**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ**

**“ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №8**

Выполнила студент 2 курса, группа 21-ИТ-1 Катушёнок И.В.

Проверила Виноградова А.Д

Полоцк, 2022 г.

**Задание 1:** Дано целое число 1≤*n*≤103 и массив  *A*[1…*n*] натуральных чисел, не превосходящих 2⋅109. Выведите максимальное  1≤*k*≤*n*, для которого найдётся подпоследовательность  1≤*i*1​<*i*2​<…<*ik*​≤*n* длины *k*, в которой каждый элемент делится на предыдущий (формально: для  всех 1≤*j*<*k*,  *A*[*ij*​]∣*A*[*ij*+1​]).

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main()

{

int n = 0;

cin >> n;

vector<int> arr;

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

{

int val = 0; cin >> val;

arr.push\_back(val);

}

auto size = arr.size();

vector<int> path\_len(size);

for (int i = 0; i < size; ++i) {

path\_len[i] = 1;

for (int j = 0; j < i; ++j) {

if ((arr[i] % arr[j] == 0) && (path\_len[j] + 1 > path\_len[i])) {

path\_len[i] = path\_len[j] + 1;

}

}

}

int max = 0;

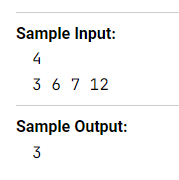
for (int i = 0; i < path\_len.size(); i++)

if (max < path\_len[i]) max = path\_len[i];

cout << max << endl;

return 0;

}



**Задание 2:** Вычислите расстояние редактирования двух данных непустых строк длины не более 102, содержащих строчные буквы латинского алфавита.

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

int main()

{

string str1, str2;

cin >> str1; cin >> str2;

int n = str1.size();

int m = str2.size();

vector<vector <int>> VE;

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

VE.push\_back(vector<int>(m + 1));

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

VE[i][0] = i;

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

VE[0][i] = i;

int c = 0;

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

{

for (int j = 1; j <= m; j++)

{

if (str1[i - 1] == str2[j - 1]) c = 0;

else c = 1;

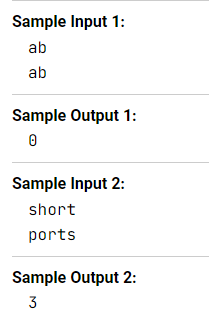
VE[i][j] = min({ VE[i - 1][j] + 1, VE[i][j - 1] + 1, VE[i - 1][j - 1] + c });

}

}

cout << VE[n][m] << endl;

}



**Задание 3:** Первая строка входа содержит целые числа 1≤*W*≤104 и 1≤*n*≤300 — вместимость рюкзака и число золотых слитков. Следующая строка содержит *n* целых чисел 0≤*w*1​,…,*wn*​≤105, задающих веса слитков. Найдите максимальный вес золота, который можно унести в рюкзаке.

#include <vector>

#include <iomanip>

#include <iostream>

using namespace std;

int main()

{

int W, n;

cin >> W >> n;

vector<int> R(n + 1);

for (int i = 1; i <= n; i++) cin >> R[i];

vector<vector<int>> D(W + 1, vector<int>(n + 1, 0));

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

for (int w = 1; w <= W; w++) {

D[w][i] = D[w][i - 1];

if (R[i] <= w) {

D[w][i] = max(D[w][i], D[w - R[i]][i - 1] + R[i]);

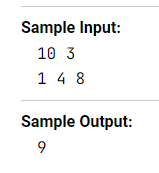
}

}

}

cout << D[W][n];

}



**Задание 4:** Даны число 1≤*n*≤102 ступенек лестницы и целые числа −104≤*a*1​,…,*an*​n≤104, которыми помечены ступеньки. Найдите максимальную сумму, которую можно получить, идя по лестнице снизу вверх (от нулевой до *n*-й ступеньки), каждый раз поднимаясь на одну или две ступеньки.

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

int main()

{

int steps; cin >> steps;

vector<int> v;

v.push\_back(0);

for (int i = 0; i < steps; i++) {

int a = 0; cin >> a;

v.push\_back(a);

}

vector<int> stairwaytohaven;

stairwaytohaven.push\_back(0);

stairwaytohaven.push\_back(v[1]);

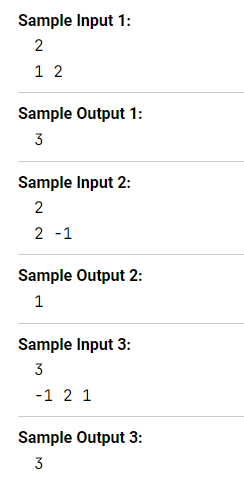
for (int i = 2; i <= steps; i++) {

stairwaytohaven.push\_back(max({ stairwaytohaven[i - 1] + v[i], stairwaytohaven[i - 2] + v[i] }));

}

cout << stairwaytohaven[steps] << endl;

}



**Задание 5:** У вас есть примитивный калькулятор, который умеет выполнять всего три операции с текущим числом *x*: заменить *x* на 2*x*, 3*x* или *x*+1. По данному целому числу 1≤*n*≤105 определите минимальное число операций *k*, необходимое, чтобы получить *n* из 1. Выведите *k* и последовательность промежуточных чисел.

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

#include <sstream>

using namespace std;

int main()

{

int N = 0;

cin >> N;

vector<size\_t> S;

vector<vector<size\_t>> path(N + 1, vector<size\_t>());

S.push\_back(0);

S.push\_back(0);

path[1].push\_back(1);

for (int i = 2; i <= N; i++) {

size\_t x\_div\_3 = numeric\_limits<int>::max();

if ((i % 3) == 0) x\_div\_3 = S[i / 3] + 1;

size\_t x\_div\_2 = numeric\_limits<int>::max();

if ((i % 2) == 0) x\_div\_2 = S[i / 2] + 1;

size\_t min\_val = min({ x\_div\_3, x\_div\_2, S[i - 1] + 1 });

S.push\_back(min\_val);

if (min\_val == x\_div\_3) { path[i] = path[i / 3]; path[i].push\_back(i); }

if (min\_val == x\_div\_2) { path[i] = path[i / 2]; path[i].push\_back(i); }

if (min\_val == S[i - 1] + 1) { path[i] = path[i - 1]; path[i].push\_back(i); }

}

cout << S[N] << endl;

for (auto elem : path[N]) cout << elem << " ";

cout << endl;

return 0;

}

