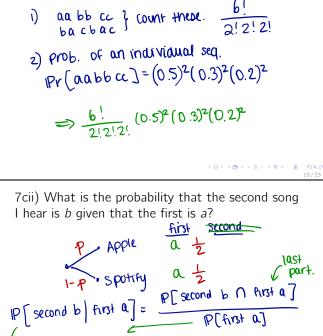


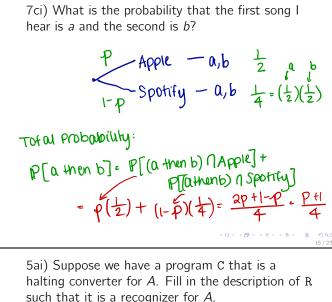
21) Continued $R_i = rumor$ is rold for ith hime w/out going back to someone. $P[R_1 \cap R_2 \cap R_3 \dots \cap R_r]$ $= P[R_1] \times P[R_2 R_1] \times P[R_3  R_1, R_2] \times \dots$ $= 1 \times 1 \times \frac{m-3}{m-2} \times \frac{m-4}{m-2} \times \dots$ $f_{corrit} tell person #1$	7a) Given a playlist, the shuffle feature on Apple Music will play songs as a series of independent <i>shuffle cycles.</i> In each shuffle cycle, all songs in the list will be reordered, with each ordering equally likely. For instance, for a playlist of four songs <i>a</i> , <i>b</i> , <i>c</i> , <i>d</i> , one possible sequence of plays could be a b c d   b d c a   d a c b   where we use   to separate the shuffle cycles.	7ai) Suppose I have an Apple Music playlist with <b>exactly two songs</b> , <i>a</i> and <i>b</i> . I have this playlist on shuffle while I'm away, so when I return, I could be at any position within a shuffle cycle with equal probability. When I return, <i>a</i> is playing. What is the probability that the next song is <i>b</i> ?
7aii) The next song played happened to be <i>b</i> . What is the probability that when I returned (i.e. when <i>a</i> was playing), it was the start of a shuffle cycle?	7b) Spotify's shuffle feature works a little differently. It instead selects any copy of any song from the playlist uniformly at random to play each time. I have a Spotify playlist with 5 copies of song a, 3 copies of song b, and 2 copies of song c (10 copies total). Select Song in a play ust uniformly at random with replacement.	7bi) I shuffle my Spotify playlist for 6 song plays. If I <i>ignore their play order</i> , how many different sets of 6 plays could I have gotten? Give your answer as an integer. SUS: $\{a,a,a,b,b,c\} \neq \{a,b,b,c,c\}$ $\begin{pmatrix} & \\ & \\ & \\ & \\ & \\ & \\ & \\ & \\ & \\ & $
<ロ><(ロ)><(ワ)><(E)><(E)>(E)>(E)>(E)>(2) 10/23	<ロ> (費) (言) (言) (言) (?) 11/23	<ロ>< () ・ロ><() ・ () ・ ()

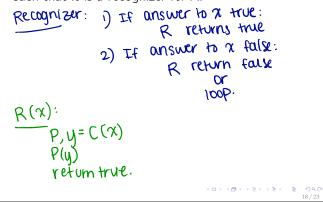
7bii) What is the probability that across the 6 songs played on my Spotify shuffle, I get song *a* twice, song *b* twice, and song *c* twice? (*You may leave your answer unsimplified*.)

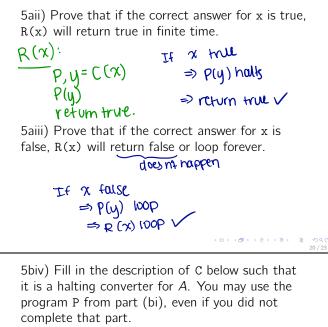


 $IP[first a] = \frac{p+1}{4} = \frac{p+1}{2}$ 

7c) YouTube Music's (YTM) shuffle functionality is somewhere in between Apple Music's and Spotify's. Specifically, given a playlist of *n* songs, YTM will still play songs as a series of *independent* length-*n* shuffle cycles. However, each YTM cycle will behave like Apple Music's shuffle feature (from part (a)) with probability  $p_{1}$ and behave like Spotify's shuffle feature (from part (b)) with probability 1 - p. I have a playlist with exactly two songs (one copy of each), a and b. I return when a (YTM) shuffle cycle is about to begin. (Note: Each of the following answers may be in terms of p.) 5) A "halting converter" for a problem A is a program C that takes an instance of A as input nd: program string x" If the correct answer for x is true, C(x) and: outputs a pair (P, y) such that P(y) halts. • If the correct answer for x is false. C(x)outputs a pair (P, y) such that P(y) loops forever.







```
C(x):
return P,X
```

```
5bi) Suppose we have a recognizer R for A. Fill in the description of P such that, for an instance x of the problem A, P(x) halts if and only if the correct answer for x is true.
```

```
P(X):
y=R(X)
if y is true:
return
else:
while lo==0) ← loop
forever.
```