

DAY 1 BASIC LAB PROGRAMS

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

```
name = readline(prompt="Input your name: priya ")
age = readline(prompt="Input your age: 23 ")
print(paste("My name is",name, "and I am",age ,"years old."))
print(R.version.string)
```

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

```
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

```
n1 : num 10
n2 : num 0.5
name : chr "Python"
nums : num [1:6] 10 20 30 40 50 60
```

3. 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.

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

```

[1] "Sequence of numbers from 20 to 50:"
> print(seq(20,50))
[1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
[27] 46 47 48 49 50
> print("Mean of numbers from 20 to 60:")
[1] "Mean of numbers from 20 to 60:"
> print(mean(20:60))
[1] 40
> print("Sum of numbers from 51 to 91:")
[1] "Sum of numbers from 51 to 91:"
> print(sum(51:91))
[1] 2911

```

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

```

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

```

OUTPUT

```

[1] "Content of the vector:"
> print("10 random integer values between -50 and +50:")
[1] "10 random integer values between -50 and +50:"
> print(v)
[1] -13 7 0 -1 -12 26 -29 13 -18 25

```

5. R program to get the first 10 Fibonacci numbers.

```

Fibonacci <- numeric(10)
Fibonacci[1] <- Fibonacci[2] <- 1
for (i in 3:10) Fibonacci[i] <- Fibonacci[i - 2] + Fibonacci[i - 1]
print("First 10 Fibonacci numbers:")
print(Fibonacci)

```

OUTPUT

```

[1] "First 10 Fibonacci numbers:"
> print(Fibonacci)
[1] 1 1 2 3 5 8 13 21 34 55

```

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

```

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 = 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
```

7. R program to print the numbers from 1 to 100 and print "Fizz" for multiples of 3, print "Buzz" for multiples of 5, and print "FizzBuzz" for multiples of both.

```

for (n in 1:100) {
  if (n %% 3 == 0 & n %% 5 == 0) {print("FizzBuzz")}
  else if (n %% 3 == 0) {print("Fizz")}
  else if (n %% 5 == 0) {print("Buzz")}
  else print(n)
}

```

OUTPUT

```

[1] 1
[1] 2
[1] "Fizz"
[1] 4
[1] "Buzz"

```

[1] "Fizz"
[1] 7
[1] 8
[1] "Fizz"
[1] "Buzz"
[1] 11
[1] "Fizz"
[1] 13
[1] 14
[1] "FizzBuzz"
[1] 16
[1] 17
[1] "Fizz"
[1] 19
[1] "Buzz"
[1] "Fizz"
[1] 22
[1] 23
[1] "Fizz"
[1] "Buzz"
[1] 26
[1] "Fizz"
[1] 28
[1] 29
[1] "FizzBuzz"
[1] 31
[1] 32
[1] "Fizz"
[1] 34
[1] "Buzz"
[1] "Fizz"
[1] 37
[1] 38
[1] "Fizz"
[1] "Buzz"
[1] 41
[1] "Fizz"
[1] 43
[1] 44
[1] "FizzBuzz"
[1] 46
[1] 47

[1] "Fizz"
[1] 49
[1] "Buzz"
[1] "Fizz"
[1] 52
[1] 53
[1] "Fizz"
[1] "Buzz"
[1] 56
[1] "Fizz"
[1] 58
[1] 59
[1] "FizzBuzz"
[1] 61
[1] 62
[1] "Fizz"
[1] 64
[1] "Buzz"
[1] "Fizz"
[1] 67
[1] 68
[1] "Fizz"
[1] "Buzz"
[1] 71
[1] "Fizz"
[1] 73
[1] 74
[1] "FizzBuzz"
[1] 76
[1] 77
[1] "Fizz"
[1] 79
[1] "Buzz"
[1] "Fizz"
[1] 82
[1] 83
[1] "Fizz"
[1] "Buzz"
[1] 86
[1] "Fizz"
[1] 88
[1] 89

```
[1] "FizzBuzz"
[1] 91
[1] 92
[1] "Fizz"
[1] 94
[1] "Buzz"
[1] "Fizz"
[1] 97
[1] 98
[1] "Fizz"
[1] "Buzz"
```

8. R program to extract first 10 english letter in lowercase and last 10 letters in uppercase and extract letters between 22 nd to 24 th letters in upper case.

```
print("First 10 letters in lowercase:")
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 uppercase:")
t = tail(LETTERS[22:24])
print(t)
```

OUTPUT

```
print("First 10 letters in lowercase:")
[1] "First 10 letters in lowercase:"
> t = head(letters, 10)
> print(t)
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
> print("Last 10 letters in uppercase:")
[1] "Last 10 letters in upper case:"
> t = tail(LETTERS, 10)
> print(t)
[1] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z" >
print("Letters between 22nd to 24th letters in uppercase:")
[1] "Letters between 22nd to 24th letters in uppercase:" > t
= tail(LETTERS[22:24])
> print(t)
[1] "V" "W" "X"
```

9. R program to find the factors of a given number.

```
print_factors <- function(x) {  
  print(paste("The factors of",x,"are:"))  
  for(i in 1:x) {  
    if((x %% i) == 0) {  
      print(i)  
    }  
  }  
}
```

OUTPUT

```
print_factors(20)  
[1] "The factors of 20 are:"  
[1] 1  
[1] 2  
[1] 4  
[1] 5  
[1] 10  
[1] 20
```

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

```
nums = c(20, 40, 60, 80, 100, 120, 140, 160)  
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

```
nums = c(20, 40, 60, 80, 100, 120, 140, 160)  
> print('Original vector:')  
[1] "Original vector:"  
> print(nums)  
[1] 20 40 60 80 100 120 140 160  
> print(paste("Maximum value of the said vector:",max(nums)))  
[1] "Maximum value of the said vector: 160"  
> print(paste("Minimum value of the said vector:",min(nums)))  
[1] "Minimum value of the said vector: 20"
```

11. R program to get the unique elements of a given string and unique numbers of vectors.

```
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

```
str1 = "The quick brown fox jumps over the lazy dog."
> print("Original vector(string)")
[1] "Original vector(string)"
> print(str1)
[1] "The quick brown fox jumps over the lazy dog."
> print("Unique elements of the said vector:")
[1] "Unique elements of the said vector:"
> print(unique(tolower(str1)))
[1] "the quick brown fox jumps over the lazy dog."
> nums = c(1, 2, 2, 3, 4, 4, 5, 6)
> print("Original vector(number)")
[1] "Original vector(number)"
> print(nums)
[1] 1 2 2 3 4 4 5 6
> print("Unique elements of the said vector:")
[1] "Unique elements of the said vector:"
> print(unique(nums))
[1] 1 2 3 4 5 6
```

12. 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.

```
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
```

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

```
n = floor(rnorm(100, 50, 100))
print('List of random numbers in normal distribution:')
print(n)
t = table(n)
print("Count occurrences of each value:")
print(t)
```

OUTPUT

```
print(t)
n
-231 -208 -207 -187 -185 -184 -183 -179 -178 -172 -166 -163 -162 -159 -155 -154
  1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
-151 -150 -147 -144 -142 -141 -140 -139 -138 -137 -134 -132 -131 -130 -129 -128
  2 1 1 1 1 1 1 2 1 1 4 2 3 4 2 1
-127 -124 -121 -120 -118 -117 -116 -115 -114 -113 -112 -111 -110 -109 -108 -107
  2 1 1 1 1 2 3 1 1 1 2 4 2 1 1 2
-106 -105 -104 -103 -102 -98 -97 -96 -94 -93 -92 -91 -89 -88 -87 -85 2 2 4
  1 1 1 3 5 2 1 2 1 2 2 1 3
-84 -83 -82 -81 -80 -79 -78 -77 -76 -75 -74 -73 -71 -70 -68 -67 2 2 2 3 2
  2 3 1 2 2 3 1 1 1 5 4
-66 -65 -63 -62 -61 -60 -59 -58 -57 -55 -54 -53 -52 -51 -50 -49 4 3 3 2 2
  3 2 1 2 3 2 1 1 1 4 4
-48 -47 -46 -45 -44 -43 -42 -41 -40 -39 -38 -37 -36 -35 -34 -33 3 1 2 3 5
  1 4 2 4 5 4 3 3 1 3 3
-32 -31 -30 -29 -28 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 6 2 5 5 4
```

```

1 2 4 4 1 2 5 6 2 4 4
-15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 6 3 2 6 3 4 3 1
2 5 3 4 4 2 3 7
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 2 6 7 5 4 4 4 8 3 4 2 5
3 4 3 7
17 18 19 20 21 22 23 24 26 27 28 29 30 31 32 33 1 5 2 3 4 2 4 3 3 3
3 9 4 7 1 5
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 5 7 3 3 4 4 3 4 7 1
2 2 3 2 3 6
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 3 5 2 3 1 5 5 5 2 2
8 5 3 6 3 3
66 67 68 69 70 71 72 73 74 76 77 78 79 80 81 82 5 3 6 2 7 3 2 6 3 4
1 4 3 1 3 7
83 84 85 86 87 88 89 90 92 93 94 95 97 98 99 100 3 2 4 1 6 1 1 2 2
4 4 7 4 5 1 3
101 102 103 104 105 106 107 108 109 110 111 112 114 115 116 117 6 3 3 4 3
1 4 2 2 5 3 6 6 3 3 1
119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 3 3 5 7 2
2 3 3 2 4 6 3 2 3 4 7
135 136 137 138 139 140 142 143 144 145 146 147 148 149 150 151 6 4 2 1 5
1 1 2 4 2 1 2 4 1 2 3
152 153 154 156 158 159 160 161 162 163 164 165 166 167 169 170 3 3 3 2 3
3 4 5 1 1 3 4 1 3 1 2
171 172 173 174 175 176 177 180 182 183 184 185 187 188 189 190 2 1 1 2 2
2 1 1 2 5 1 3 2 1 1 2
191 192 193 195 196 197 198 199 200 201 204 205 206 207 208 209 2 1 4 1 1
2 1 1 2 1 1 1 2 3 1 1
211 213 214 216 221 222 223 227 228 229 230 231 232 233 234 235 1 2 1 2 2
1 2 1 1 1 1 2 2 1 2 1
240 242 243 245 248 249 252 256 261 262 268 270 276 277 288 296 3 2 1 1 1
1 1 1 1 1 1 1 1 1 2
298 308 332 357 366 373
1 1 1 1 1 1

```

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

```

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

```
print(a)
[1] 1 2 5 3 4 0 -1 -3
> print(typeof(a))
[1] "double"
> print(b)
[1] "Red" "Green" "White"
> print(typeof(b))
[1] "character"
> print(c)
[1] TRUE TRUE TRUE FALSE TRUE FALSE
> print(typeof(c))
[1] "logical"
```

16. 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.

```
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 × 4 matrix:"
> print(m1)
      [,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 × 3 matrix with labels, filled by rows: "
```

```
> print(m2)
```

```
  Col1 Col2 Col3
Row1 1 3 5
Row2 7 8 9
Row3 11 12 14
```

```
[1] "3 × 3 matrix with labels, filled by columns: "
```

```
> print(m3)
```

```
  Col1 Col2 Col3
Row1 1 7 11
Row2 3 8 12
Row3 5 9 14
```

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

```
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

```
., Part1
```

```
  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
```

18. R program to create an array with three columns, three rows, and two
"tables", taking two vectors as input to the array. Print the array.

```
v1 = c(1, 3, 5, 7, 9)
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 4
[2,] 3 9 6
[3,] 5 2 8
```

```
, , 2
      [,1] [,2] [,3]
[1,] 10 5 2
[2,] 1 7 4
[3,] 3 9 6
```

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

```
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")
```

20. R program to draw an empty plot and an empty plot specify the axes limits of the graphic

```
#print("Empty plot:")
```

```
plot.new()
```

```
#print("Empty plot specify the axes limits of the graphic:")
```

```
plot(1, type="n", xlab="", ylab="", xlim=c(0, 10), ylim=c(0, 10))
```

OUTPUT

