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.

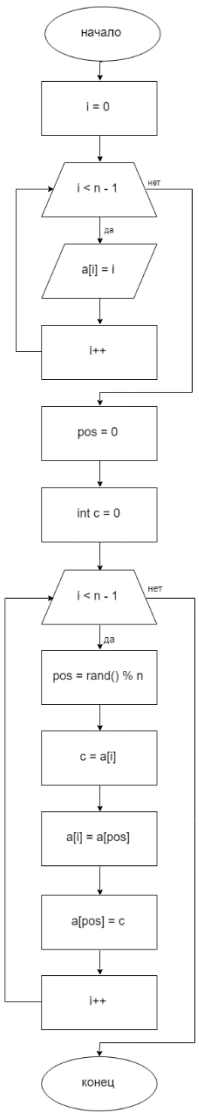
Написать программу сортировки массива длины n "пузырьком".
Длина массива в приложенном файле с вариантами.
Сгенерировать случайное заполнение массива целыми числами от 0 до n. Вычислить число итераций при сортировке "пузырьком". Повторить 50 раз.
Вычислить среднее число итераций при сортировке массива алгоритмом "пузырек".

2.Доказательство алгоритма

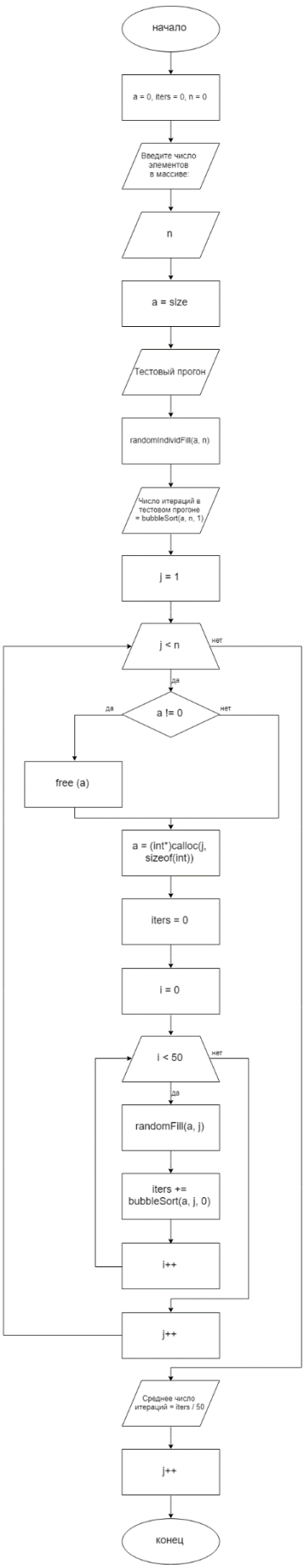
В конце каждой итерации максимальное число не отсортированной части массива оказывается на последнем месте, следующая итерация будет выполняться без учета уже отсортированных элементов, сортировка закончится, когда переменная shift = 0.

3.Блок-схема алгоритма

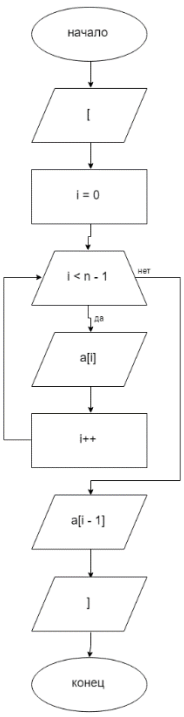
Функция void randomIndividFill



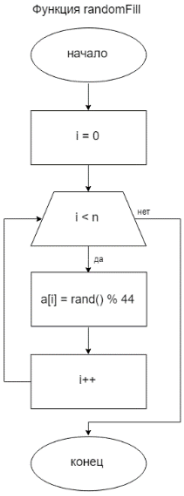
Тело алгоритма

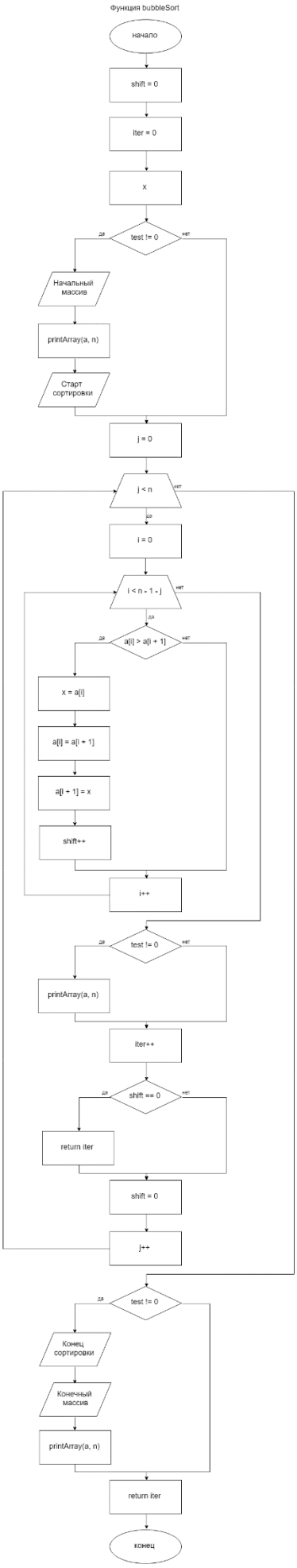


Функция printArray



Функция randomFill




Функция bubbleSort
начало
shift = 0
iter = 0
x
test != 0
да
нет
Начальный массив
printArray(a, n)
Старт сортировки
j = 0
j < n
нет
да
i = 0
i < n - 1 - j
нет
да
a[i] > a[i + 1]
да
нет
x = a[i]
a[i] = a[i + 1]
a[i + 1] = x
shift++
i++
test != 0
да
нет
printArray(a, n)
iter++
shift == 0
да
нет
return iter
shift = 0
j++
test != 0
да
Конец сортировки
Конечный массив
printArray(a, n)
return iter
конец

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++;
```

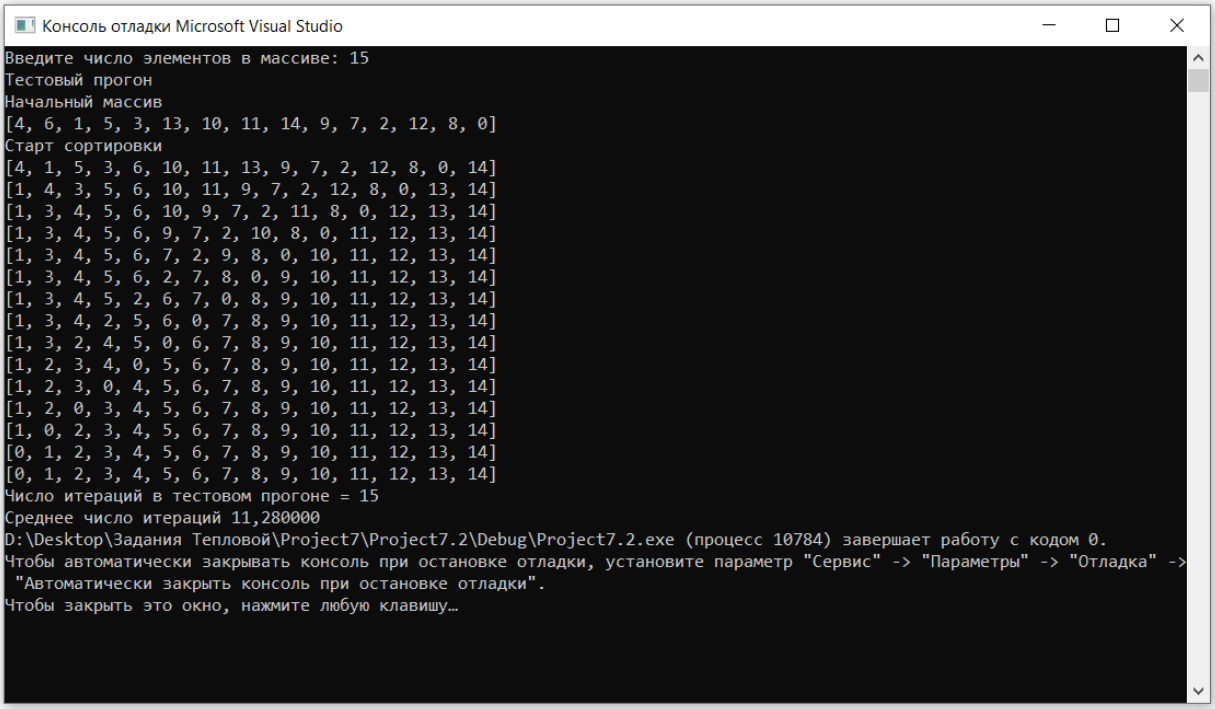
```
                        }
                }
                if (test != 0)
                {
                        printArray(a, n);
                }
                iter++;
                if (shift == 0)
                {
                        return iter;
                }
                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



6. Решение варианта