Names:

1)

I have a box with 5 tickets, $\{-1, -1, 0, 1, 2\}$

a) compute the expected value of a single draw from this box

b)Simulate 1000 draws from this box using the `sample` function. Do it in two ways

i) Write `R` code to sample using the `rep` function and sampling uniform from an uneven vector

ii) Write `R` code to sample using the `p` argument in the sample function and using a non-uniform probability vector

c) Compute the mean of your 1000 simulated draws in `R`, is it close to the calculated expected value?

2)

Here we will simulate a "non-uniform" random walk. Consider fliping a coin with probability $p = 0.6$ of heads. You start at location 0. If the coin is heads, you take one step to the right, which we call a "1". If the coin is tails, you take one step to the left which we call a "−1". If you flip the coin 20 times, where you "wind up" in space is the sum of your left steps and right steps.

a) Simulate a single 20 step instance of your random walk by generating 20 independent −1,1 "left" or "right" steps. Call your vector "my_steps"

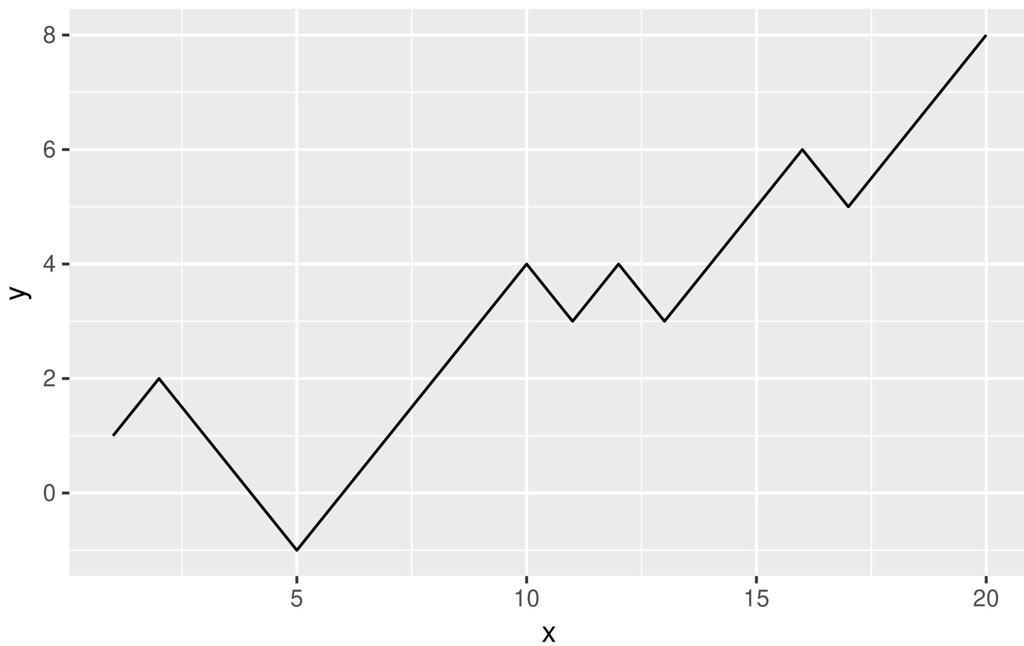
fill in this code:

```
my_steps <- sample(x = ________, p = _______, replace = _____, size = ________)
```

b) Run the following command to compute the running sum of your location after each step, "cumulate sum" adds up the elements in a vector in order, i.e. the 5th number is the sum of the first 5 numbers in my_samples.

```
my_location <- cumsum(my_steps)
```

Make a line plot of `x = c(1:20)` vs `y = my_location`, be sure to load `tidyverse` and make a data frame for plotting. You should see a plot like this (yours will looks different, every run is a unique random set of up down steps)



c) Now, simulate 10000 instances of your 20 step "random walk" and each time, compute your final location (sum the steps). Call this vector "my_walks" and run the following code. What does this code demonstrate? Look at the computed mean, is the mean 0, or something else. Why do you think the mean is like this? Compute the numnber $20^*(1^*(0.6) - 0.4(1))$ and relate it to your mean.

```
my_df <- data.frame(x = my_walks)

mu <- mean(my_walks)

sigma <- sd(my_walks)

ggplot(my_df)+
  geom_histogram(aes(x = my_walks, y = after_stat(density)), binwidth = 2)+
  stat_function(fun = dnorm, args = list(mean = mu, sd = sigma), color = "red")
```

d) In your sample, compute the proportion of random walks that ended between $\mu \pm \sigma$ (within 1 standard deviation of the mean)

Use the normal CDF function `pnorm` to approximate this probability, approximate your density as normal with mean μ and variance σ

Are they pretty close?