

```

#####Problem 25
randomWalk = function(){
x = 0
t = 0
while(x <20){
  if(as.logical(rbinom(1,1,.5))){
    x = x+1
    t = t+1
  }
  else{
    x = x-1
    t = t+1
  }
}
return(t)
}
Problem25 = replicate(250, randomWalk())
summary(Problem25)
#Min.  1st Qu.  Median    Mean  3rd Qu.    Max.
#44      294      701   209800    3151 32220000
#####Problem 30
###Part B
Problem30 = matrix(data = 0, nrow= 9, ncol = 9)
Problem30[1, c(1,2,4)] = c(.5,.25,.25)
Problem30[2, c(1,2,3,5)] = c(.5/3,.5,.5/3, .5/3)
Problem30[3, c(2,3,6)] = c(.5/2,.5,.5/2)
Problem30[4, c(1,4,5,7)] = c(.5/3,.5,.5/3, .5/3)
Problem30[5, c(4,5,2,6)] = c(.5/3,.5,.5/3, .5/3)
Problem30[6, c(5,6,3)] = c(.25,.5,.25)
Problem30[7, c(4,7,8)] = c(.25,.5,.25)
Problem30[8, c(7,8,9)] = c(.25,.5,.25)
Problem30[9, 9] = 1

Problem30b = Problem30%%Problem30
###Part C
Problem30c = Problem30b
for(i in 1:28){
  Problem30c = Problem30c%%Problem30
}
1 - Problem30c[1,9]
#[1] 0.780697
#####Problem 31
###Part B
pmfBE = function(lambda, p, k){
  return(replicate(k, rbinom(1, rpois(1, lambda), p)))
}
###Part C
Problem31C = c()
for(p in c(1:3/10)){

```

```

    for(lambda in c(12:15)){
      Problem31C = c(Problem31C, lambda*p - mean(
        replicate(1000, mean(pmfBE(lambda, p, 10)))))
    }
  }
#[1] 4.50175
###Actual Mean

#[1] 4.5
#####Problem 32
###Part A
Problem32A = matrix(nrow = 8, ncol = 10)
for(x in 1:10){
  for(y in 1:8){
    Problem32A[y,x] = x+3*y
  }
}
c = 1/sum((Problem32A))
Problem32A = c*Problem32A
c
#[1] 0.0006578947
###Part B
X = colSums(Problem32A)
X
#[1] 0.07631579 0.08157895 0.08684211 0.09210526 0.09736842 0.10263158
0.10789474
#[8] 0.11315789 0.11842105 0.12368421
Y = rowSums(Problem32A)
Y
#[1] 0.05592105 0.07565789 0.09539474 0.11513158 0.13486842 0.15460526
0.17434211
#[8] 0.19407895
###Part C
Z = Problem32A[, 5]
Z
#[1] 0.005263158 0.007236842 0.009210526 0.011184211 0.013157895
0.015131579
#[7] 0.017105263 0.019078947

```