Aim : Write a program in R for implementation of function.

Theory :

A function is a set of statements organised together to perform a specific task. R has a large number of in-built functions and the uses can create their own functions.

In R, a function is an object so the R interpreter is able to pass control to the function, along with arguments that maybe necessary for the function to accomplish the actions.

Function Definition :

```
function_name ← function (arg-1, arg-2, ---)
{
  function body
}
```

Simple examples of in-built functions are seq(), mean(), max(), sum() and paste(), etc. They are directly called by user written programs.

User - defined function :

We can create a user defined functions in R. They are specific to what a user wants and once created they can be used like the built in functions.

Calling a function with argument value :

The arguments to a function call can be supplied in the same sequence as defined in a different sequence but assigned to the names of the argument.

Code :

Built in Function

```
# Create a Sequence of number from 30 to 40.
  print (seq (30, 40))
# Find mean of numbers from 20 to 80.
  print (mean(20:80))
# Find sum of numbers from 40 to 68.
  print (sum (40:68))
```

User-defined function:

```
new_func <- function (a) {
   for (i in 1:a) {
   b <- i^2
   print (b)
  }
}
new_func (7)
```

Calling a function without an argument:

```
new_function <- function () {
for (i in 1:5) {
print (i^2)
}
}
# Call the function without supplying an argument.
  new. function ()

# To find the greatest number out of 3 nos.
greatest_number <- function (a=0, b=0, c=0) {
high = max (a,b,c)
print (high)
}
greatest_number (66, 38, 71)
```

To perform binary search on number in given vector.

```
binary_search = function (x, y) {
p = 1
q = length (y)
while (p-q > 1) {
    mid = floor ((p+q)/2)
    if (x < y [mid]) {
        q = mid
    }
    else {
        p = mid
    }
}
if (abs (x-y [q]) < abs (x-y [p])) {
    return (q)
}
    else {
    return (p)
}
}
x = 8
y = c (1, 3, 4, 7, 8)
binary search (x, y)
```

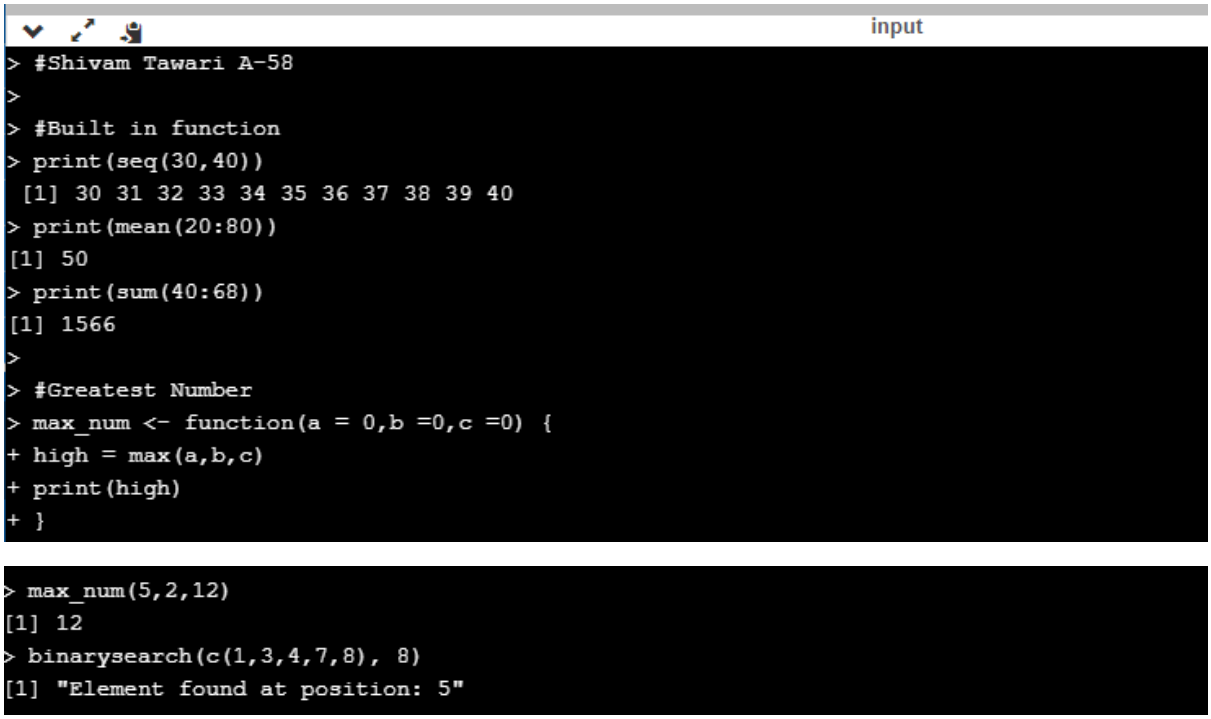
Conclusion: Hence successfully implemented functions in R programming language.

Code:

main.r

```r
#Shivam Tawari A-58

#Built in function
print(seq(30,40))
print(mean(20:80))
print(sum(40:68))

#Greatest Number
max_num <- function(a = 0,b =0,c =0) {
high = max(a,b,c)
print(high)
}

#Binary Search
binarysearch <- function(v, ele)   {
    start <- 0
    end <- length(v)
    while (start <= end) {
        mid <- as.integer((start+end)/2)
        if (v[mid] == ele)    {
            return (paste("Element found at position:", mid))
        }
        else if (v[mid] > ele) {
            end <- mid-1
        }
        else if(v[mid] < ele)    {
            start <- mid+1
        }
    }
    return ("Not Available")
}

max_num(5,2,12)
binarysearch(c(1,3,4,7,8), 8)
```

Output:


input
> #Shivam Tawari A-58
>
> #Built in function
> print(seq(30,40))
[1] 30 31 32 33 34 35 36 37 38 39 40
> print(mean(20:80))
[1] 50
> print(sum(40:68))
[1] 1566
>
> #Greatest Number
> max_num <- function(a = 0,b =0,c =0) {
+ high = max(a,b,c)
+ print(high)
+ }
> max_num(5,2,12)
[1] 12
> binarysearch(c(1,3,4,7,8), 8)
[1] "Element found at position: 5"