Assignment One (8%)

Due Date: 25 February 2024, 23:59 PM

1. This assignment contains **3** questions. You are required to finish a complete C++ program (.cpp only) for each question on E-quiz before the deadline.

2. You can check as many times as you want before the deadline. We will grade your **latest version**.

3. Only a small set of the test cases are visible for the testing, and a more comprehensive set of test cases will be used for grading. In other word, passing all the visible test cases does not mean that you can get the full mark for this question. Try your best to thoroughly check your program.

4. Hidden test cases will not be released.

5. The marking of each question is based on the percentage of the total test cases that your solution can pass. If your submitted solution leads to a compilation error on E-quiz, zero mark will be given to that question and no manual checking to your solution is provided in such a case.

6. No late submission will be accepted.

7. Plagiarism check will be performed.

8. You only need to use the material from Lectures 1 to 5. It is **NOT** necessary to include any other library, except <iostream> <iomanip>.

9. You need to check your solution and submit it on E-quiz.

To ensure timely feedback to the students' questions, each of the following TAs will be responsible for one question. If you have any questions, you can contact the corresponding TA directly.

Question	**TA responsible**	**Email Address**
Q1	MUKASHEV Alikhan	amukashev2-c@my.cityu.edu.hk
Q2	HONG Chenhao	HONG.Chenhao@my.cityu.edu.hk
Q3	SHATOKHIN Anton	ashatokhi2-c@my.cityu.edu.hk

Q1: [Print M or W]

Write a program to meet all the following requirements:

1. Receive two input values.
    a. An integer representing the size of the letter (M or W) to print. It represents the number of lines of the whole letter to be printed out.
    b. A character which indicates to print M or W or “Wrong Input!”. If this character is M, then output a shape of M (see below); if this character is W, output a shape of W. Otherwise the program will print "Wrong input!".
2. The shape of M can be seen as two isosceles triangles without the base edges, the bottom-right vertex of the first triangle overlaps the bottom-left vertex of the second triangle. The shape of W is the upside down the shape of M.

NOTE:

1: When the height is less than(or equal to) 1, the output should be "Wrong input!"

2: M or W should be case sensitive. (that means if the input char is "m" or "n" , the result should be "Wrong input!")

Example 1:

```
Please input the height (integer):
5
Do you want to print M or W?:
M
    *       *
   * *     * *
  *   *   *   *
 *     * *     *
*       *       *
```

Example 2:

```
Please input the height (integer):
5
Do you want to print M or W?:
W
*       *       *
 *     * *     *
  *   *   *   *
   * *     * *
    *       *
```

Example 3:

```
Please input the height (integer):
5
Do you want to print M or W?:
A
Wrong input!
```

Q2: [Gas Station Nearby Map Program]

Write a program to implement a nearby gas stations map with the following requirements:

1) Input exactly 10 gas station records.
    a) Each gas station record contains one character (A-Z) and one number (0-10).
    b) The character stands for the name of gas station and the number indicates the distance (km) from your location to the gas station.
2) Print the nearby map.
    a) The map should only include gas stations within 5km or less from your current location.
    b) Each line of the map can list at most three gas stations. If a line already has three stations, start a new line for additional ones.
    c) The gas station at the shortest distance should be highlighted with an asterisk (“*”).

NOTE:

○ You can assume all input will be valid and formatted correctly.

Example 1:

```
Input the gas station data:
A 1 B 2 C 6 D 8 E 9 F 10 J 0 H 4 I 3 G 9
The nearby gas stations map is:
A(1) B(2) J(0)*
H(4) I(3)
```

Example 2:

```
Input the gas station data:
A 1 M 4 V 3 H 1 K 2 L 9 N 4 S 2 P 5 G 8
The nearby gas stations map is:
A(1)* M(4) V(3)
H(1)* K(2) N(4)
S(2) P(5)
```

Q3: [Library Management System]

Implement a Library Management System that allows the user to manage books in a library. The system should be able to add new books, display all books, delete a book by ID, and sort the books by either ID or date of addition. The maximum number of books in library management system is fixed to 10. And the ID of books are unique.

Requirements:

1. Complete the definition of the structure Date to hold the day, month, and year as integers.
2. Complete the definition of the structure Book to hold an integer ID, a double price, and a Date.
3. Complete the definition of the structure Library to hold an array of Book structures and an integer to keep track of the current number of books.
4. Implement the listOptions function to print the menu options for the user as shown in the following examples.
5. Implement the add function to allow adding a new book to the library. This function should prompt the user for the unique book ID, price, and date (day, month, year). Ensure that the library does not exceed its maximum capacity (10). The ID of the new book will not conflict with the old book.
6. Implement the show function to display all the books in the library. Each book should be displayed with the ID, price, and date formatted correctly.
7. Implement the del function to delete a book from the library by its ID. The user should be prompted to enter the ID of the book to delete.
8. Implement the sort_id function that sorts the books in ascending order based on their ID.
9. Implement the sort_date function that sorts the books based on their date of addition. If two books have the same date, they should be sorted by ID.
10. Implement the sort function to allow the user to choose how to sort the books (by ID or by date).

Instructions:

- You are provided with a code framework and a partially implemented main function.
- Your task is to complete the missing parts of the code to make it functional as per the requirements.
- Do not change the provided constants and function signatures.
- You can assume all input will be valid and formatted correctly.

```
// Code framework:

#include <iostream>
#include <iomanip>
using namespace std;

const int N = 10;

struct Date {

};
```

```
struct Book {

};

struct Library {

};

void listOptions() {

}

void add(/* */) {

}

void show(/* */) {

}

void del(/* */) {

}

void sort_id(/* */) {

}

void sort_date(/* */) {

}

void sort(/* */) {
    int n;
    cout << "1: Sort by ID" << endl;
    cout << "2: Sort by Date" << endl;
    cin >> n;

}

void init(Library *p) {
    p->books[p->num].id = 3;
    p->books[p->num].price = 10.5;
    p->books[p->num].date.day = 15;
    p->books[p->num].date.month = 1;
    p->books[p->num].date.year = 1990;
    (p->num)++;

    p->books[p->num].id = 2;
    p->books[p->num].price = 20.55;
    p->books[p->num].date.day = 15;
    p->books[p->num].date.month = 2;
    p->books[p->num].date.year = 2024;
    (p->num)++;

    p->books[p->num].id = 4;
    p->books[p->num].price = 10.5;
    p->books[p->num].date.day = 20;
    p->books[p->num].date.month = 2;
    p->books[p->num].date.year = 2021;
    (p->num)++;

    p->books[p->num].id = 1;
    p->books[p->num].price = 30.1;
    p->books[p->num].date.day = 20;
    p->books[p->num].date.month = 2;
    p->books[p->num].date.year = 2021;
    (p->num)++;
}


int main() {
    Library lib;
    lib.num = 0;

    init(&lib);

    int opt;

    return 0;
}
```

Example 1: Add a book

```
~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
1
Input Book ID:
5
Input Price:
15.99
Input Date (dd mm yyyy):
21 12 2023

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
003     10.50   15-1-1990
002     20.55   15-2-2024
004     10.50   20-2-2021
001     30.10   20-2-2021
005     15.99   21-12-2023

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
0
Bye!
```

Example 2: Delete a book

```
~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
1
Input Book ID:
7
Input Price:
22.5
Input Date (dd mm yyyy):
11 11 2021

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
3
Enter the ID to be deleted
7

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
003     10.50   15-1-1990
002     20.55   15-2-2024
004     10.50   20-2-2021
001     30.10   20-2-2021

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
0
Bye!
```

Example 3: Sort books by ID

```
~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
003     10.50   15-1-1990
002     20.55   15-2-2024
004     10.50   20-2-2021
001     30.10   20-2-2021

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
4
1: Sort by ID
2: Sort by Date
1

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
001     30.10   20-2-2021
002     20.55   15-2-2024
003     10.50   15-1-1990
004     10.50   20-2-2021

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
0
Bye!
```

Example 4: Sort books by date

```
~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
003     10.50   15-1-1990
002     20.55   15-2-2024
004     10.50   20-2-2021
001     30.10   20-2-2021

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
4
1: Sort by ID
2: Sort by Date
2

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
003     10.50   15-1-1990
001     30.10   20-2-2021
004     10.50   20-2-2021
002     20.55   15-2-2024

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
0
Bye!
```

Example 5: Delete non-existent book

```
~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
2
003     10.50  15-1-1990
002     20.55  15-2-2024
004     10.50  20-2-2021
001     30.10  20-2-2021

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
3
Enter the ID to be deleted
7
The ID cannot be found

~~~~~~~~~Welcome!~~~~~~~~~~
0: Exit
1: Add
2: Show
3: Delete
4: Sort
~~~~~~~~~~~~~~~~~~~~~~~~~~~
0
Bye!
```