

1. **Цель работы**

Получение навыков работы с многомерными массивами на примере двумерных массивов.

1. **Постановка задачи**

Вариант 1: В двумерном целочисленном массиве размера N на K обнулить все строки, находящиеся ниже минимального элемента.

1. **Формализация задачи**

Массив состоит из целых чисел, его размер N на K. Если в массиве несколько минимальных элементов, программа удаляет строки после первого встреченного. Если все элементы массива одинаковы, программа выводит соответствующее сообщение. Изначально минимум задаётся как первый элемент массива. Далее каждый элемент массива сравнивается с текущими минимумом и, если он меньше минимума, то запоминается позиция этого числа, а оно само записывается как новый минимум. После проверки всего массива, к номеру строки минимального элемента прибавляется единица и все строки, идущие после строки, в которой находится минимум, обнуляются, то есть обнуляется каждый элемент этих строк. Результирующий массив выводится пользователю.

1. **Набор тестовых примеров**

|  |  |
| --- | --- |
| Входные значения | Вывод программы |
| Массив 3х3.  Элементы:  1 2 3  4 -5 6  7 8 9 | Your new array is:  1 2 3  4 -5 6  0 0 0 |
| Массив 2х4.  Элементы:  1 2 3 4  5 6 -7 8 | Your new array is:  1 2 3 4  5 6 -7 8 |
| Массив 3х3.  Элементы:  2 0 5  6 4 34  456 456 90 | Your new array is:  2 0 5  0 0 0  0 0 0 |
| Массив 3х3.  Элементы:  1 1 1  1 1 1  1 1 1 | All the elements in the array are equal. |

1. **Схема алгоритма**

Схема алгоритма программы представлена на рисунке 1.

Начало

N = 3; K = 3;

endFlag = 1;

Arr[N][K] = { 0 };

min = Arr[0][0];

minNumI = 0;

equal = 0;

A

A

Да

Ввод Arr[i][j]

j < K

Нет

Да

i < N

Нет

C

Да

Нет

equal = 0;

Нет

Да

Arr[i][j] = min

equal = 1;

j < K

Да

Нет

i < N

B

B

Нет

Да

equal = 0

C

“All the elements in the array are equal.”

Arr[i][j] < min

Нет

Да

minNumI = i;

min = Arr[i][j];

Да

j < K

Нет

Да

Нет

i < N

i = minNumI+1;

D

F

E

F

D

D

E

Arr[i][j] = 0;

Вывод Arr[i][j]

j < K

Да

Нет

i < N

Да

Нет

endFlag = 1;

memset(&Arr[0][0], 0, sizeof(Arr));

min = 2147483647;

minNumI = -1;

1 - Count again.

2 - Exit.

2

1

Конец

*Рисунок 1 – Схема алгоритма программы.*

1. **Листинг программы**

#if defined(\_WIN32) || defined(\_\_WIN32\_\_) || defined(WIN32)

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#pragma warning(disable:6031)

#endif

#define N 3

#define K 3

#include <stdio.h>

int main() {

int endFlag = 1; //a flag that stops the application

int Arr[N][K] = { 0 }; //array of integer numbers

int minNumI = 0; //number of the string with the minimum element

int equal = 0; //flag of equality of the array

//a global cycle that prevents the program from endting without the user's desire

do {

//a cycle for reading characters from a stream associated with the keyboard

printf("Please type down your array:\n");

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

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

scanf("%d", &Arr[i][j]);

}

}

int min = Arr[0][0]; //first element is chosen as minimum

//checking if all elements are equal

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

for (int j = 1; j < K; j++) {

if (Arr[i][j] == min) {

equal = 1; //sets equality flag to 1 if every element so far was equal to each other

}

else {

equal = 0; //sets equality flag to 0 and exits from the loop

break;

}

}

if (equal == 0) { //exits from the loop if equality flag if 0

break;

}

}

//if not all the elements in the array are equal

if (equal == 0) {

//finding the minimum element ant its position

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

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

if (Arr[i][j] < min) {

minNumI = i;

min = Arr[i][j];

}

}

}

//setting every line after the minimum element to zero

for (int i = minNumI + 1; i < N; i++) {

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

Arr[i][j] = 0;

}

}

//output of the result

printf("Your new array is:\n");

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

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

printf("%d ", Arr[i][j]);

}

printf("\n");

}

}

else { //if all the elements in the array are equal

printf("\nAll the elements in the array are equal.\n");

}

//resetting the variables

endFlag = 1;

memset(&Arr[0][0], 0, sizeof(Arr)); //filling the array with the specified characters

minNumI = 0;

//suggesting to user to run program again or exit.

printf("\nPlease choose from the following:\n\t1 - Count again.\n\t2 - Exit.\n");

scanf("%d", &endFlag);

switch (endFlag) {

case 1: endFlag = 1; break;

case 2: endFlag = 0; break;

default: printf("Incorrect answer. The program will be closed."); endFlag = 0; break;

}

} while (endFlag); //if flag is 0, then exiting out of cycle

return 0;

}

1. **Выводы**

Массив данных – это пронумерованный набор данных одного типа.

Элементы массива в языке Си нумеруются начиная с индекса 0.

В языке Си не существует проверки выхода значения индекса за границы массива.

Оператор for целесообразно использовать для полного, а while и do…while для частичного перебора элементов массива.

В языке Си можно описать массив переменного размера при помощи указателей.