

# Programming Assignment I

## Date Data Type

Due Date: 2023/09/19 (20 points)

### 1 Description of the Assignment

Implement a new data type, called *date*.

1. Input (initialize the date)  
Provide at least *yyyy/mm/dd* as input.
2. Output (convert the date to string)  
In addition to *yyyy/mm/dd*, generate at least *MonthName dd, yyyy* as output, where *MonthName* is the name of the month *January*, *February*, ..., *December*.
3. Operations  
Implement at least the following operations:
  - (a) DayOfWeek(date)  
Return day of the week: *Sunday*, *Monday*, ... *Saturday*.
  - (b) DateSub(date1, date2)  
Return number of days from *date1* to *date2*.
  - (c) DateAdd(date, n)  
Return the date which is *n* days after *date*.

And then write a program to demonstrate the usage of the new data type according to the following requirements.

### 2 Input Format

There are 3 types of input format:

1. *yyyy/mm/dd*
2. *yyyy/mm/dd - YYYY/MM/DD*
3. *yyyy/mm/dd + x*

### 3 Output Format

1. For the first type of input "*yyyy/mm/dd*" print out

*month date, year is weekday*

For example, on input “*2019/9/20*”, print out

September 20, 2019 is Friday

2. For the second type of input “*yyyy/mm/dd - YYYY/MM/DD*” print

*x* days from *month date, year* to *Month Date, Year*

For example, on input “*2018/9/20 - 2019/9/20*”, print out

365 days from September 20, 2019 to September 20, 2018

Note that “YYYY/MM/DD” may come before or after “yyyy/mm/dd”.

3. For the third type of input “*yyyy/mm/dd + x*” print out

*x* days after *month date, year* is *Month Date, Year*

For example, on input “*2019/9/20 + 365*”, print

365 days after September 20, 2019 is September 19, 2020

Note that *x* may be negative. For simplicity, the input can be

2019/9/20 + -65

## Notes

The format of the report of the assignment should be close to the format of a research technical report. Include at least the following in your report.

1. Title and Author.

This section should include *assignment number*, *your name*, *student number* and *email address* on the *first* page of your report.

2. Statement of the problem.

A “formal” description of the problem in this assignment. In addition to the basic requirements specified in the assignment, highlight other functions or features that you have implemented.

3. Main results.

This section should include at least the following items.

- (a) Description of the design of your program.
- (b) Describe the data structures used in the program to improve the efficiency of the program. These data structures should be implemented by you and appear in the first part of your program.

- (c) List of your program with comments.
  - i. If your program is very long, list only the main parts of the program here and the entire program in appendix.
  - ii. Additional comments can be added manually to explain the design of the program.
- (d) Outputs of the compilation and the executions of your program.

#### 4. Conclusions

Give a brief summary of what you did, and interesting thing you learned from this assignment.

Additional notes:

1. Turn in your report on or before due day.
2. The output of the program execution should indicate the correctness of your program. In other words, a set of [comprehensive](#) (but not necessarily exhaustive) annotated test data for the problem should be provided to show that your program is indeed correct. This can be done by carefully selecting a set of test data.
3. Print or write the report on A4 papers. Bind them together in the upper left corner.