Министерство образования Республики Беларусь

г. Минск

Государственное учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

**Лабораторная работа №5**

**“Указатели.”**

**Учебная группа 230501**

Выполнил: Кочеров Роман Сергеевич

Проверил: Селезнев Александр Игоревич

2022 год

**Вариант 4**

**Задача 1**

Заполнить массив с клавиатуры. Удалить все элементы, кратные 11.

Ввод: n, a[n](числа).

#include "Roman.h"

int main()

{

srand(time(NULL));

int\* a;

int i, n;

printf("vvedite ko-vo elementov\n");

n= vvod();

a = mass(n);

ydalenie(a, &n);

vvivod(a, &n);

return 0;

}

#include <stdio.h>

#include <malloc.h>

#include<stdlib.h>

#include<time.h>

#define VIBOR\_MIN 1

#define VIBOR\_MAX 2

int numinputrange(int min, int max)

{

int var;

while ((scanf\_s("%d", &var) == 0) || (var < min) || (var > max) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

int vvod()

{

int var;

while ((scanf\_s("%d", &var) == 0) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

int\* mass(int n1)

{

int\* mass;

int i, k;

mass = (int\*)malloc(n1 \* sizeof(int));

printf("cam(1) random(2)\n");

k = numinputrange(VIBOR\_MIN, VIBOR\_MAX);

if (k == 1)

{

for (i = 0; i < n1; i++)

{

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

mass[i] = vvod();

}

}

if (k == 2)

{

for (int i = 0; i < n1; i++)

{

mass[i] = rand() % 200 - 100; // диапазон случайных чисед от 0 до 100

printf("%d ", mass[i]);

}

printf("\n ");

}

return mass;

}

void ydalenie(int\* a, int\* n)

{

int razmer = \*n;

for (razmer; razmer > -1; razmer--)

{

if ((a[razmer] % 11 == 0) && (a[razmer] != 0))

{

for (int i = razmer; i < \*n; i++)

{

a[i] = a[i + 1];

}

(\*n)--;

}

}

}

void vvivod(int\* a, int\* n)

{

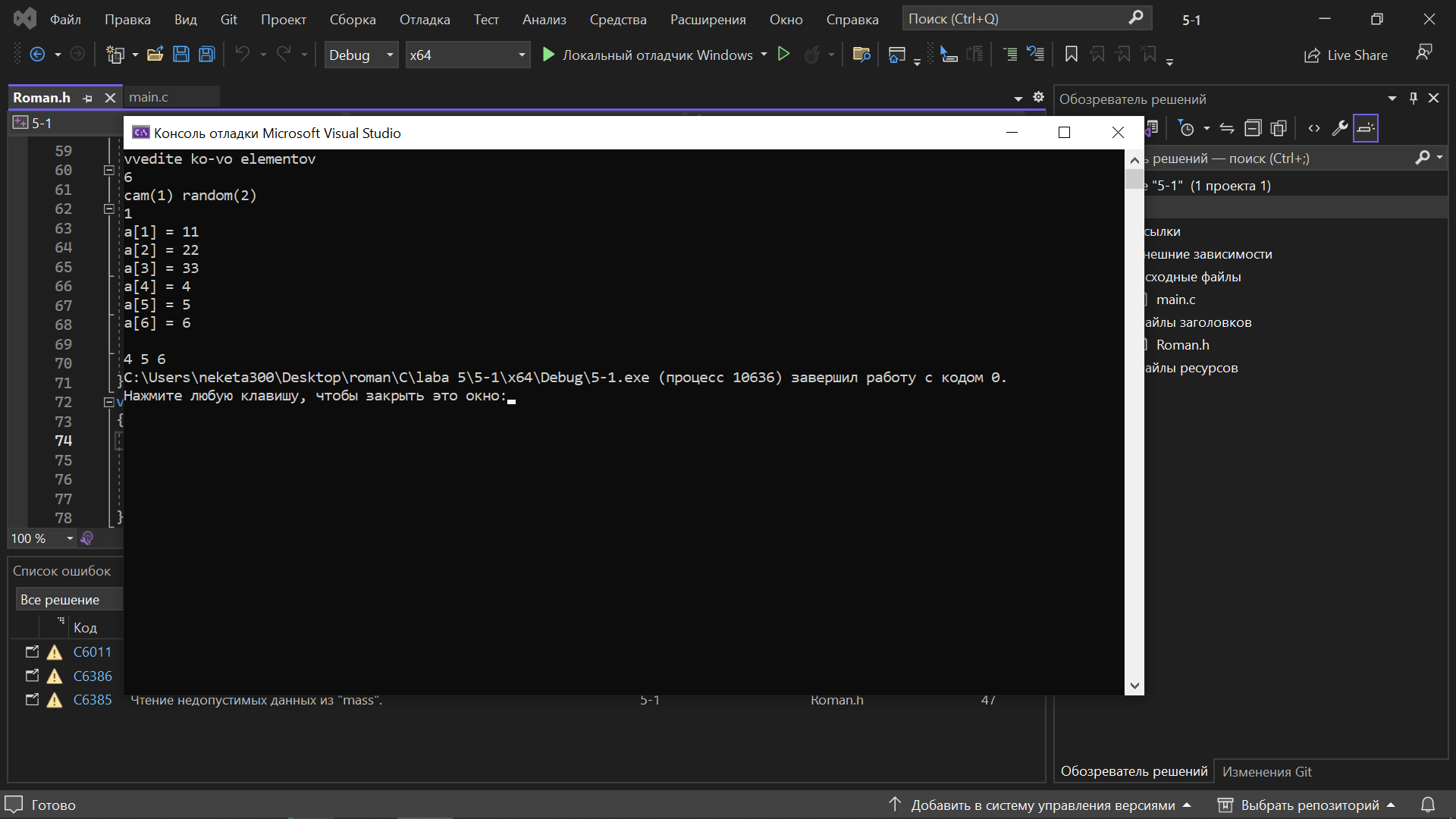
printf("\n");

for (int i = 0; i < \*n; i++)

printf("%d ", a[i]);

free(a);

}



**Задача 2**

Двумерный массив заполнить числами. Количество чисел в строке может быть различным, последнее число - 0. В каждой строке удалить все максимальные элементы

Ввод: n, \*m, a[m\*n](числа).

#include "Roman.h";

int main()

{

srand(time(NULL));

int\*\* a;

int n;

int \* m;

printf("Vvedite ko-vo strok: ");

n = vvod();

a = mass2(n);

m = mass1(n);

vvod\_2matr\_canystolbcami(a, n, m);

vivod\_2matr\_canystolbcami(a, n, m);

ydalenie\_maxelementov\_v\_strokax(a, n, m);

vivod\_2matr\_canystolbcami(a, n, m);

ochichenie(a, n, m);

return 0;

}

#include <stdio.h>

#include <malloc.h>

#include<stdlib.h>

#include<time.h>

#define VIBOR\_MIN 1

#define VIBOR\_MAX 2

int numinputrange(int min, int max)

{

int var;

while ((scanf\_s("%d", &var) == 0) || (var < min) || (var > max) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

int vvod()

{

int var;

while ((scanf\_s("%d", &var) == 0) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

void ydalenie\_maxelementov\_v\_strokax(int\*\* a, int n, int\* m)

{

int i, j, max;

printf("\n");

for (i = n-1; i > -1; i--)

{

max = a[i][0];

for (j = 0; j < m[i]; j++)//поиск максимального

{

if (a[i][j] > max)

{

max = a[i][j];

}

}

printf("max v stroke %d = %d\n", i + 1, max);

for (j = m[i]-1; j > -1; j--)//удаление

{

if (a[i][j]==max)

{

for (int w = j; w < m[i]; w++)

{

a[i][w] = a[i][w + 1];

}

--m[i];

}

}

}

printf("\n");

}

void vvod\_2matr\_canystolbcami(int\*\* a, int n, int\* m)

{

int i, j, k;

for (i = 0; i < n; i++)//ввод

{

printf("Vvedite kol-vo stolcev v stroke %d: ", i + 1);

m[i] = vvod();

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

printf("cam(1) random(2)\n");

k = numinputrange(VIBOR\_MIN, VIBOR\_MAX);

if (k == 1)

{

for (j = 0; j < m[i] - 1; j++)

{

printf("a[%d][%d]= ", i + 1, j + 1);

a[i][j] = vvod();

}

a[i][m[i] - 1] = 0;

}

if (k == 2)

{

for (j = 0; j < m[i] - 1; j++)

{

a[i][j] = rand() % 200 - 100; // диапазон случайных чисед от 0 до 100

}

a[i][m[i] - 1] = 0;

}

}

}

void vivod\_2matr\_canystolbcami(int\*\* a, int n, int\* m)

{

int i, j;

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

{

for (j = 0; j < m[i]; j++)

{

printf("%3d ", a[i][j]);

}

printf("\n");

}

}

int\*\* mass2(int n)

{

int\*\* a;

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

return a;

}

int\* mass1(int n)

{

int\* a;

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

return a;

}

void ochichenie(int\*\* a, int n, int\* m)

{

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

{

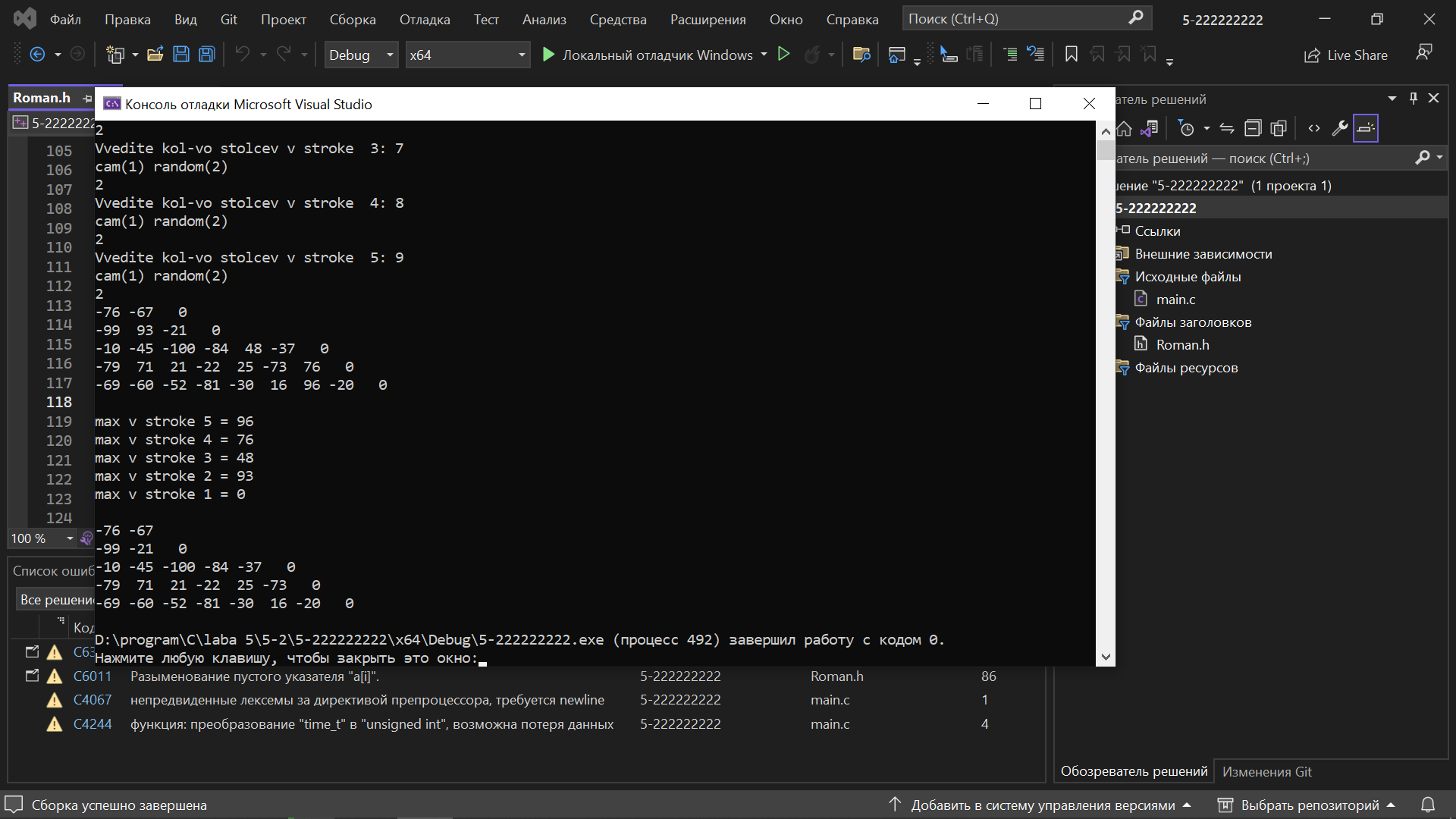
free(a[i]);

}

free(a);

free(m);

}



**Задача 3**

В двумерной матрице размером NxM удалить все строки содержащие более 2х нулей подряд.

Ввод: n, m, a[n\*m] (числа).

#include "Roman.h";

int main()

{

int\*\* a;

int n, m;

srand(time(NULL));

printf("vvedite kol-vo ctrochek:");

n = vvod();

printf("vvedite kol-vo ctolbcev:");

m = vvod();

a = mass(n, m);

vvod\_2matr(a, &n, &m);

vivod\_2matr(a, &n, &m);

ydalenie\_crtochec\_c\_0(a, &n, &m);

vivod\_2matr(a, &n, &m);

ochichenie(a, &n, &m);

return 0;

}

#include <stdio.h>

#include <malloc.h>

#include<stdlib.h>

#include<time.h>

#define VIBOR\_MIN 1

#define VIBOR\_MAX 2

int numinputrange(int min, int max)

{

int var;

while ((scanf\_s("%d", &var) == 0) || (var < min) || (var > max) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

int vvod()

{

int var;

while ((scanf\_s("%d", &var) == 0) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

void vvod\_2matr(int\*\* a, int\* n, int\* m)

{

int i, j, k;

printf("vvedite 2 chiclo(cami-1; random-2)\n");

k = numinputrange(VIBOR\_MIN, VIBOR\_MAX);//тут выбор сам или нет

if (k == 1)

{

for (int i = 0; i < \*n; i++)

for (int j = 0; j < \*m; j++)

{

printf("\nb[%d][%d]=", i + 1, j + 1);

a[i][j] = vvod();

}

}

if (k == 2)

{

for (int i = 0; i < \*n; i++)

for (int j = 0; j < \*m; j++)

a[i][j] = rand() % 199 - 99;

}

}

void vivod\_2matr(int\*\* a, int\* n, int\* m)

{

int i, j;

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

{

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

{

printf("%3d ", a[i][j]);

}

printf("\n");

}

}

void ydalenie\_crtochec\_c\_0(int\*\* a, int\* n, int\* m)

{

int popalsa=0;

for (int i = 0; i < \*n; i++)

{

for (int j = 0; j < (\*m)-2; j++)//поиск 0 0 0

{

if ((a[i][j]==0) && (a[i][j + 1]==0) && (a[i][j + 2]==0))

{

popalsa = 1;

}

}

if (popalsa == 1)//перестановка

{

for (int w = i; w < (\*n) - 1; w++)

{

for (int j = 0; j < \*m; j++)

{

a[w][j] = a[w + 1][j];

}

}

--(\*n);

}

popalsa = 0;

for (int j = 0; j < (\*m) - 2; j++)//поиск 0 0 0 для перескока

{

if ((a[i][j] == 0) && (a[i][j + 1] == 0) && (a[i][j + 2] == 0))

{

popalsa = 1;

}

}

if (popalsa == 1)

{

--i;

popalsa = 0;

}

}

printf("\n");

}

int\*\* mass(int n, int m)

{

int\*\* a;

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

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

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

return a;

}

void ochichenie(int\*\* a, int\* n, int\* m)

{

for (int i = 0; i < \*n; i++)

free(a[i]);

free(a);

}

