DAY5 ACTIVITIES

TOPIC : HIGHER ORDER FUNCTIONS

Part I

Q.1) Find the most populated city

Q.2) Find the number of movies of each director

Q.3) Find the number of genres of each director's movies

Q.4) Find the highest populated capital city

Q.5) Find the highest populated capital city of each continent

Q.6) Sort the countries by number of their cities in descending order

Q.7) Find the list of movies having the genres "Drama" and "Comedy" only

Q.8) Group the movies by the year and list them

Q.9) Sort the countries by their population densities in descending order ignoring zero population countries

Q.10) Find the richest country of each continent with respect to their GNP (Gross National Product) values.

Q.11) Find the minimum, the maximum and the average population of world countries.

Q.12) Find the minimum, the maximum and the average population of each continent.

Q.13) Find the countries with the minimum and the maximum population.

Q.14) Find the countries of each continent with the minimum and the maximum population.

Q.15) Group the countries by continent, and then sort the countries in continent by number of cities in each continent.

Q.16) Find the cities with the minimum and the maximum population in countries.

Q.17) Find the minimum, the maximum, the average, and the standard deviation of GNP values.

Q.18) Find the year where the maximum number of movie is available

1.
```
class City {
    private val id = 0
    private val name: String? = null
    private val population = 0
    private val countryCode: String? = null //getter & setter
}
```

2.

```
class Country {
    private val code: String? = null
    private val name: String? = null
    private val continent: String? = null
    private val surfaceArea = 0.0
    private val population = 0
    private val gnp = 0.0
    private val capital = 0
    private var cities: List<City>? = null

    init {
        cities = ArrayList()
    }
    //getter & setter
}
```

```
3. class Director {
    private val id = 0
    private val name: String? = null
    private val imdb: String? = null //getter & setter
}
```

```
4. class Genre {
    private val id = 0
    private val name: String? = null //getter & setter
}
```

```
5. class Movie {
    private val id = 0
    private val title: String? = null
    private val year = 0
    private val imdb: String? = null
    private var genres: List<Genre>? = null
    private var directors: List<Director>? = null

    init {
        genres = ArrayList()
        directors = ArrayList<Director>()
    }
}
```

Part - II

This is a short exercise in using the lambda functions. Suppose that the class Score is defined as

```
data class ScoreInfo(var lastName: String, var firstName: String, var score: Int)
```

and that scoreData is an array of ScoreInfos containing information about the scores of students on a test. Use the stream API to do each of the following tasks:

print the number of students (without using scoreData.length)

print the average score for all of the students

print the number of students who got an A (score greater than or equal to 90)

use the collect() stream operation to create a List<String> that contains the names of students whose score was less than 70; the names should be in the form first name followed by last name

print the names from the List that was generated in the previous task

print out the student's names and scores, ordered last name

print out the student's names and scores, ordered by score