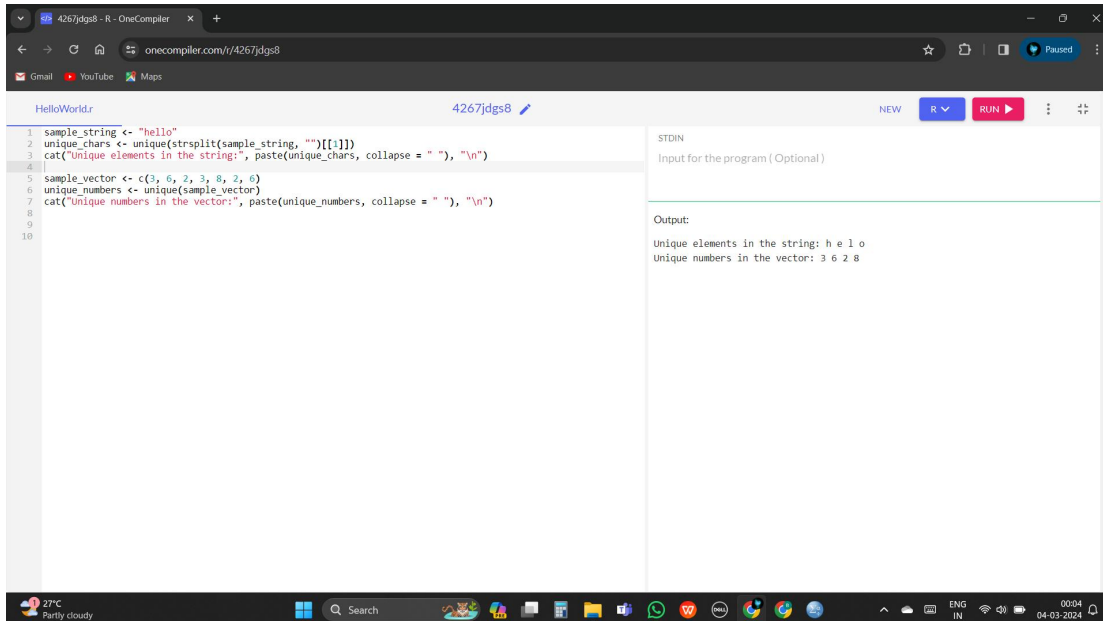


DAY - 2

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

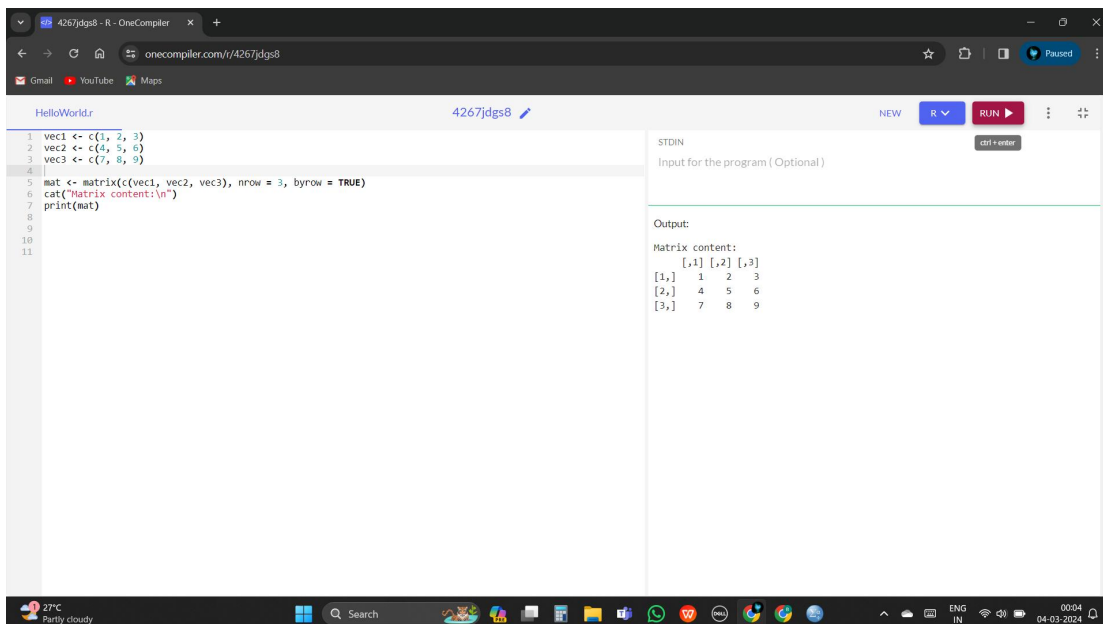


```
1 sample_string <- "hello"
2 unique_chars <- unique(strsplit(sample_string, "")[[1]])
3 cat("Unique elements in the string:", paste(unique_chars, collapse = " "), "\n")
4
5 sample_vector <- c(3, 6, 2, 3, 8, 2, 6)
6 unique_numbers <- unique(sample_vector)
7 cat("Unique numbers in the vector:", paste(unique_numbers, collapse = " "), "\n")
8
9
10
```

Output:

```
Unique elements in the string: h e l l o
Unique numbers in the vector: 3 6 2 8
```

2. 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.

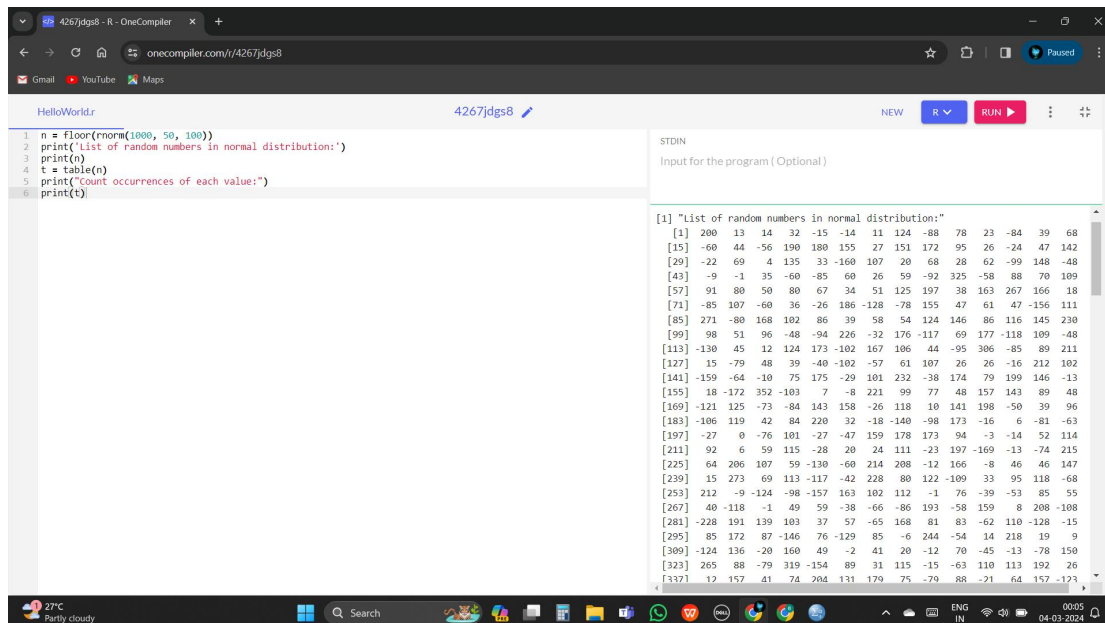


```
1 vec1 <- c(1, 2, 3)
2 vec2 <- c(4, 5, 6)
3 vec3 <- c(7, 8, 9)
4
5 mat <- matrix(c(vec1, vec2, vec3), nrow = 3, byrow = TRUE)
6 cat("Matrix content:\n")
7 print(mat)
8
9
10
11
```

Output:

```
Matrix content:
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

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



```

1 n = floor(rnorm(1000, 50, 100))
2 print("List of random numbers in normal distribution:")
3 print(n)
4 t = table(n)
5 print("Count occurrences of each value:")
6 print(t)

```

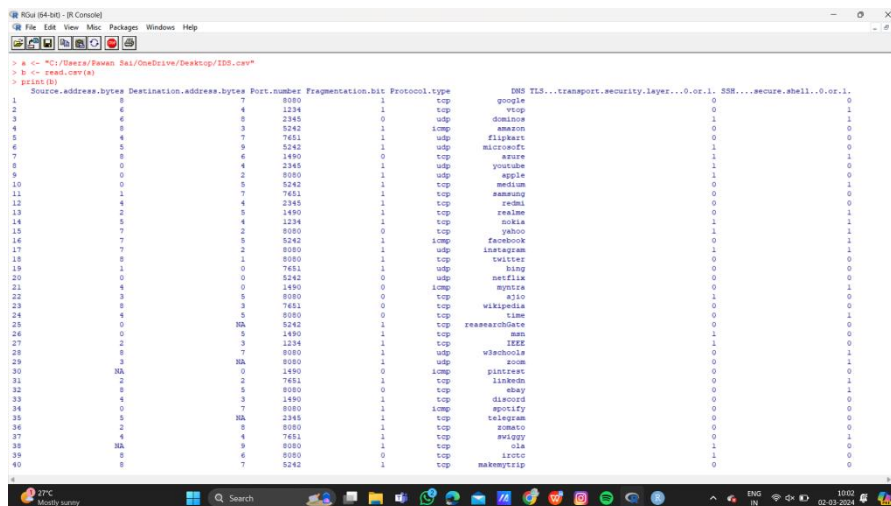
STDIN
Input for the program (Optional)

```

[1] "List of random numbers in normal distribution:"
[1] 200 13 14 32 -15 -14 11 124 -88 78 23 -84 39 68
[15] -60 44 -56 190 180 155 27 151 172 95 26 -24 47 142
[29] -22 69 4 135 33 -160 107 20 68 28 62 -99 148 -48
[43] -9 -1 35 -60 -85 60 26 59 -92 325 -58 88 70 109
[57] 91 80 50 80 67 34 51 125 197 38 163 267 166 18
[71] -85 107 -60 36 -26 186 128 78 155 47 61 47 -156 111
[85] 271 -80 168 102 86 39 58 54 124 146 86 116 145 230
[99] 98 51 96 -48 -94 226 -32 176 -117 69 177 -118 109 -48
[113] -130 45 12 124 173 -102 167 106 44 -95 306 -85 89 211
[127] 15 -79 48 39 -40 -102 -57 61 107 26 26 -16 212 102
[141] -159 -64 -10 75 175 -29 101 232 -38 174 79 199 146 -13
[155] 18 -172 352 -103 7 -8 221 99 77 48 157 143 89 48
[169] -121 125 -73 -84 143 158 -26 118 10 141 198 -50 39 96
[183] -106 119 42 84 220 32 -18 -140 -98 173 -16 6 -81 -63
[197] -27 0 -76 101 -27 -47 159 178 173 94 -3 -14 52 114
[211] 92 6 -50 115 -20 20 24 111 -23 107 -169 -13 -74 215
[225] 64 206 107 59 -130 -60 214 208 -12 166 -8 46 46 147
[239] 15 273 69 113 -117 -42 228 80 122 -109 33 95 118 -68
[253] 212 -9 -124 -98 -157 163 102 112 -1 76 -39 -53 85 55
[267] 40 -118 -1 49 59 -38 -66 -86 193 -58 159 8 208 -108
[281] -228 191 139 103 37 57 -65 168 81 83 -62 110 -128 -15
[295] 85 172 87 -146 76 -129 85 -6 244 -54 14 218 19 9
[309] -124 136 -20 160 49 -2 41 20 -12 70 -45 -13 -78 150
[323] 265 88 -79 319 -154 89 31 115 -15 -63 110 113 192 26
[337] 12 157 41 74 204 131 179 75 -79 88 -21 64 157 -173

```

4. Write a R program to read the .csv file and display the content



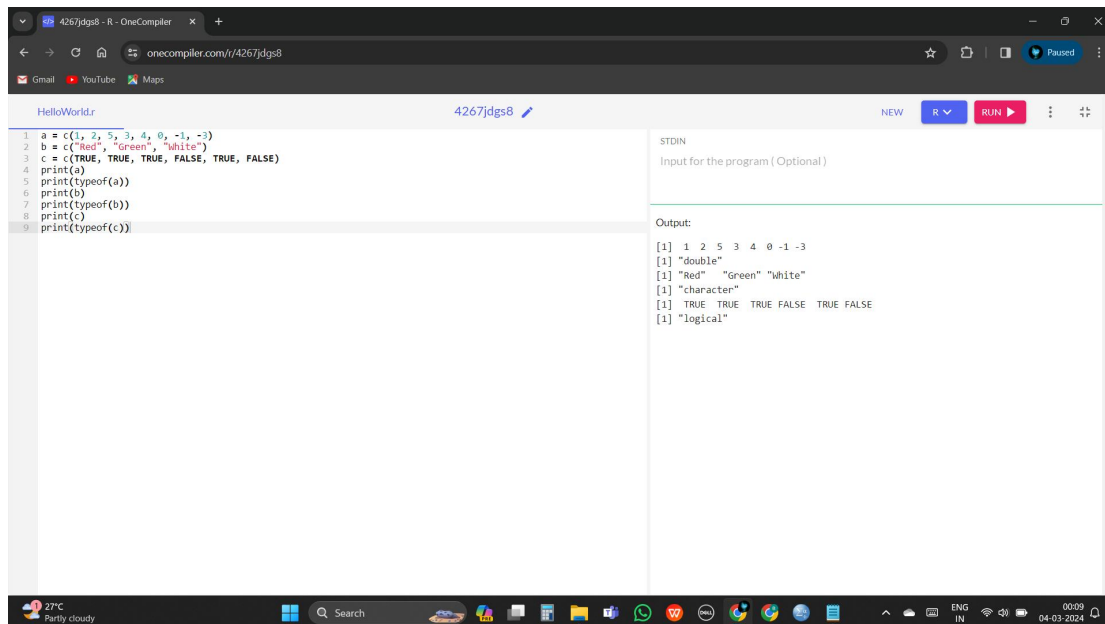
```

> a <- read.csv("C:/Users/Anan/Desktop/IDS.csv")
> b <- read.csv(a)
> print(b)

```

	Source.address.bytes	Destination.address.bytes	Port.number	Fragmentation.bit	Protocol.type	DNS.TLS...transport.security.layer...0.or.1
1	8	7	8080	1	tcp	google
2	6	4	1234	1	tcp	vkmp
3	6	5	2345	0	udp	amazon
4	8	3	5242	1	icmp	amazon
5	4	7	7451	1	udp	flipkart
6	5	9	5242	1	udp	microsoft
7	8	4	1490	0	tcp	azure
8	0	4	2345	1	udp	youtube
9	0	2	8080	1	udp	apple
10	0	5	5242	1	tcp	medium
11	1	7	7451	1	tcp	amazon
12	4	4	2345	1	tcp	redmi
13	2	5	1490	1	tcp	realme
14	5	4	1234	1	tcp	nokia
15	7	2	8080	0	tcp	jaboo
16	7	5	5242	1	icmp	facebook
17	7	5	8080	1	udp	instagram
18	8	1	8080	0	tcp	twitter
19	1	0	7451	1	udp	bing
20	0	0	5242	0	udp	netflix
21	4	0	1490	0	icmp	myntre
22	3	5	8080	0	tcp	asio
23	8	3	7451	0	tcp	wikipedia
24	4	5	8080	0	tcp	time
25	0	NA	5242	1	tcp	researchgate
26	0	5	1490	1	tcp	san
27	2	3	1234	1	tcp	ifsc
28	8	7	8080	1	udp	wishbooks
29	3	NA	8080	1	udp	scum
30	NA	0	1490	0	icmp	pintrtest
31	2	2	7451	1	tcp	linkedin
32	8	5	8080	0	tcp	etay
33	5	3	1490	1	tcp	discord
34	0	7	8080	1	icmp	spotify
35	5	NA	2345	1	tcp	telegram
36	2	5	8080	1	tcp	tomato
37	4	4	7451	1	tcp	swiggy
38	NA	8	8080	1	tcp	ole
39	8	6	8080	0	tcp	iroct
40	8	7	5242	1	tcp	makemytrip

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

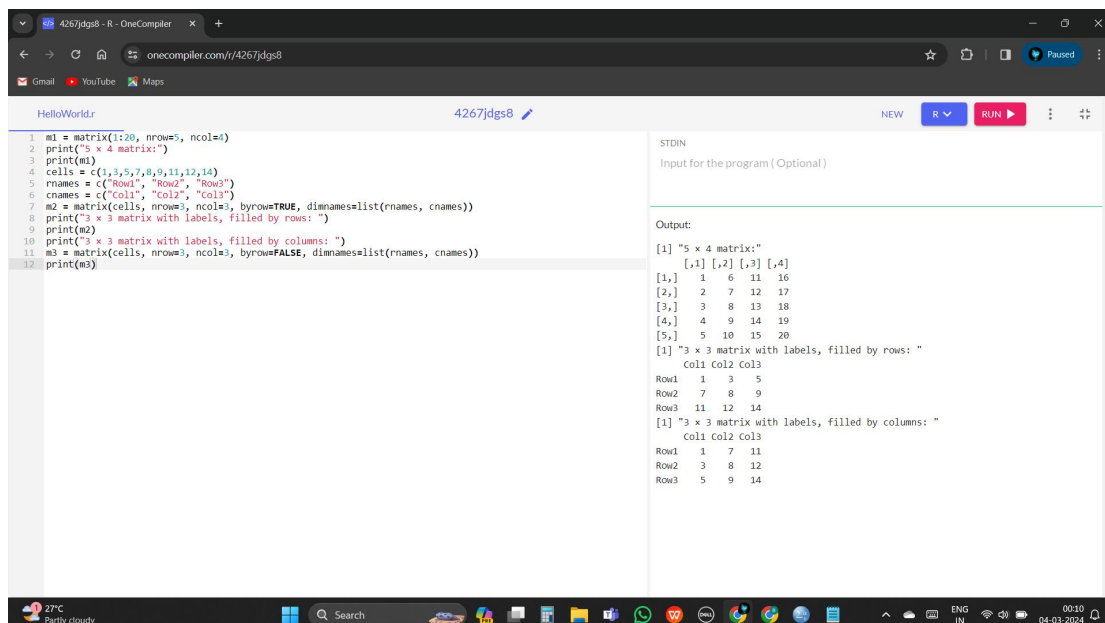


```
1 a = c(1, 2, 5, 3, 4, 0, -1, -3)
2 b = c("Red", "Green", "White")
3 c = c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)
4 print(a)
5 print(typeof(a))
6 print(b)
7 print(typeof(b))
8 print(c)
9 print(typeof(c))
```

Output:

```
[1] 1 2 5 3 4 0 -1 -3
[1] "double"
[1] "Red" "Green" "White"
[1] "character"
[1] TRUE TRUE TRUE FALSE TRUE FALSE
[1] "logical"
```

6. 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 x 2 matrix with labels and fill the matrix by columns.

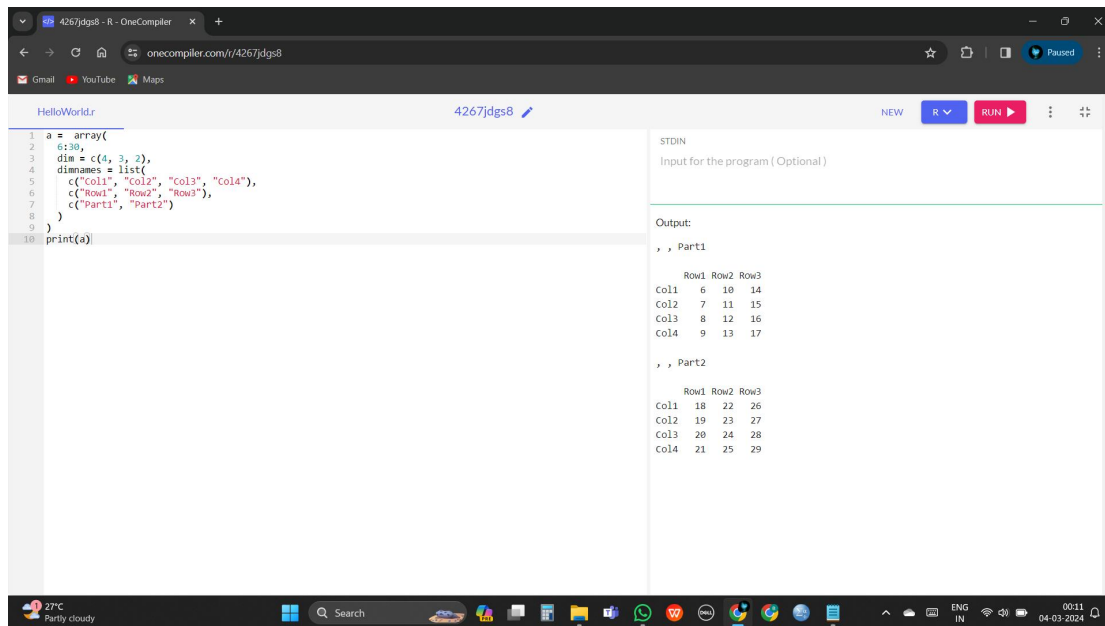


```
1 m1 = matrix(1:20, nrow=5, ncol=4)
2 print("5 x 4 matrix:")
3 print(m1)
4 cells = c(1,3,5,7,8,9,11,12,14)
5 rnames = c("Row1", "Row2", "Row3")
6 cnames = c("Col1", "Col2", "Col3")
7 m2 = matrix(cells, nrow=3, ncol=3, byrow=TRUE, dimnames=list(rnames, cnames))
8 print("3 x 3 matrix with labels, filled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

Output:

```
[1] "5 x 4 matrix:"
[1] [,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 x 3 matrix with labels, filled by rows: "
Col1 Col2 Col3
Row1 1 3 5
Row2 7 8 9
Row3 11 12 14
[1] "3 x 3 matrix with labels, filled by columns: "
Col1 Col2 Col3
Row1 1 7 11
Row2 3 8 12
Row3 5 9 14
```

7. 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



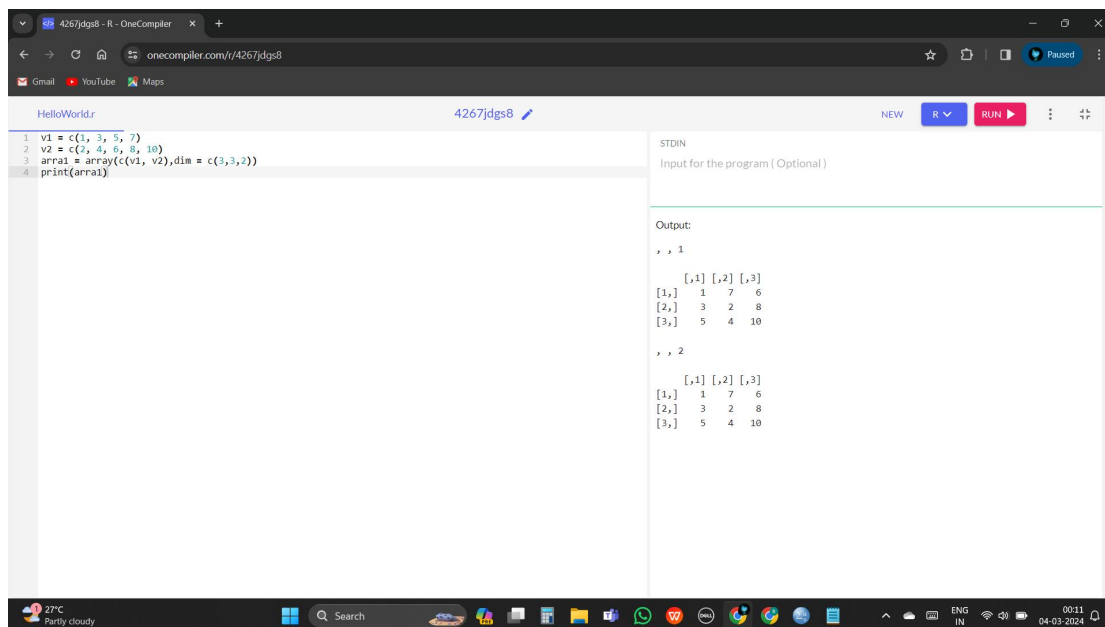
```
1 a = array(
2   6:30,
3   dim = c(4, 3, 2),
4   dimnames = list(
5     c("Col1", "Col2", "Col3", "Col4"),
6     c("Row1", "Row2", "Row3"),
7     c("Part1", "Part2")
8   )
9 )
10 print(a)
```

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
```

8. Write a 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.



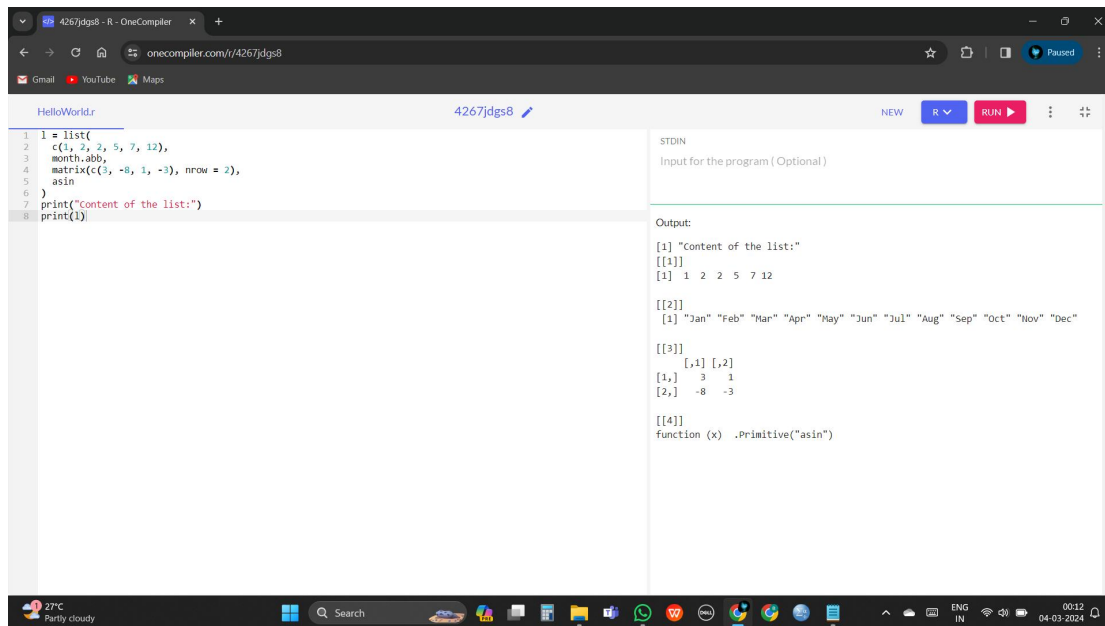
```
1 v1 = c(1, 3, 5, 7)
2 v2 = c(2, 4, 6, 8, 10)
3 arrai = array(c(v1, v2), dim = c(3,3,2))
4 print(arrai)
```

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
```

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



```
1 l = list(  
2   c(1, 2, 2, 5, 7, 12),  
3   month.abb,  
4   matrix(c(3, -8, 1, -3), nrow = 2),  
5   asin  
6 )  
7 print("Content of the list:")  
8 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,] [1, 2]  
[1,] 3 1  
[2,] -8 -3  
  
[[4]]  
function (x) .Primitive("asin")
```

10. Write a R program to draw an empty plot and an empty plot specify the axes limits of the graphic

```
> plot.new()  
> plot(1, xlim=c(0, 20), ylim=c(0, 20))  
> 
```

