PC4a – Numerical Accuracy

1. It is always wise to start cleaning the global environment!

```
rm(list = ls())   # clear memory

f <- function(x) {
  return(x/(1+x^2))
}

g <- function(x) {
  return(tan(x))
}

x=seq(-pi/2,pi/2,length=100)
plot(x,f(g(x)),type="l",lty=2,col="blue",lwd=2,ylab="")
lines(x,g(f(x)),type="l",lty=3,col="red",lwd=2,ylab="")
legend("bottomright",legend=c("f(g(x))","g(f(x))"),
       lty=c(2,3),col=c("blue","red"),lwd=c(2,2))
```

2. Again, `sprint()` is handy to format your output. The first statement clears the output screen in R!

```
cat("\f")
x = 0:3
cosx=cos(x)
cat(sprintf('x:      %12.8f %12.8f %12.8f %12.8f\n',x[1],x[2],x[3],x[4]))
cat(sprintf('cos(x): %12.8f %12.8f %12.8f %12.8f\n',
                     cosx[1],cosx[2],cosx[3],cosx[4]))
cosx = rep(0,length(x))
for(n in 0:3){
  cosx = cosx + (-1)^n*x^(2*n)/factorial(2*n)
  if(n>=1){
    cat(sprintf('N=%d:    %12.8f %12.8f %12.8f %12.8f\n',
                    n,     cosx[1],cosx[2],cosx[3],cosx[4]))
  }
}
```

3. We reuse most of the code of exercise 2!

```
x <- seq(-4,4,by=0.05)
plot(x,cos(x),type="l",col=1,lwd=2)
cosx <- rep(1,length(x))
for(n in 1:3){
  cosx = cosx + (-1)^n*x^(2*n)/factorial(2*n)
  lines(x,cosx,type="l", lty=n+1,col=n+1,lwd=2)
}
legend(2.7,1, legend=c("n=1", "n=2","n=3"),
       lty=c(2,3,4), col=c(2,3,4),lwd=2)
```

4. Just a small exercise in using double loops! I really like `sprintf` to display the results.

```
for(p in 1:4){
  sum <- 0
  for(j in 1:(p+1)){
    sum <- sum + j^p
  }
  cat(sprintf('Sum for p=%d is %d\n',p,sum))
}
```

5. Write a program to calculate the binary expansion of 1.1 to, say, 30 binary places [Exercise 1 of Ch.9 spuRs].

Below find my quick-and-dirty solution. Other solutions are possible and likely to be more efficient!
The main idea is to split the integer from the fractional part. Hence, the first for-loop determines the bits before the decimal, while the second for-loop determines the bits after the decimal.

```
x <- 1.1
int <- trunc(x)      # integer part before the decimal
frac <- x-int        # fractional part after the decimal
if(int>0) {
  str <- ""
  for(i in trunc(log(int,2)):0){     # trunc(log(int,2) is the
largerst power of 2
    dec <- 2^i
    if(dec<=x){
      x <- x-dec
      str <- paste(str,"1",sep="")
    } else str <- paste(str,"0",sep="")
  }
} else str <- "0"
str <- paste(str,".",sep="")
for(i in 1:30){
  dec <- 2^(-i)
  if(dec<=frac){
    frac <- frac-dec
    str <- paste(str,"1",sep="")
  } else str <- paste(str,"0",sep="")
}
cat(paste(str))
```