

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий

Отчет № 7

по дисциплине «Информатика»

на тему: «Чтение из файла. Запись в файл. Программа приближенного вычисления логарифмов
с заданной точностью»

Выполнил:
студент группы 3530902/90001

_____ Непушкин Сергей Александрович

Проверил:
Доцент ВШКФСИУ

_____ Теплова Наталья Витальевна

Санкт-Петербург
2019 г.

Оглавление	
1.Задание.....	3
2.Доказательство алгоритма.....	3
3.Блок-схема алгоритма.....	4
4.Текст кода.....	6
5.Пример работы программы.....	9
6.Решение варианта.....	10

1. Задание.

Написать программу сортировки массива длины n "пузырьком".

Длина массива в приложенном файле с вариантами.

Сгенерировать случайное заполнение массива целыми числами от 0 до n . Вычислить число итераций при сортировке "пузырьком". Повторить 50 раз.

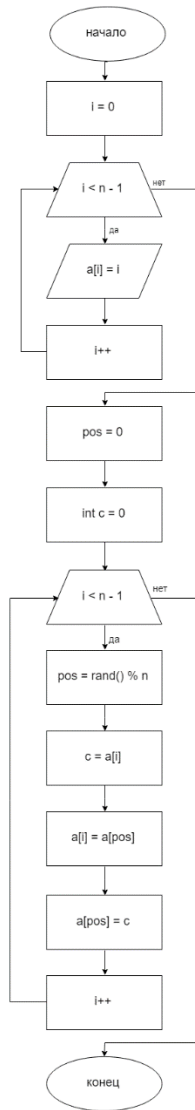
Вычислить среднее число итераций при сортировке массива алгоритмом "пузырек".

2. Доказательство алгоритма

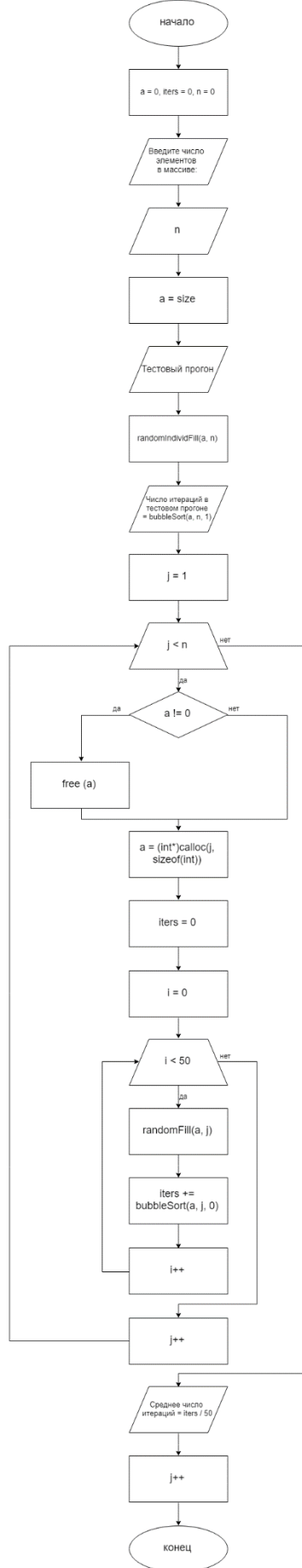
В конце каждой итерации максимальное число не отсортированной части массива оказывается на последнем месте, следующая итерация будет выполняться без учета уже отсортированных элементов, сортировка закончится, когда переменная $shift = 0$.

3.Блок-схема алгоритма

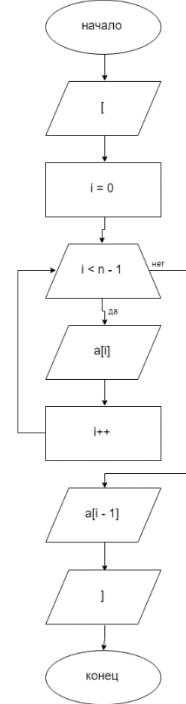
Функция void randomIndividFill



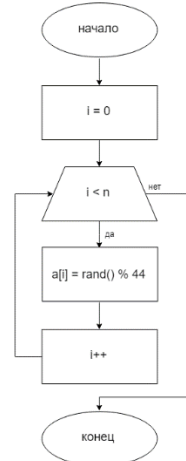
Тело алгоритма

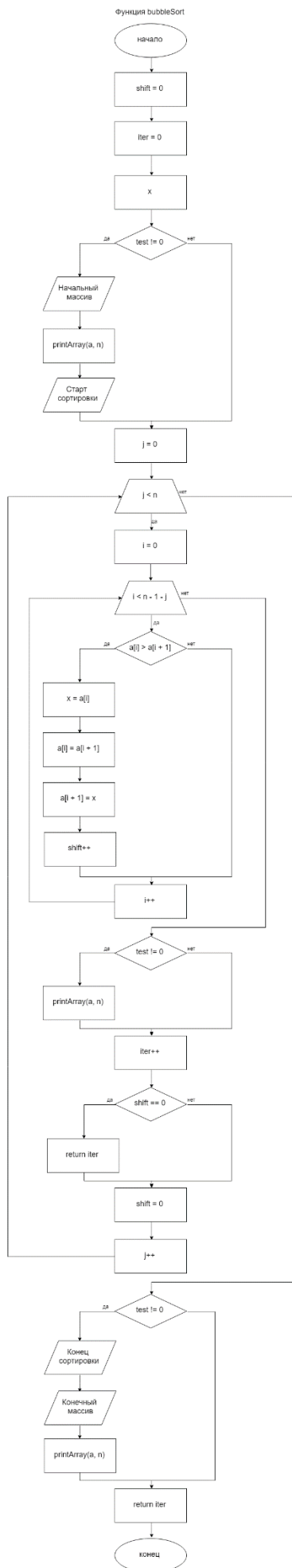


Функция printArray



Функция randomFill





4.Текст кода.

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<math.h>
#include<time.h>
#include<stdlib.h>
#include<locale.h>
void printArray(int* a, int n)
{
    printf("[");
    for (int i = 0; i < n - 1; i++)
    {
        printf("%d, ", a[i]);
    }
    printf("%d", a[n - 1]);
    printf("]\n");
}

void randomFill(int* a, int n)
{
    for (int i = 0; i < n; i++)
    {
        a[i] = rand() % 44;
    }
}

void randomIndividFill(int* a, int n)
{
    for (int i = 0; i < n; i++)
    {
        a[i] = i;
    }
    int pos = 0;
    int c = 0;
    for (int i = 0; i < n; i++)
    {
        pos = rand() % n;
        c = a[i];
        a[i] = a[pos];
        a[pos] = c;
    }
}

int bubbleSort(int* a, int n, int test)
{
    int shift = 0;
    int iter = 0;
    int x;
    if (test != 0)
    {
        printf("Начальный массив\n");
        printArray(a, n);
        printf("Старт сортировки\n");
    }
    for (int j = 0; j < n; j++)
    {
        for (int i = 0; i < n - 1 - j; i++)
        {
            if (a[i] > a[i + 1])
            {
                x = a[i];
                a[i] = a[i + 1];
                a[i + 1] = x;
                shift++;
            }
        }
    }
}
```

```

        }
    }
    if (test != 0)
    {
        printArray(a, n);
    }
    iter++;
    if (shift == 0)
    {
        return iter;
    }
    shift = 0;
}
if (test != 0)
{
    printf("Конец сортировки\n");
    printf("Конечный массив\n");
    printArray(a, n);
}
return iter;
}

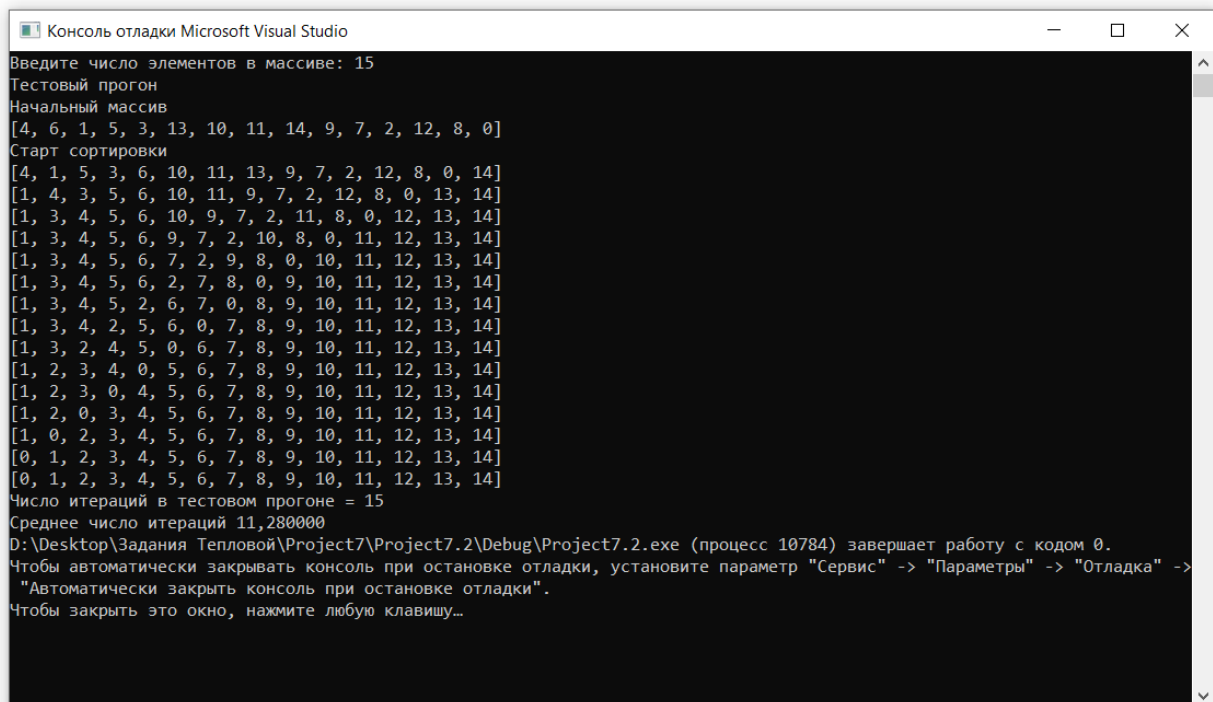
int main()
{
    setlocale(LC_ALL, "Russian");
    srand(time(0));
    int* a = 0;
    int iters = 0;
    int n = 0;
    printf("Введите число элементов в массиве: ");
    scanf("%d", &n);

    a = (int*)calloc(n, sizeof(int));
    printf("Тестовый прогон\n");
    randomIndividFill(a, n);
    printf("Число итераций в тестовом прогоне = %d\n", bubbleSort(a, n, 1));

    for (int j = 1; j <= n; j++)
    {
        if (a != 0)
        {
            free(a);
        }
        a = (int*)calloc(j, sizeof(int));
        iters = 0;
        for (int i = 0; i < 50; i++)
        {
            randomFill(a, j);
            iters += bubbleSort(a, j, 0);
        }
    }
    printf("Среднее число итераций %f", (float)iters / 50);
    free(a);
    return 0;
}

```


5.Пример работы программы



```
Консоль отладки Microsoft Visual Studio
Введите число элементов в массиве: 15
Тестовый прогон
Начальный массив
[4, 6, 1, 5, 3, 13, 10, 11, 14, 9, 7, 2, 12, 8, 0]
Старт сортировки
[4, 1, 5, 3, 6, 10, 11, 13, 9, 7, 2, 12, 8, 0, 14]
[1, 4, 3, 5, 6, 10, 11, 9, 7, 2, 12, 8, 0, 13, 14]
[1, 3, 4, 5, 6, 10, 9, 7, 2, 11, 8, 0, 12, 13, 14]
[1, 3, 4, 5, 6, 9, 7, 2, 10, 8, 0, 11, 12, 13, 14]
[1, 3, 4, 5, 6, 7, 2, 9, 8, 0, 10, 11, 12, 13, 14]
[1, 3, 4, 5, 6, 2, 7, 8, 0, 9, 10, 11, 12, 13, 14]
[1, 3, 4, 5, 2, 6, 7, 0, 8, 9, 10, 11, 12, 13, 14]
[1, 3, 4, 2, 5, 6, 0, 7, 8, 9, 10, 11, 12, 13, 14]
[1, 3, 2, 4, 5, 0, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[1, 2, 3, 4, 0, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[1, 2, 3, 0, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[1, 2, 0, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[1, 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
Число итераций в тестовом прогоне = 15
Среднее число итераций 11,280000
D:\Desktop\Задания Тепловой\Project7\Project7.2\Debug\Project7.2.exe (процесс 10784) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" ->
"Автоматически закрыть консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...
```

6. Решение варианта

```
Консоль отладки Microsoft Visual Studio
Введите число элементов в массиве: 43
Тестовый прогон
Начальный массив
[25, 7, 38, 9, 20, 15, 19, 23, 40, 36, 14, 4, 21, 1, 31, 6, 37, 17, 8, 39, 30, 42, 34, 22, 29, 27, 41, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10]
Старт сортировки
[25, 9, 20, 15, 19, 23, 38, 36, 14, 4, 21, 1, 31, 6, 37, 17, 8, 39, 30, 40, 34, 22, 29, 27, 41, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 42]
[7, 9, 20, 15, 19, 23, 25, 36, 14, 4, 21, 1, 31, 6, 37, 17, 8, 38, 30, 39, 34, 22, 29, 27, 40, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 41, 42]
[7, 9, 15, 19, 20, 23, 25, 14, 4, 21, 1, 31, 6, 36, 17, 8, 37, 30, 38, 34, 22, 29, 27, 39, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 40, 41, 42]
[7, 9, 15, 19, 20, 23, 14, 4, 21, 1, 25, 6, 31, 17, 8, 36, 30, 37, 34, 22, 29, 27, 38, 20, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 39, 40, 41, 42]
[7, 9, 15, 19, 20, 14, 4, 21, 1, 23, 6, 25, 17, 8, 31, 30, 36, 34, 22, 29, 27, 37, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 38, 39, 40, 41, 42]
[7, 9, 15, 19, 14, 4, 20, 1, 21, 6, 23, 17, 8, 25, 30, 31, 34, 22, 29, 27, 36, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 37, 38, 39, 40, 41, 42]
[7, 9, 15, 14, 4, 19, 1, 20, 6, 21, 17, 8, 23, 25, 30, 31, 22, 29, 27, 34, 28, 11, 33, 24, 2, 16, 35, 26, 12, 13, 18, 0, 3, 32, 5, 10, 36, 37, 38, 39, 40, 41, 42]
[7, 9, 14, 4, 15, 1, 19, 6, 20, 17, 8, 21, 23, 25, 30, 22, 29, 27, 31, 28, 11, 33, 24, 2, 16, 34, 26, 12, 13, 18, 0, 3, 32, 5, 10, 35, 36, 37, 38, 39, 40, 41, 42]
[7, 9, 4, 14, 1, 15, 6, 19, 17, 8, 20, 21, 23, 25, 22, 29, 27, 30, 28, 11, 31, 24, 2, 16, 33, 26, 12, 13, 18, 0, 3, 32, 5, 10, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[7, 4, 9, 1, 14, 6, 15, 17, 8, 19, 20, 21, 23, 22, 25, 27, 29, 28, 11, 30, 24, 2, 16, 31, 26, 12, 13, 18, 0, 3, 32, 5, 10, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[4, 7, 1, 9, 6, 14, 15, 8, 17, 19, 20, 21, 22, 23, 25, 27, 28, 11, 29, 24, 2, 16, 30, 26, 12, 13, 18, 0, 3, 31, 5, 10, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[4, 1, 7, 6, 9, 14, 8, 15, 17, 19, 20, 21, 22, 23, 25, 27, 11, 28, 24, 2, 16, 29, 26, 12, 13, 18, 0, 3, 30, 5, 10, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 9, 8, 14, 15, 17, 19, 20, 21, 22, 23, 25, 11, 27, 24, 2, 16, 28, 26, 12, 13, 18, 0, 3, 29, 5, 10, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 17, 19, 20, 21, 22, 23, 11, 25, 24, 2, 16, 27, 26, 12, 13, 18, 0, 3, 28, 5, 10, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 17, 19, 20, 21, 22, 11, 23, 24, 2, 16, 25, 26, 12, 13, 18, 0, 3, 27, 5, 10, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 17, 19, 20, 21, 11, 22, 23, 2, 16, 24, 25, 12, 13, 18, 0, 3, 26, 5, 10, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 17, 19, 20, 11, 21, 22, 2, 16, 23, 24, 12, 13, 18, 0, 3, 25, 5, 10, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 17, 19, 11, 20, 21, 2, 16, 22, 23, 12, 13, 18, 0, 3, 24, 5, 10, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 17, 11, 19, 20, 2, 16, 21, 22, 12, 13, 18, 0, 3, 23, 5, 10, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 15, 11, 17, 19, 2, 16, 20, 21, 12, 13, 18, 0, 3, 22, 5, 10, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 14, 11, 15, 17, 2, 16, 19, 12, 13, 18, 0, 3, 21, 5, 10, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 11, 14, 15, 17, 2, 16, 17, 19, 12, 13, 18, 0, 3, 20, 5, 10, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 11, 14, 2, 15, 16, 17, 12, 13, 18, 0, 3, 19, 5, 10, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 11, 2, 14, 15, 16, 12, 13, 17, 0, 3, 18, 5, 10, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 9, 2, 11, 14, 15, 12, 13, 16, 0, 3, 17, 5, 10, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 8, 2, 9, 11, 14, 12, 13, 15, 0, 3, 16, 5, 10, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 7, 2, 8, 9, 11, 12, 13, 14, 0, 3, 15, 5, 10, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 2, 7, 8, 9, 11, 12, 13, 0, 3, 14, 5, 10, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 4, 6, 2, 7, 8, 9, 11, 12, 0, 3, 13, 5, 10, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 4, 6, 7, 8, 9, 11, 0, 3, 12, 5, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 4, 6, 7, 8, 9, 0, 3, 11, 5, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 4, 6, 7, 8, 0, 3, 9, 5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 4, 6, 7, 0, 3, 8, 5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 4, 6, 0, 3, 7, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 4, 0, 3, 6, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 2, 0, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[1, 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
Число итераций в тестовом прогоне = 39
Среднее число итераций 36,888002
D:\Desktop\Задания Тепловой\Project7\Project7.2\Debug\Project7.2.exe (процесс 7888) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...
```