

Lab Questions

Part 1: Loops

1. The marla is a traditional unit of area that is commonly used in Pakistan where one marla contains 30.25 square yards. Write a program `marla2yards.cpp` that generates the following table for converting area in marla to square yards. You must use one loop and a single `cout` statement.

```
5 marla = 151.25 square yards
10 marla = 302.5 square yards
15 marla = 453.75 square yards
20 marla = 605 square yards
25 marla = 756.25 square yards
30 marla = 907.5 square yards
```

2. Write a program `factors.cpp` that print all prime factors of a `long long` number given by user as input.

Examples:

- $3,757,208 = 2 \times 2 \times 2 \times 7 \times 13 \times 13 \times 397$
 - $98 = 2 \times 7 \times 7$
 - $17 = 17$
 - $11,111,111,111,111,111 = 2,071,723 \times 5,363,222,357$
3. Write a program `all_digits_odd.cpp` that determines whether every digit of a given positive integer is odd. Print `true` if the number consists entirely of odd digits (1, 3, 5, 7, 9) and `false` if any of its digits are even (0, 2, 4, 6, 8). For example,
 - On input 135319, the output should be `true`.
 - On input 9145293, the output should be `false`.
 4. Write a program called `gcd.cpp` that accepts two integers as input and returns the *greatest common divisor* (GCD) of the two numbers. The GCD of two integers a and b is the largest integer that is a factor of both a and b .

One efficient way to compute the GCD of two numbers is to use Euclid's algorithm, which states the following:

- If b is 0, then the GCD of a and b is absolute value of a . That is, $\text{GCD}(a, 0) = |a|$.
- Otherwise, the GCD of a and b is the same as the GCD of b and $a \% b$. That is, $\text{GCD}(a, b) = \text{GCD}(b, a \% b)$.

5. Write a program **boxed.cpp** that take a **string** as input and print it inside a box. For example, if the input string is "Hello World", the output should be:

```
+-----+
| Hello World |
+-----+
```

Note:

- For a variable **s** of type **std::string**, **s.size()** returns the length of the string.
- To input a sentence using **cin** in variable **s** of type **std::string**, you can use **getline(cin, s)**.

Part 2: Arrays

6. Write a program **count_even.cpp** that read **int** values from input, put them in an array, and print the count of even integers in the array.

Hint: First ask user the number of values n , and then in a loop read n values. You may assume that $n \leq 50$;

7. Write a program **is_sorted.cpp** that given an array of real numbers prints **true** if the list is in sorted (non-decreasing) order and **false** otherwise. For example, if arrays store {16.1, 12.3, 22.2, 14.4} and {1.5, 4.3, 7.0, 19.5, 25.1, 46.2} respectively, your program should print **false** and **true** respectively. Assume the array has at least one element. A one-element array is considered to be sorted.

Note: You may use hard-coded array in this exercise. I.e, you do not need to take input from user.

8. Write a program **reverse_array.cpp** that reverses the order of values in a one-dimensional **string** array. For instance, the following array {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"} would be transformed to {"nine", "eight", "seven", "six", "five", "four", "three", "two", "one", "zero"}.

Do not create another array to hold the result. Use exchanges/swap to transform the array.

Note: You may use hard-coded array in this exercise.

9. *Find a duplicate*: Given an integer array of length n , with each value between 1 and n , write a program **duplicate.cpp** to determine whether there are any duplicate values in the given array. Print **true** if there are any duplicates and **false** otherwise.

Note: You do not need to take input from the user in this exercise.

10. Write a program `max_ones.cpp` that given an array of integers, find the maximum number of consecutive 1's present in the array.

Example:

- For the array `{1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1}`, the maximum number of consecutive 1's is 5.
- For the array `{1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0}`, the maximum number of consecutive 1's is 4.
- For the array `{0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1}`, the maximum number of consecutive 1's is 1.

Note: You may use hard-coded array in this exercise.

11. *Birthday problem*: Suppose that people enter an empty room until a pair of people share a birthday. On average, how many people will have to enter before there is a match? Write a program `birthday.cpp` to simulate one experiment. Write another program `birthdays.cpp` to repeat the experiment many times and estimate the average value. Assume birthdays to be uniform random integers between 0 and 364.

Hint: Use an array of `bool` values to mark the birthdays of people present in the room.