers from 51 to
91:")
print(sum(51:91))
```

output:

```
> print(sum(51:91))
[1] 2911
```

4. Write a R program to create a vector which contains 10 random integer values between -50 and +50.

code:

```
v = sample(-50:50, 10, replace=TRUE)
print("Content of the vector:")
print("10 random integer values
between -50 and +50:")
print(v)
```

output:

```
> v = sample(-50:50, 10,
replace=TRUE)
```

5. Write a R program to get all prime numbers up to a given number (based on
the sieve of Eratosthenes).

Code:

Input:

```
prime_numbers <- function(n) {
if (n >= 2) {
 x = seq(2, n)
 prime_nums = c()
 for (i in seq(2, n)) {
 if (any(x == i)) {
 prime_nums = c(prime_nums, i)
 x = c(x[(x %% i) != 0], i)
 }
 }
 return(prime_nums)
 }
 else
 {
 stop("Input number should be at
least 2.")
 }
 }
prime_numbers(12)
```

Output:

```
[1]  2  3  5  7 11
```

6. Write a R program to extract first 10 english letter in lower case and last 10 letters in upper case and extract letters between 22 nd to 24 th letters in upper case.

```
Input:print("First 10 letters in lower case:")
t = head(letters, 10)
print(t)
print("Last 10 letters in upper case:")
t = tail(LETTERS, 10)
print(t)
print("Letters between 22nd to 24th letters in upper case:")
e = tail(LETTERS[22:24])
print(e)
```

Output:

```
[1]"First 10 letters in lower case:"[1]"a""b""c""d""e""f""g""h""i""j"[1]"Last 10 letters in upper case:"[1]"Q""R""S""T""U""V""W""X""Y""Z"
```

7. Write a R program to find the maximum and the minimum value of a given
vector.

Code:

Input:

```
nums = c(10, 20, 30, 40, 50, 60)
print('Original vector:')
print(nums)
print(paste("Maximum value of the said vector:",max(nums)))
print(paste("Minimum value of the said vector:",min(nums)))
```

Output:

```
[1] "Original vector:"
[1] 10 20 30 40 50 60
[1] "Maximum value of the said vector: 60"
[1] "Minimum value of the said vector: 10"
```

8. Write a R program to get the unique elements of a given string and unique numbers of vector.

Code:

Input:

```
str1 = "The quick brown fox jumps over the lazy dog."
```

```
print("Original vector(string)")
print(str1)
print("Unique elements of the said
vector:")
print(unique(tolower(str1)))
nums = c(1, 2, 2, 3, 4, 4, 5, 6)
print("Original vector(number)")
print(nums)
print("Unique elements of the said
vector:")
print(unique(nums))
```

Output:

```
[1] "Original vector(string)"
[1] "The quick brown fox jumps over
the lazy dog."
[1] "Unique elements of the said
vector:"
[1] "the quick brown fox jumps over
the lazy dog."
[1] "Original vector(number)"
[1] 1 2 2 3 4 4 5 6
[1] "Unique elements of the said
vector:"
[1] 1 2 3 4 5 6
```

9. Write a R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector.
Print the content of the matrix.

Code:

Input:

```
a<-c(1,2,3)
b<-c(4,5,6)
c<-c(7,8,9)
m<-cbind(a,b,c)
print("Content of the said matrix:")
print(m)
```

Output:

```
[1] "Content of the said matrix:"
      a b c
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

10. Write a R program to create a list of random numbers in normal distribution
and count occurrences of each value.

Code:

Input:

```
n = floor(rnorm(10, 5, 10))
print('List of random numbers in
normal distribution:')
print(n)
t = table(n)
print("Count occurrences of each
value:")
print(t)
```

Output:

```
[1] "List of random numbers in normal
distribution:"
 [1]    8   14 -10    1   15    6   11
19    3    3
[1] "Count occurrences of each value:"
n
-10    1    3    6    8   11   14   15
19
  1    1    2    1    1    1    1    1
1
```

11. Write a R program to create three vectors numeric data, character data and logical data. Display the content of the vectors and their type.

Code:

Input:

```
a = c(1, 2, 5, 3, 4, 0, -1, -3)
b = c("Red", "Green", "White")
c = c(TRUE, TRUE, TRUE, FALSE, TRUE,
FALSE)
print(a)
print(typeof(a))
print(b)
print(typeof(b))
print(c)
print(typeof(c))
```

Output:

```
[1]125340-1-3[1]"double"[1]"Red"
"Green""White"[1]"character"[1]TRUET
RUETRUEFALSETRUEFALSE
```

12. Write a R program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.

Code:

Input:

```
m1 = matrix(1:20, nrow=5, ncol=4)
print("5 × 4 matrix:")
print(m1)
cells = c(1,3,5,7,8,9,11,12,14)
rnames = c("Row1", "Row2", "Row3")
cnames = c("Col1", "Col2", "Col3")
m2 = matrix(cells, nrow=3, ncol=3,
byrow=TRUE, dimnames=list(rnames,
cnames))
print("3 × 3 matrix with labels, filled by
rows: ")
print(m2)
print("3 × 3 matrix with labels, filled by
columns: ")
m3 = matrix(cells, nrow=3, ncol=3,
byrow=FALSE, dimnames=list(rnames,
cnames))
print(m3)
```

Output:

```
[1] "5 \303\227 4 matrix:"
[,1] [,2] [,3] [,4]
[1,]     1     6    11    16
[2,]     2     7    12    17
[3,]     3     8    13    18
[4,]     4     9    14    19
[5,]     5    10    15    20
[1] "3 \303\227 3 matrix with labels,
filled by rows: "
Col1 Col2 Col3
Row1     1     3     5
Row2     7     8     9
Row3    11    12    14
[1] "3 \303\227 3 matrix with labels,
filled by columns: "
Col1 Col2 Col3
Row1     1     7    11
Row2     3     8    12
Row3     5     9    14
```

13. Write a R program to create an array, passing in a vector of values and a
vector of dimensions. Also provide names for each dimension.

Code:

Input:

```
a =    array(
   6:30,
   dim = c(4, 3, 2),
   dimnames = list(
        c("Col1", "Col2", "Col3", "Col4"),
        c("Row1", "Row2", "Row3"),
        c("Part1", "Part2")
    )
)
print(a)
```

Output:

```
          Row1 Row2 Row3
Col1        6     10     14
Col2        7     11     15
Col3        8     12     16
Col4        9     13     17, , Part2

          Row1 Row2 Row3
Col1      18     22     26
Col2      19     23     27
Col3      20     24     28
Col4      21     25     29
```

14. Write a R program to create an array with three columns, three rows, and two
&quot;tables&quot;, taking two vectors as input to the array. Print the array.

Code:

Input:

```
v1 = c(1, 3, 5, 7)
v2 = c(2, 4, 6, 8, 10)
arra1 = array(c(v1, v2),dim = c(3,3,2))
print(arra1)
```

Output:

```
, , 1

          [,1] [,2] [,3]
[1,]1       7       6
[2,]       3       2       8
[3,]       5       4     10

, , 2

          [,1] [,2] [,3]
[1,]       1       7       6
[2,]       3       2       8
[3,]       5       4     10
```

15. Write a R program to create a list of elements using vectors, matrices and a
functions. Print the content of the list.

Code
Input:

```
l = list(
   c(1, 2, 2, 5, 7, 12),
   month.abb,
   matrix(c(3, -8, 1, -3), nrow = 2),
   asin
)
print("Content of the list:")
print(l)
```

Output:

```
[1] "Content of the list:"
[[1]]
[1]    1    2    2    5    7 12

[[2]]
  [1] "Jan" "Feb" "Mar" "Apr"
"May""Jun""Jul" "Aug" "Sep" "Oct" "Nov"
"Dec"

[[3]]
          [,1] [,2]
[1,]        3        1
[2,]      -8      -3
[[4]]
function (x)    .Primitive("asin")
```