**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**COURSE CODE / SUBJECT: ITA 0448 / STATISTICS WITH R PROGRAMMING FOR VECTORIZED EXPRESSIONS**

**DAY 1 – LAB ASSESSMENT**

**Reg No:192121058**

**Name: B. Gayathri devi**

**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. 

**> name = readline(prompt="Input your name:")**

**Input your name:gayu**

**> age = readline(prompt="Input your age:")**

**Input your age:18**

**> print(paste("my name is" ,name, "and i am" ,age, "years old."))**

**[1] "my name is gayu and i am 18 years old."**

**> print(R.version.string)**

**[1] "R version 4.2.3 (2023-03-15 ucrt)**"

**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())**

**[1] "n1" "n2" "name" "nums"**

**> print("details of the objects in memory:")**

**[1] "details of the objects in memory:"**

**> print(ls.str())**

**n1 : num 10**

**n2 : num 0.5**

**name : chr "python"**

**nums : num [1:6] 10 20 30 40 50 60**

**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. 

**> 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**

**[26] 45 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 number from 51 to 91:")**

**[1] "sum of number from 51 to 91:"**

**> 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. 

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

**> print("content of the vector:")**

**[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] -36 5 -28 8 -49 18 -40 -47 35 -8**

**5.** Write a 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:")**

**[1] "first 10 Fibonacci numbers:"**

**> print(Fibonacci)**

**[1] 1 1 2 3 5 8 13 21 34 55**

**6.** Write a 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 = c(x[(x %% i) !=0] ,i)**

**+ }**

**+ }**

**+ return(prime\_nums)**

**+ }**

**+ else**

**+ {**

**+ stop("input number should be at least 2.")**

**+ }**

**+ }**

**> prime\_numbers(12)**

**[1] 2 3 5 7 1**

**7.** Write a 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("Buzz")}**

**+ else print(n)**

**+ }**

**[1] 1**

**[1] 2**

**[1] "Buzz"**

**[1] 4**

**[1] 5**

**[1] "Buzz"**

**[1] 7**

**[1] 8**

**[1] "Buzz"**

**[1] 10**

**[1] 11**

**[1] "Buzz"**

**[1] 13**

**[1] 14**

**[1] "FizzBuzz"**

**[1] 16**

**[1] 17**

**[1] "Buzz"**

**[1] 19**

**[1] 20**

**[1] "Buzz"**

**[1] 22**

**[1] 23**

**[1] "Buzz"**

**[1] 25**

**[1] 26**

**[1] "Buzz"**

**[1] 28**

**[1] 29**

**[1] "FizzBuzz"**

**[1] 31**

**[1] 32**

**[1] "Buzz"**

**[1] 34**

**[1] 35**

**[1] "Buzz"**

**[1] 37**

**[1] 38**

**[1] "Buzz"**

**[1] 40**

**[1] 41**

**[1] "Buzz"**

**[1] 43**

**[1] 44**

**[1] "FizzBuzz"**

**[1] 46**

**[1] 47**

**[1] "Buzz"**

**[1] 49**

**[1] 50**

**[1] "Buzz"**

**[1] 52**

**[1] 53**

**[1] "Buzz"**

**[1] 55**

**[1] 56**

**[1] "Buzz"**

**[1] 58**

**[1] 59**

**[1] "FizzBuzz"**

**[1] 61**

**[1] 62**

**[1] "Buzz"**

**[1] 64**

**[1] 65**

**[1] "Buzz"**

**[1] 67**

**[1] 68**

**[1] "Buzz"**

**[1] 70**

**[1] 71**

**[1] "Buzz"**

**[1] 73**

**[1] 74**

**[1] "FizzBuzz"**

**[1] 76**

**[1] 77**

**[1] "Buzz"**

**[1] 79**

**[1] 80**

**[1] "Buzz"**

**[1] 82**

**[1] 83**

**[1] "Buzz"**

**[1] 85**

**[1] 86**

**[1] "Buzz"**

**[1] 88**

**[1] 89**

**[1] "FizzBuzz"**

**[1] 91**

**[1] 92**

**[1] "Buzz"**

**[1] 94**

**[1] 95**

**[1] "Buzz"**

**[1] 97**

**[1] 98**

**[1] "Buzz"**

**[1] 100**

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

**> print("first 10 letters in lower case:")**

**[1] "first 10 letters in lower case:"**

**> t = head(letters, 10)**

**> print(t)**

**[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"**

**> print("last 10 letters in upper case:")**

**[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 upper case:")**

**[1] "letters between 22nd to 24th letters in upper case:"**

**> e = tail(LETTERS[22:24])**

**> print(e)**

**[1] "V" "W" "X"**

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

**> print\_factors = function(n) {**

**+ print(paste("the factors of" ,n,"are:"))**

**+ for(i in 1:n) {**

**+ if((n %% i) == 0) {**

**+ print(i)**

**+ }**

**+ }**

**+ }**

**> print\_factors(4)**

**[1] "the factors of 4 are:"**

**[1] 1**

**[1] 2**

**[1] 4**

**> print\_factors(7)**

**[1] "the factors of 7 are:"**

**[1] 1**

**[1] 7**

**> print\_factors(12)**

**[1] "the factors of 12 are:"**

**[1] 1**

**[1] 2**

**[1] 3**

**[1] 4**

**[1] 6**

**[1] 12**

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

**> nums = c(10, 20, 30, 40, 50, 60)**

**> print('original vector:')**

**[1] "original vector:"**

**> print(nums)**

**[1] 10 20 30 40 50 60**

**> print(paste("maximum value of the said vector:",max(nums)))**

**[1] "maximum value of the said vector: 60"**

**> print(paste("minimum value of the said vector:",min(nums)))**

**[1] "minimum value of the said vector: 10"**

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

**> 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, 3, 4, 5, 6)**

**> print("original vector(number)")**

**[1] "original vector(number)"**

**> print(nums)**

**[1] 1 2 3 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.** 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. 

**> 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:")**

**[1] "content of the said matrix:"**

**> print(m)**

**a b c**

**[1,] 1 4 7**

**[2,] 2 5 8**

**[3,] 3 6 9**

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

**> n - floor(rnorm(1000, 50, 100))**

**Error: object 'n' not found**

**> n = floor(rnorm(1000, 50, 100))**

**> print('list of random numbers in normal distribution:')**

**[1] "list of random numbers in normal distribution:"**

**> print(n)**

**[1] 49 105 88 199 96 104 -101 221 132 245 -77 121 156 56**

**[15] 11 56 24 100 -129 53 0 -102 -93 -3 71 -29 101 24**

**[29] -22 85 51 97 63 11 -3 110 66 -11 -42 187 109 17**

**[43] 84 42 -9 75 186 32 165 -4 195 191 57 48 47 66**

**[57] 102 315 9 -56 -69 113 78 44 -123 109 269 269 161 -92**

**[71] 109 70 192 -25 46 117 216 334 118 180 190 -5 -40 150**

**[85] 152 57 144 97 148 144 -229 -47 -9 24 161 -6 2 -27**

**[99] 112 135 142 116 41 13 60 202 52 231 100 25 189 111**

**[113] 26 152 102 -28 37 4 25 -29 254 19 -16 31 -91 96**

**[127] 98 264 147 -86 69 43 149 43 210 -45 177 87 63 -5**

**[141] 265 19 66 65 122 -18 39 151 -23 108 -84 173 126 55**

**[155] 1 50 4 -4 -14 147 -81 225 -10 -6 134 162 82 63**

**[169] 34 -99 30 45 83 78 -25 61 168 -37 56 -75 -60 -60**

**[183] 117 199 -78 38 71 -169 61 64 11 -108 -34 156 -23 70**

**[197] -42 182 92 123 14 148 -45 142 -7 10 124 274 19 -79**

**[211] 111 -104 -34 37 34 84 29 -75 -69 125 -60 50 -58 -88**

**[225] 79 -27 19 -33 31 79 71 153 -18 -31 -69 66 71 81**

**[239] 86 20 -127 82 178 78 48 17 72 44 -81 118 -22 78**

**[253] -157 357 -38 33 51 -21 141 71 28 -49 -145 13 69 -102**

**[267] 1 -51 32 17 134 202 188 -2 147 2 31 22 164 73**

**[281] -5 -55 7 95 16 -5 210 41 -1 3 78 137 -62 -45**

**[295] -273 55 38 251 -6 89 73 140 161 134 112 -15 63 85**

**[309] -21 5 -83 -1 -25 199 71 -7 202 57 -142 60 243 -28**

**[323] 111 86 19 138 -95 -36 71 16 -197 -1 140 -47 107 -103**

**[337] 40 -149 36 78 -61 7 290 87 -40 45 -66 11 153 84**

**[351] 57 82 148 63 -111 46 37 225 36 71 155 24 71 -48**

**[365] 47 97 109 -128 209 152 201 -8 5 -48 -59 95 -120 106**

**[379] 62 163 -227 45 -116 167 45 -75 -40 94 4 96 26 -1**

**[393] 63 52 255 78 125 58 -110 172 56 -26 63 27 100 6**

**[407] 164 163 -11 45 92 -47 107 89 93 -32 31 122 -146 162**

**[421] -56 102 119 -75 -50 128 -94 -82 13 22 121 43 26 260**

**[435] -31 86 90 159 -67 -125 27 174 155 240 206 231 156 57**

**[449] 43 -59 147 117 10 132 -22 -18 39 97 11 -64 -196 123**

**[463] 12 21 24 105 150 121 26 194 9 124 30 61 177 55**

**[477] 16 214 65 146 -37 101 39 29 77 132 -76 48 73 48**

**[491] 182 60 -132 105 244 -164 24 225 129 -101 168 -102 -19 -139**

**[505] -190 -109 -35 154 -13 -39 116 12 249 -55 129 83 51 -63**

**[519] -122 249 70 100 -6 -24 50 111 -50 120 0 1 -60 312**

**[533] 144 -100 19 33 42 -76 -10 178 90 -35 -205 -94 52 -38**

**[547] 116 10 223 -50 -159 100 76 -100 140 -23 67 45 182 183**

**[561] 94 174 54 -9 292 -14 197 66 95 79 -13 -188 103 63**

**[575] -83 -10 139 67 -6 -143 -16 41 -49 75 29 -125 167 25**

**[589] 152 181 -36 -84 105 34 82 11 -71 -75 -36 61 33 -281**

**[603] -1 55 151 84 144 -5 134 -141 47 121 29 6 57 108**

**[617] -159 -29 -29 65 113 7 109 159 55 -163 -233 -42 -64 16**

**[631] 31 -5 -69 101 -26 14 122 181 -126 133 68 194 107 74**

**[645] 147 86 171 217 290 208 173 86 -82 21 -145 -142 88 151**

**[659] -144 26 -97 66 -51 49 202 60 136 78 31 178 -116 65**

**[673] -109 141 33 -51 147 117 206 -15 217 81 49 41 280 -45**

**[687] -18 -86 129 25 -60 62 17 115 4 83 330 30 42 -107**

**[701] 227 188 71 60 -58 125 35 190 201 -171 54 132 126 274**

**[715] 48 -15 90 28 -137 39 68 183 33 133 140 -30 -122 -63**

**[729] -51 179 135 -12 44 217 84 144 117 -197 -57 153 244 243**

**[743] -6 -98 77 110 -40 133 71 -92 -49 -11 30 103 94 -46**

**[757] 23 82 147 -8 132 99 208 5 70 64 -69 -69 55 -62**

**[771] -143 42 -28 137 -119 -22 109 38 -11 -34 26 120 151 15**

**[785] 84 -68 200 278 30 89 292 201 138 265 33 67 -118 114**

**[799] 47 171 40 35 -71 63 13 -163 120 -58 -111 156 13 75**

**[813] -36 128 79 143 42 49 28 241 241 211 107 75 131 -157**

**[827] -119 131 71 -88 55 30 152 31 145 -50 176 40 228 110**

**[841] 127 68 -14 206 152 -135 105 -30 105 -63 17 82 248 -119**

**[855] -3 4 19 -106 -39 -56 96 36 175 -18 169 -28 127 43**

**[869] 94 153 148 50 104 44 16 -41 -128 27 229 -84 130 -112**

**[883] -191 24 -76 -41 -71 -103 -7 217 42 137 4 36 114 -32**

**[897] 40 97 -89 -14 167 118 78 118 317 39 110 -96 25 34**

**[911] -6 -51 232 121 107 82 61 -74 21 171 24 69 -28 53**

**[925] 180 54 203 23 54 97 32 27 62 59 97 -12 -41 54**

**[939] 41 -115 181 145 30 -77 163 -58 -89 15 -36 118 128 -112**

**[953] 156 129 -93 -73 46 156 69 -155 58 -186 -60 57 153 128**

**[967] 137 187 109 58 17 -83 -11 -123 155 235 -19 -72 -41 -147**

**[981] 18 45 106 114 234 271 132 -45 -6 -70 60 111 84 22**

**[995] 137 -36 -29 173 68 105**

**> t = table(n)**

**> print("count occurrences of each value:")**

**[1] "count occurrences of each value:"**

**> print(t)**

**n**

**-281 -273 -233 -229 -227 -205 -197 -196 -191 -190 -188 -186 -171 -169 -164 -163**

**1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2**

**-159 -157 -155 -149 -147 -146 -145 -144 -143 -142 -141 -139 -137 -135 -132 -129**

**2 2 1 1 1 1 2 1 2 2 1 1 1 1 1 1**

**-128 -127 -126 -125 -123 -122 -120 -119 -118 -116 -115 -112 -111 -110 -109 -108**

**2 1 1 2 2 2 1 3 1 2 1 2 2 1 2 1**

**-107 -106 -104 -103 -102 -101 -100 -99 -98 -97 -96 -95 -94 -93 -92 -91**

**1 1 1 2 3 2 2 1 1 1 1 1 2 2 2 1**

**-89 -88 -86 -84 -83 -82 -81 -79 -78 -77 -76 -75 -74 -73 -72 -71**

**2 2 2 3 3 2 2 1 1 2 3 5 1 1 1 3**

**-70 -69 -68 -67 -66 -64 -63 -62 -61 -60 -59 -58 -57 -56 -55 -51**

**1 6 1 1 1 2 3 2 1 6 2 4 1 3 2 5**

**-50 -49 -48 -47 -46 -45 -42 -41 -40 -39 -38 -37 -36 -35 -34 -33**

**4 3 2 3 1 5 3 4 4 2 2 2 6 2 3 1**

**-32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -19 -18 -16 -15**

**2 2 2 5 5 2 2 3 1 3 4 2 2 5 2 3**

**-14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1**

**4 2 2 5 3 3 2 3 8 6 2 3 1 5 2 3**

**2 3 4 5 6 7 9 10 11 12 13 14 15 16 17 18**

**2 1 6 3 2 3 2 3 6 2 5 2 2 5 6 1**

**19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34**

**7 1 3 3 2 8 5 6 4 3 4 7 7 3 6 4**

**35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50**

**2 4 3 3 5 4 5 6 5 4 7 3 4 5 4 4**

**51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66**

**3 3 2 5 7 4 7 3 1 6 5 3 9 2 4 6**

**67 68 69 70 71 72 73 74 75 76 77 78 79 81 82 83**

**3 4 4 4 12 1 3 1 4 1 2 9 4 2 7 3**

**84 85 86 87 88 89 90 92 93 94 95 96 97 98 99 100**

**7 2 5 2 2 3 3 2 1 4 3 4 7 1 1 5**

**101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116**

**3 3 2 2 7 2 5 2 7 4 5 2 2 3 1 3**

**117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132**

**5 5 1 3 5 3 2 2 3 2 2 4 4 1 2 6**

**133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148**

**3 4 2 1 5 2 1 4 2 2 1 5 2 1 7 4**

**149 150 151 152 153 154 155 156 159 161 162 163 164 165 167 168**

**1 2 4 6 5 1 3 6 2 3 2 3 2 1 3 2**

**169 171 172 173 174 175 176 177 178 179 180 181 182 183 186 187**

**1 3 1 3 2 1 1 2 3 1 2 3 3 2 1 2**

**188 189 190 191 192 194 195 197 199 200 201 202 203 206 208 209**

**2 1 2 1 1 2 1 1 3 1 3 4 1 3 2 1**

**210 211 214 216 217 221 223 225 227 228 229 231 232 234 235 240**

**2 1 1 1 4 1 1 3 1 1 1 2 1 1 1 1**

**241 243 244 245 248 249 251 254 255 260 264 265 269 271 274 278**

**2 2 2 1 1 2 1 1 1 1 1 2 2 1 2 1**

**280 290 292 312 315 317 330 334 357**

**1 2 2 1 1 1 1 1 1**

**14.** Write a 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)**

**[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"**

**15.** 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. 

**> m1 = matrix(1:20, nrow=5, ncol=4)**

**> print("5 x 4 matrix:")**

**[1] "5 x 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**

**> 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 x 3 matrix with labels, filled by rows:")**

**[1] "3 x 3 matrix with labels, filled by rows:"**

**> print(m2)**

**col1 col2 col3**

**row1 1 3 5**

**row2 7 8 9**

**row3 11 12 14**

**> print("3 x 3 matrix with lables, filled by columns:")**

**[1] "3 x 3 matrix with lables, filled by columns:"**

**> m3 = matrix(cells, nrow=3, ncol=3, byrow=FALSE, dimnames=list(rnames, cnames))**

**> print(m3)**

**col1 col2 col3**

**row1 1 7 11**

**row2 3 8 12**

**row3 5 9 14**

**16.** 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. 

**> 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)**

**, , 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**

**17.** 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. 

**> 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)**

**, , 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**

**18.** Write a R program to create a list of elements using vectors, matrices and a function. 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:")**

**[1] "content of the list:"**

**> print(1)**

**[1] 1**