Tutorial 4 Memo

Dr. Niladri Chakraborty

2023-04-24

Q1.Write a while loop that prints out the Fibonacci sequence up to the 10th term. The Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding numbers. The sequence starts with 0 and 1, and the next number is the sum of the previous two numbers. The sequence goes as follows: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, and so on.

Let us consider the first 20 Fibonacci numbers.

```
n <- 20
a <- 0
b <- 1
i <- 1
while (i <= n) {
  print(a)
  # we store the sum in the variable s
  s <- b + a
  a <- b
  b <- s
  i <- i + 1
}
```

```
## [1] 0
## [1] 1
## [1] 1
## [1] 2
## [1] 3
## [1] 5
## [1] 8
## [1] 13
## [1] 21
## [1] 34
## [1] 55
## [1] 89
## [1] 144
## [1] 233
## [1] 377
## [1] 610
```

```
## [1] 987
## [1] 1597
## [1] 2584
## [1] 4181
```

Q2.Write a while loop that calculates the sum of the first 50 odd numbers.

```
sum <- 0
i <- 1
while (i <= 100) {
  if (i %% 2 != 0) {
    sum <- sum + i
  }
  i <- i + 1
}
print(sum)
```

```
## [1] 2500
```

Q3.Write a for loop that creates a vector of the first 20 multiples of 3.

```
multiples <- numeric(20)
for (i in 1:20) {
  multiples[i] <- i * 3
}
print(multiples)
```

```
##  [1]  3  6  9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60
```

Q4. Write a while loop that checks if a given number is prime.

There are many ways to check if a number is prime. Here we will use the Wilson's theorem that says a number $p > 1$ is prime if and only if $(p-1)!$ divided by p has the remainder -1 or $(p-1)$.

```
is_prime <- function(n) {
  if (n <= 1) {
    return(FALSE)
  }

  if (n == 2) {
    return(TRUE)
  }

  if (n %% 2 == 0) {
    return(FALSE)
  }
```

```
  if(n %% 2 == 1){
    fact <- 1
    for (i in 2:(n-1)) {
      fact <- fact*i
    }
    if(fact %% n == (n-1) || fact %% n == (-1)){
      return(TRUE)
    }else{return(FALSE)}
  }

}

is_prime(17)

## [1] TRUE
```

Q5. Write a for loop that creates a matrix of size 5x5, with each element equal to its row number times its column number. Check if it is a singular matrix by the determinant of the matrix.

```
mat <- matrix(nrow=5, ncol=5)
for (i in 1:5) {
  for (j in 1:5) {
    mat[i,j] <- i * j
  }
}
print(mat)

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    2    4    6    8   10
## [3,]    3    6    9   12   15
## [4,]    4    8   12   16   20
## [5,]    5   10   15   20   25

det(mat)

## [1] 0
```

The determinant is 0. So the matrix is a singular matrix.