УО «Брестский государственный технический университет»

Кафедра интеллектуальных информационных технологий

Лабораторная работа №7

По дисциплине: "Основы алгоритмизации и программирования"

Тема: "Обработка динамических массивов"

Вариант 6

Выполнил: Комиссаров А.Е,

Малинин Е.А.

Брест 2020 ПО-7, 1 курс

Проверил:

Цель работы

Приобретение практических навыков в составлении программ по обработке динамических массивов.

Ход работы

**Задание** 1) Задан массив А из N чисел. Удалить из этого массива все нули, т.е. если массив А был (1,0,2,0,4), то получим новый массив Б (1,2,4). Также реализовать функцию удаления элемента из массива с заданным значением.

**Решение:**



#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <malloc.h>

void main(){

int n, i, b, x;

int\* a;

printf("\n razmer massiva = ");

scanf("%d", &n);

a = (int\*)calloc(n, sizeof(int));

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

{ printf("\n a[%d]= ", i+1);

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

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

{

if (a[i]==0){

for (b=i; b<n; b++) { a[b]=a[b+1]; }

i=i-1;

n=n-1;

}

}

printf("\n massiv: ");

for (i=0; i<n; i++){printf("%d; ",a[i]);}

printf("\nvvedite znachenie dlya udalenija:");

scanf("%d", &x);

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

{

if (a[i]==x){

for (b=i; b<n; b++) { a[b]=a[b+1]; }

i=i-1;

n=n-1;

}

}

printf("\n massiv: ");

for (i=0; i<n; i++){printf("%d; ",a[i]);}

printf("\n------- vvedite luboe chislo :");

scanf("%d", &n);

}

**Вывод программы:**



**Задание** 2) Элемент двухмерного массива называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером NxN, найти максимум среди всех локальных минимумов.

**Решение**



#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <malloc.h>

int main(){

int \*a;

int i, j, n, m, dop, sum;

printf("\nvvedite kol-vo strok: ");

scanf("%d", &n);

printf("\nvvedite kol-vo stolbcov: ");

scanf("%d", &m);

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

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

{

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

{

printf("a[%d + %d = ", i, j);

scanf("%d", (a + i\*m + j));

}

}

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

{

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

{

printf("%5d ", \*(a + i\*m + j));

}

printf("\n");

}

printf("\nvvedite kol-vo dop. strok: ");

scanf("%d", &dop)

sum = (n\*m)+(dop\*m);

a = (int\*)realloc(sum \*sizeof(int));

// Вывод элементов массива с доп. строкой(ами)

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

{

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

{

printf("%5d ", \*(a + i\*m + j));

}

printf("\n");

}

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

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

if (i == 1 && j == 1){if( \*(a + i\*m + j) < (a + i\*m + j+1) && \*(a + i\*m + j) < (a + i\*m+1 + j) && \*(a + i\*m + j) < (a + i\*m+1 + j+1 ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (i == 1 && j != 1 && j != n){if ( \*(a + i\*m + j) < (a + i\*m + j-1) && \*(a + i\*m + j) < (a + i\*m + j+1) && \*(a + i\*m + j) < (a + i\*m+1 + j-1) && \*(a + i\*m + j) < (a + i\*m+1 + j) && \*(a + i\*m + j) < (a + i\*m+1 + j+1 ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (i == 1 && j == n){if ( \*(a + i\*m + j) < (a + i\*m + j-1 ) && \*(a + i\*m + j) < (a + i\*m+1 + j-1 ) && \*(a + i\*m + j) < (a + i\*m+1 + j ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (j == 1 && i != 1){if ( \*(a + i\*m + j) < (a + i\*m-1 + j) && \*(a + i\*m + j) < (a + i\*m-1 + j+1) && \*(a + i\*m + j) < (a + i\*m + j+1 ) && \*(a + i\*m + j) < (a + i\*m+1 + j+1) && \*(a + i\*m + j) < (a + i\*m+1 + j ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (j == n && i != 1 && i != n){if ( \*(a + i\*m + j) < (a + i\*m-1 + j && \*(a + i\*m + j) < (a + i\*m-1 + j-1 && \*(a + i\*m + j) < (a + i\*m + j-1 && \*(a + i\*m + j) < (a + i\*m+1 + j-1 && \*(a + i\*m + j) < (a + i\*m+1 + j ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}}

if (i == n && j == 1){if ( \*(a + i\*m + j) < (a + i\*m-1 + j) && \*(a + i\*m + j) < (a + i\*m-1 + j+1) && \*(a + i\*m + j) < (a + i\*m + j+1 ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (i == n && j != 1 && j != n){if ( \*(a + i\*m + j) < (a + i\*m + j-1 ) && \*(a + i\*m + j) < (a + i\*m-1 + j-1) && \*(a + i\*m + j) < (a + i\*m-1 + j) && \*(a + i\*m + j) < (a + i\*m-1 + j+1) && \*(a + i\*m + j) < a + i\*m + j+1 ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (i == n && j == n){if ( \*(a + i\*m + j) < (a + i\*m + j-1) && \*(a + i\*m + j) < (a + i\*m-1 + j-1) && \*(a + i\*m + j) < (a + i\*m-1 + j ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

if (i != 1 && i != n && j != 1 && j != n){if (\*(a + i\*m + j) < (a + i\*m-1 + j-1) && \*(a + i\*m + j) < (a + i\*m-1 + j) && \*(a + i\*m + j) < (a + i\*m-1 + j+1) && \*(a + i\*m + j) < (a + i\*m + j-1) && \*(a + i\*m + j) < (a + i\*m + j+1) && \*(a + i\*m + j) < (a + i\*m+1 + j-1) && \*(a + i\*m + j) < (a + i\*m+1 + j) && \*(a + i\*m + j) < (a + i\*m+1 + j+1 ) {numbermax = numbermax + 1; maxim[numbermax = \*(a + i\*m + j);}};

}

}

free(a);

printf("\nkol-vo minimumov: %d\n", numbermax);

if (numbermax != 0){ for (c=0; c<numbermax; c++){

if (c==0) {if ( maxim[c] > maximum ) { maxim[c] = maximum; }}

if (c!=0 && c!=numbermax) {if ( maxim[c+1] > maxim[c] ) { maxim[c+1] = maximum; }}

}printf("\nmaximum iz minimumov: %d", maximum);}

printf("\n------- vvedite luboe chislo :");

scanf("%d", &n);

}

**Вывод программы:**



**Вывод:** Я приобрёл практические навыки в составлении программ по обработке динамических массивов.