Ensure you answer all questions, each question separately.

You should submit your assignment as a fully commented code listing with any plots attached. Independent of the format it should be clear where plots are created. Points will be lost if no plots are submitted. A report should be submitted clearly explaining the results. It should also be clear where any plots produce originate.

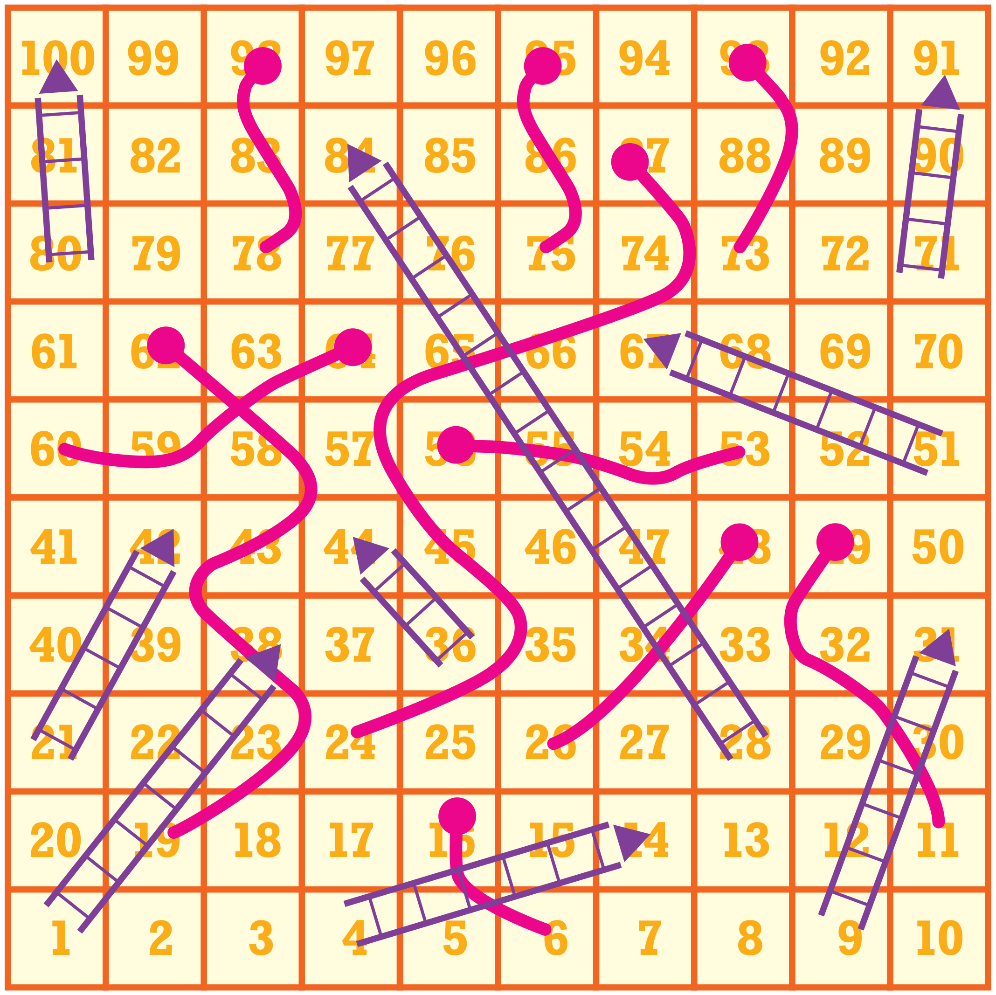
The ideal format is .Rmd and the accompanying .html file including plots and comments. Both files need to be submitted together. Other formats are acceptable as long as code can be copied and tested for accuracy but presentation is important.

The use of AI tools to generate text is not permitted. The use of AI tools to generate code is permitted but the code must be fully commented and explained. If you do use AI to generate code you must state this in your report, and submit the prompt you use and the code generated. Your aim is to add to any code you generate using an AI tool and interpret the results.

The *total number of marks* available is **25**.

1 Snakes and Ladders

You produce a simulation of a game of snakes and ladders (<https://en.wikipedia.org/wiki/Snakes_and_Ladders>) based on the board configuration shown below.



Snakes and Ladders board for this game

Q 1 Commenting Code

Study the following code example which simulates movement around the board, and add comments to describe what what each line does. [3 MARKS]

*Note, do not change the code before commenting*

set.seed(42)

max\_turns <- 20

position <- 0

max\_position <- 120

n\_sides\_die <- 4

**for** (turn **in** 1:max\_turns) {

die\_roll <- sample.int(n\_sides\_die, 1)

position <- position + die\_roll

**if** (position >= max\_position) {

**break**

}

}

Q 2 Extend the code above to incorporate the effect of the snakes and ladders. Fix any potential issues with the code and simulate the game for 1000 turns. Plot the position of the player at each turn. Compute the position at wich the player exceeds position 60 on the board. [4 MARKS]

Q 3 Use your code from Q2 or otherwise repeat the game for a single player and work out how many turns it takes to reach the 100th square. Repeat this simulation 1000 times and plot the distribution of the number of turns taken to reach the 100th square. Explain what you observe in the distribution briefly. [3 MARKS]

# 2 Data Analysis

For this part of the coursework you will need the data file (assess\_data\_resit-24.Rdata), instructions to download the file can be found on the course canvas page. This file contains the following objects:

* Y: gene expression matrix with 3,465 genes (along the rows) and 30 samples (along the columns) from 15 tumour-normal tissue pairs.
* patient\_data: Matrix with metadata for each patient.

You will perform the analysis in two steps, first you will look at the data to identify visually any problems with the data followed by a differential expression analysis to identify genes that are differentially expressed between tumour and normal tissue using regression.

Q 4 Study the following code example and add comments to describe what it does: [2 MARKS]

Note, do not change the code before commenting

**library**(tidyverse)

**library**(MASS)

load("data/assess\_data\_24-resit.Rdata")

idx <- 20

c\_cl <- 15:30

boxplot(log2(Y[, c\_cl] + 1))

x <- patient\_data$tissue[c\_cl]

z <- patient\_data$patient[c\_cl]

tmp <- data.frame(y = Y[idx, c\_cl], x = x, z = z, lib\_size = colSums(Y[, c\_cl]))

out <- glm(y ~ x + z + lib\_size, data = tmp, family = "poisson")

p\_val <- summary(out)$coefficients[2, 4]

## **2.1 Data Exploration**

Q 5 There is one problematic sample in the data, identify the problematic sample and explain why it is problematic. Provide a plot to support your answer. You could consider using some statistics on the data to identify the problematic sample. [2 MARKS]

## **2.2 Expression Analysis**

Using the remaining sample pairs, we will perform a regression-based differential expression analysis. We will use the glm function from the MASS package to perform the regression.

Q 6 Using the code from Q4, perform a regression-based differential expression analysis between **all** normal and tumour samples using Poisson regression. Use the tissue type as the only covariate. Plot the appropriate log10𝑙𝑜𝑔10 p-value from your analysis. [2 MARKS]

Q 7 Perform a regression-based analysis to identify genes differentially expressed between normal and tumour samples including the tissue variable indicating if it is tumour or normal sample. Plot the appropriate log10𝑙𝑜𝑔10p-value from your analysis. Compare the p-values with and without inclusion of the tissue type as a covariate, what do you observe? Which of the covariate has the biggest effect? Explain your answer with supporting plots, tables and further analysis if required. [4 MARKS]