C++ Programming

STL Practice #2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Practice: 3 Stack max sum

- Given 3 stack of positive numbers. You may remove some items from the top of them. The target is the sum of all stacks is equal and maximum.
- Let's say the 3 stacks as following (remove from end)
  - A = {1, 2, 3, 4};
  - B = {2, 2, 2, 4, 0};
  - C = {0, 3, 3, 5};
  - Initial sums are: 10, 10, 11 ⇒ Not equal
  - If we remove top element from A (4), top 2 from B (0, 4) and top from C (5) ⇒ 6, 6, 6 = max
- What if we have initial C as {12, 3, 3, 5}. Maximum possible sum is 0 :(
- Implement: int max_3stack_sum(vector<int> &a, vector<int> &b, vector<int> &c)
  - A, B, C acts like a stack. E.g. a values = {1, 2, 3, 4} and b = {2, 2, 2, 4, 0}

Practice: 3 Stack max sum

```
int sum_vec(vector<int> &a) {
    int sum = 0;
    for (auto x : a)
        sum += x;
    return sum;
}

int max_3stack_sum(vector<int> &a, vector<int> &b, vector<int> &c) {
    int sa = sum_vec(a), sb = sum_vec(b), sc = sum_vec(c);

    while(!a.empty() && !b.empty() && !c.empty())
    {
        if(sa == sb && sb == sc)
            return sa;  // as all +ve, this is tha max possible

        // Remove top element from max-sum stack
        if (sa >= sb && sa >= sc)
            sa -= a.back(), a.pop_back();
        else if (sb >= sa && sb >= sc)
            sb -= b.back(), b.pop_back();
        else
            sc -= c.back(), c.pop_back();
    }
    return 0;
}
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."