**CS431: Programming Languages Lab (2020)**

**Department of Computer Science and Engineering**

**Indian Institute of Technology Guwahati**

## Assignment Set III

- **Assignments will be evaluated by the TAs.**
- **You should submit report (for answering non-code related questions, if any), complete source codes and executable files (whichever applicable).**
- **All codes must be properly documented and good code writing practice should be followed (carry marks).**
- **Copying is strictly prohibited. Any case of copying will automatically result in F grade for the whole course, irrespective of your performance in the other parts of the lab.**
- **Submission deadline: November 27, 2020.**
- **Marks distribution: 20, 30, 50.**

### 1. Implement Haskell Functions for Basic Set Operations (20 Marks)

Implement the Haskell functions for the following operation on Set.

A) Check whether a set is empty (NULL SET),
B) Union of two sets,
C) Intersection of two sets,
D) Subtraction one set from another set and
E) Addition of two sets.

In each of the cases check the possibility of the particular operation.

### 2. IITG Football League (30 Marks)

The IITG sports board has decided to promote a healthy campus lifestyle in the campus amongst the academic personal. So they have decided to inaugurate a yearly football league amongst the various academic departments of IITG. The event will start from 1st December to 7th December 2020. The sports board has received positive response from the campus residents. 12 teams have registered for the inaugural season out of which 11 teams are from different departments and 1 team is from the staff members. The tournament will be knock out in the first round followed by a league in the second round out of which top 4 teams will reach the playoffs.

Write a program in Haskell to carry out the draw for the fixtures. Your program should consider the following.

a. The draw will be for the initial round only where each team will play only one match.
b. Use acronyms of the registered teams as follows:

| Sl. No | Team | Acronym |
|---|---|---|
| 1 | Biosciences & Bio-Engineering | BS |
| 2 | Chemical Engineering | CM |
| 3 | Chemistry | CH |
| 4 | Civil Engineering | CV |
| 5 | Computer Science & Engineering | CS |
| 6 | Design | DS |
| 7 | Electronics and Electrical Engineering | EE |
| 8 | Humanities and Social Sciences | HU |
| 9 | Mathematics | MA |
| 10 | Mechanical Engineering | ME |
| 11 | Physics | PH |
| 12 | Staff Members | ST |

c. After the draw, generate the fixtures for the first round with date and time.
d. The first round will commence from 1st to 3rd December 2020. Schedule the fixtures in this period only.
e. There will be two matches per day. The first match will start at 9:30 AM and the second match will start at 7:30 PM
f. Your program should be able to display the entire fixture, match details about any particular team and the next match details.

***Sample Input/ Output:***

```
*Main> fixture 'all'
  CS vs PH      1-12-2020        9:30
  MA vs ST      1-12-2020        7:30
        ----
        ----
  CV vs DS      3-12-2020        7:30

*Main> fixture 'DS'
  ME vs DS      3-12-2020        7:30

*Main> nextMatch 1  13.25    [This means what is the next match when
                             current date is 1/12 and current time is 1:15 PM]
  MA vs ST       1-12-2020       7:30
```

**3. House Planner** **(50 Marks)**

To create a dream house with the best utilization of space, a good design is necessary. Write a program in Haskell to suggest the best possible design of a house in the available space. Consider the following assumptions:
The house can have bedroom(s), hall(s), kitchen(s), bathroom(s), balcony and garden. Their dimension can be of the following range (the numbers represent some unit):

Bedroom → (10 × 10) to (15 × 15)

Hall → (15 × 10) to (20 × 15)

Kitchen → (7 × 5) to (15 × 13)

Bathroom → (4 × 5) to (8 × 9)

Balcony → (5 × 5) to (10 × 10)

Garden → (10 × 10) to (20 × 20)

Given a space (assume a square area) in square units and the number of the two components (bedroom and hall); find the dimensions of all the components (bedroom, hall, kitchen, bathroom, garden and balcony). Note the following while designing:

1) The dimension of a kitchen must not be larger than that of a hall or a bedroom

2) The dimension of a bathroom must not be larger than that of a kitchen

3) There can be one kitchen for up to three bedrooms

4) Number of bathrooms is one more than the number of bedrooms

5) There must be one garden and one balcony

6) If some space remains such that no further increase in any component is possible, then return it as 'Unused Space'

If no design is possible with the given constraint, return "No design possible for the given constraints" as output.

***Sample input/output***

***Input:***

Design (1000, 3, 2)

***Output:***

Bedroom: 3 (11 × 11)

Hall: 2 (16 × 11)

Kitchen: 1 (8 × 6)

Bathroom: 4 (4 × 5)

Garden: 1 (11 × 11)

Balcony: 1 (6 × 6)

Unused Space: 0

**Answer the following in your report.**

1. Write the algorithm (in pseudo-code) that you devised to solve the problem (**you must not write the code for the algorithm in the report**).

2. How many functions did you use?

3. Are all those pure?

4. If not, why? (Means, why the problem can't be solved with pure functions only).

**In addition, write short notes on the following in the report.**

a) Do you think the *lazy* evaluation feature of Haskell can be exploited for better performance in the solutions to the assignments? If so, which solution(s) and how?

b) We can solve the problems using any imperative language as well. Do you find any advantage of using Haskell for these problems (w.r.t the property of lack of side effect)? If your answer is no, elaborate on why not?

---------- *********0********* ----------