SUMMARY

```
The Symposium Scheduler is used to schedule panels that have some
constraints on them into venues across time slots.

These are the user instructions for running the Symposium Scheduler.
```

REQUIREMENTS

```
In order to run the scheduler, you will need
1) Symposium
2) A properly formatted JSON file containing the necessary information
    (More on this under Input Formatting)
```

INSTALLATION

```
Simply double click Symposium to run the algorithm after ensuring
all requirements have been met.
```

RUN

```
There is a three options to running the application. The first one is double
click Symposium.jar to get a user interface. The other two options are using the
command line as follows:
$ Symposium input.json
$ Symposium input.json output.json
```

INPUT FORMATTING

```
In order for the Rener-Alshakh algorithm to run properly, the user
must supply a JSON file following the format as described below.
If there is any confusion, please refer to example_info.txt for an
example.

The formatting should be as follows, noting that
1) whitespace is not needed and
2) replace any word in ALL CAPS by an actual value
3) "..." indicates that this pattern can go on indefinitely.
4) All times (00:00-00:00) are 24-hour based and must always have 4
    digits seperated by a ":"

{
    "Venues":[
        {"name": "NAME", "size": SIZE, "priority": PRIORITY},
        {"name": "NAME", "size": SIZE, "priority": PRIORITY},
        ...
        {"name": "NAME", "size": SIZE, "priority": PRIORITY}
    ],
    "Venue-Times":[
        { "name": "NAME", "time": "DAY_NUMBER, 00:00-00:00"},
        { "name": "NAME", "time": "DAY_NUMBER, 00:00-00:00"},
        ...
        { "name": "NAME", "time": "DAY_NUMBER, 00:00-00:00"}
    ],
    "Panelists":[
        {
            "name": "PERSON",
            "times": [
                "DAY_NUMBER, 00:00-00:00",
                "DAY_NUMBER, 00:00-00:00",
                "DAY_NUMBER, 00:00-00:00",
                ...
                "DAY_NUMBER, 00:00-00:00"
            ],
            "new": "YES OR NO"
        },
        ...
        {
            "name": "PERSON",
            "times": [
                "DAY_NUMBER, 00:00-00:00",
                "DAY_NUMBER, 00:00-00:00",
                "DAY_NUMBER, 00:00-00:00",
                ...
                "DAY_NUMBER, 00:00-00:00"
            ],
            "new": "YES OR NO"
        }
    ],
    "Panels":[
        {
            "name": "TITLE",
            "panelists": [
                "PERSON",
                "PERSON",
                ...
                "PERSON"
            ],
            "constraints": [
                "CONSTRAINT:PRIORITY",
                "CONSTRAINT:PRIORITY",
                ...
                "CONSTRAINT:PRIORITY"
            ],
            "category": "CATEGORY"
        },
        ...
        {
            "name": "TITLE",
            "panelists": [
                "PERSON",
                "PERSON",
                ...
                "PERSON"
            ],
            "constraints": [
                "CONSTRAINT:PRIORITY",
                "CONSTRAINT:PRIORITY",
                ...
                "CONSTRAINT:PRIORITY"
            ],
            "category": "CATEGORY"
        }
    ]
}
```

CONSTRAINTS

```
For every panel there are 4 assumed constraints:

Panelist constraint:
A panelist can’t be assigned to multiple locations at the same time.

Consecutive Panels Constraint:
A panelist may not be assigned to three panels that are scheduled very close
together.

Preferred Venues Filter:
This filter is used to order venue times based on priorities given in the
input file for each venue.

Venue Time Duration Filter:
This filter checks the venue times and increase their score based on the length
of the venue time. If there are two or fewer panelists on the panel, a short
venue time is preferred. If a panel has more than two panelists, a long venue
time is preferred.

These are the available constraints that can be added to panels with these
exact formats:
PRIORITY ranges from 1 to 3. 1 = REQUIRED, 2 = VERY_IMPORTANT, 3 = DESIRED.
If a constraint has a priority of REQUIRED that is violated, the panel won’t
be scheduled.  As the priority number increases, the importance of the
constraint decreases.

What is between the quotes must be put instead of the CONSTRAINT above

“New-Panelist”: This constraint indicates that the panel includes a panelist who
is new to the conference.

“Paired-Panelists”: This constraint is to prevent any two panelists from
appearing together twice or more in a single day.

"Single-Category": This constraint is to prevent any panels having the same
category from appearing at the same time.

“Max-Panels(MAX_NUMBER)”: This constraint is to prevent a panelist from
appearing more than the MAX_NUMBER of times per day. MAX_NUMBER should be
replaced with a number.
```

```
“Min-Panels()”: This filter prioritizes scheduling panelists in days they
haven’t been assigned yet.

“Minimum-Capacity(MINIMUM_SIZE)”: This constraint is for the panel to be
scheduled at a room with at least size MINIMUM_SIZE. MAX_NUMBER should be
replaced with a number.

“Availability”: This constraint is to prevent a panel from being scheduled at a
time where one or more panelists on the panel are not available.

“Venue(VENUE_NAME)”: This constraint is to schedule a panel at the given
VENUE_NAME. VENUE_NAME must be replaced by the name of the venue, and it must
be declared as a venue.

“Time(DAY_NUMBER;00:00)”: This constraint is to schedule a panel at the given
time. DAY_NUMBER is the number of the day for the panel to be scheduled at.
```

OUTPUT FORMAT

```
Once finished, the program will create a json file in the directory it
is running from. For reference, the file will follow
the below format:

{
    "Scheduled":[
        {
            "Panel":"PANEL_NAME",
                "Messages":[
                    "MESSAGE1",
                    “MESSAGE2“,
                    ...
                ],
            "Venue":"VENUE_NAME",
            "Time":"DAY_NUMBER, 00:00-00:00"
        },
        {
            "Panel":"PANEL_NAME",
                "Messages":[
                    "MESSAGE1",
                    “MESSAGE2“,
                    ...
                ],
            "Venue":"VENUE_NAME",
            "Time":"DAY_NUMBER, 00:00-00:00"
        },
        ...
    ],
    “Messages”:[
        "MESSAGE1",
        “MESSAGE2“,
        ...
    ],
    "Unscheduled":[
        {
            "Panel":"PANEL_NAME",
            "Messages":[
                "MESSAGE1",
                “MESSAGE2“,
                ...
            ]
        },
        {
            "Panel":"PANEL_NAME",
            "Messages":[
                "MESSAGE1",
                “MESSAGE2“,
                ...
            ]
        },
        ...
    ]
    "Underutilized_Panelists":{
        "PANELISTS_NAME":[DAY_NUMBER1, DAY_NUMBER2, ... ],
        "PANELISTS_NAME":[DAY_NUMBER1, DAY_NUMBER2, ... ],
        ...
    }
}


Note that the messages given in the output are error related, or explain
unnecessary constraints that were violated, and why.
```