test

Kerui Du

23/08/2021

Q1

```
x =  3 : 10

# (ai)
cumsum(x)
```

```
## [1]  3  7 12 18 25 33 42 52
```

```
# (aii)
ifelse(x<7, 6, x)
```

```
## [1]  6  6  6  6  7  8  9 10
```

```
# (aiii)
ifelse(abs(5.5-x)>2, TRUE, FALSE)
```

```
## [1]  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
```

```
# (aiv)
x[ifelse(abs(7-x)<2, TRUE, FALSE)]
```

```
## [1] 6 7 8
```

```
y = c("EGO", "STOOD", "GLUE", "CHECK")

# (bi)
paste(substring(y, c(2,4), c(3,5)),collapse = '')
```

```
## [1] "GOODLUCK"
```

```
# (bii)
rep(paste(y, ':-)', sep = ''),2)
```

```
## [1] "EGO:-)"   "STOOD:-)" "GLUE:-)"  "CHECK:-)" "EGO:-)"   "STOOD:-)" "GLUE:-)"
## [8] "CHECK:-)"
```

```r
# (biii)
names(y) = 1:4;y
```

```
##       1       2       3       4
##   "EGO" "STOOD"  "GLUE" "CHECK"
```

Q2

```r
nth.na <- function(x, n) {
  len = length(x[is.na(x)])
  if (len >= n){
    return(which(is.na(x))[n])
  }
  else{
    return(NA)}
}
nth.na(c(NA, 0, NA, 7, 9, NA), 2)
```

```
## [1] 3
```

```r
nth.na(c(1, NA, 0, NA, 7, 9, NA), 2)
```

```
## [1] 4
```

```r
nth.na(c(1, NA, 0, NA, 7, 9, NA), 5)
```

```
## [1] NA
```

```r
nth.na.element <- function(x, n) {
  len = length(x[is.na(x)])
  if (len >= n){
    return(x[which(is.na(x))[n]+1])
  }
  else
    return(NA)
}
nth.na.element(c(1, NA, 0, NA, 7, 9, NA), 1)
```

```
## [1] 0
```

```r
nth.na.element(c(1, NA, 0, NA, 7, 9, NA), 2)
```

```
## [1] 7
```

```r
nth.na.element(c(1, NA, 0, NA, 7, 9, NA), 3)
```

```
## [1] NA
```

```
nth.na.element(c(1, NA, 0, NA, 7, 9, NA), 5)
```

```
## [1] NA
```

```
nth.na.element(c(1, NA, 0, NA, NA, 9, NA), 2)
```

```
## [1] NA
```

Q3

```
fun <- function(x, y) {
  m = outer(x, y, '+')
  ifelse(m%%2==0, 0, 1)
}
A = fun (1:3 , 1:5)
A
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    0    1    0
## [2,]    1    0    1    0    1
## [3,]    0    1    0    1    0
```

```
A = fun (1:3 , 0:7)
A
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    1    0    1    0    1    0    1    0
## [2,]    0    1    0    1    0    1    0    1
## [3,]    1    0    1    0    1    0    1    0
```

```
A = fun (2:3 , 7:5)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    1
## [2,]    0    1    0
```