Programming Assignment I
Define New Data Type (New Object)

Due Date: 2025/09/10 (20 points)

1 Description of the Assignment

This assignment contains 2 parts.

1. Choose a programming language, such as C or C++, and set up your programming environment.
2. Create a new data type, called *date*, and implement the new data type as described below.

Implement *date* data type

1. Input (initialize the date)
   Provide at least *yyyy/mm/dd* as input to the creation of a data.
2. Output (convert the date to string)
   In addition to *yyyy/mm/dd*, generate at least *MonthName dd, yyyy* as output string, where *MonthName* is the name of the moth 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, 2018 to September 20, 2019.

   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 each assignment report should be closse to a technical research report and must include at least the following sections:

1. **Title and Author**
   On the first part of the first page, clearly include:

   - Assignment number
   - Your name

- Student number
- Email address

2. **Description of the Problem**
   Provide a clear and formal description of the problem of this assignment. In addition to the basic requirements given in the assignment, highlight any extra functions or features you have implemented. Do not simply copy the assignment instructions into this section.

3. **Main Results**
   This section should include at least the following:

   (a) Program Design.
   Explain the overall design of your program. If any part of the design was obtained from references, discussions with other people, or other sources, proper citations must be provided.

   (b) Data Structures.
   Describe the data structures you implemented to improve efficiency. These should be your own implementations and appear in the first part of your program.

   (c) Program Listing with Comments.

   i. If the program is long, include only the main parts in the report body, and place the complete program in the appendix.

   ii. Add explanatory comments where appropriate to clarify your design.

   (d) Program Outputs.
   Include compilation results and execution outputs. Whenever possible, provide screen dumps.

4. **Performance Evaluation**

   (a) Report execution times of your program with various input sizes, such as $n = 100, 200, \ldots, 1000$.

   (b) Indicate the maximum input size your program can handle within a reasonable time limit (e.g., 1, 5, or 10 minutes).

5. **Conclusions**

   (a) Summarize what you accomplished and any interesting insights gained from this assignment.

   (b) Describe the challenges you encountered during development and how you overcame them. (This provides strong evidence that you completed the work independently.)

Additional Notes

1. Submit your report on or before the due date.
2. The program output should clearly demonstrate correctness. That is, provide a set of comprehensive (but not necessarily exhaustive) annotated test cases to show that your program works correctly.
3. Print the report on A4 paper and staple it in the upper-left corner.