Rule: lazy evaluation

|  |  |
| --- | --- |
| l |  |
| string s1 = "lazy"  string s2 = s1 | string s1 = "lazy"  string s2 = "lazy" |

#include <iostream>

#include <algorithm>

#include <array>

using namespace std;

int main() {

int a[] = {10,1,2,3,4,5,6,7,8,9};

std::array<int, 2000> v;

static const size\_t v\_size = 2000;

int ve[v\_size];

vector <int> vv[100];

//std::for\_each;

sort(begin(a),end(a));

//best way to make string

const string authorName("Scott Meyers");

sort(a, a + sizeof a / sizeof a[0]);

std::cout << a;

}

Effective programming some cute points to talk over code…

<https://www.geeksforgeeks.org/find-subarray-with-given-sum/>

<https://ide.geeksforgeeks.org/lYWD7Dlubm>

<https://www.geeksforgeeks.org/print-paths-root-specified-sum-binary-tree/>

<https://www.geeksforgeeks.org/print-k-sum-paths-binary-tree/>

<https://www.geeksforgeeks.org/given-a-binary-tree-print-all-root-to-leaf-paths/>