Assignment 1:

- A figure skating competition is held at the Winter Olympics. You are hired to design a system to store the medals won by different countries. There are 3 medal category- GOLD, SILVER and BRONZE. Once the Olympics is over, you must provide various functionalities as requested.
  For Example:

	Gold	Silver	Bronze
Canada	1	0	1
China	1	1	0
Germany	0	0	1
Korea	1	0	0
Japan	0	1	1
Russia	0	1	1
United States	1	1	0

Figure 13 Figure Skating Medal Counts

TASK:

- Take the number of countries as input and create an array of string to store the names of those many countries.
- Create a 2-D array to store the number of medals in different categories for every country.
- Print the data as shown in the figure above
- Print the count of countries that secured gold, silver and bronze medals
  Ex: gold – 4
  Silver – 4
  Bronze - 4
- Take the country's name as input and display the country's medal counts for different categories.

Assignment 2:

- An electric mountain railway makes a return trip every day. In a trip, the train goes up the mountain and back down. The train leaves from the foot of the hill at 09:00, 11:00, 13:00 and 15:00. The train returns from the top of the hill at 10:00, 12:00, 14:00 and 16:00. Train can accommodate 200 passengers.
- Passengers can only purchase a return ticket; all tickets must be purchased on the day of travel. The cost is $25 for the journey up and $25 for the journey down.
- Passengers must book their return train journey, as well as the departure train journey, when they purchase their ticket. Passengers can return on the next train down the mountain or a later train.

Task:

- Create 2 arrays to store the uptime and downtimes.
- Create a class Passenger with name, nooftickets, amount, date, downtime and uptime.
- Create an Array List of passengers and add the passenger object once that passenger is allocated tickets.
- Ask for passenger name, number of tickets to purchase, and if the number of tickets is available, ask time slots to book.
- Calculate total and store this passenger detail in the list. Repeat this until all tickets exhausted.
- Once the 200 available tickets are booked, display the passengers' details in descending order of the tickets purchased.
- Can create helper methods or classes as needed

If time permits, add below validations:

- Ask user to enter departure time until they enter the right time as the time slots available
- Ask user to enter return time until they enter the right time as the time slots available and it should be after the departure time.