Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Нижегородский Государственный Университет им.

Н.И.Лобачевского» (ННГУ)

Институт Информационных Технологий Математики и Механики

Отчёт по лабораторной работе №2

Генерация случайных чисел. Сравнение сортировок.

Выполнила:

обучающаяся группы 3821Б1ФИ3

Кухарева А.Р.

Проверил:

заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г

Нижний Новгород

2021г.

**Содержание**

Введение....................................................................................................2

Постановка задачи....................................................................................3

Руководство пользователя........................................................................4-6

Руководство программиста………………………………………………7-14

Описание структуры программы.........................................12

Описание алгоритмов.....................................................................12-14

Эксперименты.............................................................................................15

Заключение..................................................................................................16

Список литературы.....................................................................................17

Приложение 1...............................................................................................18

Приложение 2...............................................................................................19-22

**Введение**

**Программи́рование** — [процесс](https://ru.wikibooks.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81) и [искусство](https://ru.wikibooks.org/w/index.php?title=%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%BE&action=edit&redlink=1) создания компьютерных программ с помощью [языков программирования](https://ru.wikibooks.org/w/index.php?title=%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F&action=edit&redlink=1).

Программирование сочетает в себе элементы искусства, науки, математики и инженерии.

В узком смысле слова, программирование рассматривается как кодирование — реализация одного или нескольких взаимосвязанных алгоритмов на некотором языке программирования. В более широком смысле, программирование — процесс создания программ, то есть разработка программного обеспечения.

Существуют десятки алгоритмов сортировки, каждый из которых хорош в чём-то своём. На одних легко объяснять принципы сортировки, другие хороши при работе с большими массивами, третьи оптимизированы по скорости, четвёртые — по числу процессорных циклов, компактности кода и т. д.

**Постановка задачи**

Реализовать сортировки массивов данных (тип данных определяется преподавателем(float)) задаваемых: обязательно случайно, дополнительно с клавиатуры или из файла.

Реализовать сортировки: пузырьком, вставкой, быстрая. Сравнить время работы, сделать выводы.

Первая программа создает текстовый файл с записанными в него числами. Программа принимает количество чисел n, максимальное и минимальное значение.

Вторая программа читает текстовый файл с набором чисел, выводит консольный интерфейс (печать, сортировка, сброс, выход), выполняет выбранные действия.

**Руководство пользователя**

* **Программа 1.**

Пользователю после запуска программы нужно ввести с клавиатуры следующие значения: количество случайных чисел, минимальное и максимальное числа.



Рисунок 1. Консоль при запуске программы.

Здесь пользователь вводит число элементов массива случайных чисел (количество должно быль больше нуля).

Затем пользователю нужно ввести минимальное и максимальное значения (максимальный элемент должен быть обязательно больше минимального).

Если пользователь все-таки нарушит условие и введет количество чисел <=0, то программа выдаст следующее сообщение:” Ошибка. Количество элементов должно быть больше нуля!”

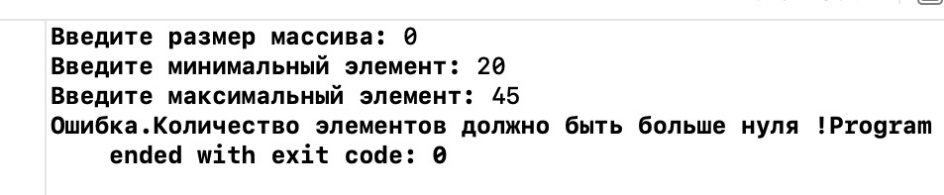


Рисунок 2. Консоль после ввода количества чисел, равное нулю.

Так же программа выдаст сообщение:” Ошибка. Максимальный элемент не может быть меньше минимального!”, если ввести минимальное значение больше максимального.

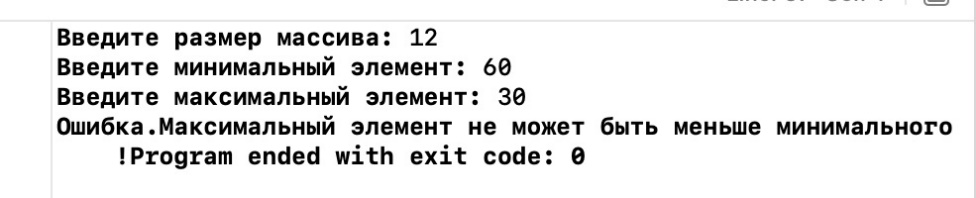


Рисунок 3. Консоль после ввода минимального элемента больше максимального.

Если пользователь выполнил все условия и данные введены корректно, то программа создаст массив рандомных чисел и запишет его в файл.

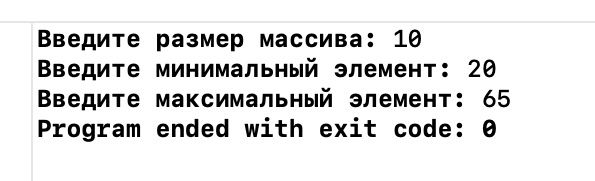


Рисунок 4. Консоль после того, как ввели корректные данные.

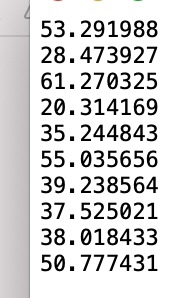


Рисунок 5. Результат программы. Массив рандомных чисел, записанный в файл.

* **Программа 2.**

После запуска программы будет выведен интерфейс, (1. печать, 2. сортировка, 3. сброс, 4. выход) в котором ему необходимо выбрать нужное действие.

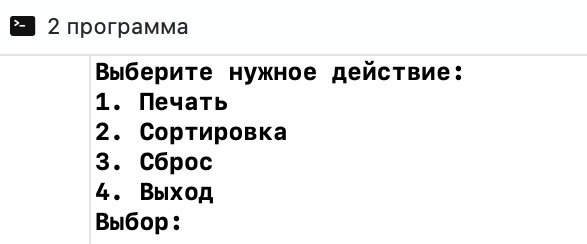


Рисунок 6. Консоль после запуска программы.

Если пользователь выберет “1. Печать”, то выводится текущий массив данных. Если была проведена сортировка, то выводится отсортированный массив. Если сортировка не проводилась, то выводится исходный массив ( из файла).

Если пользователь выберет “2. Сортировка”, то перед пользователем будет следующий интерфейс (1. Сортировка пузырьком, 2. Быстрая сортировка, 3. Сортировка вставкой). Пользователю нужно выбрать какую сортировку он хочет осуществить.

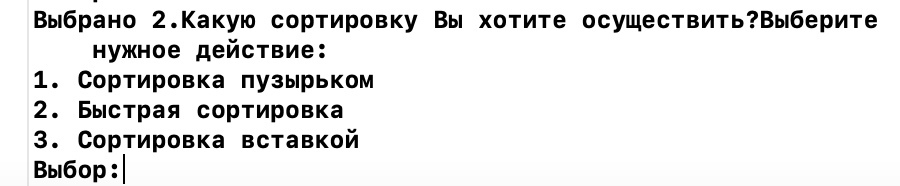


Рисунок 7. Консоль после выбора “2. Сортировка”.

После того, как пользователь выберет любую из предложенных сортировок, на консоли появится время, которое потребовалось на сортировку данного массива в милисекундах. Перед пользователем останется прежнее меню (1. печать, 2. сортировка, 3. сброс, 4. выход).

Если пользователь выберет “3. Сброс”, тогда отсортированный массив сбросит к исходному (из файла).

При выборе “4. Выход” программа завершается и меню закрывается.

**Руководство программиста**

* **Программа 1.**

Первая программа создает текстовый файл с записанными в него числами. Программа принимает количество чисел n, максимальное и минимальное значение.

1) Подключение всех библиотек, с которыми предстоит работать.

[#include](https://vk.com/im?sel=176031583&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <time.h>  
const float RAND\_MAX\_F = RAND\_MAX;

2) Объявление функции main, после чего объявляем все переменные.

int main()  
{  
int i,max,min,N;

3) Получение и проверка данных от пользователя.

printf("Введите размер массива: ");  
scanf("%d",&N);  
printf("Введите минимальный элемент: ");  
scanf("%d", &min);  
printf("Введите максимальный элемент: ");  
scanf("%d", &max);  
  
if (N <= 0 ){  
printf("Ошибка.Количество элементов должно быть больше нуля !");  
return 0;  
}  
if (max < min){  
printf("Ошибка.Максимальный элемент не может быть меньше минимального !");  
return 0;  
}

4) Создание массива рандомных чисел и запись его в файл.

srand(time(NULL));  
FILE\* file = 0;  
file = fopen("/Users/alina/Desktop/first/1 программа/1 программа/textFile.txt", "w");  
for(i=0;i<N;i++){  
fprintf(file, "%lf\n", (rand() / RAND\_MAX\_F) \* ((float)max - (float)min) + min);  
}  
fclose(file);  
  
return 0;  
}

* **Программа 2.**

Вторая программа читает текстовый файл с набором чисел, выводит консольный интерфейс (печать, сортировка, сброс, выход), выполняет выбранные действия.

1. Подключение всех библиотек, с которыми предстоит работать.

[#include](https://vk.com/im?sel=176031583&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <time.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <string.h>

2) Сортировка пузырьком.

void bubbleSort(float\* arrb, int size)  
{  
int current,j,i;  
for(j=1;j<size-1;j++){  
for(i = 0;i < size-j;i++){  
if(arrb[i]>arrb[i+1]){  
current = arrb[i+1];  
arrb[i+1] = arrb[i];  
arrb[i] = current;  
}  
}  
}  
}

3) Быстрая сортировка.

void quickSort(float\* mas, int start, int end)  
{  
int left = start;  
int right = end;  
float pivot = mas[(left + right) / 2];  
while (left <= right)  
{  
while (mas[left] < pivot)  
{  
left++;  
}  
while (mas[right] > pivot)  
{  
right--;  
}  
if (left <= right)  
{  
float buff = mas[left];  
mas[left] = mas[right];  
mas[right] = buff;  
left++;  
right--;  
}  
}  
if (start < right)  
quickSort(mas, start, right);  
if (end > left)  
quickSort(mas, left, end);  
}

4) Сортировка вставкой.

void insertionSort(float\* mas, int size)  
{  
for (int i = 1; i < size; i++)  
{  
for (int j = i; j > 0 && mas[j - 1] > mas[j]; j--)  
{  
float buff = mas[j];  
mas[j] = mas[j - 1];  
mas[j - 1] = buff;  
}  
}  
}

5) Объявление функции outputResult, которая выводит массив.

void outputResult(float\*c,int countFile)  
{  
int i;  
printf("Массив после сортировки:\n");  
for(i=0;i<countFile;i++){  
printf("|%f",c[i]);  
}  
printf("|\n");  
}

**7)** Объявление функцииcountNumbers, которая подсчитывает количество элементов в массиве из файла.

int countNumbers(FILE\* file)  
{  
int count;  
float s;  
  
count = 0;  
while ((fscanf(file, "%f",&s)!=EOF))  
{ if(!file) break;  
count += 1;  
}

**6)** Объявление функции main(). Объявление всех переменных. Считывание файла.

int main()  
{  
int countFile,i,checkbox = 0;  
double startTime, endTime;  
float \*c, \*copy;  
FILE\* file = fopen("/Users/alina/Desktop/first/1 программа/1 программа/textFile.txt", "r+");  
  
countFile = countNumbers(file);  
  
c = (float\*) malloc(countFile\*sizeof(float));  
copy = (float\*) malloc(countFile\*sizeof(float));  
  
rewind(file);  
  
for(i=0;i<countFile;i++)  
{  
fscanf(file, "%g",&c[i]);  
}  
fclose(file);  
  
memcpy(copy, c , sizeof(float) \* countFile);  
  
free(c);

7) Интерфейс программы.

do{  
  
printf("Выберите нужное действие:\n1. Печать\n2. Сортировка\n3. Сброс\n4. Выход\nВыбор:");  
do  
{  
scanf("%d", &checkbox);  
if (checkbox < 1 || checkbox > 4)  
{  
printf("\nВы неверно ввели номер. Введите снова\n\nВыбор:");  
}  
} while (checkbox < 1 || checkbox > 4);  
  
if (checkbox == 1)  
{  
// clearScrs();  
printf("Выбрано 1.Осуществляется печать данных.");  
outputResult(c, countFile);  
}  
else if (checkbox == 2)  
{  
// clearScrs();  
printf("Выбрано 2.Какую сортировку Вы хотите осуществить?\n");  
printf("Выберите нужное действие:\n\n1. Сортировка пузырьком\n2. Быстрая сортировка\n3. Сортировка вставкой\nВыбор:");  
do  
{  
scanf("%d", &checkbox);  
if (checkbox < 1 || checkbox > 3)  
{  
printf("\nВы неверно ввели номер. Введите снова\n\nВыбор:");  
}  
} while (checkbox < 1 || checkbox > 3);  
  
if (checkbox == 1)  
{  
printf("Выбрано 1.Осуществляется сортировка пузырьком.\n");  
startTime = clock();  
bubbleSort(c, countFile);  
endTime = clock();  
printf("Время затраченное на сортировку %lf\n", (endTime - startTime)/1000);  
}  
else if (checkbox == 2)  
{  
printf("Выбрано 2.Осуществляется быстрая сортировка.\n");  
startTime = clock();  
quickSort(c, 0,countFile- 1);  
endTime = clock();  
printf("Время затраченное на сортировку %lf\n", (endTime - startTime)/1000);  
}  
else if (checkbox == 3)  
{  
printf("Выбрано 3.Осуществляется сортировка вставкой.\n");  
startTime = clock();  
insertionSort(c,countFile);  
endTime = clock();  
printf("Время затраченное на сортировку %lf\n", (endTime - startTime)/1000);  
}  
  
}  
else if (checkbox == 3)  
{  
printf("Выбрано 3.Сброс\n");  
memcpy(c, copy , sizeof(float) \* countFile);  
printf("Массив сброшен в исходное состояние\n");  
}  
else if (checkbox == 4)  
{  
return 0;  
}  
} while (checkbox >=1 && checkbox<=4);  
}

**Описание структуры программы.**

Первая и вторая программы состоят из одного модуля int main().

В первой программе там находится код генерации рандомных чисел и запись их в файл.

Во второй программе находится код считывания файла и сортировки чисел из него.

**Описание алгоритмов**

1. Сортировка «пузырьком».

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется перестановка элементов. Проходы по массиву повторяются {\displaystyle N-1} раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырёк в воде — отсюда и название алгоритма).

void bubbleSort(float\* arrb, int size)  
{  
int current,j,i;  
for(j=1;j<size-1;j++){  
for(i = 0;i < size-j;i++){  
if(arrb[i]>arrb[i+1]){  
current = arrb[i+1];  
arrb[i+1] = arrb[i];  
arrb[i] = current;  
}  
}  
}  
}

1. Быстрая Сортировка.

Алгоритм состоит из трёх шагов:

1. Выбрать элемент из массива. Назовём его опорным.
2. Разбиение: перераспределение элементов в массиве таким образом, что элементы, меньшие опорного, помещаются перед ним, а большие или равные - после.
3. Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один элемент или отсутствуют элементы.
4. void quickSort(float\* mas, int start, int end)  
   {  
   int left = start;  
   int right = end;  
   float pivot = mas[(left + right) / 2];  
   while (left <= right)  
   {  
   while (mas[left] < pivot)  
   {  
   left++;  
   }  
   while (mas[right] > pivot)  
   {  
   right--;  
   }  
   if (left <= right)  
   {  
   float buff = mas[left];  
   mas[left] = mas[right];  
   mas[right] = buff;  
   left++;  
   right--;  
   }  
   }  
   if (start < right)  
   quickSort(mas, start, right);  
   if (end > left)  
   quickSort(mas, left, end);  
   }

3.Сортировка вставкой.

[Алгоритм сортировки](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B8), в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов.

void insertionSort(float\* mas, int size)  
{  
for (int i = 1; i < size; i++)  
{  
for (int j = i; j > 0 && mas[j - 1] > mas[j]; j--)  
{  
float buff = mas[j];  
mas[j] = mas[j - 1];  
mas[j - 1] = buff;  
}  
}  
}

**Эксперименты**

Проведём эксперимент. Узнаем какая сортировка работает быстрее с масивом из рандомных чисел.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип сортировки** | **Сложность** | **Размер входных данных** | **Время работы, с** |
| Пузырьком | О(n²) | 1000 | 3.248000 |
| Быстрая | O(n log n) | 1000 | 0.153000 |
| Вставками | О(n²) | 1000 | 2.010000 |
| Пузырьком | О(n²) | 5000 | 74.068000 |
| Быстрая | O(n log n) | 5000 | 0.925000 |
| Вставками | О(n²) | 5000 | 41.353000 |
| Пузырьком | О(n²) | 10000 | 373.245000 |
| Быстрая | O(n log n) | 10000 | 1.810000 |
| Вставками | О(n²) | 10000 | 141.836000 |

Таблица 1. Результат эксперимента.

Для проведения эксперимента я взяла массивы из 1000, 5000 и 10000 элементов.

В таблице указано время, за которое была произведена каждая сортировка.

Эксперимент показал, что самая быстрая сортировка- это “быстрая сортировка”.

**Заключение**

У меня получилось создать 2 программы на языке С, которые выполняют следующее техническое задание:

“Реализовать сортировки массивов данных (тип данных определяется преподавателем(float)) задаваемых: обязательно случайно, дополнительно с клавиатуры или из файла.

Реализовать сортировки: пузырьком, вставкой, быстрая. Сравнить время работы, сделать выводы.

Первая программа создает текстовый файл с записанными в него числами. Программа принимает количество чисел n, максимальное и минимальное значение.

Вторая программа читает текстовый файл с набором чисел, выводит консольный интерфейс (печать, сортировка, сброс, выход), выполняет выбранные действия.”

В процессе эксперимента я выяснила, что из всех трёх сортировок, самая быстрая – “быстрая сортировка”.

**Список литературы**

1. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. — СПб.: Питер, 2003.
2. Брайан Керниган, Деннис Ритчи. Язык программирования C. — Москва: Вильямс, 2015.
3. Лекции Лебедева Ильи Геннадьевича и Сысоева Александра Владимировича, 2021.

**Приложение 1.**

[#include](https://vk.com/im?sel=176031583&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <time.h>  
const float RAND\_MAX\_F = RAND\_MAX;  
  
int main()  
{  
int i,max,min,N;  
  
printf("Введите размер массива: ");  
scanf("%d",&N);  
printf("Введите минимальный элемент: ");  
scanf("%d", &min);  
printf("Введите максимальный элемент: ");  
scanf("%d", &max);  
  
if (N <= 0 ){  
printf("Ошибка.Количество элементов должно быть больше нуля !");  
return 0;  
}  
if (max < min){  
printf("Ошибка.Максимальный элемент не может быть меньше минимального !");  
return 0;  
}  
  
srand(time(NULL));  
FILE\* file = 0;  
file = fopen("/Users/alina/Desktop/first/1 программа/1 программа/textFile.txt", "w");  
for(i=0;i<N;i++){  
fprintf(file, "%lf\n", (rand() / RAND\_MAX\_F) \* ((float)max - (float)min) + min);  
}  
fclose(file);  
  
return 0;  
}

**Приложение 2.**

[#include](https://vk.com/im?sel=176031583&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <time.h>  
[#include](https://vk.com/im?sel=176031583&st=%23include) <string.h>  
  
void bubbleSort(float\* arrb, int size)  
{  
int current,j,i;  
for(j=1;j<size-1;j++){  
for(i = 0;i < size-j;i++){  
if(arrb[i]>arrb[i+1]){  
current = arrb[i+1];  
arrb[i+1] = arrb[i];  
arrb[i] = current;  
}  
}  
}  
}  
  
void quickSort(float\* mas, int start, int end)  
{  
int left = start;  
int right = end;  
float pivot = mas[(left + right) / 2];  
while (left <= right)  
{  
while (mas[left] < pivot)  
{  
left++;  
}  
while (mas[right] > pivot)  
{  
right--;  
}  
if (left <= right)  
{  
float buff = mas[left];  
mas[left] = mas[right];  
mas[right] = buff;  
left++;  
right--;  
}  
}  
if (start < right)  
quickSort(mas, start, right);  
if (end > left)  
quickSort(mas, left, end);  
}  
  
void insertionSort(float\* mas, int size)  
{  
for (int i = 1; i < size; i++)  
{  
for (int j = i; j > 0 && mas[j - 1] > mas[j]; j--)  
{  
float buff = mas[j];  
mas[j] = mas[j - 1];  
mas[j - 1] = buff;  
}  
}  
}  
void outputResult(float\* c,int countFile)  
{  
int i;  
printf("Вывод результата:\n");  
for(i=0;i<countFile;i++){  
printf("%f\n",c[i]);  
}  
printf("\n");  
}  
  
int countNumbers(FILE\* file)  
{  
int count;  
float s;  
  
count = 0;  
while ((fscanf(file, "%f",&s)!=EOF))  
{ if(!file) break;  
count += 1;  
}  
  
return count;  
}  
  
//void clearScrs()  
//{  
//system("clear");  
//}  
  
int main()  
{  
int countFile,i,checkbox = 0;  
double startTime, endTime;  
float \*c, \*copy;  
FILE\* file = fopen("/Users/alina/Desktop/first/1 программа/1 программа/textFile.txt", "r+");  
  
countFile = countNumbers(file);  
  
c = (float\*) malloc(countFile\*sizeof(float));  
copy = (float\*) malloc(countFile\*sizeof(float));  
  
rewind(file);  
  
for(i=0;i<countFile;i++)  
{  
fscanf(file, "%g",&c[i]);  
}  
fclose(file);  
  
memcpy(copy, c , sizeof(float) \* countFile);  
  
free(c);  
  
  
  
// clearScrs();  
  
do{  
  
printf("Выберите нужное действие:\n1. Печать\n2. Сортировка\n3. Сброс\n4. Выход\nВыбор:");  
do  
{  
scanf("%d", &checkbox);  
if (checkbox < 1 || checkbox > 4)  
{  
printf("\nВы неверно ввели номер. Введите снова\n\nВыбор:");  
}  
} while (checkbox < 1 || checkbox > 4);  
  
if (checkbox == 1)  
{  
// clearScrs();  
printf("Выбрано 1.Осуществляется печать данных.");  
outputResult(c, countFile);  
}  
else if (checkbox == 2)  
{  
// clearScrs();  
printf("Выбрано 2.Какую сортировку Вы хотите осуществить?\n");  
printf("Выберите нужное действие:\n\n1. Сортировка пузырьком\n2. Быстрая сортировка\n3. Сортировка вставкой\nВыбор:");  
do  
{  
scanf("%d", &checkbox);  
if (checkbox < 1 || checkbox > 3)  
{  
printf("\nВы неверно ввели номер. Введите снова\n\nВыбор:");  
}  
} while (checkbox < 1 || checkbox > 3);  
  
if (checkbox == 1)  
{  
printf("Выбрано 1.Осуществляется сортировка пузырьком.\n");  
startTime = clock();  
bubbleSort(c, countFile);  
endTime = clock();  
printf("Время затраченное на сортировку %lf\n", (endTime - startTime)/1000);  
}  
else if (checkbox == 2)  
{  
printf("Выбрано 2.Осуществляется быстрая сортировка.\n");  
startTime = clock();  
quickSort(c, 0,countFile- 1);  
endTime = clock();  
printf("Время затраченное на сортировку %lf\n", (endTime - startTime)/1000);  
}  
else if (checkbox == 3)  
{  
printf("Выбрано 3.Осуществляется сортировка вставкой.\n");  
startTime = clock();  
insertionSort(c,countFile);  
endTime = clock();  
printf("Время затраченное на сортировку %lf\n", (endTime - startTime)/1000);  
}  
  
}  
else if (checkbox == 3)  
{  
printf("Выбрано 3.Сброс\n");  
memcpy(c, copy , sizeof(float) \* countFile);  
printf("Массив сброшен в исходное состояние\n");  
}  
else if (checkbox == 4)  
{  
return 0;  
}  
} while (checkbox >=1 && checkbox<=4);  
}