Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №15**

Дисциплина: основы алгоритмизации и программирования

Тема: Методы внутренней сортировки массивов: быстрая сортировка, сортировка подсчетом

Вариант: 25

Выполнил работу

студент группы РИС-20-1б

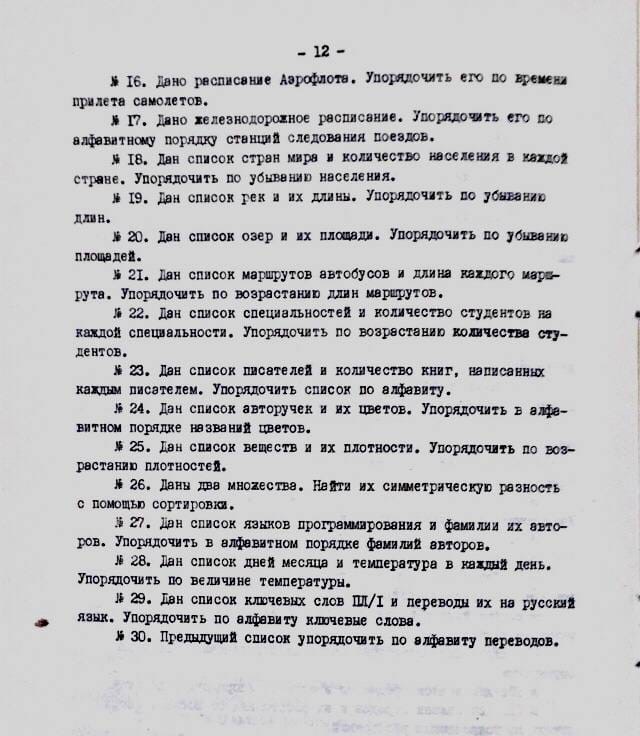
Рябов Никита Андреевич

Проверила

Доцент кафедры ИТАС Полякова О.А.

Пермь, 2021

**Постановка задачи**



**Анализ задачи**

*Какие предстоит выполнить действия.*

Основная функция считывает длину массива и сам массив. Получает результаты от функций и выводит их на экран.

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int len=-1;

cout << "Введите кол-во записей\n";

while (len<2) cin>> len;

string chem1[len];

float P1[len];

string chem2[len];

float P2[len];

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

{

cout << "Введите вещество " <<i+1<<endl;

cin >> chem1[i];

chem2[i]=chem1[i];

cout << "Введите плотность " <<i+1<<endl;

cin >> P1[i];

P2[i]=P1[i];

}

SortShell(P1,chem1,len);

SortChoar(P2,chem2,0,len-1);

cout << "Результат сортировки Шелла\n";

for (int i=0;i<len;i++) cout << chem1[i] << " " << P1[i] << endl;

cout << "Результат сортировки Хоара\n";

for (int i=0;i<len;i++) cout << chem2[i] << " " << P2[i] << endl;

system("pause");

}

Функция сортировки Шелла ничего не возвращает, так как напрямую изменяет массив. Во внешнем цикле перебираются все возможные значения d от len/2 до 1. Внутренний цикл перебирает все возможные выборки элементов. В каждой выборке элементы отстоят друг от друга на d. Еще 2 внутренних цикла представляют собой обычную сортировку вставками.

void SortShell(float P[],string chem[],int len)

{

for(int d=len/2; d!=0; d/=2)

{

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

{

for (int j=i; j+d<len; j+=d)

{

int num=j+d;

while (num>=i+d && P[num-d]>P[num])

{

swap(P[num-d],P[num]);

swap(chem[num-d],chem[num]);

num-=d;

}

}

}

}

}

Сортировка Хоара. Выбирается средний элемент. Создаются 3 стека: для меньших, равных и больших элементов. Они записываются последовательно в указанный промежуток массива. Функция рекурсивно повторяется относительно отрезков массива с меньшими и большими элементами. Также создаются еще 3 стека, чтобы одновременно сортировать и массив названий веществ.

void SortChoar(float P[],string chem[],int first,int last)

{

if (last>first)

{

float x=P[(last-first)/2];

stack <float> STACKminP;

stack <float> STACKmaxP;

stack <string> STACKminC;

stack <string> STACKmaxC;

stack <float> STACKcenP;

stack <string> STACKcenC;

for (int i=first;i<=last;i++)

{

if (P[i]<x)

{

STACKminP.push(P[i]);

STACKminC.push(chem[i]);

}

else if (P[i]>x)

{

STACKmaxP.push(P[i]);

STACKmaxC.push(chem[i]);

}

else

{

STACKcenP.push(P[i]);

STACKcenC.push(chem[i]);

}

}

int num=STACKminP.size();

int num2=STACKcenP.size();

for (int i=first;i<first+num;i++)

{

P[i]=STACKminP.top();

STACKminP.pop();

chem[i]=STACKminC.top();

STACKminC.pop();

}

for (int i=first+num;i<first+num+num2;i++)

{

P[i]=STACKcenP.top();

STACKcenP.pop();

chem[i]=STACKcenC.top();

STACKcenC.pop();

}

for (int i=first+num+num2;i<=last;i++)

{

P[i]=STACKmaxP.top();

STACKmaxP.pop();

chem[i]=STACKmaxC.top();

STACKmaxC.pop();

}

SortChoar(P,chem,first,first+num-1);

SortChoar(P,chem,first+num+num2,last);

}

}

*С каким типом данных действуем*

Вещественные числа, обозначающие плотность, и строки, обозначающие названия веществ.

*В каком виде представлены данные*

Массивы float и string соответственно

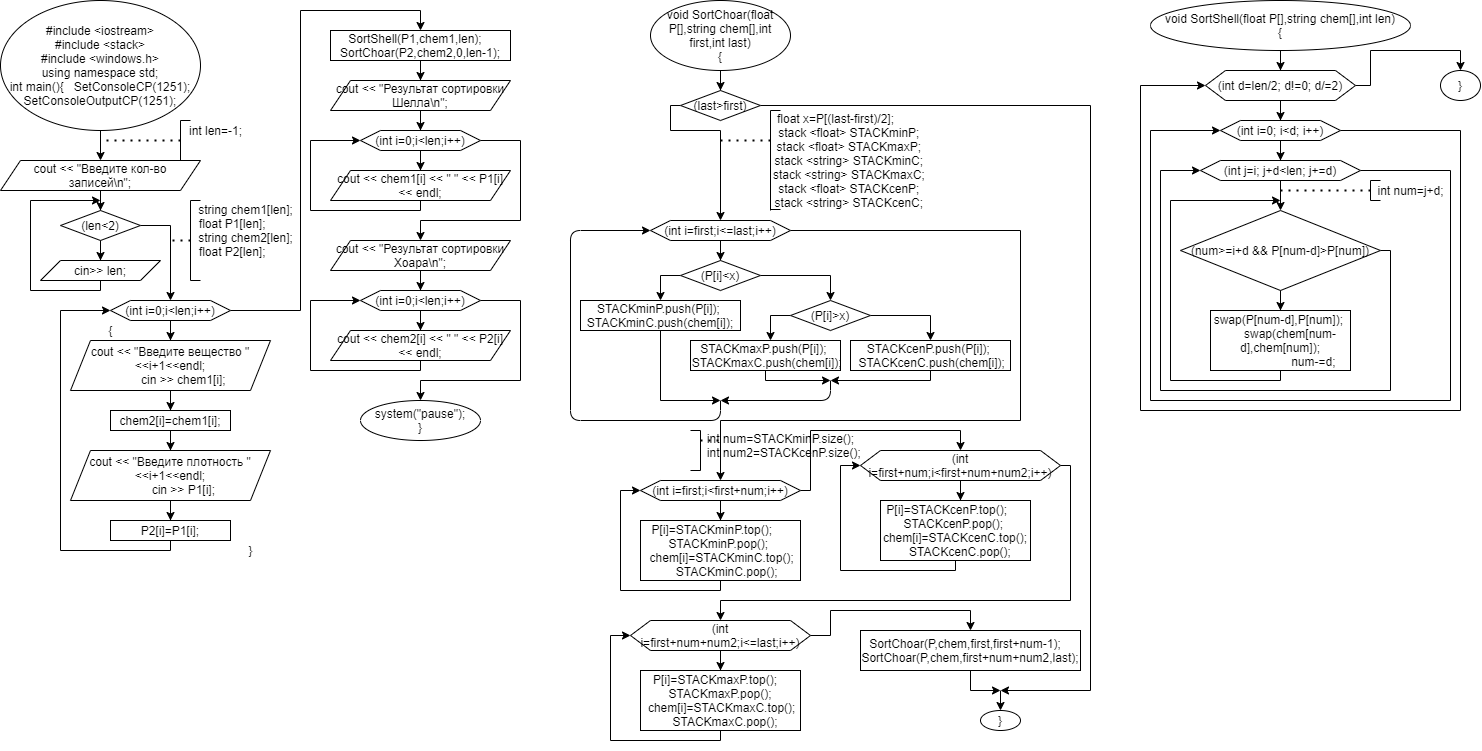
*Какие поля*

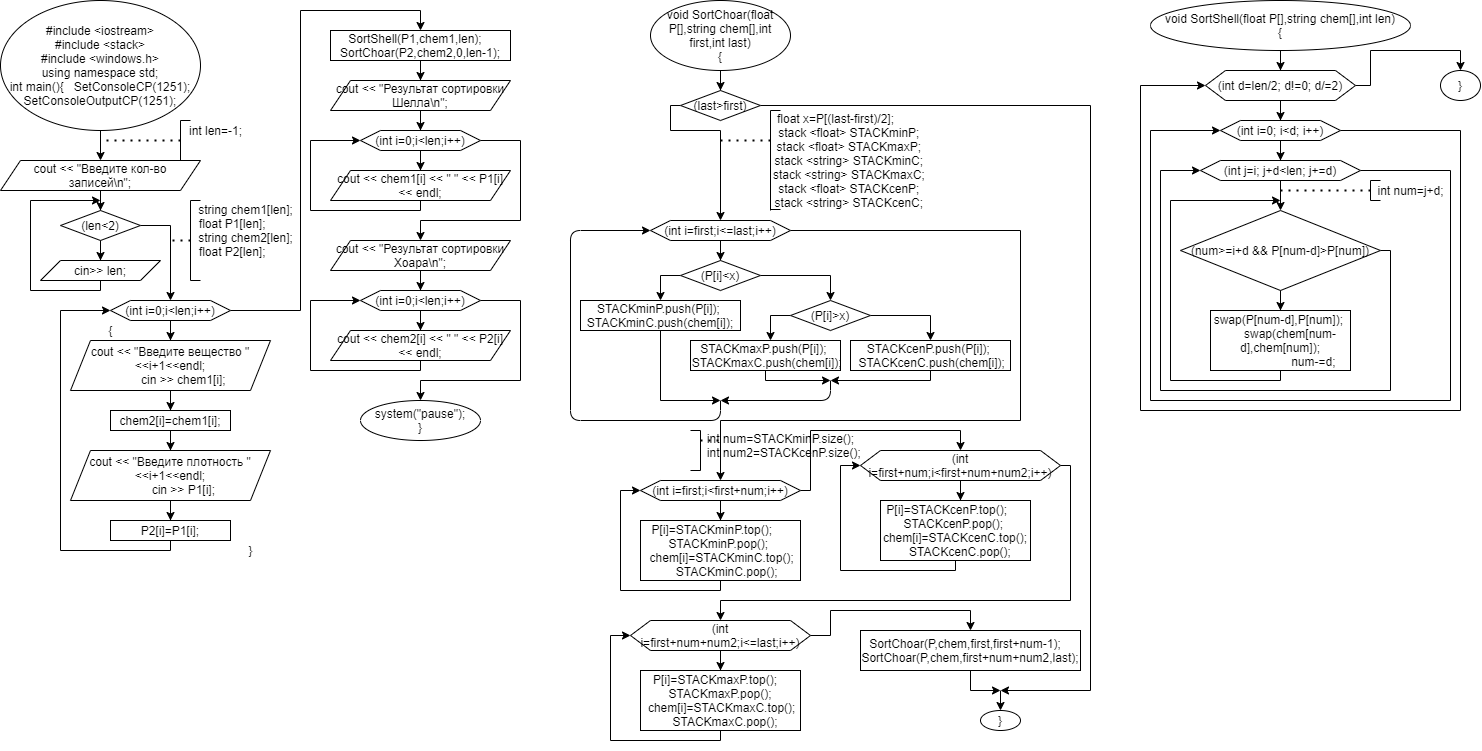
Структуры в данной программе отсутствуют

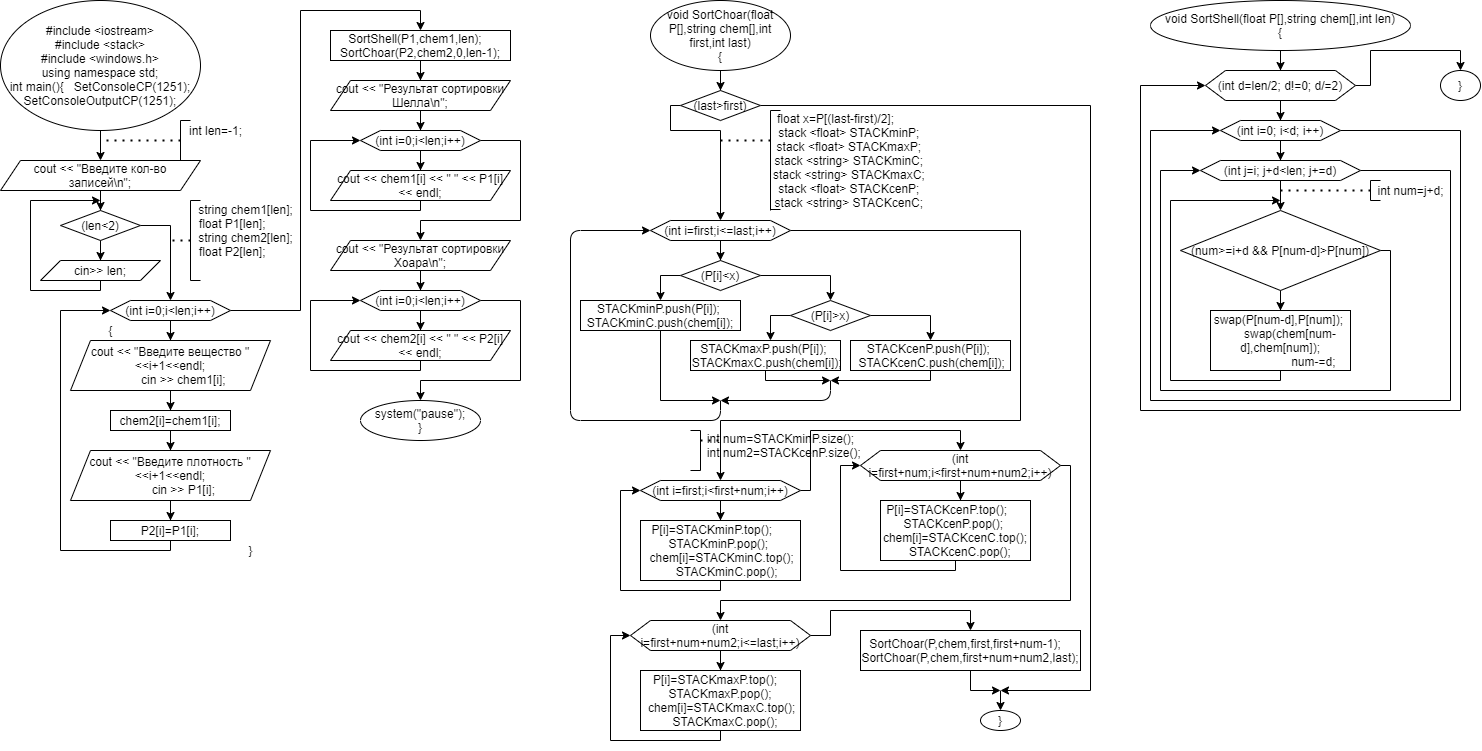
*Операторы ввода/вывода*

В программе используются операторы cin/cout.

**Блок-схема**

****

****

****

**Код**

#include <iostream>

#include <stack>

#include <windows.h>

using namespace std;

void SortShell(float P[],string chem[],int len)

{

for(int d=len/2; d!=0; d/=2)

{

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

{

for (int j=i; j+d<len; j+=d)

{

int num=j+d;

while (num>=i+d && P[num-d]>P[num])

{

swap(P[num-d],P[num]);

swap(chem[num-d],chem[num]);

num-=d;

}

}

}

}

}

void SortChoar(float P[],string chem[],int first,int last)

{

if (last>first)

{

float x=P[(last-first)/2];

stack <float> STACKminP;

stack <float> STACKmaxP;

stack <string> STACKminC;

stack <string> STACKmaxC;

stack <float> STACKcenP;

stack <string> STACKcenC;

for (int i=first;i<=last;i++)

{

if (P[i]<x)

{

STACKminP.push(P[i]);

STACKminC.push(chem[i]);

}

else if (P[i]>x)

{

STACKmaxP.push(P[i]);

STACKmaxC.push(chem[i]);

}

else

{

STACKcenP.push(P[i]);

STACKcenC.push(chem[i]);

}

}

int num=STACKminP.size();

int num2=STACKcenP.size();

for (int i=first;i<first+num;i++)

{

P[i]=STACKminP.top();

STACKminP.pop();

chem[i]=STACKminC.top();

STACKminC.pop();

}

for (int i=first+num;i<first+num+num2;i++)

{

P[i]=STACKcenP.top();

STACKcenP.pop();

chem[i]=STACKcenC.top();

STACKcenC.pop();

}

for (int i=first+num+num2;i<=last;i++)

{

P[i]=STACKmaxP.top();

STACKmaxP.pop();

chem[i]=STACKmaxC.top();

STACKmaxC.pop();

}

SortChoar(P,chem,first,first+num-1);

SortChoar(P,chem,first+num+num2,last);

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int len=-1;

cout << "Введите кол-во записей\n";

while (len<2) cin>> len;

string chem1[len];

float P1[len];

string chem2[len];

float P2[len];

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

{

cout << "Введите вещество " <<i+1<<endl;

cin >> chem1[i];

chem2[i]=chem1[i];

cout << "Введите плотность " <<i+1<<endl;

cin >> P1[i];

P2[i]=P1[i];

}

SortShell(P1,chem1,len);

SortChoar(P2,chem2,0,len-1);

cout << "Результат сортировки Шелла\n";

for (int i=0;i<len;i++) cout << chem1[i] << " " << P1[i] << endl;

cout << "Результат сортировки Хоара\n";

for (int i=0;i<len;i++) cout << chem2[i] << " " << P2[i] << endl;

system("pause");

}

**Скриншоты**

****