hw_1

Sonish Lamsal, Emmie Jenkins

1/27/2021

Question 1(a)

```
divide <- function(d,a) {
  if (a==0) stop("division by zero is undefined.")
  if (d<0 || a<0) stop("divide() only works for positive ‘a‘ and ‘d‘.")
  quotient <- 0
    while (a<=d){
      d = d - a
      quotient = quotient + 1
    }
  c(q = quotient, r = d)
}
divide(22, 7)
```

```
## q r
## 3 1
```

```
22 %/% 7
```

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

```
22 %% 7
```

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

```
divide(22, 0)
```

```
## Error in divide(22, 0): division by zero is undefined.
```

```
divide(-22,7)
```

```
## Error in divide(-22, 7): divide() only works for positive ‘a‘ and ‘d‘.
```

```
divide(21,7)
```

```
## q r
## 3 0
```

Question 1(b)

```
mod <- function(d,a) {
  a<- divide(d,a)
  unname(a[2])
}
mod(23, 7)
```

```
## [1] 2
```

```
mod(21,7)
```

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

Question 1(c)

Question 1(d)

```
is.divisor <- function(d,a) {
  if (a==0){
    rem <- 1
  }
  else rem <- mod(abs(d),abs(a))
  rem == 0
}
is.divisor(6, 3)
```

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

```
is.divisor(6, 4)
```

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

```
is.divisor(-6, 3)
```

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

```
is.divisor(6, 0)
```

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

Question 1(e)

```
divisors <- function(d) {
  if (d==0) stop("Input cannot be zero")
  num <- c(-abs(d):abs(d))
  check <- lapply(num, is.divisor, d=d)
  num[unlist(check)]
}
divisors(7)
```

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

```
divisors(18)
```

```
##  [1] -18  -9  -6  -3  -2  -1   1   2   3   6   9  18
```

Question 1(f)

```
gcd_naive <- function(a,b) {
  if ( a!=0 && b!=0){
    div_a <- divisors(a)
    div_b <- divisors(b)
    common <- intersect(div_a, div_b)
    gcd <- max(common)
  }
```

```
  else{
    gcd <-  max(abs(a),abs(b))
  }
  gcd
}
gcd_naive(64, 28)
```

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

```
gcd_naive(64, -28)
```

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

```
gcd_naive(64,0)
```

```
## [1] 64
```

```
gcd_naive(-64,0)
```

```
## [1] 64
```

```
gcd_naive(0,0)
```

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

Question 1(g)

```
is.prime_naive <- function(p) {
  div <- divisors(p)
  length(div) == 4
}
is.prime_naive(5)
```

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

```
is.prime_naive(20)
```

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

```
is.prime_naive(1)
```

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

```
is.prime_naive(-3)
```

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

Question 1(h)

Question 2(a)

```
inv <- function(b, x0 = 10^-(ceiling(log10(abs(b)))), tol = sqrt(.Machine$double.eps),
                message = FALSE) {
  dif <- 100
  while (dif > tol){
    y <- x0
    x0 <- x0 * (2 - b * x0)
    dif <- x0 - y
    if (message) print(x0)
```

```
  }
  x0
}

inv(1000, message = TRUE)
```

```
## [1] 0.001
```

```
## [1] 0.001
```