IT5001 Scala OOP Assignment
Cruise Loaders

Topic Coverage

- Inheritance
- Polymorphism
- Method overriding
- String formatting

Problem Description

The Kent Ridge Cruise Centre has just opened and you are required to design a Scala application to decide how much equipment are needed based on a given daily cruise schedule. The cruises should be served as soon as they arrive. There are two types of cruises:

- Normal Cruise:
  - Requires only one loader and takes 30 min for a loader to fully load
- Big Cruise (with a cruise code that starts with 'B'):
  - Requires two loaders and each takes 60 min for each loader to fully load

The program determines the number of loaders for a cruise schedule and the allocation scheduling follows the following steps (sub-optimal):

- For each cruise that needs to be served, check through the inventory of loaders, starting from first loader in sequence.
- The first (or first two) available loader(s) will be allocated to the cruise based on its type (small or big).
- If there are not enough loaders, purchase a new one(s) to serve the cruise.

Task

Write a program that reads in the number of cruises in the schedule as an integer, and a list of cruises that will arrive for that day.

The program will output

- The number of normal cruises
- The number of big cruises
- The number of loaders required
- The loader allocation schedule

Do note the following assumptions:

- Input cruises are presented chronologically by arrival time
- There are up to 30 cruises in one day with no duplicates.

Although this problem can be done procedurally, use OOP as much as you can.

This task is divided into four levels.

Level 1: Represent a Cruise

Write a program that reads in the number of cruises that arrive daily, and the daily cruise schedule. Then, print out the cruise schedule in the format as `cruiseCode@time` such as `A1234@1330` which means that cruise A1234 arrives at 1330hrs.

The following is a sample run of the program. User input is underlined.

```
> scala Main_Level_1
1
A1234 0
A1234@0000
```

```
> scala Main_Level_1
6
A1111 0900
A1111@0900
B1112 0901
B1112@0901
B1113 0940
B1113@0940
C2030 1000
C2030@1000
B1115 1030
B1115@1030
D1115 1130
D1115@1130
```

Level 2: Find the allocation schedule for loaders

Write a program that reads in the number of cruises that arrive daily, and the daily cruise schedule. Then, print out the allocation schedule for each loader.

Ensure that each line denoting the cruise served is prefixed with 4 spaces.

The following is a sample run of the program. User input is underlined.

```
> scala Main_Level_2
1
A1234 0
Loader 1 serves:
    A1234@0000
```

```
> scala Main_Level_2
6
```

```
A1111 0900
A1112 0901
A1113 0940
C2030 1000
A1115 1030
D1115 1130
Loader 1 serves:
    A1111@0900
    A1113@0940
    A1115@1030
    D1115@1130
Loader 2 serves:
    A1112@0901
    C2030@1000
```

Level 3: Serve Big Cruises with purchase of loaders whenever necessary

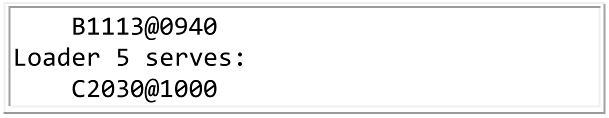
Write a program that reads in the number of cruises that arrive daily, followed by the daily cruise schedule.

Then, print out the allocation schedule for each loader.

The following is a sample run of the program. User input is underlined.

```
> scala Main_Level_3
1
B1111 0
Loader 1 serves:
    B1111@0000
Loader 2 serves:
    B1111@0000
```

```
> scala Main_Level_3
6
A1111 0900
B1112 0901
B1113 0940
C2030 1000
B1115 1030
D1115 1130
Loader 1 serves:
    A1111@0900
    B1113@0940
    D1115@1130
Loader 2 serves:
    B1112@0901
    B1115@1030
Loader 3 serves:
    B1112@0901
    B1115@1030
Loader 4 serves:
```



```
    B1113@0940
Loader 5 serves:
    C2030@1000
```

Level 4: Print statistics

Write a program that reads in the number of cruises that arrive daily, and the daily cruise schedule. Then, print out the following details:

- number of normal cruises
- number of big cruises
- number of loaders
- allocation schedule for the loaders

Note the following:

- The width of the containing box in the output should be 36 characters.
- Thick horizontal borders should use '='
- Thin horizontal borders should use '-'
- There should at least be a space between the first and last characters of each line
- Use the format specifier %3d to print integers of width = 3

The following is a sample run of the program. User input is underlined.

```
> scala Main_Level_4
1
A1111 1300
+==================================+
| Cruise Statistics                |
+----------------------------------+
| Number of normal cruises   =   1 |
| Number of big cruises      =   0 |
+==================================+
| Equipment statistics             |
+----------------------------------+
| Number of loaders          =   1 |
+==================================+
| Loader 1 serves:                 |
|     A1111@1300                   |
+==================================+
> scala Main_Level_4
4
A1111 1300
B1112 1300
C1113 1359
B1114 1400
+==================================+
| Cruise Statistics                |
+----------------------------------+
| Number of normal cruises   =   2 |
| Number of big cruises      =   2 |
+==================================+
| Equipment statistics             |
+----------------------------------+
| Number of loaders          =   3 |
+==================================+
| Loader 1 serves:                 |
|     A1111@1300                   |
|     C1113@1359                   |
+==================================+
| Loader 2 serves:                 |
|     B1112@1300                   |
|      B1114@1400                  |
+==================================+
| Loader 3 serves:                 |
|     B1112@1300                   |
|     B1114@1400                   |
+==================================+
> scala Main_Level_4
6
A1111 0900
B1112 0901
B1113 0940
C2030 1000
B1115 1030
D1115 1130
+==================================+
| Cruise Statistics                |
+----------------------------------+
| Number of normal cruises   =   3 |
| Number of big cruises      =   3 |
+==================================+
| Equipment statistics             |
+----------------------------------+
| Number of loaders          =   5 |
+==================================+
| Loader 1 serves:                 |
|     A1111@0900                   |
|     B1113@0940                   |
|     D1115@1130                   |
+==================================+
| Loader 2 serves:                 |
|     B1112@0901                   |
|     B1115@1030                   |
+==================================+
| Loader 3 serves:                 |
|     B1112@0901                   |
|     B1115@1030                   |
+==================================+
| Loader 4 serves:                 |
|     B1113@0940                   |
+==================================+
| Loader 5 serves:                 |
|     C2030@1000                   |
```

+==================================+

Submission Instruction:

Submit to coursmology a single zip file named yourname_studentnumber where yourname is your name in the student record and student_number is your student number.

The zip file contains a main folder also named yourname_studentnumber and has the following folder structure:

```
yourname_studentnumber
|__ Level1
      |_ files for Level1
|__ Level2
      |_ files for Level2
|__ Level3
      |_ files for Level3
|__ Level4
      |_ files for Level4
```