CS-431 PROGRAMMING LANGUAGES LAB PROJECT README

# QUESTION-1 BASIC SET OPERATIONS

**NOTE** : Since it is not specified, we are taking any kind of input, the set can be a number set or a string set,etc. This is achived by using the **Ord** class as can be seen in the code.

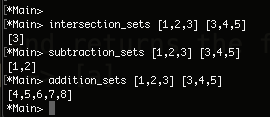
#### Instructions to run

1. ghci
2. :l ques\_1

#### Check if set is empty

1. check\_empty [1,3,4]

#### Find Union of Sets

1. union\_sets [1,2,3] [3,4,5]

#### Find Intersection of Sets

1. intersection\_sets [1,2,3] [3,4,5]

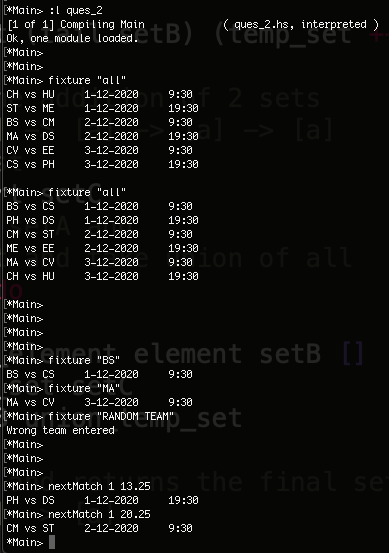
#### Find Subtraction of Sets

1. subtraction\_sets [1,2,3] [3,4,5]

#### Find Additon of Sets

1. additon\_sets [1,2,3] [3,4,5]

# QUESTION-2 IITG FOOTBALL LEAGUE

**NOTE:** As specified in the discussion forum, **everytime** the user enters ‘fixture “all”’, we **generate** a **new** **random** **schedule**.

Moreover, to avoid confusion we have stuck to the 24 hour clock format.

So instead of writing 7:30, we display 19:30.

#### Instructions to run

1. ghci
2. :l ques\_2

#### To display all fixtures

1. fixture “all”

#### To display fixture for a specific team

1. fixture “BS”

#### To display next fixture for a given date and time

1. nextMatch 1 13.25

# QUESTION-3 HOUSE PLANNER

**NOTE:** We have used some optimizations as given in the report to prune the search space and return the output. Still some input may take a little longer to run. I have provided 3 inputs and the time it took to run on my machine to get a rough idea.

#### Instructions to run

1. ghci
2. :l ques\_3

#### Design an Architecture for a given input

1. 1) design 1000 3 2 (it took 26 seconds to execute)

2) design 730 1 1 (it took 8 seconds to execute)

3) design 2450 4 1 (it took 55 seconds to execute)