

Assignment 1: Set Operations

Due: 23rd June 2022, 11:59 pm

- Write a program in C++ to compute the union, intersection, or difference of given input sets.
- Elements of the sets are restricted to single digits (0 to 9). For the input and output, the sets are represented as strings of digits.
 - For example, the set $\{4, 8\}$ is represented as either the string "48" or "84" (note that the order is not relevant).
- The empty set (\emptyset) is represented by the string "empty".
- The input can be one of two types:
 - The form "[first set] [op] [second set]", where [op] is either the character "U", "I", or "D" representing the union, intersection, or difference operations respectively. There is a space on both sides of the operation character. This type of input can be entered multiple times.
 - The character "X" only, to indicate the end of the program.
- If the user inputs "X", the program should terminate. Otherwise, the program should output the result of the operation and allow the user to enter new input.
- There should be a new line after each output.

Sample Input:	Sample Output:
1234 U 2468	132468
1234 I 2468	24
1234 D 2468	13
2468 D 1234	68
605 U empty	605
empty I 605	empty
605 D empty	605
empty D 605	empty
X	

The next page contains important guidelines

Guidelines:

- The file name should be your student ID, with the extension `.cpp`, submitted in the Google Classroom assignment.
 - If you are not comfortable with programming in C++ for this assignment, please contact the instructor to discuss exceptional alternatives.
- The allowed standard libraries are `iostream`, `cstdio`, and `cstring`. **No other libraries or external code are allowed.**
- There are **no duplicates** in a set, and there is **no required order** for the elements. This applies to both the expected input and the generated output.
- You can assume that **all inputs will be valid**, i.e., you do not have to deal with invalid input (such as when the set contains duplicates or non-numerical characters, or when the format is not followed).
- It is strongly recommended that you write a function to check whether a given character appears in a given string or not.
- Efficiency is not considered here, except for extreme cases (like generating a power set, or getting stuck in an infinite loop when trying to compute a required output).