

Fall 2026 COP 3330 Program #2:

Program A: Total Touring Distance (tourdist.java)

When musical acts become successful and famous, they give world-wide tours! A band will play at some shows in one city, then travel to a different city, and repeat. For this program, we'll compute the total travel distance for a band.

We'll ask the user to enter the number of locations for a band's tour. This will be followed by asking the user to enter the distance between each pair of locations (#1 and #2, #2 and #3, ..., #(n-1) and #n, where n is the number of locations for the band's tour.) After the user completes inputting all of this information, your program should compute and print out the total travel distance for the tour.

Input Specification

The first value the user will enter is, **n**, the number of locations for the band's tour. It's guaranteed that $2 \leq n \leq 10$. (We know that real tours are longer, but I don't want to slow down my graders too much!!!)

The user will then enter an additional **n – 1**, positive integers, representing the lengths (in miles) of each leg of their journey. Each of these integers will be less than 12,500. (The circumference of the earth is roughly 25,000 miles, so hopefully, any two locations on Earth can be reached with a travel distance of no more than half of the circumference!)

Output Specification

Print out a single line of output with the following format:

The band will travel a total of X miles for their tour.

where X is a positive integer representing the total travel distance of the band.

Sample Program Run (User Input in Bold and Italics)

How many locations are on the tour?

3

What is the distance, in miles, between location 1 and location 2?

198

What is the distance, in miles, between location 2 and location 3?

592

The band will travel a total of 790 miles for their tour.

Program B: Secret Song (secretsong.java)

Many bands, to build some excitement for individual shows, will add some randomness to their set list. For example, on their latest tour, Five Seconds of Summer has fans vote online for the secret song for each show. For a few hours before each show, fans are presented with four possible choices for the secret song. The song with the most votes gets revealed in the middle of the show and the band plays the song immediately after finding out what the fans have voted for.

For this program, instead of tallying up votes, your program will **randomly** choose the secret song from a list of choices provided by the user.

Note: Many song titles have spaces in them. Thus, to make this program more fun, please ONLY use the Scanner method `nextLine()` to read a whole line of input at a time. Since this method returns a `String`, if your intent is to read in a single integer on a whole line, you'll have the built-in method `Integer.parseInt`, to convert the string read in to its corresponding integer value.

Your program should first ask the user how many candidates there are for the secret song. Next, you should ask the user to enter each of the song titles, one by one.

Finally, print out the title of the selected song based on a randomly generated integer.

Note: Although you can solve this problem easily with an array of `String`, there's a way to solve it without using an array of `String`. Full credit will ONLY be given for students who can figure out how to solve the problem without an array of `String`.

Input Specification

The integer entered by the user first to indicate the number of possible song choices will be in between 2 and 6, inclusive.

Each song title entered by the user will consist of printable characters only (primarily letters, digits, spaces, dashes and apostrophes.) We'll only test your program on actual song titles that are easy to type on a keyboard.

Output Specification

Output a single sentence of the form:

The secret song for the concert is: X.

where X is the title of the randomly selected song.

Sample Program Run (User Input in Bold and Italics)

How many secret song choices are there?

4

What is the title of the choice #1?

Wishful Dreaming

What is the title of the choice #2?

Caramel

What is the title of the choice #3?

Chest

What is the title of the choice #4?

Blender

The secret song for the concert is: Wishful Dreaming.

Program C: Randomized Seating Chart? (seating.java)

For this question we'll simplify the seating chart for a concert venue as a rectangle with rows and columns, with the first row representing the seats closest to the stage. For this simulation, your program will ask the user to enter the number of rows and columns in the seating arrangement. (The columns are just the number of seats in a single row.) In addition, you'll ask the user to enter the percentage of seats that have already been sold.

Aisles for walking between some columns and some rows must exist, so for the purposes of this question, we'll assume that we have a walking space after every five rows and every 20 columns.

Your program should print out a seating chart for the venue, printing an 'X' for a taken seat and 'O' for an empty seat. You should use the generation of a random integer in between 0 and 99, inclusive, to determine whether or not each individual seat is taken, assigning it to be taken based on the percentage of seats that have already been sold. When you simulate the situation in this manner, it's entirely possible that the simulation won't have the precise percentages of seats indicated as sold. (If we randomly generate 100 integers in between 0 and 99, it may not be the case that exactly 80 of them are less than 80. The actual number less than 80 could vary some.)

Input Specification

The user will enter three values:

1. r – the number of rows in the concert venue
2. c – the number of columns in the concert venue
3. p – the percentage of seats that are already sold

r is an integer with $0 < r \leq 50$, and r is a multiple of 5

c is an integer with $0 < c \leq 100$, and c is a multiple of 20

p is an integer with $0 < p < 100$

Output Specification

Print out a seating chart with a blank line after every five rows, and a blank line after every 20 columns. Follow the output format in the sample run.

Sample Program Run #1 (User Input in Bold and Italics)

How many rows in the concert venue?

15

How many columns in the concert venue?

40

What percentage of the tickets are already sold?

74

Here is the seating chart:

```
XOXOOXOXXXXXXXXXXXXXXX XXXXXOXXXOXXXXXXXXXXXXX
XXOXXXXXXXXXXXXXXXXXXX XXXOXXXOXXXOXXXOXXX
XXOOXXOXXXXXXXXXXXXXXX OOOXXXOXXOXXXXXXXXXX
XXXOXXOXXOXXXXXXXOX XXXOOOXXXOXOXXXOXX
XXXXXOXOXXOXOOXXXXXXX XXXOXXOXOXXXXXOOXOXO

XXXXXXOXXXOXXOXXXXOX XXXXOXXXXXOXXXXXOXOX
XXXXOXXXXXOXOXXXXXXX XXXXOXXOOXXOXOXXXXXO
XXOXXXXOXXXXOXXXOXX XXOOOXXXXXXXXXXOXXXX
XXOXXOXXXOXXOXXXXXXX OOXXXXXXOXXXOXXOXXXX
XXXXOXXXOXXOOXXOXOOX XXXXXOOXXOXXXXXOOXX

XXXXOXXXXOXXXXXOOOXO OXXXXXXXXXXOXXXXXXXXXO
XXXOXOXXXOXXXOXXXXXX XXXOXXXXXXXXOXXXOXXXO
XOXXXOXOXXXXXXOXXXOX XXXXXXXXXXXXXXXXXXXXX
XOXOXXOXOXOXOOOXOXXX OOXOXOXXXOXXXXXOXXXX
OXXXXXXOXOOXXOXXXXXO XOOOOXXXXOXXXOXOOOXO
```

Sample Program Run #2 (User Input in Bold and Italics)

How many rows in the concert venue?

30

How many columns in the concert venue?

60

What percentage of the tickets are already sold?

8

Here is the seating chart:

```
0000000000000000000000 00X0X000XX0X00000000 000000000000X00000000
0000000000000000000000 00000000000000000000 0000X00000000000X0000
0000000000000000000000 0000XX000X0000000000 00000000000000000X000
00000X0X000000000000X0 00000X000000000000000 0000X0000000000000000
00000000000X0000000000 000X0000000000000000 0000000000X0000000000

00000000000000000000X0 000000000000000000000 000000000000000000000
0000000000000000000000 000X0X000000000000000X 000000000000000000000
00000000000000000000X0 000000000000000000000 000000000000000000000
0000000000000000000000 00000000000X000000000 0000XX0000000000000X0
0000000X00000000000000 000X00000000000000000 00000000000000X000000

0000000000000000000000 000000000000000000000 00000000000000X0XX000
0000000000000000000000 0XX0000000000000000X0X 0X0000000000000000000
0000000000000000000000 000000000000000000X000 00000X000000X00000000
0000000000X00X00000000 00000000000000000X000X 00000XX0000000000X000
0000000000000000000000 X0000000000X0000X000 000000000000000000000

000X00X000000000000000 00X00000000000000X0000 000XX00000000000000000
0000000000000000000000 00000000000000XX000000 0000000000XX000000000
00000000X0000000000000 X0000000X000000X0000 0X0000000000000000000
00000000000000000000X0 0000X000000000000000X 000000000000XX0000000
00X000000000000X00000 0000X0000000000000000 000X0000X000000X00000

00000000000000000000X0 000000X0000000000X000 000000000000000000000
000X00000000000000X0X0 000000000000000000000X 0000X0000000000000X00
0000000000000000000000 000000000000000000000 00000000000000000000X
0000000000000000000000 000000X00000X00000000 0000XX000000X00000000
00000000000X000000000 0000000000XX000000000 000000000000000000000

0000XX0000X0000000000 00000000X00000X00000 000000000000000000000
000000000000X00000X0 00000000000000X00000 00000000000000X000000
0000000000XX00000000 00000000000000000000X 000000000000000000000
00X000000000000000000 000X00X0000000000000 000000X00000000000000
0X00000X000000000000 000000000000000000X0 000000X00000000000000
```

Deliverables

Please submit three separate .java files for your solutions to these problems via WebCourses by the designated due date:

Program A: **tourdist.java**

Program B: **secretsong.java**

Program C: **seating.java**

Please make sure to include a header comment and internal comments in your code and indent when appropriate.