**ЛАБОРАТОРНАЯ РАБОТА №7**

**«ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД СТРУКТУРАМИ И НАД МАССИВАМИ СТРУКТУР»**

**Цель работы**

Изучение способов описания структур данных на языке С.

**Задания**

1. Написать программу в соответствии с заданием из методических указаний (вариант 12):

Описать структуру с именем product, содержащую следующие поля: порядковый номер, название продукта, дата изготовления, срок годности, производитель, цена. Данные сортировать по полю «название продукта». Выполнить поиск индекса продукта заданного пользователем производителя с наибольшей ценой.

1. Разработать структурные схемы алгоритма для всех функций решения задачи;
2. Разработать тестовые примеры и выполнить тестирование и отладку написанной программы;

**Текст программы**

#include<iostream>

#include<string.h>

**using namespace** std;

**int** i, j, k=0; //глобальные параметры

//структура задания даты

**struct** date

{

**int** day, //день

month, //месяц

year; //год

};

//структура описания продукта

**struct** product

{

**int** num, //порядковый номер

price; //цена

date di, //дата изготовления

sg; //срок годности

**char** name[30], //название продукта

mdby[30]; //производитель

};

//объявление функций

**void** menu(product \*a, **int** N);

**void** input(product \*a, **int** N);

**void** sort(product \*a, **int** N);

**void** output(product \*a, **int** N);

**int** search(product \*a, **int** N, **char** who[]);

//меню программы

**void** menu(product \*a, **int** N)

{

**int** x;

cout<<"\nProgram menu:\n\n";

cout<<"To enter array enter 1\n";

cout<<"To sort array enter 2\n";

cout<<"To search in array enter 3\n";

cout<<"To output array enter 4\n";

cout<<"To close program enter 5\n\n";

cin>>x;

**switch**(x) {

**case** 1: input(a, N); menu(a, N); **break**;

**case** 2: sort(a, N); menu(a, N); **break**;

**case** 3:

**char** who[30];

cin.clear();

cin.ignore();

cout<<"\nSearch max price of -> "; //ввод производителя

cin.getline(who,30);

**if** (search(a, N, who)!=-1)

//вывод индекса макс. по цене продукта производителя

cout<<"Index of max price - "<<search(a, N, who)<<"\n";

**else**

//сообщение о ненахождении продуктов от производителя

cout<<"No products of this manufacturer\n";

menu(a, N); **break**;

**case** 4: output(a, N); menu(a, N); **break**;

**case** 5: **return**;

**default**: cout<<"\nError: no related command\n"; menu(a, N);

}

}

//функция ввода продуктов

**void** input(product \*a, **int** N)

{

**for** (i=0;i<N;i++)

{

cout<<"\n";

cout<<i+1<<") Product:\n";

cout<<" number -> "; //ввод номера продукта

cin>>a[i].num;

cout<<" name -> "; //ввод названия

cin.clear();

cin.ignore();

cin.getline(a[i].name,30);

cout<<" date of manufacture:\n"; //ввод даты изготовления

cout<<" day -> "; //день

cin>>a[i].di.day;

cout<<" month -> "; //месяц

cin>>a[i].di.month;

cout<<" year -> "; //год

cin>>a[i].di.year;

cout<<" shelf life:\n"; //ввод срока годности

cout<<" day -> "; //день

cin>>a[i].sg.day;

cout<<" month -> "; //месяц

cin>>a[i].sg.month;

cout<<" year -> "; //год

cin>>a[i].sg.year;

cout<<" manufacturer -> "; //ввод производителя

cin.clear();

cin.ignore();

cin.getline(a[i].mdby,30);

cout<<" price -> "; //ввод цены продукта

cin>>a[i].price;

k=1;

}

}

//функция сортировки массива продуктов по названиям (алфавит)

**void** sort(product \*a, **int** N)

{

**if** (!k) //проверка на наличие массива

{

cout<<"\nNo array entered\n";

**return**;

}

product temp; //переменная для обмена

**for** (i=0;i<N-1;i++) //сортировка прямым обменом

**for** (j=1;j<N-i;j++)

{

**if** ((strcmp(a[j-1].name,a[j].name))>0) //сравнение названий

{

temp=a[j]; //изменение продуктов местами

a[j]=a[j-1];

a[j-1]=temp;

}

}

cout<<"\nArray sorted\n";

}

//функция вывода продуктов

**void** output(product \*a, **int** N)

{

**if** (!k) //проверка на наличие массива

{

cout<<"\nNo array entered\n";

**return**;

}

cout<<"\nOutput of array\n";

**for** (i=0;i<N;i++)

{

cout<<"\n";

cout<<i+1<<") Product:\n";

cout<<" number -> "; //вывод номера продукта

cout<<a[i].num<<"\n";

cout<<" name -> "; //вывод названия продукта

cout<<a[i].name<<"\n";

cout<<" date of manufacture: "; //вывод даты изготовления

cout<<a[i].di.day<<".";

cout<<a[i].di.month<<".";

cout<<a[i].di.year<<"\n";

cout<<" shelf life: "; //вывод срока годности

cout<<a[i].sg.day<<".";

cout<<a[i].sg.month<<".";

cout<<a[i].sg.year<<"\n";

cout<<" manufacturer -> "; //вывод производителя

cout<<a[i].mdby<<"\n";

cout<<" price -> "; //вывод цены

cout<<a[i].price<<"\n";

}

}

//функция поиска индекса макс. по цене продукта заданного производителя

**int** search(product \*a, **int** N, **char** who[])

{

**if** (!k) //проверка на наличие массива

{

cout<<"\nNo array entered\n";

**return**;

}

**int** max=-1;

//поиск индекса первого продукта данного производителя

**for** (i=0;i<N;i++)

**if** (strcmp(a[i].mdby,who)==0)

{

max=i;

**break**;

}

**if** (max>=0)

{

//поиск индекса максимального по цене продукта производителя

**for** (i=max+1;i<N;i++)

**if** ((strcmp(a[i].mdby,who)==0) && (a[i].price>a[max].price))

max=i;

**return**(max+1);

}

**else** **return**(-1);

}

main()

{

**int** N;

cout<<"N of products -> "; //ввод количества продуктов

cin>>N;

product \*a=**new** product [N]; //выделение памяти под массив

menu(a, N); //вызов меню

cout<<"\nProgram ended";

**delete**[]a; //возвращение памяти из под массива

}

**Структурная схема программы**

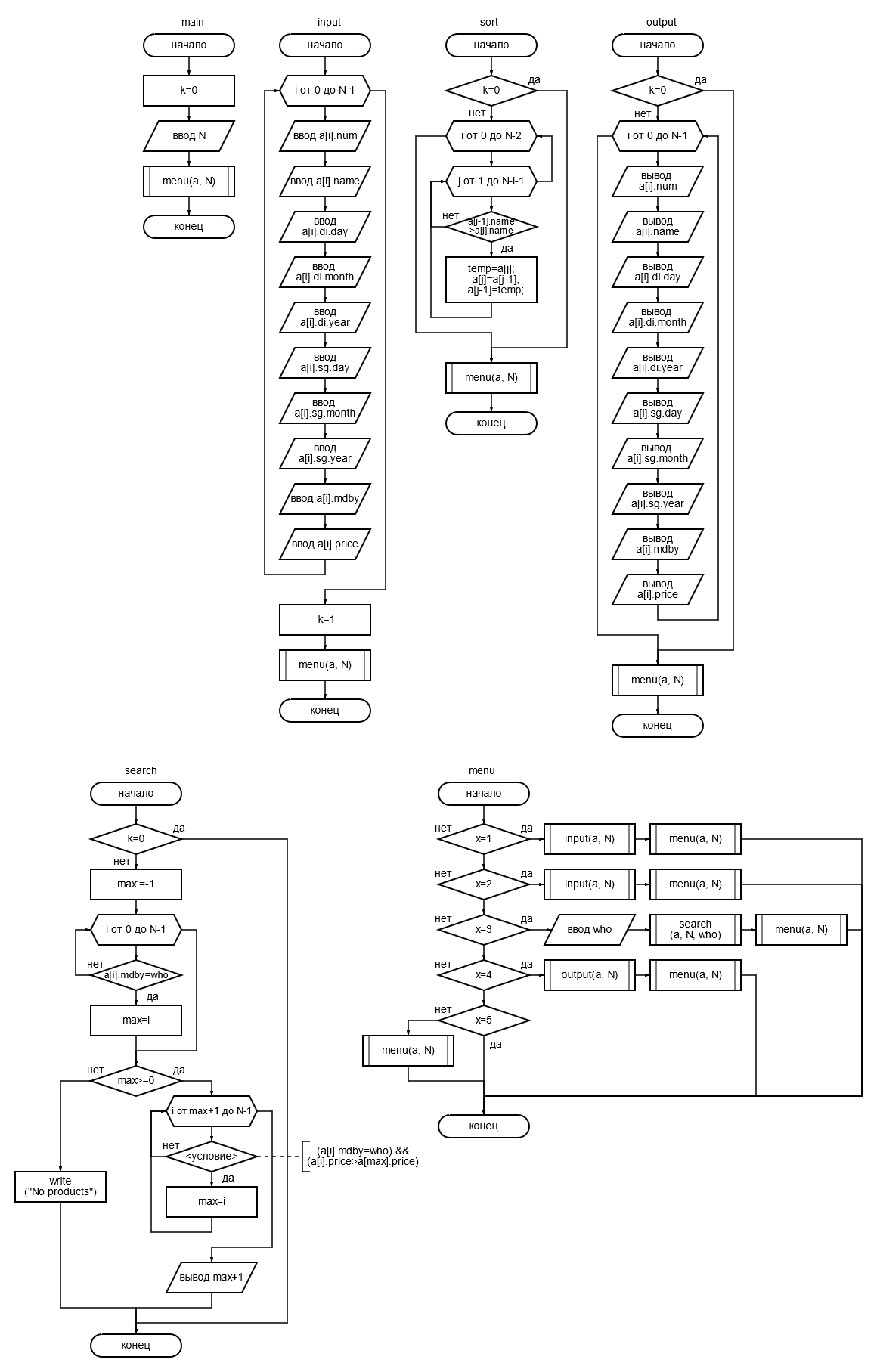


Рисунок 1 – Структурная схема функций программы

**Тестовые примеры**

N=3. Проверка того, как ведёт себя программа при вводе в меню номера, которому не соответствует ни одной команды. Ввод 6:

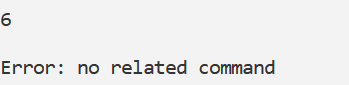


Рисунок 2 – Тестирование default-ветви

Проверка того, как ведёт себя программа при попытке вывода массива в случае, если он ещё не введён. Ввод 4:



Рисунок 3 – Тестирование ошибки для вывода массива

Проверка того, как ведёт себя программа при попытке сортировки массива в случае, если он ещё не введён. Ввод 2:



Рисунок 4 – Тестирование ошибки для сортировки массива

Проверка того, как ведёт себя программа при попытке поиска в массиве в случае, если он ещё не введён. Ввод 3, who=“Q”:

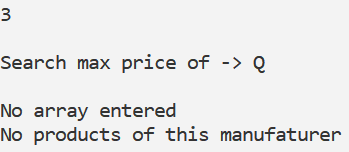


Рисунок 5 – Тестирование ошибки для поиска внутри массива

Проверка работы ввода массива. Ввод 1:

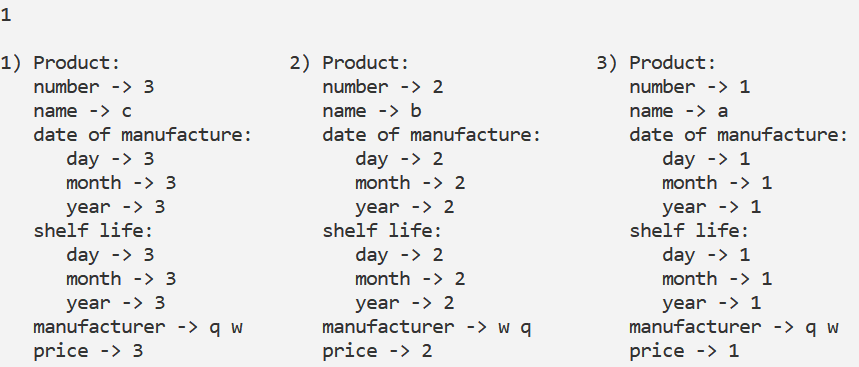


Рисунок 6 – Ввод продуктов в массив

Проверка сортировки массива. Ввод 2, затем 4:



Рисунок 7 – Сортировка массива

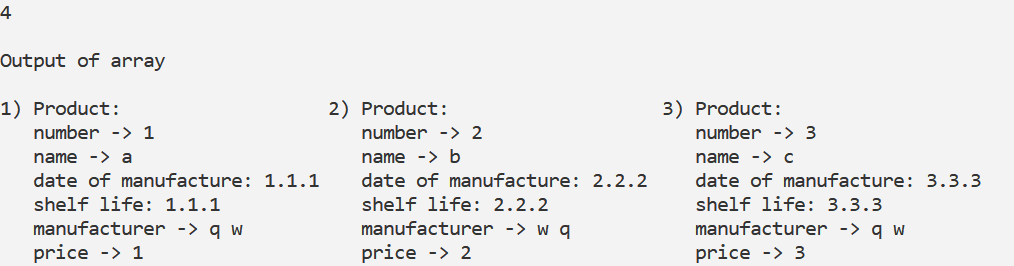


Рисунок 8 – Вывод отсортированного массива

Проверка поиска внутри массива. Ввод 3, who= “q w”:

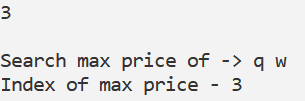


Рисунок 9 – Поиск внутри массива

Проверка сообщения о ненахождении ни одного продукта введённого производителя. Ввод 3, who= “qwerty”:

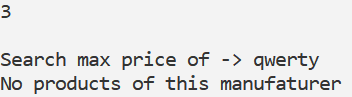


Рисунок 10 – Сообщение о ненахождении нужных продуктов

Проверка окончания работы программы. Ввод 5:

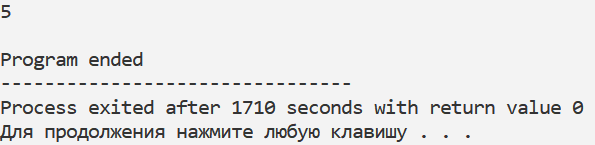


Рисунок 11 – Конец работы программы

На основании выведенных результатов можно сделать вывод о том, что алгоритм работает верно и написанная программа справляется со своей задачей.

**Вывод**

В ходе работы были изучены способы написания и применения структур в языках C/C++. Результатом работы стала программа на языке C++, производящая ввод, сортировку и вывод динамического массива структур, а также поиск внутри него.