HW3

John Rothen

9/19/2020

Problem 1.3.3

```
.Machine$double.xmax
```

```
## [1] 1.797693e+308
```

The maximum double value would be given in the floating point represenation as having sign bit = 0, exponent bit as large as possible (all bits =1, except the final), and all mantisa bits = 1. The floating point would be evaluated at $2^{1023} * (1 + (1 - 2^{-52}))$

64bit : 0 11111111110 1111111111111111111111111111111111111111111111111111

```
.Machine$double.xmin
```

```
## [1] 2.225074e-308
```

The minimum double value above zero is represented with a signbit of 0, a mantissa all 0's (52 bits of 0), and an exponent of (000000000001). This results in a floating point of $2^{-1022} * 1$

64bit : 0 00000000001 0000000000000000000000000000000000000000000000000000

```
.Machine$double.eps
```

```
## [1] 2.220446e-16
```

This value is the same as 2^{-52}, which can be represented as floating point using the following binary:

64bit: 0 01111001010 1111111111111111111111101000001011111000110110100011

```
.Machine$double.neg.eps
```

```
## [1] 1.110223e-16
```

This is the same as 2^{-53}, which is represented as a floating point with the following binary:

64bit: 0 01111001001 1111111111111111111111101000001011111000110110100011

Double Precision for the number 0 in 64bit : 0 01111111111 0000000000000000000000000000000000000000000000000000

Double Precision for the number 1 in 64bit : 0 00000000000 0000000000000000000000000000000000000000000000000000 (the sign bit can be 1 or 0, as +0 = -0).

Double precision for pi in 64bit : 0 10000000000 1001001000011111101101010100010001000010110100011000

Problem 1.3.4

In order to get an arbitrarily accurate value of e, it was saved as an exceptionally long string obtained from https://www.math.utah.edu/~pa/math/e.html. From this string, 10 consecutive numbers were extracted starting from the decimal point until the first prime was found. This is given in the code following.

```
e<- '71828 18284 59045 23536 02874 71352 66249 77572 47093 69995 95749 66967 62772 40766 30353 54759 45

e1<- gsub(' ' , '' , e, fixed=TRUE)

#library(stringr)#for string length checking, not necessary
library(gmp)
```

```
##
## Attaching package: 'gmp'

## The following objects are masked from 'package:base':
##
##     %*%, apply, crossprod, matrix, tcrossprod
```

```
for (i in 1:2455){
  j=i+9
  chunk<-substr(e1,i,j)
  chunk<- as.numeric(chunk)
    if(isprime(chunk) == 1| isprime(chunk == 2)){
    print(chunk)
    break
    }

}
```

```
## [1] 7427466391
```