Lab 12

Text Processing Using Strings

https://github.com/mmujtaba25/cs-110

Muhammad Mujtaba

CMD ID: 540040

mmujtaba.bese25seecs@seecs.edu.pk

Class: BESE 16B

Batch: 2k25

Task 1 [CLO 2]

CODE:

```cpp
#include <iostream>
#include <string.h>

int main()
{
    std::string first_city = "";
    std::string second_city = "";
    std::string third_city = "";

    std::cout << "Enter first city name: ";
    std::getline(std::cin, first_city);
    std::cout << "Enter second city name: ";
    std::getline(std::cin, second_city);
    std::cout << "Enter third city name: ";
    std::getline(std::cin, third_city);

    // compare first and second
    if (second_city < first_city)
        std::swap(first_city, second_city);

    // compare first and third
    if (third_city < first_city)
        std::swap(first_city, third_city);

    // compare second and third
    if (third_city < second_city)
        std::swap(second_city, third_city);

    std::cout << "Cities in alphabetical order: ";
    std::cout << first_city << " - ";
    std::cout << second_city << " - ";
    std::cout << third_city << "\n";

    return 0;
}
```

OUTPUT:

```
obscure@Obscures-MacBook-Air output % ./"task1"
Enter first city name: Chicago
Enter second city name: Los Angeles
Enter third city name: Atlanta
Cities in alphabetical order: Atlanta - Chicago - Los Angeles
obscure@Obscures-MacBook-Air output %
```

Task 2 [CLO 3]

CODE:

```cpp
#include <iostream>
#include <cstring>

constexpr size_t ISBN_STR_SIZE = 13;

int getChecksum(const char s[ISBN_STR_SIZE - 1]);

int main()
{
    // testing with example values
    char testValues[][ISBN_STR_SIZE] = {
        "978030640615", // 7
        "978014028657", // 1
        "978160234061"  // 9
        // values tested from https://freeisbn.com/check-digit/
    };

    for (auto &arr : testValues)
    {
        int checksum = getChecksum(arr);

        std::cout << "The ISBN-13 number is: ";
        for (size_t i = 0; i < ISBN_STR_SIZE - 1; ++i)

            std::cout << arr[i];

        std::cout << checksum << "\n";
    }

    char isbn[ISBN_STR_SIZE]; // 12 + null digit
    std::cout << "Enter first 12 digits of ISBN: ";
    std::cin >> isbn;

    int checksum = getChecksum(isbn);
    std::cout << "The ISBN-13 number is: " << isbn << checksum << "\n";

    return 0;
}

int getChecksum(const char s[ISBN_STR_SIZE - 1])
{
    // works because of ascii values ordering
    // '0' translates to ascii value of 0
    int d1 = s[0] - '0';
    int d2 = s[1] - '0';
    int d3 = s[2] - '0';
    int d4 = s[3] - '0';
    int d5 = s[4] - '0';
    int d6 = s[5] - '0';
    int d7 = s[6] - '0';
    int d8 = s[7] - '0';
    int d9 = s[8] - '0';
    int d10 = s[9] - '0';
    int d11 = s[10] - '0';
    int d12 = s[11] - '0';

    // applying formula
    int checksum = 10 - ((d1 + 3 * d2 + d3 + 3 * d4 + d5 + 3 * d6 + d7 + 3 * d8 + d9 + 3 * d10 + d11
+ 3 * d12) % 10);

    // return 0 if 10
    return (checksum == 10) ? 0 : checksum;
}
```

OUTPUT:

```
obscure@Obscures-MacBook-Air output % ./"task2"
The ISBN-13 number is: 9780306406157
The ISBN-13 number is: 9780140286571
The ISBN-13 number is: 9781602340619
Enter first 12 digits of ISBN: 000100100001
The ISBN-13 number is: 0001001000013
obscure@Obscures-MacBook-Air output %
```

Calculate the check digit for ISBN-10 or ISBN-13. The tool will automatically detect whether you're entering an ISBN-10 or ISBN-13.

Single ISBN Calculation | Multiple ISBN Calculation

ISBN Numbers

978030640615
978014028657
978160234061
000100100001

Calculate Check Digits

Results:

978030640615 → ISBN-13: 9780306406157 (Check Digit: 7)
978014028657 → ISBN-13: 9780140286571 (Check Digit: 1)
978160234061 → ISBN-13: 9781602340619 (Check Digit: 9)
000100100001 → ISBN-13: 0001001000013 (Check Digit: 3)

Do NOT generate ISBNs using a calculator. The check digit tool is for verification/correction only—unregistered ISBNs are invalid and may cause legal/distribution problems.