МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

по лабораторной работе №7

Выполнил:

студент группы ПО-9

Харитонович Захар Сергеевич

Проверила:

Гирель Т. Н.

Брест 2021

**Указатели и динамические массивы**

Вариант 9

***Задание А***

**Текст задания**

Задан массив А из N чисел. В массиве А рядом могут стоять одинаковые числа. Образовать массив В, подсчитав количество повторяющихся чисел в массиве А, стоящих рядом. Для полученного массива реализовать удаление всех элементов с заданным значением (значение запрашивается с клавиатуры).

**Описание входных и выходных данных**

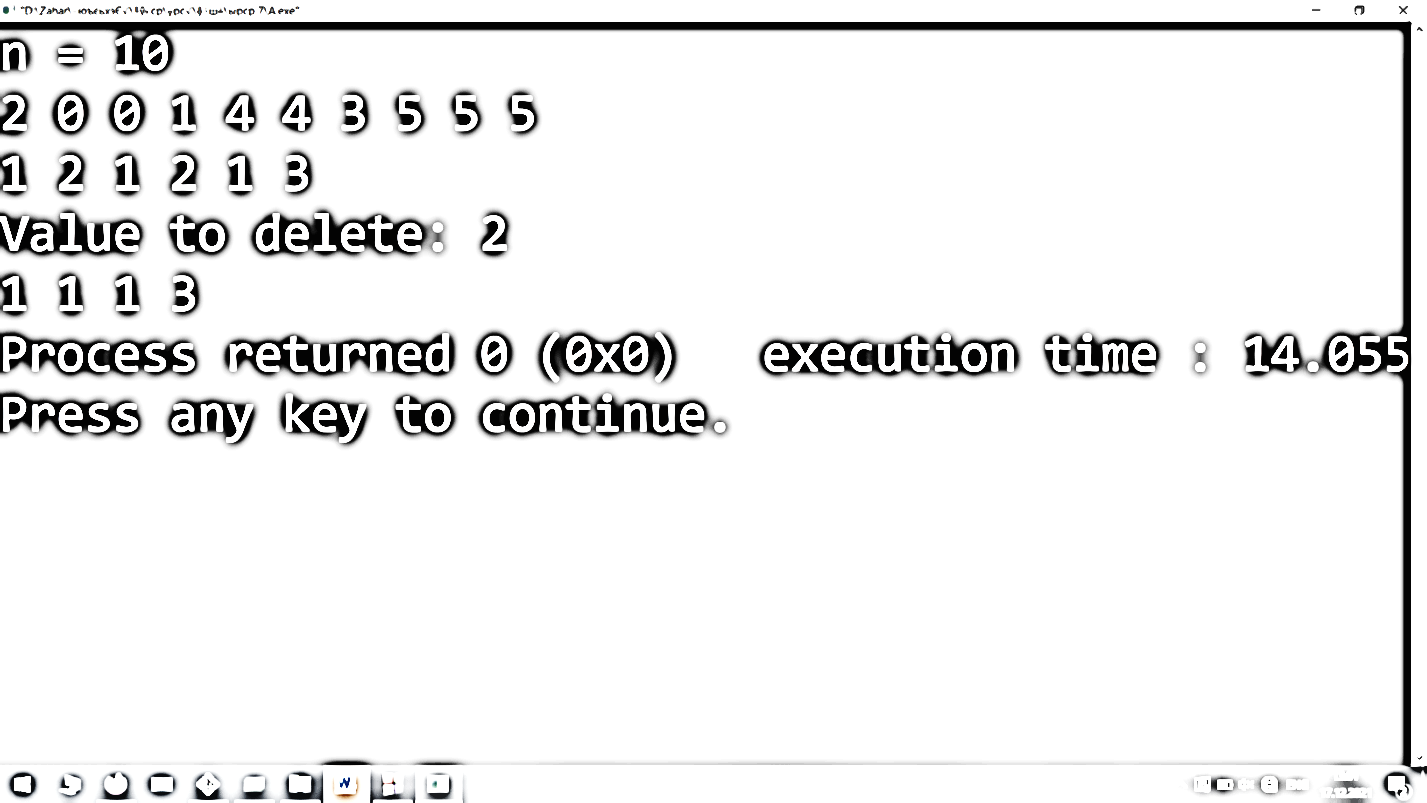
Входные данные – целое число n – размер массива, целочисленный массив a[n], целое число – значение, подлежащее удалению из итогового массива.

Выходные данные – получившийся целочисленный массив b, массив b после удаления заданного элемента.

**Описание способа решения задачи**

Начинаем перебирать массив со второго элемента, сравниваем с предыдущим. Если элементы одинаковы, то продолжаем подсчёт одинаковых идущих подряд элементов. Если элементы отличаются, то делаем запись в массив b, сбрасываем счётчик и начинаем подсчёт для следующей группы одинаковых элементов. Перебираем массив и убираем заданное значение.

**Текст программы Результат работы**

#include <stdio.h>

#include <stdlib.h>

int main() {

int n, k, i, reps = 1, j = 0;

int \*pa;

int \*pb;

printf("n = "); scanf("%d", &n);

pa = (int \*)malloc(n\*sizeof(int));

pb = (int \*)malloc(n\*sizeof(int));

for(i = 0; i < n; i++) scanf("%d", (pa+i));

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

if(\*(pa+i) == \*(pa+(i-1))) {

reps++;

} else {

\*(pb+j) = reps; j++; reps = 1;

}

}

for(i = 0; i < j; i++) printf("%d ", \*(pb+i));

printf("\nValue to delete: ");

scanf("%d", &n);

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

if(\*(pb+i) == n) {

j--;

for(k = i; k < j; k++) {

\*(pb+k) = \*(pb+k+1);

}

}

}

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

printf("%d ", \*(pb+i));

}

free(pa); free(pb);

return 0;

}

***Задание Б***

**Текст задания**

Даны два числа k1 и k2 и матрица A[1..n,1..m]. Поменять местами столбцы матрицы с номерами k1 и k2, а потом поменять соответствующие строки. Для полученного массива реализовать добавление строки элементов в заданной позиции (значение

позиции и элементов строки вводятся с клавиатуры).

**Описание входных и выходных данных**

Входные данные – целые числа k1 и k2 – номера столбцов и строк, которые нужно поменять местами, целые числа n и m – размеры двумерного массива, исходный целочисленный массив a[n][m], целое число – номер строки для вставки, целочисленный массив – строка для вставки.

Выходные данные – целочисленный видоизменённый массив a[n][m].

**Описание способа решения задачи**

Сначала идём вниз по матрице и меняем местами соответствующие элементы столбцов k1 и k2. Потом идём по матрице слева направо и меняем местами соответствующие элементы строк k1 и k2.

**Текст программы Результат работы**

#include <stdio.h>

#include <stdlib.h>

int main() {

int el, sw = 0, n, m, k1, k2, i = 0, j = 0, \*\*a;

printf("k1 = "); scanf("%d", &k1); printf("k2 = "); scanf("%d", &k2);

printf("n = "); scanf("%d", &n); printf("m = "); scanf("%d", &m);

k1--; k2--;

a = (int \*\*)malloc((n+1)\*sizeof(int \*));

for(i = 0; i < n + 1; i++) a[i] = (int \*)malloc((m)\*sizeof(int));

for(i = 0; i < n; i++) for(j = 0; j < m; j++) scanf("%d", &a[i][j]);

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

sw = a[i][k1]; a[i][k1] = a[i][k2]; a[i][k2] = sw; sw = 0;

}

printf("\n");

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

for(j = 0; j < m; j++) printf("%d ", a[i][j]);

printf("\n");

}

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

sw = a[k1][i]; a[k1][i] = a[k2][i]; a[k2][i] = sw; sw = 0;

}

printf("\n");

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

for(j = 0; j < m; j++) printf("%d ", a[i][j]);

printf("\n");

}

printf("Line number: "); scanf("%d", &k1);

n++; k1--;

printf("Line: ");

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

scanf("%d", &el);

for(i = n - 1; i > k1; i--) a[i][j] = a[i-1][j];

a[k1][j] = el;

}

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

for(j = 0; j < m; j++) printf("%d ", a[i][j]);

printf("\n");

}

for(i = 0; i < n; i++) free(a[i]);

free(a);

return 0;

}