Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №10**

**По дисциплине ОАиП за 2 семестр**  
**Тема: «Структуры, перечисления, объединения»**

**Выполнил:**

Студент группы ПО-6(1)  
 1-го курса

Мартынович Даниил

**Проверила:**

Хацкевич М.В.

Брест 2021

**Цель работы:** изучить синтаксис и правила работы со структурами. Реализовать программу с применением структур, перечислений и объединений.

Создать тип структуры согласно варианту, организовать поля этой структуры, создать массив структур, содержащий информацию согласно варианту индивидуального задания. Реализовать работу с массивом структур через меню: ввод данных в массив, вывод содержимого массива на экран, сортировка по одному полю, удаления записи по заданному значению поля, выборка записей согласно индивидуальному заданию.

**Вариант 18**

Структура содержит следующие поля: название процессора, внутренняя частота процессора, количество ядер, объем кэш-памяти. Написать функцию для нахождения двухъядерного процессора с самой низкой внутренней частотой.

#include<iostream>

#include<string>

#include<stdio.h>

#include<ctime>

#include<utility>

#include <iomanip>

using namespace std;

struct PC

{

char processor[100];

float GHZ;

int Yadra;

int cache\_memory;

union

{

int map;

int crash;

};

};

enum vid{Laptop,PC};

void VvodStruct(struct PC \*compon,int \*size);

void VivodStruct(struct PC \*compon, int\* size);

void Sortirovka(struct PC \*compon, int\* size);

void DeleteStroks( struct PC \*compon, int\* size);

void PoiskStroks( struct PC \*compon, int\* size);

int main()

{

setlocale(LC\_ALL, "ru");

srand(time(0));

int size = 10;

struct PC\* compon = new struct PC[size];

int a;

cout << setw(10) << "МЕНЮ" << endl;

do {

cout << "\t1 - ввод элементов в структуру\t2 - вывод структуры на экран\t\n\t3 - сортировка по полю\t4 - удаление выбранной строки\t\n\t5 - функция\t6 - exit\t\n";

cin >> a;

switch (a)

{

case 1:system("cls"); VvodStruct(compon, &size); system("cls"); break;

case 2:VivodStruct(compon, &size); system("pause"); system("cls"); break;

case 3:Sortirovka(compon, &size); system("pause"); system("cls"); break;

case 4:DeleteStroks(compon, &size); system("pause"); system("cls"); break;

case 5:PoiskStroks(compon, &size); system("pause"); system("cls"); break;

}

} while (a != 6);

delete[]compon;

system("pause");

return 0;

}

void VvodStruct(struct PC \*compon, int\* size)

{

////////ВВОД ЭЛЕМЕНТОВ СТРУКТУРЫ

int a;

bool vod = false;

do

{

if (\*size == 10)

{

\*size = 0;

}

cout << "Введите названия процессоров: ";

cin >> compon[\*size].processor;

cout << "Введите внутренняя частота процессоров(ГГц): ";

cin >> compon[\*size].GHZ;

cout << "Введите Количества ядер: ";

cin >> compon[\*size].Yadra;

cout << "Введите Объемы кэш-памяти(Мб): ";

cin >> compon[\*size].cache\_memory;

\*size = \*size + 1;

system("cls");

cout << "Продолжить ввод структур\n1 - Да\n2 - Нет" << endl;

cin >> a;

switch (a)

{

case 1:vod = false; break;

case 2:vod = true; break;

}

system("cls");

} while (vod==false);

cout << endl;

}

void VivodStruct(struct PC \*compon, int\* size)

{

int stroka = 0;

cout << "№\tНаз\tГГц\tЯд\tМб" << endl;

for (int i = 0; i < \*size; i++)

{

cout << stroka++ << " \t" << compon[i].processor << " \t" << compon[i].GHZ << " \t" << compon[i].Yadra << " \t" << compon[i].cache\_memory << endl;

}

cout << endl;

}

void Sortirovka(struct PC \*compon, int\* size)

{

///////////СОРТИРОВКА

cout << "Структура отсортирована по внутренней часоте процессора (ГГц)" << endl;

cout << endl;

for (int i = 0; i < \*size; i++)

{

for (int j = 0; j < \*size; j++)

{

if (compon[i].GHZ < compon[j].GHZ)

{

swap(compon[i], compon[j]);

}

}

}

///////////СОРТИРОВКА

}

void DeleteStroks(struct PC \*compon, int\* size)

{

int stroka = 0;

cout << "№\tНаз\tГГц\tЯд\tМб" << endl;

for (int i = 0; i < \*size; i++)

{

cout << stroka++ << " \t" << compon[i].processor << " \t" << compon[i].GHZ << " \t" << compon[i].Yadra << " \t" << compon[i].cache\_memory << endl;

}

cout << endl;

int del;

cout << "Введите № строки для удаления: ";

cin >> del;

cout << endl;

\*size = \*size - 1;

struct PC\* newCompon = new struct PC[\*size];

for (int i = 0; i < \*size; i++)

{

if (i != del)

{

newCompon[i] = compon[i];

}

else

{

newCompon[i] = compon[i + 1];

}

}

for (int i = 0; i < \*size; i++)

{

compon[i] = newCompon[i];

}

delete[]newCompon;

}

void PoiskStroks(struct PC \*compon, int\* size)

{

int min;

for (size\_t i = 0; i < \*size; i++)

{

if (compon[i].Yadra == 2)

{

min = compon[i].GHZ;

break;

}

}

for (size\_t i = 0; i < \*size; i++)

{

if (compon[i].Yadra == 2)

{

if (compon[i].GHZ < min)

{

min = compon[i].GHZ;

}

}

}

for (size\_t i = 0; i < \*size; i++)

{

if (compon[i].Yadra == 2)

{

if (min == compon[i].GHZ)

{

min = i;

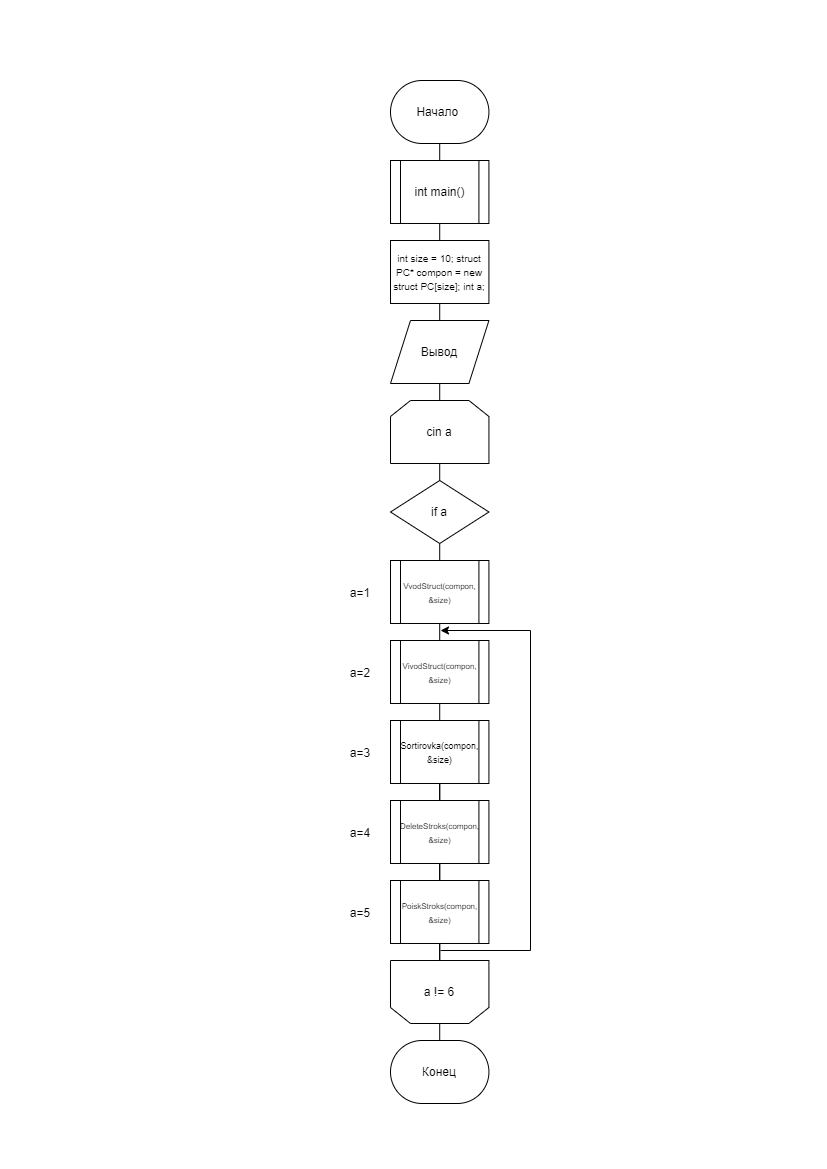
}

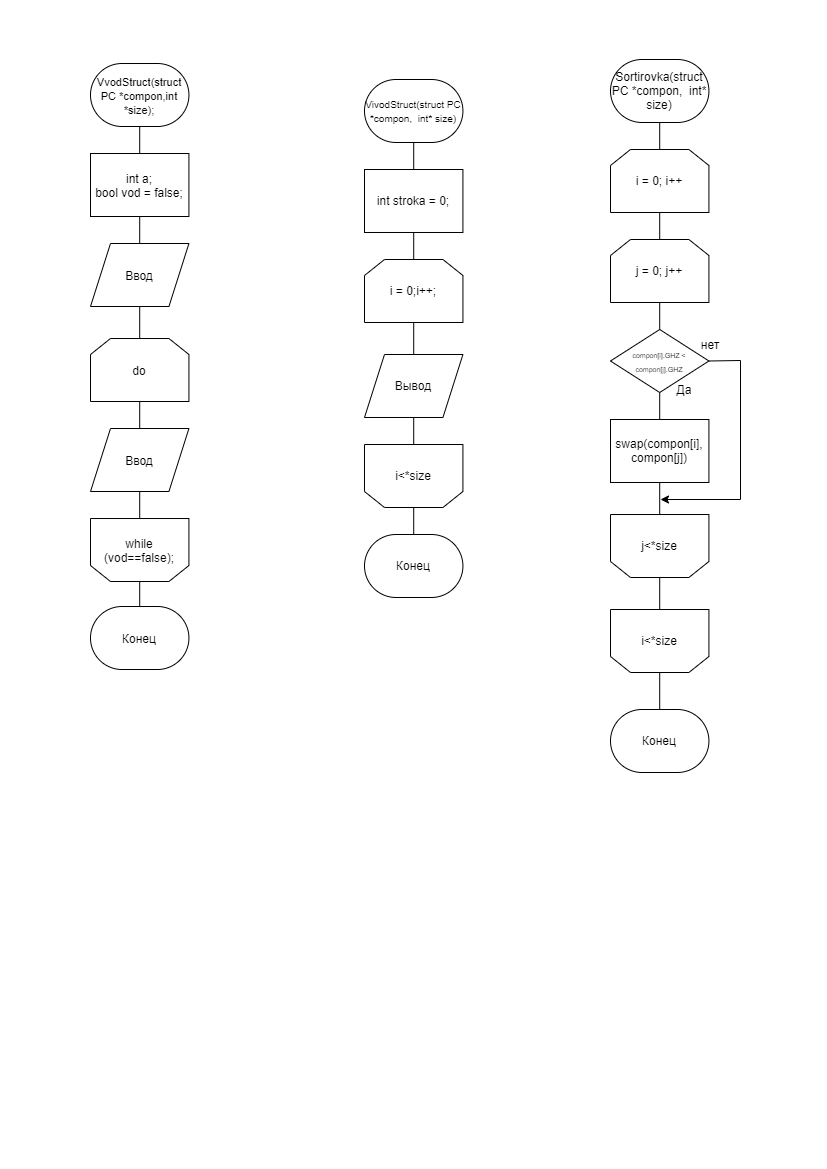
}

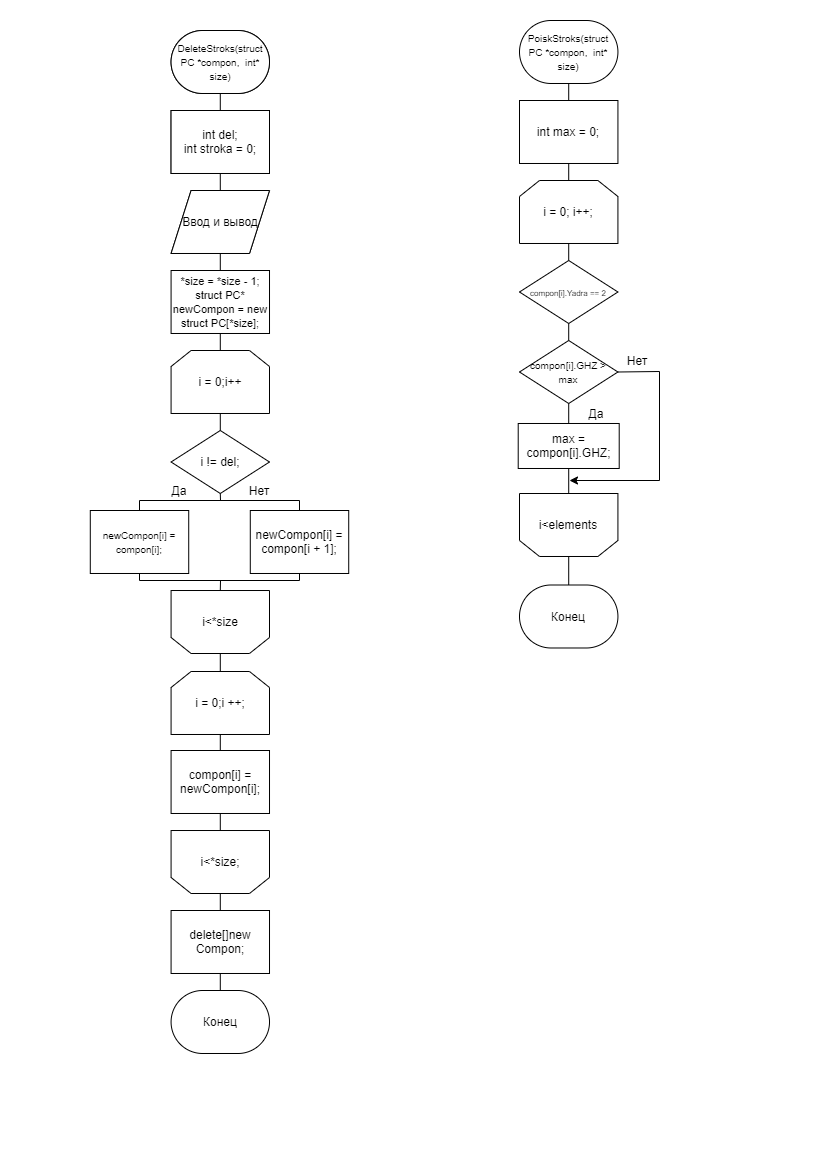
}

cout << "Двухъядерный процессор с самой низкой внутренней частотой " << compon[min].processor<<endl;

}







Вывод: изучил синтаксис и правила работы со структурами. Реализовал программу с применением структур, перечислений и объединений.