

Assignment 1

Austin Dollar

9/7/2021

Question 1

```
#func definition

sum_mults <- function(nums, n)
{
  f <- 0

  for(x in 1:n-1)
  {
    for(i in nums)
    {
      if(x %% i == 0)
      {
        f <- f+x
        break
      }
    }
  }
  return(f)
}
```

```
#test function
Q1_1<- sum_mults(c(3,5), 30)
Q1_2<- sum_mults(c(2,8), 40)
Q1_3<- sum_mults(c(4,6), 50)
```

Test of $\text{sum_mults}(c(3,5), 30) = 195$

Test of $\text{sum_mults}(c(2,8), 40) = 380$

Test of $\text{sum_mults}(c(4,6), 50) = 408$

Question 2

```

collatz_len <- function(n)
{
  t <- 0

  while(n > 1)
  {
    if (n%%2 == 0)
    {
      n = n/2
      t = t+1

    }else
    {
      n = 3*n+1
      t = t+1
    }

  }

  t = t+1

  return(t)
}

collat_1 <- collatz_len(17)
collat_2 <- collatz_len(15)
collat_3 <- collatz_len(21)

```

Test of collatz_len(17) = 13

Test of collatz_len(15) = 18

Test of collatz_len(21) = 8

Question 3

```

reverse<- function(v)
{
  return(rev(v))
}

reversal_1 <- reverse(c(1,2,3))
reversal_2 <- reverse(c(2,2,3))
reversal_3 <- reverse(c(4,5,6))

```

Test of reverse(c(1,2,3)) = 3, 2, 1

Test of reverse(c(2,2,3)) = 3, 2, 2

Test of reverse(c(4,5,6)) = 6, 5, 4

Question 4

```
drop <- function(v, n)
{
  c <- 0

  v1 <- character()

  for(i in v)
  {
    c <- c + 1

    if(c == n)
    {
      v1 <- c(v1, v[i])

      c = 0
    }
  }

  v <- setdiff(v, v1)
  return(v)
}

dropper_1 <- drop(c(1,2,3,4,5), 2)
dropper_2 <- drop(c(1,2,3,4,5), 4)
dropper_3 <- drop(c(1,2,3,4,5), 3)
```

Test of $\text{drop}(c(1,2,3,4,5), 2) = 1, 3, 5$

Test of $\text{drop}(c(1,2,3,4,5), 4) = 1, 2, 3, 5$

Test of $\text{drop}(c(1,2,3,4,5), 3) = 1, 2, 4, 5$

Question 5

```
intersect_3 <- function(v, w, x)
{
  v1 <- intersect(v,w)
  v1 <- intersect(v1, x)

  return(v1)
}

inter <- intersect_3(c(1,2,3,1), c(1,1,3,2), c(3,1,9,1))
inter_1 <- intersect_3(c(1,2,3), c(3,4,5), c(1,2,3,4))
inter_2 <- intersect_3(c(4,5,6,7), c(3,4,5), c(5,7,8))
```

Test of $\text{intersect_3}(c(1,2,3,1), c(1,1,3,2), c(3,1,9,1)) = 1, 3$

Test of intersect_3(c(1,2,3), c(3,4,5), c(1,2,3,4)) = 3

Test of intersect_3(c(4,5,6,7), c(3,4,5), c(5,7,8)) = 5

Question 6

```
filter_vec <- function(v,p)
{
  v1 <- integer()

  for (i in v)
  {
    if(p(v[i]) == TRUE)
    {
      v1 <- c(v1, v[i])
    }
  }

  return(v1)
}

p <- function(x){return(x>3)}
l <- 1:6
m <- filter_vec(l,p)
```

Test of filter_vec(l,p) = 4, 5, 6

Question 7

```
n_fibs <- function(n)
{
  v <- integer()

  for(i in 1:n)
  {
    if(i == 1 | i == 2)
    {
      v <- c(v, 1)
    }else
    {
      cur <- v[i-1] + v[i-2]
      v <- c(v, cur)
    }
  }

  return(v)
}
```

```

}

fib <- n_fibs(2)
fib_1 <- n_fibs(3)
fib_2 <- n_fibs(8)

```

Test of $n_fibs(2) = 1, 1$

Test of $n_fibs(3) = 1, 1, 2$

Test of $n_fibs(8) = 1, 1, 2, 3, 5, 8, 13, 21$

Question 8

```

shift <- function(v,n)
{

  if (n == 0)
  {

    return(v)

  }else if(n > 0)
  {
    return(c(tail(v, -n), head(v, n)))
  }
  else
  {
    return(c(tail(v, n), head(v, -n)))
  }

}

shifter1 <- shift(c(1,2,3,4), 2)
shifter2 <- shift(c(1,2,3,4), -3)
shifter3 <- shift(c(1,2,3,4), -1)

```

Test of $shift(c(1,2,3,4), 2) = 3, 4, 1, 2$

Test of $shift(c(1,2,3,4), -3) = 4, 1, 2, 3$

Test of $shift(c(1,2,3,4), -1) = 2, 3, 4, 1$

Question 9

```

rem_consec_dups <- function(v)
{
  v1 = integer()
  k <- 99999

```

```

for(i in 1:length(v))
{
  if(v[i] != k)
  {
    v1 <- c(v1, v[i])
  }

  k <- v[i]
}
return(v1)
}

remmer <- rem_consec_dups(c(1,1,1,2,3,3,1,2,2))
remmer_1 <- rem_consec_dups(c(1,1,1,2,3,3,1,2,2))
remmer_2 <- rem_consec_dups(c(1,1,1,2,3,3,1,2,2))

```

Test of `rem_consec_dups(c(1,1,1,2,3,3,1,2,2))` = 1, 2, 3, 1, 2

Test of `rem_consec_dups(c(1,1,2,2,3,3,1,2))` = 1, 2, 3, 1, 2

Test of `rem_consec_dups(c(1,1,1,2,3,3,1,2,2,4,5,5,6,6,6,7))` = 1, 2, 3, 1, 2

Question 10

```

n_even_fibs <- function(n)
{
  v <- integer()

  for(i in 1:30)
  {
    if(i == 1 | i == 2)
    {

      v <- c(v, 1)

    }else
    {
      cur <- v[i-1] + v[i-2]
      v <- c(v, cur)

    }

  }

  fiblist <- list(name = 1)

  for(x in 1:length(v))
  {
    if(v[x] %% 2 == 0)
    {
      zz = as.character(x)
      list.append(fiblist, zz = v[x])
    }
  }
}

```

```
    }  
  }  
  print(fiblist)  
}  
  
efibs <- n_even_fibs(5)
```

```
## $name  
## [1] 1
```