

1. Write a R program to take input from the user (name and age) and display the values.

Program:

```
name <- readline(prompt = "Enter your name: ")
age <- as.integer(readline(prompt = "Enter your age: "))

cat("Name:", name, "\n")
cat("Age:", age, "\n")

cat("R Version:", R.version.string, "\n")
```

output:

```
Enter your name: John
Enter your age: 25
Name: John
Age: 25
R Version: R version 4.1.2 (2021-11-01) -- "Bird Hippie"
```

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

Program:

```
x <- 5
y <- c(1, 2, 3, 4, 5)
z <- matrix(1:9, nrow = 3)
cat("Objects in memory:\n")
print(ls())
cat("\nStructure of each object:\n")
str(x)
str(y)
str(z)
cat("\nMemory size of each object:\n")
cat("Size of x:", object.size(x), "bytes\n")
cat("Size of y:", object.size(y), "bytes\n")
```

```
cat("Size of z:", object.size(z), "bytes\n")
cat("\nTotal memory used by R (Windows only):\n")
cat("Memory used:", memory.size(), "bytes\n")
```

output:

Objects in memory:

```
[1] "x" "y" "z"
```

Structure of each object:

```
num 5
```

```
num [1:5] 1 2 3 4 5
```

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

Memory size of each object:

```
Size of x: 40 bytes
```

```
Size of y: 112 bytes
```

```
Size of z: 120 bytes
```

Total memory used by R (Windows only):

```
Memory used: 3456 bytes
```

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.

Program:

```
sequence_20_50 <- 20:50
cat("Sequence from 20 to 50:\n")
print(sequence_20_50)
numbers_20_60 <- 20:60
mean_20_60 <- mean(numbers_20_60)
cat("\nMean of numbers from 20 to 60:\n")
print(mean_20_60)
numbers_51_91 <- 51:91
sum_51_91 <- sum(numbers_51_91)
cat("\nSum of numbers from 51 to 91:\n")
print(sum_51_91)
```

output:

Sequence from 20 to 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 46 47  
48 49 50
```

Mean of numbers from 20 to 60:

```
40
```

Sum of numbers from 51 to 91:

```
1716
```

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

Program:

```
random_integers <- sample(-50:50, 10, replace = TRUE)  
cat("Random integers between -50 and +50:\n")  
print(random_integers)
```

output:

Random integers between -50 and +50:

```
[1] 7 -24 15 4 30 -42 47 21 -6 0
```

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

Program:

```
fibonacci <- function(n) {
```

```

fib <- numeric(n) # Create an empty numeric vector to store Fibonacci numbers

fib[1] <- 0 # First Fibonacci number
fib[2] <- 1 # Second Fibonacci number


for (i in 3:n) {
  fib[i] <- fib[i - 1] + fib[i - 2] # Fibonacci relation
}


return(fib) # Return the Fibonacci sequence
}

first_10_fibonacci <- fibonacci(10)
cat("First 10 Fibonacci numbers:\n")
print(first_10_fibonacci)

```

output:

First 10 Fibonacci numbers: [1] 0 1 1 2 3 5 8 13 21 34

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

Program:

```

sieve_of_eratosthenes <- function(n) {
  primes <- rep(TRUE, n + 1) # True means prime, false means not prime
  primes[1] <- FALSE # 1 is not a prime
  for (i in 2:sqrt(n)) {
    if (primes[i]) {
      primes[seq(i * i, n, by = i)] <- FALSE
    }
  }
}

```

```

}
n <- as.integer(readline(prompt = "Enter the number up to which you want to find
primes: "))
prime_numbers <- sieve_of_eratosthenes(n)
cat("Prime numbers up to", n, "are:", prime_numbers, "\n")

```

output:

Enter the number up to which you want to find primes: 30

Prime numbers up to 30 are: 2 3 5 7 11 13 17 19 23 29

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.

Program:

```

fizz_buzz <- function() {
  # Loop through numbers from 1 to 100
  for (i in 1:100) {
    if (i %% 3 == 0 & i %% 5 == 0) {
      cat("FizzBuzz\n")
    } else if (i %% 3 == 0) {
      cat("Fizz\n")
    } else if (i %% 5 == 0) {
      cat("Buzz\n")
    } else {
      cat(i, "\n")
    }
  }
}

```

fizz_buzz()

output:

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

16

17

Fizz

19

Buzz

Fizz

22

23

Fizz

Buzz

26

Fizz

28

29

FizzBuzz

31

32

Fizz

34

Buzz

Fizz

37

38

Fizz

Buzz

41

Fizz

43

44

FizzBuzz

46

47

Fizz

49

Buzz

Fizz

52

53

Fizz

Buzz

56

Fizz

58

59

FizzBuzz

61

62

Fizz

64

Buzz

Fizz

67

68

Fizz

Buzz

71

Fizz

73

74

FizzBuzz

76

77

Fizz

79

Buzz

Fizz

82

83

Fizz

Buzz

86

Fizz

88

89

FizzBuzz

91

92

Fizz

94

Buzz

Fizz

97

98

Fizz

Buzz

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

Program:

```
<- function() {
```

```
letters_lower <- letters
```

```
first_10_lower <- letters_lower[1:10]
```

```
last_10_upper <- toupper(letters_lower[17:26])
```

```
cat("First 10 English letters in lowercase: ", paste(first_10_lower, collapse = ""), "\n")
```

```
cat("Last 10 English letters in uppercase: ", paste(last_10_upper, collapse = ""), "\n")
```

```
cat("Letters between 22nd and 24th in uppercase: ", paste(between_22_24_upper, collapse = ""), "\n")
```

```
}
```

```
extract_letters()
```

output:

First 10 English letters in lowercase: abcdefghij

Last 10 English letters in uppercase: QRSTUVWXYZ

Letters between 22nd and 24th in uppercase: VWX

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

Program:

```
find_factors <- function(n) {  
  factors <- c() # Create an empty vector to store factors
```

```

for (i in 1:n) {
  if (n %% i == 0) { # If n is divisible by i
    factors <- c(factors, i) # Add i to the factors vector
  }
}
return(factors)
}

number <- as.integer(readline(prompt = "Enter a number to find its factors: "))

factors <- find_factors(number)

cat("The factors of", number, "are:", factors, "\n")

```

output:

```

Enter a number to find its factors: 36
The factors of 36 are: 1 2 3 4 6 9 12 18 36

```

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

Program:

```

find_max_min <- function(vec) {
  max_value <- max(vec) # Find the maximum value in the vector
  min_value <- min(vec) # Find the minimum value in the vector
  return(list(max_value = max_value, min_value = min_value))
}vec_input <- as.numeric(unlist(strsplit(readline(prompt = "Enter a vector of numbers
(separated by spaces): "), " ")))

# Get the maximum and minimum values using the function
result <- find_max_min(vec_input)
cat("The maximum value of the vector is:", result$max_value, "\n")
cat("The minimum value of the vector is:", result$min_value, "\n")

```

output:

Enter a vector of numbers (separated by spaces): 4 7 2 9 5 1

The maximum value of the vector is: 9

The minimum value of the vector is: 1