

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**ОТЧЕТ по практической работе**

**по дисциплине “Методы программирования”**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автор |  | | |  | Карпов Д.И. | | | |
|  | (подпись, дата) | | |  |  | | | |
| Обозначение | 10.05.01.550000.000 О | | Группа | | | | ВКБ31 |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | | | | |
| Профиль | Компьютерная безопасность | | | | | | |
| Преподаватель |  | | |  | | Савельев В.А. | |
|  | (подпись, дата) | | |  | |  | |

г. Ростов-на-Дону

2021 год

**Лабораторная работа №3**

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.

Входные данные

В первой строке входных данных содержится число NN – длина первой последовательности (1 ≤ NN ≤ 1000). Во второй строке заданы члены первой последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.

В третьей строке записано число MM – длина второй последовательности (1 ≤ MM ≤ 1000). В четвертой строке задаются члены второй последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.

Выходные данные

Требуется вывести наибольшую общую подпоследовательность данных последовательностей, через пробел.

**Примеры**

|  |  |
| --- | --- |
| входные данные  3  1 2 3  3  2 3 1 | выходные данные  2 3 |

#include <stdio.h>

#include <string.h>

#define SIZE 1002

int x[SIZE], y[SIZE], dp[SIZE][SIZE];

int n, m, i, j, res;

int max(int i, int j)

{

return (i > j) ? i : j;

}

int lcs(int \*x, int \*y, int m, int n)

{

if (m == 0 || n == 0)

return 0;

if (dp[m][n] != -1) return dp[m][n];

if (x[m] == y[n])

return dp[m][n] = 1 + lcs(x, y, m - 1, n - 1);

else

return dp[m][n] = max(lcs(x, y, m, n - 1), lcs(x, y, m - 1, n));

}

int main(void)

{

scanf("%d", &n);

for (i = 1; i <= n; i++) scanf("%d", &x[i]);

scanf("%d", &m);

for (i = 1; i <= m; i++) scanf("%d", &y[i]);

memset(dp, -1, sizeof(dp));

res = lcs(x, y, n, m);

printf("%d\n", res);

return 0;

}