NAME: . . . . MODEL ANSWERS ....

STUDENT NUMBER: . . . .

It is known from *the theory of random walks* that if two gamblers with capitals p and $1 - p$, respectively, engage in a fair game until one of the gamblers goes broke, then the gambler with initial capital p will win the game with probability p.

In this exam we will construct a fair game simulation to test this theory.

Note that if your solutions are **not functional** then they must be **dysfunctional**.

Question 1 (5 marks)

Consider the following R code snippet:

```
x <- sample(c(-1, 1), 10, replace=TRUE)
y <- 2 * ( ( ! (x+1) ) - (1/2) )
```

If after executing this code `x` prints as

```
[1]  1 -1  1  1  1  1 -1 -1 -1 -1
```

what will `y` print as ? (write your answer in the space below)

```
[1]  -1 1 -1 -1 -1 -1 1 1 1 1
```

hint: `!0` evaluates to `TRUE` and `!2` evaluates to `FALSE`.

Question 2 (15 marks)

To model a *fair game* we will consider a *pot* of money of say $R100$. If player `X` starts with $R60$ then player `Y` will start with $R40$.

At each thow of the dice both `X` and `Y` bet $R1$. If `X` wins this particular bet then he takes $R1$ from `Y` and visa versa if `Y` wins this bet.

Bets continue until one of the players runs out of money.

Construct an R function called `xWinsEventually` that simulates repeated bets until one of the two players runs out of money. Your function should take as input

`xStart` the amount of money that player `X` starts with.

`pot` the total value of the pot of money in play

Your function should return `TRUE` if `X` eventually wins all the money in play and `FALSE` if `Y` eventually wins all the money.

hint: think about using `cumsum` on the sequences generated in question 1.

write your question 2 solution on this page

```
xWinsEventually <- function(xStart=50,pot=100)
{

    n <- 1000
    x <- sample(c(-1, 1), n, replace=TRUE)

    xp <- cumsum(c(xStart,x))

    xw <- match(pot,xp)
    xl <- match(0,xp)

    if (is.na(xw) & is.na(xl)) {
          # continuing ....
      return(xWinsEventually(xp[length(xp)],pot))
    }
    if (is.na(xw)) return(FALSE)
    if (is.na(xl)) return(TRUE)
    return(xw<xl)

}
```

Question 3 (10 marks)

Write an R function called `xWinProbability` that makes use of your function from question 2 to compute the probability of `X` wining a fair game from any user supplied starting point.

Your function should take three parameters:

`xStart` the amount of money `X` starts the game with.

`pot` the total amount of money in the game

`nTrials` the number of trial games to simulate before computing and returning the probability of `X` wining the game.

write your question 3 solution on this page

```
xWinProbability <- function(
            xStart=50, pot=100, nTrials=1000)
{
    t <- rep(xStart,nTrials)
    return(sum(
            sapply(t,xWinsEventually,pot)
            ) / nTrials)
}
```

Question 4 (10 marks)

Write down a snippet of R code that will make use of your function constructed in Question 3 to generate probabilities of `X` wining from **different** starting positions. Use a total pot of $R100$.

After generating the different probabilities write another snippet of R code that makes use of the `lm` function from the `stats` package to fit a straight line to your probability results and test the assertion made in the opening remarks of this question paper.

Include R code to plot your data with the fit superimposed on the plot.

write your question 4 solution on this page

```
pot <- 100
xStarts <- seq(0,pot,1)
winProbs <- sapply(xStarts,xWinProbability)
plot(xStarts/pot,winProbs)

pst <- xStarts/pot
fit <- lm(winProbs ~ pst)
coef(fit)

abline(coef(fit))
```