Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 4

«Многомерные массивы»

Проверил: Выполнил:

асс. каф. ЭВМ ст. гр. 250504

И.Г. Скиба А.Д.Яцкевич

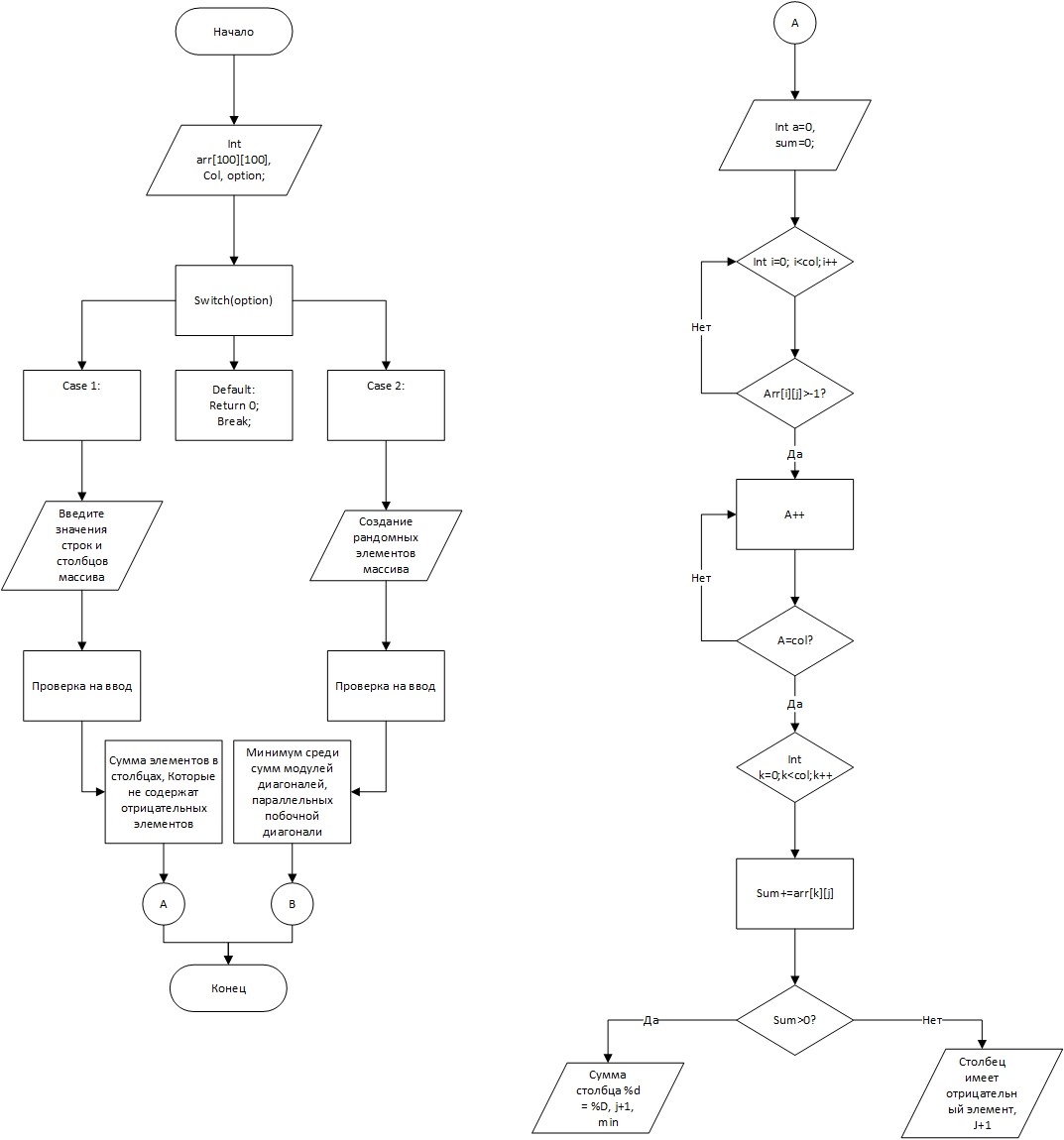
МИНСК 2022

Цель работы: научиться разрабатывать алгоритмы для работы с двумерными массивами и писать код на языке Си по составленному алгоритму. Лабораторная работа включает в себя 3 задачи для выполнения. Задачи решаются последовательно.

**Задание 1**

Дана целочисленная квадратная матрица. Определить: сумму элементов в тех столбцах, которые не содержат отрицательных элементов; минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

* 1. **Блок-схема алгоритма**



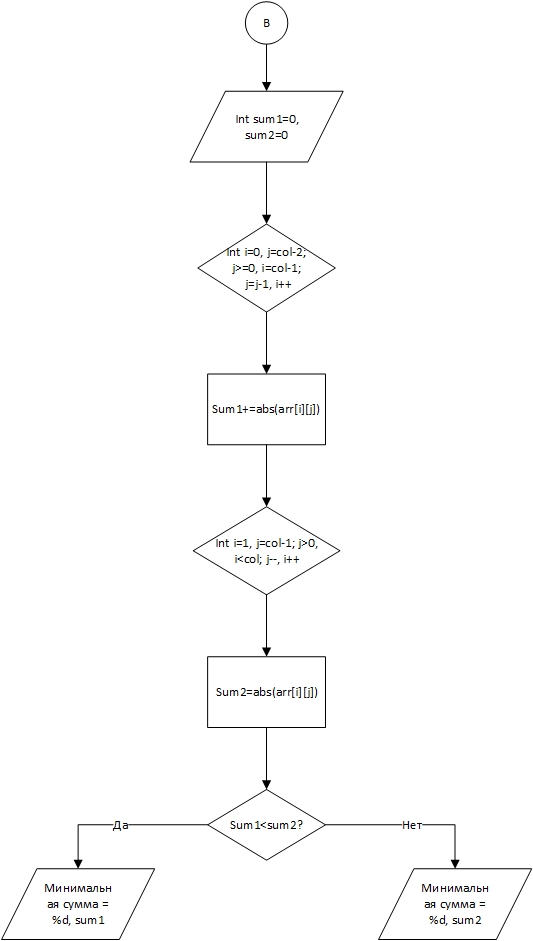


Рисунок 1.1 – Блок-схема алгоритма

* 1. **Код**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include <locale.h>

int input\_check(int\* col);

int choice\_input\_check();

void arr\_input\_check(int arr[100][100], int col);

void arr\_rand\_input(int arr[100][100], int col);

void output\_arr(int arr[100][100], int col);

int sum(int arr[100][100], int col);

int minn(int arr[100][100], int col);

int main()

{

int arr[100][100], col, choice = 0;

input\_check(&col);

choice = choice\_input\_check();

switch (choice)

{

case 1:

arr\_input\_check(arr, col);

break;

case 2:

arr\_rand\_input(arr, col);

break;

default:

return 0;

break;

}

output\_arr(arr, col);

printf(sum(arr, col), minn(arr, col));

}

int input\_check(int\* col)

{

printf("Enter arr size\n");

while (scanf\_s("%d", col) == 0 || \*col >= 101 || \*col < 1)

{

printf("Wrong input.\n");

rewind(stdin);

}

}

int choice\_input\_check()

{

int input;

printf("Print 1 for manually 2 random\n");

while (scanf\_s("%d", &input) < 1 || input > 2)

{

printf\_s("Wrong input\n");

rewind(stdin);

}

return input;

}

void arr\_input\_check(int arr[100][100], int col)

{

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

{

printf("Enter the elements of the %d row\n", i + 1);

for (int j = 0; j < col; j++)

{

while (scanf\_s("%d", &arr[i][j]) == 0)

{

printf("Wrong input.\n");

rewind(stdin);

}

}

}

}

void arr\_rand\_input(int arr[100][100], int col)

{

srand(time(NULL));

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

{

for (int j = 0; j < col; j++)

{

arr[i][j] = rand() % 200 - 100;

}

}

}

void output\_arr(int arr[100][100], int col)

{

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

{

for (int j = 0; j < col; j++)

{

printf("%d \t", arr[i][j]);

}

printf("\n");

}

}

int sum(int arr[100][100], int col)

{

for (int j = 0; j < col; j++)

{

int a = 0;

int sum = 0;

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

{

if (arr[i][j] > -1)

{

a++;

}

}

if (a == col)

{

for (int k = 0; k < col; k++)

{

sum += arr[k][j];

}

}

if (sum > 0)

{

printf("\nSumma stolbca %d = %d", j + 1, sum);

}

else

{

printf("\nCol %d has a negative element", j + 1);

}

}

}

int minn(int arr[100][100], int col)

{

int sum1 = 0, sum2 = 0;

for (int i = 0, j = col - 2; j >= 0, i < col - 1; j = j - 1, i++)

{

sum1 += abs(arr[i][j]);

}

for (int i = 1, j = col - 1; j > 0, i < col; j--, i++)

{

sum2 += abs(arr[i][j]);

}

if (sum1 < sum2)

{

printf("Min sum %d", sum1);

}

else

{

printf("Min sum %d", sum2);

}

return 0;

}

* 1. **Результат выполнения программы**

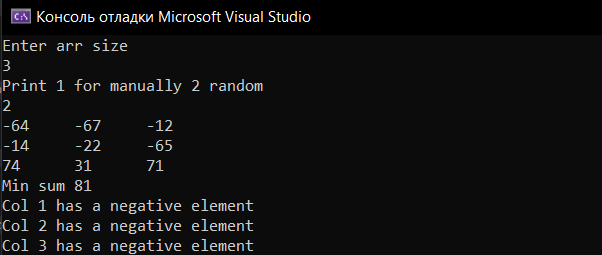


Рисунок 1.2 - Скриншот результата выполнения программы

**Задание 2**

Найти в матрице первый столбец, все элементы которого равны нулю. Знаки элементов строки с таким же номером изменить на противоположные.

**2.1 Блок-схема алгоритма**

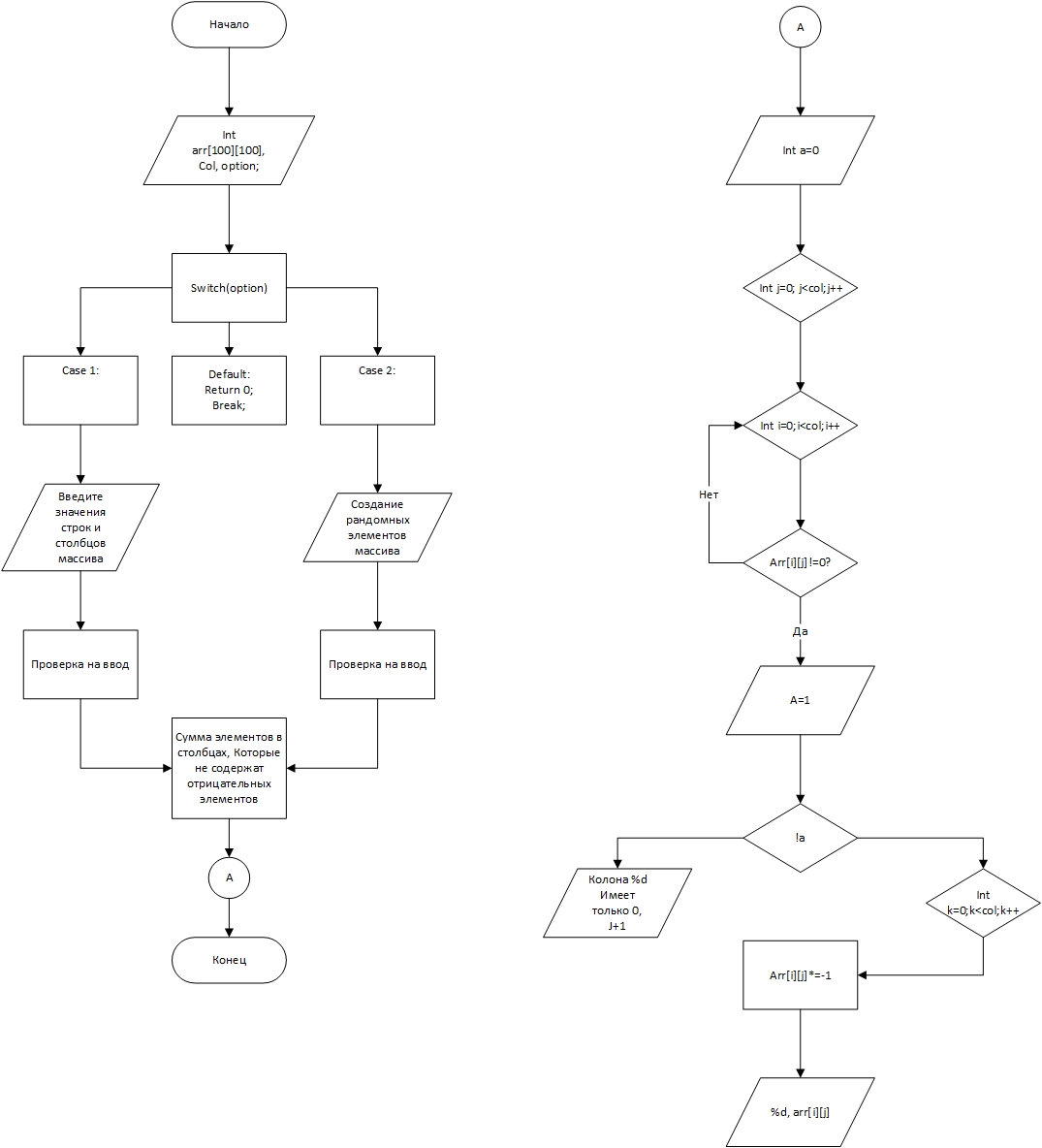
****

Рисунок 2.1 – Блок-схема алгоритма

**2.2 Исходный код**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

void arr\_rand\_input(int arr[100][100], int col, int row);

int find\_strings\_without\_zero(int arr[100][100], int row, int col);

void output\_arr(int arr[100][100], int row, int col);

void arr\_input\_check(int arr[100][100], int col, int row);

int choice\_input\_check();

int input\_check(int\* row, int\* col);

int main()

{

int arr[100][100], col, row, choice = 0, max\_elem = 0;

input\_check(&row, &col);

choice = choice\_input\_check();

switch (choice)

{

case 1:

arr\_input\_check(arr, col, row);

break;

case 2:

arr\_rand\_input(arr, col, row);

break;

default:

return 0;

break;

}

output\_arr(arr, row, col);

printf(find\_strings\_without\_zero(arr, row, col));

}

int input\_check(int\* row, int\* col)

{

printf("Enter arr cols\n");

while (scanf\_s("%d", col) == 0 || \*col >= 101 || \*col < 1 || getchar == '\n')

{

printf("Wrong input\n");

rewind(stdin);

}

printf("Enter arr rows\n");

while (scanf\_s("%d", row) == 0 || \*row >= 101 || \*row < 1 || getchar == '\n')

{

printf("Wrong input\n");

rewind(stdin);

}

}

int choice\_input\_check()

{

int input;

printf("Print 1 for manually 2 to random\n");

while (scanf\_s("%d", &input) < 1 || input > 2)

{

printf\_s("Wrong input\n");

rewind(stdin);

}

return input;

}

void arr\_input\_check(int arr[100][100], int col, int row)

{

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

{

printf("Enter the elements of the %d row\n", i + 1);

for (int j = 0; j < col; j++)

{

while (scanf\_s("%d", &arr[i][j]) == 0)

{

printf("Wrong input.\n");

rewind(stdin);

}

}

}

}

void arr\_rand\_input(int arr[100][100], int col, int row)

{

srand(time(NULL));

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

{

for (int j = 0; j < col; j++)

{

arr[i][j] = rand() % 200 - 100;

}

}

}

void output\_arr(int arr[100][100], int row, int col)

{

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

{

for (int j = 0; j < col; j++)

{

printf("%d \t", arr[i][j]);

}

printf("\n");

}

}

int find\_strings\_without\_zero(int arr[100][100], int row, int col)

{

int a=0;

int sum = 0;

for (int j = 0; j < col; j++)

{

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

{

if (arr[i][j] != 0)

a = 1;

}

if (!a)

{

printf("Column %d has only zero\n", j+1);

for (int k = 0; k < col; k++)

{

arr[j][k] \*= -1;

printf("%d ", arr[j][k]);

}

break;

}

}

}

**2.3 Результат выполнения программы**

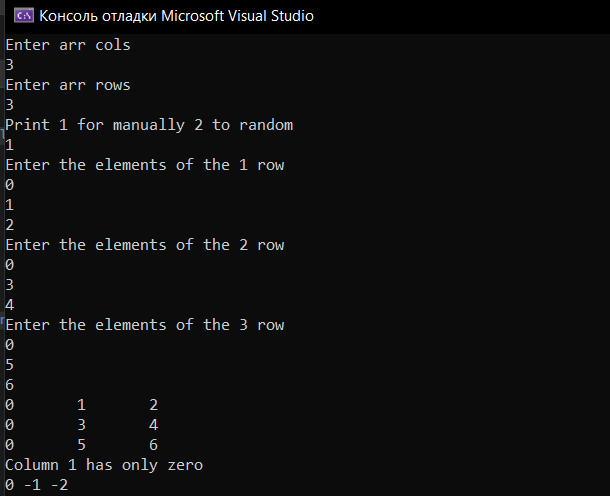


Рисунок 2.2 - Скриншот результата выполнения программы

**Задание 3**

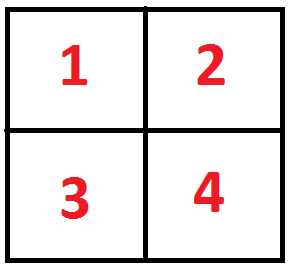


Рисунок 3.1 – Матрица разделённая на области

Условие: В квадратной матрице размером NxN найти минимальный четный элемент в 3-ой области.

**3.1 Блок-схема алгоритма**

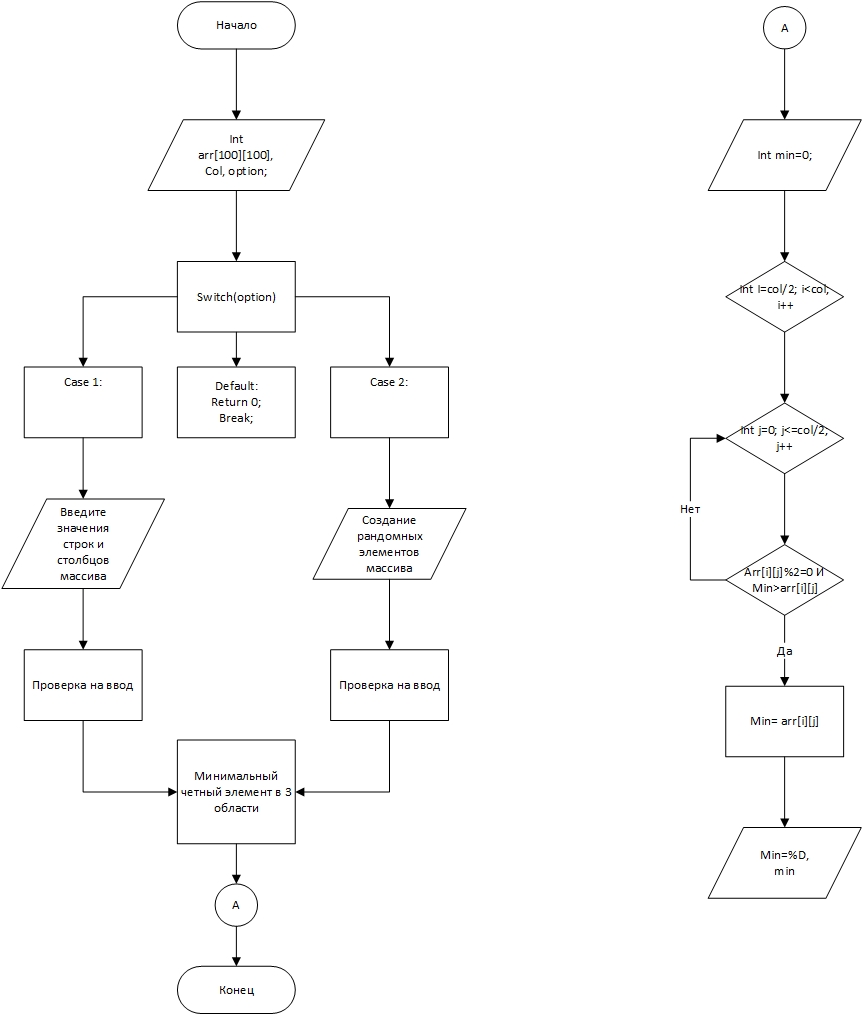


Рисунок 3.2 – Блок-схема алгоритма

**3.2 Исходный код**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include <locale.h>

int input\_check(int\* col);

int choice\_input\_check();

void arr\_input\_check(int arr[100][100], int col);

void arr\_rand\_input(int arr[100][100], int col);

void output\_arr(int arr[100][100], int col);

int sum(int arr[100][100], int col);

int main()

{

int arr[100][100], col, choice = 0;

input\_check(&col);

choice = choice\_input\_check();

switch (choice)

{

case 1:

arr\_input\_check(arr, col);

break;

case 2:

arr\_rand\_input(arr, col);

break;

default:

return 0;

break;

}

output\_arr(arr, col);

printf(sum(arr, col));

}

int input\_check(int\* col)

{

printf("Enter arr\n");

while (scanf\_s("%d", col) == 0 || \*col >= 101 || \*col < 1)

{

printf("Wrong input.\n");

rewind(stdin);

}

}

int choice\_input\_check()

{

int input;

printf("Print 1 for manually 2 to random\n");

while (scanf\_s("%d", &input) < 1 || input > 2)

{

printf\_s("Wrong input\n");

rewind(stdin);

}

return input;

}

void arr\_input\_check(int arr[100][100], int col)

{

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

{

printf("Enter the elements of the %d row and cols\n", i + 1);

for (int j = 0; j < col; j++)

{

while (scanf\_s("%d", &arr[i][j]) == 0)

{

printf("Wrong input.\n");

rewind(stdin);

}

}

}

}

void arr\_rand\_input(int arr[100][100], int col)

{

srand(time(NULL));

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

{

for (int j = 0; j < col; j++)

{

arr[i][j] = rand() % 200 - 100;

}

}

}

void output\_arr(int arr[100][100], int col)

{

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

{

for (int j = 0; j < col; j++)

{

printf("%d \t", arr[i][j]);

}

printf("\n");

}

}

int sum(int arr[100][100], int col)

{

int min = 0;

for (int i = col / 2; i < col; i++)

{

for (int j = 0; j <= col / 2; j++)

{

printf("%d\t", arr[i][j]);

}

printf("\n");

}

printf("\n");

for (int i = col / 2; i < col; i++)

{

for (int j = 0; j <= col / 2; j++)

{

if (arr[i][j]%2 == 0 && min > arr[i][j])

{

min = arr[i][j];

}

}

}

printf("min = %d\n", min);

}

**3.3 Результат выполнения программы**

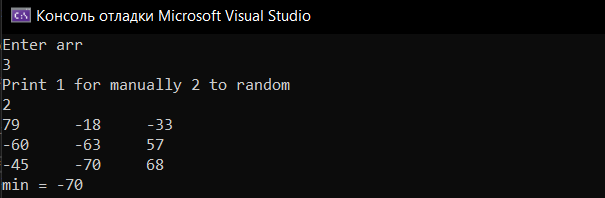
****

Рисунок 3.3 - Скриншот результата выполнения программы