Министерство образования Республики Беларусь

г. Минск

Государственное учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

**Лабораторная работа №6**

**“Сортировка.”**

**Учебная группа 230501**

Выполнил: Кочеров Роман Сергеевич

Проверил: Селезнев Александр Игоревич

2022 год

**Вариант 4**

**Задача 1**

В одномерном массиве выполнить сортировку нечетных элементов до последнего положительного методом Шелла.

Ввод: n, a[n](числа).

#include "Roma.h"

int main()

{

srand(time(NULL));

int\* a;

int\* nechet;

int i, n, stop=0, razmer\_nechet=0;

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

n = vvod();

a = mass(n);

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

sortirovka(a, nechet, n, &stop, &razmer\_nechet);

vvivod(a, &n);

printf("\n kol-vo nechetnix - %d", razmer\_nechet);

printf("\n stop - %d", stop+1);

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 sortirovka(int\* a, int\* nechet, int n, int \*stop, int\* razmer\_nechet)

{

int j = 0, tmp, step, i;

for (int i = 0; i < n; i++)//находим стоп

{

if (a[i] > 0)

{

\*stop = i;

}

}

for (int i = 0; i < \*stop; i++)//заполняем массив

{

if (a[i] % 2 != 0)

{

nechet[j] = a[i];

j++;

(\*razmer\_nechet)++;

}

}

for (step = (\*razmer\_nechet) / 2; step > 0; step /= 2) // Перечисление элементов, которые сортируются на определенном шаге

{

for (i = step; i < (\*razmer\_nechet); i++)

{ // Перестановка элементов внутри подсписка, пока i-тый не будет отсортирован

for (j = i - step; (j >= 0) && (nechet[j] > nechet[j + step]); j -= step)

{

tmp = nechet[j];

nechet[j] = nechet[j + step];

nechet[j + step] = tmp;

}

}

}

j = 0;

for (int i = 0; i < \*stop; i++)//вставка

{

if (a[i] % 2 != 0)

{

a[i]= nechet[j];

j++;

}

}

}

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

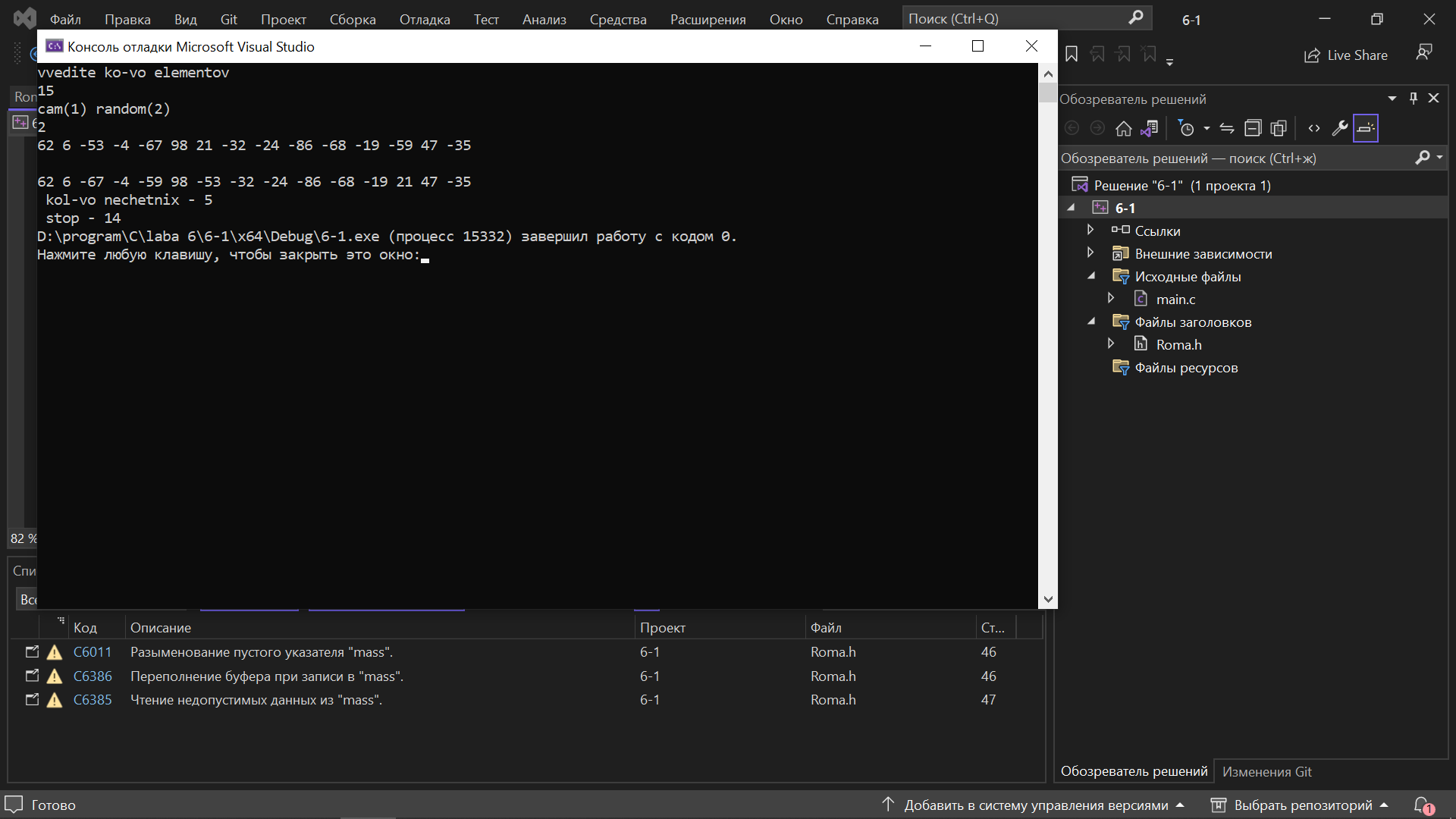
{

printf("\n");

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

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

free(a);}



**Задача 2**

В матрице размером NxM выполнить сортировку столбцов по возрастанию суммы четных элементов методом слияния(пузырёк).

Ввод: n, \*m, a[m\*n](числа).

#include "Roma.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);

chetnie(a, n, m);

bubbleSort(a, n, m);

n = n + 1;

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

ochichenie(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;

}

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;

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

{

a[\*n][i] = 0;

}

}

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("%5d ", a[i][j]);

}

printf("\n");

}

}

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

{

int i, j;

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

{

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

{

if (a[i][j] % 2 == 0)

{

a[n][j] = a[n][j] + a[i][j];

}

}

}

printf("\n");

return 0;

}

void bubbleSort(int\*\* num, int n, int m)

{

// Для всех элементов

for (int i = 0; i < m - 1; i++)

{

for (int j = (m - 1); j > i; j--) // для всех элементов после i-ого

{

if (num[n][j - 1] > num[n][j]) // если текущий элемент меньше предыдущего

{

for (int y = n; y > -1; y--)

{

int temp = num[y][j - 1]; // меняем их местами

num[y][j - 1] = num[y][j];

num[y][j] = temp;

}

}

}

}

}

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

{

n = n + 1;

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)

{

n = n + 1;

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

free(a[i]);

free(a);}

