

CST 370 – Spring (B) 2022
Homework 2
Due: 3/15/2022 (Tuesday) (11:55 PM)

How to turn in: Write **three programs** in **either C++ or Java** and submit them on Canvas before the due.

- You **can submit** your programs **multiple times** before the due. However, the **last submission will be used for grading**.
- You have to **submit three programs together**, especially at your last submission. **If you submit, for example, only one program** at the last submission, **we are able to see only that program when we grade** your homework.
- Due time is 11:55(PM). Since there could be a long delay between your computer and Canvas, you should submit it early.
- When you submit your homework program, don't forget to include "Title", "Abstract", "ID", "Name", and "Date".

1. Write a program called **hw2_1.cpp (or hw2_1.java)** that reads two timestamps of two events from a user and displays the difference between the two timestamps. For the program, you can assume that each timestamp is composed of the hour (0 ~ 23), minute (0 ~ 59), and second (0 ~ 59) format. Your program should present the difference from the second event (= second timestamp) to the first event (= first timestamp). Note that **the second event always happens after the first event** and your program should display the time difference of the events.

Sample Run 0: Assume that the user typed the following two lines.

```
18:45:30
20:50:59
```

This is the correct output of your program.

```
02:05:29
```

Sample Run 1: Assume that the user typed the following two lines.

```
20:18:59
04:25:17
```

This is the correct output of your program.

```
08:06:18
```

Sample Run 2: Assume that the user typed the following two lines.

```
02:00:25
15:30:00
```

This is the correct output of your program.

```
13:29:35
```

2. Write a C++ program called **hw2_2.cpp (or hw2_2.java)** that reads a number of elements in a set first. Then, your program should read the elements of the set. After that, your program should display all possible decimal numbers, corresponding binary numbers, and subsets one by one. For the program, you can assume that **the number of elements in a set is less than 10**.

Sample Run 0: Assume that the user typed the following input. Note that there are three elements in the set with the elements A, B, and C.

```
3
A B C
```

This is the correct output.

```
0:000:EMPTY
1:001:C
2:010:B
3:011:B C
4:100:A
5:101:A C
6:110:A B
7:111:A B C
```

The first line indicates that the first decimal number and its binary number are “0” and “000”. Note that the first bit ‘0’ in the binary number is for the element ‘A’, the second bit ‘0’ for the element B, and the last bit ‘0’ for the element C. Since all three bits in the binary number are ‘0’, the corresponding subset is “EMPTY”.

The next line indicates that the second decimal number and its binary number are “1” and “001”. The corresponding subset is {C} because only the last bit of the binary number is ‘1’.

This way, your program should display a decimal number, its binary number, and corresponding subset one by one. For instance, the last line indicates that the last decimal number and its binary number are “7” and “111”. The corresponding subset is {A, B, C} because all bits of the binary number are ‘1’.

Sample Run 1: Assume that the user typed the following input.

```
2
CST238 CST370
```

This is the correct output.

```
0:00:EMPTY
1:01:CST370
2:10:CST238
3:11:CST238 CST370
```

Sample Run 2: Assume that the user typed the following input.

```
0
```

This is the correct output of your program.

```
0:0:EMPTY
```

[**Hint**]: Refer to this program to convert a decimal number to corresponding binary number - <https://www.geeksforgeeks.org/program-decimal-binary-conversion/>

3. Write a C++ (or Java) program named **hw2_3.cpp** (or **hw2_3.java**) which checks if an input string is a palindrome or not. In the problem, a palindrome means an alphanumeric string which reads the same front to back. For the problem, you should ignore case and remove all non-alphanumeric characters in the input string. For example, “**Race car**”, “**I did, did I?**”, “**7...8 Don't nod 87.**” are all palindromes. But “**CSUMB**” is not a palindrome.

In this program, you **have to use a recursive function to check the palindrome**. For the grading, we will read your source code. **If you do not use a recursive function, you will get zero even if your program passes all test cases.**

Sample Run 0: Assume that the user typed the following one line

```
Race car
```

This is the correct output of your program.

```
TRUE
```

Sample Run 1: Assume that the user typed the following one line

```
7...8 Don't nod 78.
```

This is the correct output of your program.

```
FALSE
```

Sample Run 2: Assume that the user typed the following one line

```
7 77 777 7777 UFO tofu?      7777777777
```

This is the correct output of your program.

TRUE

[Hint] First, convert the input string to a string with only alphanumeric characters in uppercase. After that, check if the converted string is a palindrome or not using a recursive function.