МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» им.В.И.УЛЬЯНОВА (ЛЕНИНА)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчет по лабораторной работе № 4

по дисциплине «Программирование»

Тема: «Указатели в языке Си»

|  |  |
| --- | --- |
| Выполнила: Студент гр. 9308 | Семенов А.И. |
|  |  |
| Приняла: К. т. н., доцент | Сискович Т.И. |

Содержание

Введение3

1. [Задание 3](#__RefHeading___Toc320_2605875897)
2. Уточнение задания3
3. [Описание переменных 4](#__RefHeading___Toc326_2605875897)
4. Контрольные примеры [5](#__RefHeading___Toc488_2605875897)
5. Краткое описание алгоритма [6](#__RefHeading___Toc488_2605875897)
6. [Текст программы](#__RefHeading___Toc488_2605875897) 7
7. [Результат работы программы 10](#__RefHeading___Toc330_2605875897)

Заключение 11

Введение

Целью выполнения лабораторной работы №4 «Указатели» является приобретение практических навыков в использовании указателей в программировании

## 1. Задание

Ввести построчно элементы двумерного массива чисел заданного размера. Если в процессе ввода чисел в строке произойдет превышение произведения введенных в текущей строке чисел заданного в исходных данных числа, то остальные элементы в этой строке должны быть автоматически сформированы равными минимальному значению из элементов текущей строки. Вывести исходный массив.

Из столбцов исходного массива, в которых сумма элементов неотрицательная, сформировать столбцы результирующего массива. Вывести результирующий массив.

## 2. Уточнение задания

Перед вводом элементов последовательности необходимо ввести число – предел произведения чисел в строке массива. При достижении данного значения, оставшиеся элементы в строке автоматически становятся равны минимальному значению в этой же строке. Массив задается пользователем после ввода предела произведения. Необходимо сначала вывести исходный массив. Далее отбираются столбцы, удовлетворяющие условию, которые составляют результирующий массив, он и является ответом на задачу.

## 3. Описание данных

Описание переменных программы содержится в Таблице 1, приведенной ниже. Описание констант программы содержится в Таблице 2.

Таблица 1. Описание переменных

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| a | \*\*int | Указатель для двумерного массива |
| b | \*\*int | Указатель для результирующего массива |
| mlimit | int | Предел произведение чисел в строке массива |
| mult | int | Произведение чисел в строке массива |
| min | int | Минимальное значение в строке массива |
| sum | int | Сумма чисел столбца массива |
| repeat | int | Флаг повтора программы |
| m | int | Количество столбцов исходного массива |
| n | int | Количество строк массива |
| k | int | Количество столбцов результирующего массива |
| c | int | Счетчик для создания динамических массивов |
| i | int | Счетчик строк массива |
| j | int | Счетчик столбцов массива |

## 4. Контрольные примеры

Контрольные примеры программы содержатся в Таблице 2, приведенной ниже.

Таблица 2. Контрольные примеры

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Исходные данные** | | | | **Результат** |
| **mlimit** | **m n** | **Вводимый массив** | **Исходный массив** |
| **1** | **5** | **3 3** | **5 5 5**  **5 2 2**  **-5 -5 -5** | **5 5 5**  **5 2 2**  **-5 -5 -5** | **5 5 5**  **5 2 2**  **-5 -5 -5** |
| **2** | **3** | **3 2** | **1 4 5**  **-6 2 1** | **1 4 1**  **-6 2 1** | **4 1**  **2 1** |
| **3** | **-1** | **1 5** | **2**  **-1**  **1**  **6**  **-20** | **2**  **-1**  **1**  **6**  **-20** | **Ни один столбец массива не удовлетворяет условию** |
| **4** | **-4** | **2 4** | **1 -6**  **-6 1**  **1 0**  **-4 0** | **1 1**  **-6 1**  **1 1**  **-4 0** | **1**  **1**  **1**  **0** |
| **5** | **27** | **3 4** | **9 4 -1**  **6 6 1**  **1 1 -10**  **-10 -3 1** | **9 4 4**  **6 6 6**  **1 1 -10**  **-10 -3 -10** | **9 4**  **6 6**  **1 1**  **-10 -3** |
| **6** | **2** | **2 3** | **-2 -3**  **71 0**  **0 30** | **-2 -3**  **71 71**  **0 30** | **-2 -3**  **71 71**  **0 30** |
| **7** | **7** | **4 2** | **4 1 7 2**  **-7 -1 1 1** | **4 1 7 1**  **-7 -1 1 1** | **1 7 1**  **-1 1 1** |
| **8** | **-8** | **6 3** | **4 8 7 1 4 7**  **-8 1 7 5 1 1**  **-1 -2 -3 2 3 1** | **4 4 4 4 4 4**  **-8 1 7 5 1 1**  **-1 -1 -1 -1 -1 -1** | **4 4 4 4 4**  **1 7 5 1 1**  **-1 -1 -1 -1 -1** |
| **9** | **1** | **7 2** | **-1 7 5 -2 4 9 7**  **2 2 2 2 2 2 2** | **-1 7 5 -2 -2 -2 -2**  **2 2 2 2 2 2 2** | **-1 7 5 -2 -2 -2 -2**  **2 2 2 2 2 2 2** |
| **10** | **34** | **2 5** | **1 75**  **54 1**  **34 2**  **-300 1**  **20 20** | **1 75**  **54 54**  **34 2**  **-300 1**  **20 20** | **75**  **54**  **2**  **1**  **20** |

## 5. Краткое описание алгоритма

Шаг 1. На вход подается число – предел произведения чисел в строке массива.

Шаг 2. Вводится размер массива.

Шаг 3. Построчно вводится двумерный массив.

Шаг 4. Массив видоизменяется при выполнении следующего условия – если произведение чисел в строке превышает заданный предел, то все оставшиеся элементы строки автоматически принимают минимальное значение, уже встречавшееся в строке.

Шаг 5. Выводится исходный массив.

Шаг 6. Проверяется каждый столбец массива: если сумма элементов неотрицательная, то заносим данный столбец в столбец массива результирующего.

Шаг 7. Если найдется хотя бы один столбец, удовлетворяющий условию, то выводится результирующий массив. В противном случае выводится сообщение «Ни один столбец массива не удовлетворяет условию».

Шаг 8. Пользователю предлагается повторить работу программы.

## 6. Текст программы

Текст программы представлен ниже.

#include <stdio.h>

#include <locale.h>

#include <malloc.h>

int main()

{

setlocale(LC\_ALL, "russian");

int \*\*a, //Исходный массив

\*\*b, //Результирующий массив

mlimit, //Предел произведения чисел в строке массива

mult, //Произведение чисел в строке массива

min, //Минимальное число в строке массива

sum, //Сумма чисел столбца массива

repeat, //Флаг повторения работы программы

m, //Количество столбцов исходного массива

n, //Количество строк массива

k, //Количество столбцов результирующего массива

c = 0, //Счетчик для создания динамических массивов

i = 0, //Счетчик строк в цикле

j = 0; //Счетчик стобцов в цикле

do

{

sum = 0; k = 0;

puts("Введите число - предел произведения чисел в строке массива");

scanf("%d", &mlimit);

printf("Введите размера массива. Сначала количество столбцов, далее - строк.\n");

do

{

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

if(m < 1 || n < 1) printf("Размер массива должен быть минимум 1х1!\n");

} while(m < 1 || n < 1);

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

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

if(a != NULL && b != NULL)

{

for(i = 0, c = 0; i < n; i++, c++)

{

a[i] = (int\*)malloc(m\*sizeof(int));

b[i] = (int\*)malloc(m\*sizeof(int));

if(a == NULL || b == NULL)

{

i = n;

}

}

if(c != n)

{

for(i = 0; i < c; i++)

{

if(a[i] != NULL){free(a[i]);}

if(b[i] != NULL){free(b[i]);}

}

free(a);

free(b);

printf("Ошибка при создании массивов, повторите с меньшим количеством строк/столбцов. \n");

}

else

{

printf("Построчно введите элементы массива размером столбцы: %d строки: %d\n", m, n);

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

{

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

{

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

if(j == 0)

{

min = a[i][j];

mult = a[i][j];

}

if(mult > mlimit) a[i][j] = min;

if(a[i][j] < min) min = a[i][j];

if(j && mult <= mlimit) mult \*= a[i][j];

}

}

puts("Исходный массив:");

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

{

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

puts("");

}

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

{

sum = 0;

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

if(sum >= 0)

{

for(i = 0; i < n; i++) b[i][k] = a[i][j];

k++;

}

}

if(k > 0)

{

puts("Результирующий массив:");

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

{

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

puts(" ");

}

}

else puts("Ни один столбец массива не удовлетворяет условию");

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

{

free(a[i]);

free(b[i]);

}

free(a);

free(b);

a = NULL;

b = NULL;

}

}

puts("Введите любое целое число, не равное нулю, чтобы повторить программу");

scanf("%d", &repeat);

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

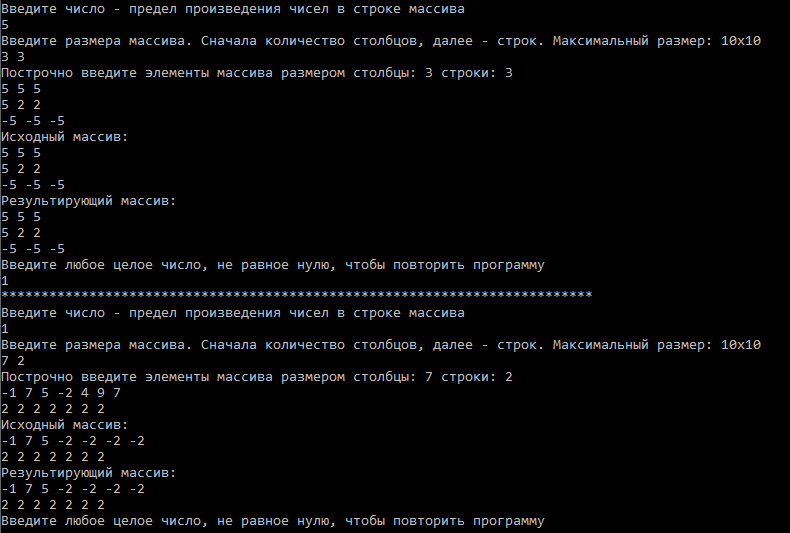
} while(repeat);

return 0;

}

## 7. Результат работы программы

В результате работы программы были получены результаты, которые приведены в таблице контрольный примеров (таблица 2). Пример выполнения программы приведен на рисунке 1.



*Рисунок 1. Пример выполнения программы*

Заключение

При выполнении лабораторной работы получены практические навыки в написании программы на языке Си с использованием указателей, а также получены результаты, совпадающие с контрольными примерами в таблице 2. Ошибок не обнаружено.