**Задача 3.** Кристальные бабочки

Недалеко от Мондштадта есть чудесное место, наполненное ароматом трав и теплом вечного лета – Одуванчиковое море. Конечно, поэтичное название «море», данное этому месту бардами, немного преувеличено. На самом же деле это живописная поляна, сплошь покрытая золотистыми цветами. Одуванчиковое море очень приглянулось редкому виду насекомых – кристальным бабочкам.

Поле представляет собой полосу из N условных клеток. На каждой клетке поля живёт определенное количество бабочек *x.* Бабочки не сидят на месте – они то прилетают, то улетают сразу по *y* штук с некоторых отрезков поля вида [*l, r*], где оба конца отрезка включены. Всего таких перелётов – M штук.

Требуется написать программу, которая определит, сколько останется бабочек на каждой из клеток после всех изменений. Считается, что за один запрос с участка от *l* до *r* улетает (или прилетает) одинаковое количество бабочек *y*.

**Входные данные.**

На вход подаются целые числа: на первой строке вводится число клеток на поле N, далее с новой строки вводится N значений (, соответствующих количеству бабочек наклетке. На следующей строке вводится число запросов M, а на последующих M строчках вводятся через пробел два числа: *l* – левая граница диапазона, *r* – правая граница (1 ≤ *l* ≤ *r*, a 1 ≤ *r* ≤ N) и *y*  – численное изменение бабочек на каждой клетке (≤ y ≤ . В конце ввода Enter не ставится.

**Выходные данные.**

Количество бабочек на каждой клетке поля после выполнения всех запросов, через пробел. Если бабочек совсем не осталось, вывести для этой клетки «0».

**Пример 1**

*На входе:*  
3  
34 42 88  
1  
0 2 2  
*На выходе:*  
36 44 90

**Решение**.

Для ответа на запросы о добавлении или вычитании числа на определенном диапазоне удобно использовать массив разниц, назовём его *diff* и после получения основного массива *a* пересчитаем разницу между имеющимися значениями по формуле (5):

Далее воспользуемся свойством массива разниц и изменим в нём пограничные значения: левую границу изменим на величину +x, а правую на обратную величину –х. Таким образом выполним все M запросов, а после по массиву *diff* восстановим интересующий нас изменённый массив, использовав для вычисления *i-*го элемента массива *a* равенство (6), полученное из формулы (5):

Алгоритм решения данной задачи представлен также в виде блок-схемы (Приложение Б). Ввод и вывод данных в программу осуществляется с помощью текстовых файлов input и output.

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

ifstream input("input.txt");

ofstream output("output.txt");

if (!input.is\_open()) { cout << "ERROR: FILE IS NOT OPEN"; }

else

{

int n, k = 1, l, r, x, m;

input >> n;

long long int\* a = new long long int[n];

for (int i = 0; i < n; i++)

{

input >> a[i];

}

long long int\* diff = new long long int[n + 1];

diff[0] = a[0];

diff[n] = 0;

for (int i = 1; i < n; i++)

{

diff[i] = a[k] - a[k - 1];

k++;

}

input >> m;

for (int j = 0; j < m; j++)

{

input >> l >> r >> x;

diff[l - 1] += x;

diff[r] -= x;

for (int i = 0; i < n; i++)

{

if (i == 0) a[i] = diff[i];

else a[i] = diff[i] + a[i - 1];

}

}

for (int i = 0; i < n; i++)

{

if (a[i] >= 0) output << a[i] << " ";

else output << 0 << " ";

}

}

input.close();

output.close();

return 0;

}