

1906188
Pratyush Kumar
IT

Q1. Print Average of Three Numbers

Source Code:

```
average <- function(a, b, c) {  
  return ((a+b+c)/3);  
}
```

```
print(average(3, 4, 5));
```

Output:

```
> source('q1.r');  
[1] 4  
> █
```

Q2. Find Factorial of an number

Source Code:

```
fact <- function(num) {  
  product = 1  
  for (i in 2:num){  
    product = product*i;  
  }  
  return (product);  
}
```

```
print(fact(5));
```

Output:

```
> source('q2.r');  
[1] 120  
> █
```

1906188
Pratyush Kumar
IT

Q3. Find LCM and HCF of Two Numbers

Source Code:

```
hcf = function(a, b) {  
  return (a*b/lcm(a, b));  
}
```

```
lcm = function(a, b) {  
  if(a > b) {  
    max = a;  
  } else {  
    max = b;  
  }  
}
```

```
while(TRUE) {  
  if(max %% a == 0 & max %% b == 0) {  
    return (max);  
  }  
  max = max+1;  
}  
}
```

```
print(lcm(4, 5));  
print(hcf(4, 16));
```

Output:

```
> source('q3.r');  
[1] 20  
[1] 4  
> █
```

Q4: find sum of N numbers using Recursion

Source Code:

```
sum = function(N) {  
  if (N > 1) {  
    return (N + sum(N-1));  
  } else {  
    return (N);  
  }  
}
```

```
print(sum(5));
```

Output:

```
> source('q4.r');  
[1] 15  
> █
```

1906188
Pratyush Kumar
IT

Q5. Reverse A number

Source Code:

```
reverse = function(num) {  
  if(num > 0) {  
    digits = as.integer(log10(num));  
    rev = num%%10 * 10**(digits);  
    return (rev + reverse(num%%10));  
  }  
  return (0);  
}  
  
print(reverse(532324));
```

Output:

```
> source('q5.r');  
[1] 423235  
> █
```

Q6: Calculate SI

Source Code:

```
simpleIntrest = function(p, r, t) {  
  return (p+(p*r*t)/100);  
}  
print(simpleIntrest(4000, 10, 3));
```

Output:

```
> source('q6.r');  
[1] 5200  
> █
```

1906188
Pratyush Kumar
IT

Q7: Print Decimal To Binary

Source Code:

```
dec2bin = function(n) {  
  if (n >= 1) {  
    rem = n %% 2;  
    dec2bin(n%%2);  
    print(rem);  
  }  
}
```

```
dec2bin(89);
```

Output:

```
> source('q7.r');  
[1] 1  
[1] 0  
[1] 1  
[1] 1  
[1] 0  
[1] 0  
[1] 1  
>
```

Q8 Print Factorial using Recursion function

Source Code:

```
fact = function(num) {  
  if(num > 1) {  
    return (num * fact(num-1));  
  } else {  
    return (1);  
  }  
}
```

```
print(fact(5));
```

Output:

```
> source('q8.r');  
[1] 120  
>
```

1906188

Pratyush Kumar

IT

Q9: Sum of Series $1 + 4 + 9 + \dots$ using recursion

Source Code

```
sum = function(N) {  
  if (N > 1) {  
    return (N**2 + sum(N-1));  
  } else {  
    return (N**2);  
  }  
}
```

```
print(sum(10));
```

Output:

```
> source('q9.r');  
[1] 385  
> █
```

Q10: Print sum, mean, and sd of an vector

Source Code:

```
arr = c(1, 2, 3, 4, 5);  
print(sum(arr));  
print(mean(arr));  
print(sd(arr));
```

Output:

```
> source('q10.r');  
[1] 1 4 9 16 25  
[1] 3  
[1] 1.581139  
Warning message:  
In if (N > 1) { :  
  the condition has length > 1 and only the first element will be used  
> █
```

1906188
Pratyush Kumar
IT

Q11: Find mean, median, var, sd, scale, summary, rank, quantile

```
arr = c(23, 41234, 124, 412, 421, 32, 2341, 415, 2, 451, 5, 51441, 45, 1345);  
print(mean(arr));  
print(median(arr));  
print(var(arr));  
print(sd(arr));  
print(scale(arr));  
print(summary(arr));  
print(rank(arr));  
print(quantile(arr));
```

Output:

```
> source('q11.r');  
[1] 7020.786  
[1] 413.5  
[1] 281874622  
[1] 16789.12  
      [,1]  
[1,] -0.4168047  
[2,]  2.0378203  
[3,] -0.4107889  
[4,] -0.3936350  
[5,] -0.3930989  
[6,] -0.4162687  
[7,] -0.2787392  
[8,] -0.3934563  
[9,] -0.4180556  
[10,] -0.3913120  
[11,] -0.4178769  
[12,]  2.6457735  
[13,] -0.4154944  
[14,] -0.3380633  
attr(,"scaled:center")  
[1] 7020.786  
attr(,"scaled:scale")  
[1] 16789.12  
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
 2.00   35.25   413.50 7020.79 1121.50 51441.00  
[1] 3 13 6 7 9 4 12 8 1 10 2 14 5 11  
    0%   25%   50%   75%  100%   
 2.00   35.25   413.50 1121.50 51441.00  
>
```