Системное программирование

ИС-33

Зулумханов Гаджи Андреевич

Вариант 8,9

Содержание

Лабораторная №1 Линейныйалгоритм…………………………………………………………………………………… 3

Лабораторная №2 *Программирование разветвляющихся алгоритмов*...…………………………….. 3

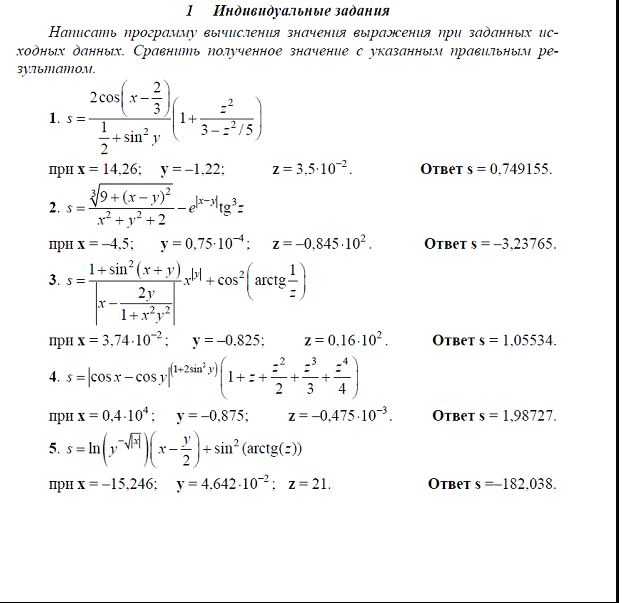
Лабораторная №3 Программирование циклических алгоритмов …………………………………………….3

Лабораторная №6 Динамические массивы…………………………………………………………………………… 3

Лабораторная №7 Указатели…………………………………………………………………………………………………… 3

Лабораторная №8 Разработка программ на языке С++ с использованием структур……..3

**Лабораторная №1 Линейный алгоритм**



1)

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double a = 14.26;

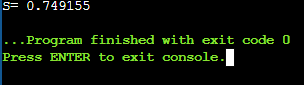
double b = -1.22;

double c = 3.5e-2;

double d = (2.\*cos(a - 2./3.) / (1./2. + pow(sin(b), 2))) \* ((1. + (pow(c,2) / (3. - pow(c, 2) / 5))));

cout << "S= " << d;

}



2)

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double a = -4.5;

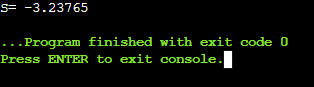
double b = 0.75e-4;

double c = -0.845e2;

double d = ((cbrt(9.+pow((a-b), 2))) / (pow(a, 2) + 2.)) - ((exp(abs(a-b))) \* pow(tan(c), 3));

cout << "S= " << d;

}



3)

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double a = 3.74e-2;

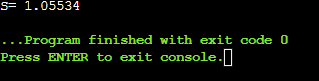
double b = -0.825;

double c = 0.16e+2;

double d = (1 + pow(sin((a+b)), 2)) / (abs(a - (2\*b / (1+pow(a,2) \* pow(b, 2))))) \* pow(a, abs(b)) + pow(cos(atan(1/c)), 2);

cout << "S= " << d;

}



4)

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double a = 0.4e4;

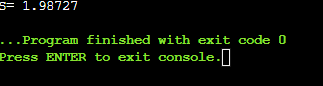
double b = -0.875;

double c = -0.475e-3;

double d = (pow(abs(cos(a) - cos(b)), (1. + 2 \* pow(sin(b), 2)))) \* (1. + c + (pow(c, 2) / 2.) + (pow(c, 3) / 3.) + (pow(c, 4) / 4.));

cout << "S= " << d;

}



5)

#include <iostream>

#include <cmath>

using namespace std;

int main(){

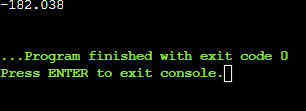
double a = -15.246; double b = 4.642e-2; int c = 21;

double d = log(pow(b, -sqrt(abs(a)))) \* (a - (b/2));

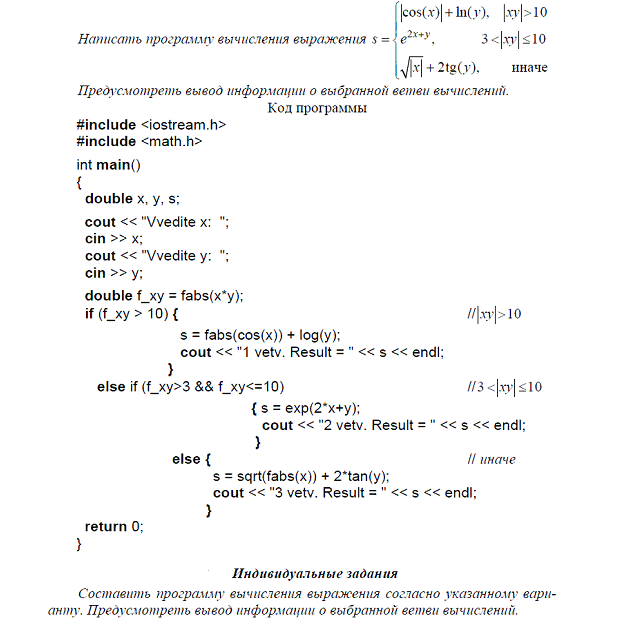
double e = pow(sin(atan(c)), 2);

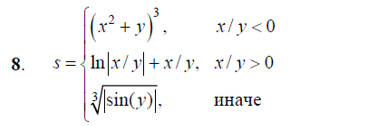
cout << d + e << endl;

}



**Лабораторная №2 *Программирование разветвляющихся алгоритмов***





#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double x,y;

cout << "Введите число x: ";

cin >> x;

cout << "Введите число y: ";

cin >> y;

if(x/y<0){

cout<< pow(x,2) + pow(y,3);

}

else if(x/y>0){

cout<<log(abs(x/y)) + x/y;

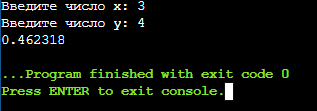
}

else{

cout << cbrt(abs(sin(y)));

}

}



**Лабораторная №3 Программирование циклических алгоритмов**





#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double a= 0.1;

double b=1.2;

double h=0.1;

for(double x = a; x <= b; x+=h ){

double y = 0;

for(int n = 1; n <= 20; n++){

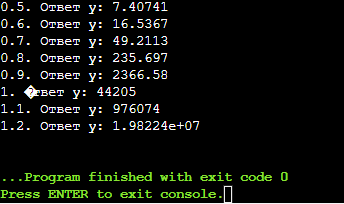
y+= ((2\*pow(n,2) + 1) / 2 \* n) \* (pow(x, 2 \* n - 2));

}

cout << x << ". Ответ y: " << y << endl;

}

}



**Лабораторная №6 *Динамические массивы***

1. Создать динамические массивы, используя указатели. Дан двумерный массив a, размером (n⋅m). Заполнить одномерный массив, найдя среднее геометрическое положительных элементов в каждом столбце матрицы.

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int n, m;

cout << "Введите количество строк (n): ";

cin >> n;

cout << "Введите количество столбцов (m): ";

cin >> m;

double\*\* a = new double\*[n];

for (int i = 0; i < n; i++) {

a[i] = new double[m];

}

cout << "Введите элементы массива a:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cin >> a[i][j];

}

}

cout << "Массива a:" << endl;

for(int i = 0; i < n; ++i) {

for(int j = 0; j < m; ++j)

cout << a[i][j] << " ";

cout << endl;

}

double\* avgGeo = new double[m];

for (int j = 0; j < m; j++) {

double product = 1.0;

int count = 0;

for (int i = 0; i < n; i++) {

if (a[i][j] > 0) {

product \*= a[i][j];

count++;

}

}

if (count > 0) {

avgGeo[j] = pow(product, 1.0 / count);

} else {

avgGeo[j] = 0.0;

}

}

cout << "Средние геометрические элементов в каждом столбце:" << endl;

for (int j = 0; j < m; j++) {

cout << avgGeo[j] << " ";

}

for (int i = 0; i < n; i++) {

delete[] a[i];

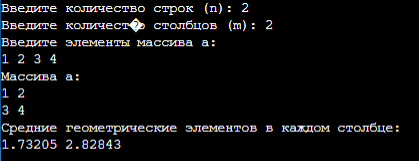
}

delete[] a;

delete[] avgGeo;

return 0;

}



1. Создать динамические массивы, используя указатели. Дан

двумерный массив a, размером (n⋅n) (или квадратная

матрица a). Найти произведение элементов

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Введите размерность квадратной матрицы (n): ";

cin >> n;

double\*\* a = new double\*[n];

for (int i = 0; i < n; i++) {

a[i] = new double[n];

}

cout << "Введите элементы квадратной матрицы a:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cin >> a[i][j];

}

}

cout << "Массива a:" << endl;

for(int i = 0; i < n; ++i) {

for(int j = 0; j < n; ++j)

cout << a[i][j] << " ";

cout << endl;

}

double product = 1.0;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

product \*= a[i][j];

}

}

cout << "Произведение всех элементов в квадратной матрице: " << product << endl;

for (int i = 0; i < n; i++) {

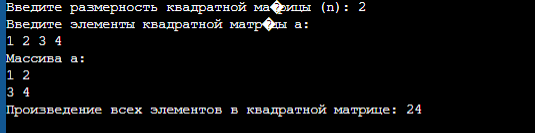
delete[] a[i];

}

delete[] a;

return 0;

}



1. Создать динамические массивы, используя указатели. Дан

двумерный массив a, размером (n⋅m). Найти сумму

положительных элементов в каждой строке матрицы. Затем

упорядочить по убыванию созданный массив.

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int n, m;

cout << "Введите количество строк (n): ";

cin >> n;

cout << "Введите количество столбцов (m): ";

cin >> m;

double\*\* a = new double\*[n];

for (int i = 0; i < n; i++) {

a[i] = new double[m];

}

cout << "Введите элементы массива a:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cin >> a[i][j];

}

}

cout << "Массива a:" << endl;

for(int i = 0; i < n; ++i) {

for(int j = 0; j < m; ++j)

cout << a[i][j] << " ";

cout << endl;

}

double\* sums = new double[n];

for (int i = 0; i < n; i++) {

sums[i] = 0.0;

for (int j = 0; j < m; j++) {

if (a[i][j] > 0) {

sums[i] += a[i][j];

}

}

}

sort(sums, sums + n, greater<double>());

cout << "Суммы положительных элементов в каждой строке:" << endl;

for (int i = 0; i < n; i++) {

cout << sums[i] << " ";

}

for (int i = 0; i < n; i++) {

delete[] a[i];

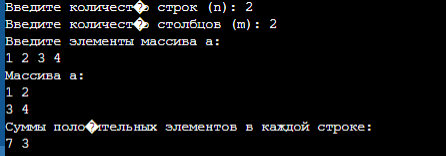
}

delete[] a;

delete[] sums;

return 0;

}



1. Создать динамические массивы, используя указатели. Дан двумерный массив a, размером (n⋅n). Упорядочить по убыванию побочную диагональ.

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int n;

cout << "Введите размерность квадратной матрицы (n): ";

cin >> n;

double\*\* a = new double\*[n];

for (int i = 0; i < n; i++) {

a[i] = new double[n];

}

cout << "Введите элементы квадратной матрицы a:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cin >> a[i][j];

}

}

cout << "Массива a:" << endl;

for(int i = 0; i < n; ++i) {

for(int j = 0; j < n; ++j)

cout << a[i][j] << " ";

cout << endl;

}

double\* diagonalValues = new double[n];

for (int i = 0; i < n; i++) {

diagonalValues[i] = a[i][n - i - 1];

}

sort(diagonalValues, diagonalValues + n, greater<double>());

for (int i = 0; i < n; i++) {

a[i][n - i - 1] = diagonalValues[i];

}

cout << "Матрица с упорядоченной побочной диагональю:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cout << a[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++) {

delete[] a[i];

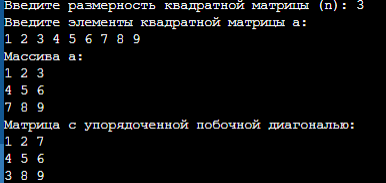
}

delete[] a;

delete[] diagonalValues;

return 0;

}



**Лабораторная №7 *Указатели***

***1***Ввести значение 2-х вещественных переменных a и b. Направить два указателя на эти переменные. Большее из них с помощью указателя увеличить на 7 и меньшее уменьшить на 3

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a ,b;

double\* ptrA = &a;

double\* ptrB = &b;

cout << "введите число a ";

cin >> a;

cout << "введите число b ";

cin >> b;

double\* ptrLarger;

double\* ptrSmaller;

if (\*ptrA > \*ptrB) {

ptrLarger = ptrA;

ptrSmaller = ptrB;

}

else {

ptrLarger = ptrB;

ptrSmaller = ptrA;

}

\*ptrLarger += 7;

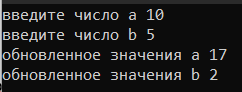
\*ptrSmaller -= 3;

cout << "обновленное значения a " << a << endl;

cout << "обновленное значения b " << b << endl;

return 0;

}



***2*** Описать 2 указателя на вещественный тип. Выделить для них динамическую память. Ввести значения в выделенную память с клавиатуры. Поменять местами их значения.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a, b, c;

cout << "введите число a ";

cin >> a;

cout << "введите число b ";

cin >> b;

double\* ptrA = &a;

double\* ptrB = &b;

c = \*ptrB;

\*ptrB = \*ptrA;

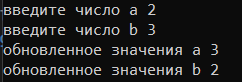
\*ptrA = c;

cout << "обновленное значения a " << \*ptrA << endl;

cout << "обновленное значения b " << \*ptrB << endl;

return 0;

}



***3*** Создать динамические массивы, используя указатели. Задан одномерный массив a(n). Найти номер последнего элемента равного 5 и переставить его с первым элементом массива. Найти среднее арифметическое элементов массива больших заданного числа alfa.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

int n;

double alfa;

cout << "введите размер массива: ";

cin >> n;

int\* a = new int[n];

cout << "Введите элементы массива a: " << endl;

for (int i = 0; i < n; i++) {

cin >> a[i];

}

int lastFiveIndex = -1;

for (int i = n - 1; i >= 0; i--) {

if (a[i] == 5) {

lastFiveIndex = i;

break;

}

}

if (lastFiveIndex != -1) {

int temp = a[0];

a[0] = a[lastFiveIndex];

a[lastFiveIndex] = temp;

}

cout << "Введите значение alfa: ";

cin >> alfa;

double sum = 0;

int count = 0;

for (int i = 0; i < n; i++) {

if (a[i] > alfa) {

sum += a[i];

count++;

}

}

double average = (count > 0) ? (sum / count) : 0;

cout << "Массив после перестановки:" << endl;

for (int i = 0; i < n; i++) {

cout << a[i] << " ";

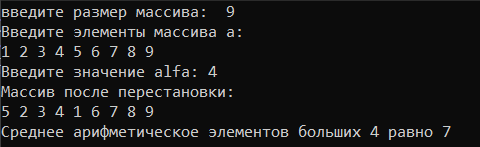
}

cout << endl;

cout << "Среднее арифметическое элементов больших " << alfa << " равно " << average << std::endl;

delete[] a;

return 0;



***4*** Создать динамические массивы, используя указатели. Дан массив c(n). Переписать в массив x(n) все ненулевые элементы массива, умноженные на 44 (со сжатием, без пустых элементов внутри). Затем упорядочить методом «выбора и перестановки» по возрастанию новый массив.

#include <iostream>

using namespace std;

void createAndSortArray(int\* c, int n) {

int\* x = new int[n];

int xSize = 0;

for (int i = 0; i < n; i++) {

if (c[i] != 0) {

x[xSize] = c[i] \* 44;

xSize++;

}

}

for (int i = 0; i < xSize - 1; i++) {

int minIndex = i;

for (int j = i + 1; j < xSize; j++) {

if (x[j] < x[minIndex]) {

minIndex = j;

}

}

int temp = x[i];

x[i] = x[minIndex];

x[minIndex] = temp;

}

cout << "Отсортированный массив x(n): ";

for (int i = 0; i < xSize; i++) {

cout << x[i] << " ";

}

cout << endl;

delete[] x;

}

int main() {

setlocale(LC\_ALL, "RU");

int n;

cout << "Введите размер массива c(n): ";

cin >> n;

int\* c = new int[n];

std::cout << "Введите элементы массива c(n): ";

for (int i = 0; i < n; i++) {

cin >> c[i];

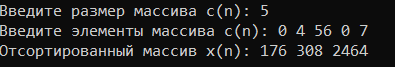
}

createAndSortArray(c, n);

delete[] c;

return 0;

}



**Лабораторная №8 Разработка программ на языке С++ с использованием структур.**

1. Определить комбинированный (структурный) тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по 20 вершинам. Вывести среднее значение высот всех 20 вершин. Далее вывести названия всех вершин выше среднего.

struct Mountain

{

char name\_mountain[50];

int height\_mountain;

};

Mountain\* AddStruct(Mountain\* Obj, const int amount);

void setData(Mountain\* Obj, const int amount);

void showData(const Mountain\* Obj, const int amount);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//setlocale(LC\_ALL, "RU");

Mountain\* OurMountain = 0;

int mountainAmount = 0;

int YesOrNot = 0;

do {

OurMountain = AddStruct(OurMountain, mountainAmount);

setData(OurMountain, mountainAmount);

mountainAmount++;

cout << "Для продолжения нажмите 1 для завершения 0: ";

cin >> YesOrNot;

cin.get();

} while (YesOrNot != 0);

showData(OurMountain, mountainAmount);

int totalHeight = 0;

for (int i = 0; i < mountainAmount; i++) {

totalHeight += OurMountain[i].height\_mountain;

}

double averageHeight = static\_cast<double>(totalHeight) / mountainAmount;

cout << "Среднее значение высот всех вершин: " << averageHeight << endl;

cout << "Названия вершин выше среднего значения высоты:\n";

for (int i = 0; i < mountainAmount; i++) {

if (OurMountain[i].height\_mountain > averageHeight) {

cout << OurMountain[i].name\_mountain << endl;

}

}

delete[] OurMountain;

return 0;

}

Mountain\* AddStruct(Mountain\* Obj, const int amount) {

if (amount == 0)

Obj = new Mountain[amount + 1];

else {

Mountain\* tempObj = new Mountain[amount + 1];

for (int i = 0; i < amount; i++) {

tempObj[i] = Obj[i];

}

delete[] Obj;

Obj = tempObj;

}

return Obj;

}

void setData(Mountain\* Obj, const int amount) {

cout << amount + 1 << " вершина:\n";

cout << "Название вершины: ";

cin.getline(Obj[amount].name\_mountain, 50);

cout << "Высота: ";

cin >> Obj[amount].height\_mountain;

cin.get();

cout << endl;

}

void showData(const Mountain\* Obj, const int amount) {

cout << endl;

for (int i = 0; i < amount; i++) {

cout << " № " << "Название вершины";

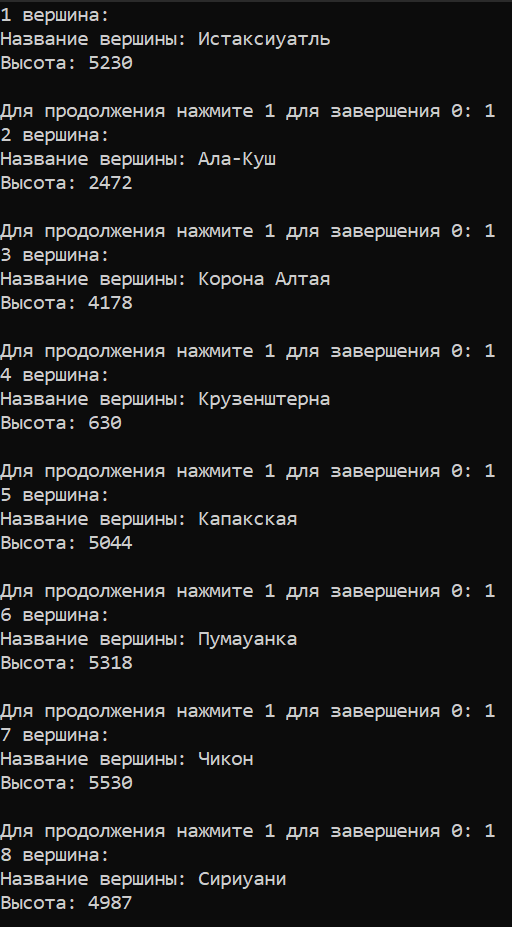
cout << " № " << "Высота в км\t" << endl;

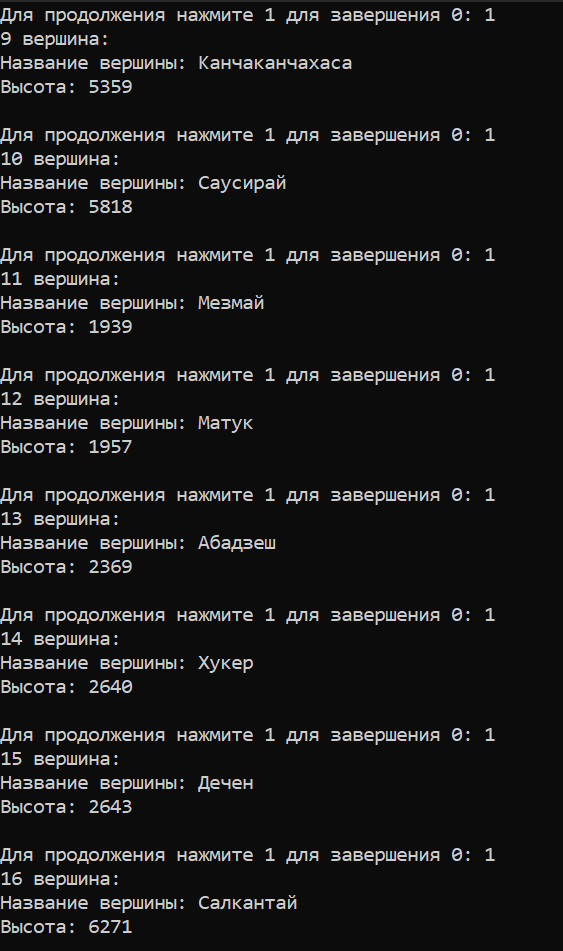
cout << "| " << i + 1 << " " << Obj[i].name\_mountain << "\t\t" << Obj[i].height\_mountain << endl;

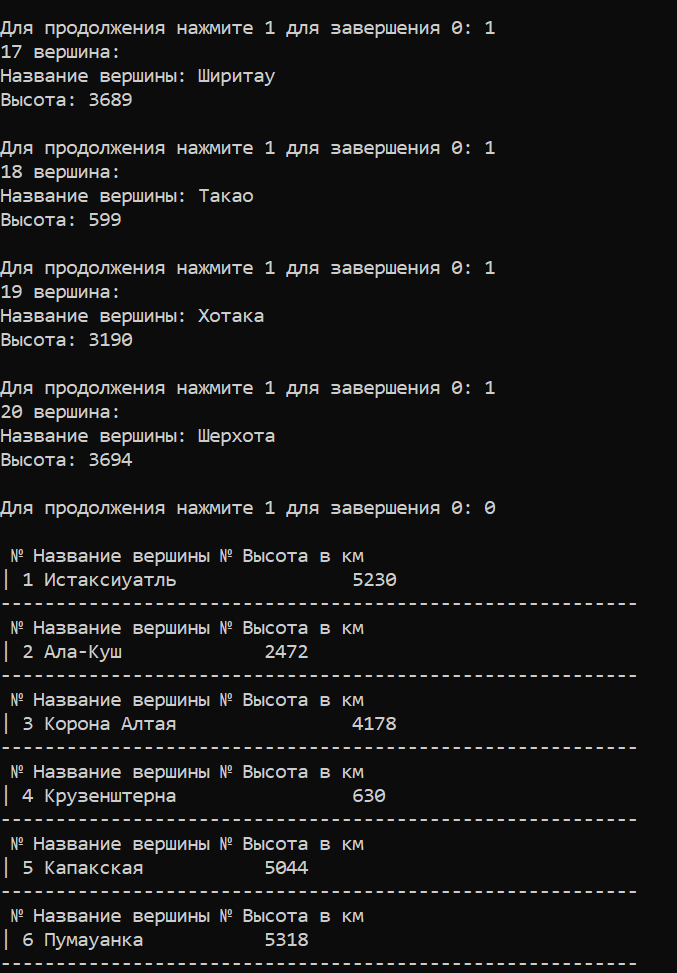
cout << "----------------------------------------------------------" << endl;

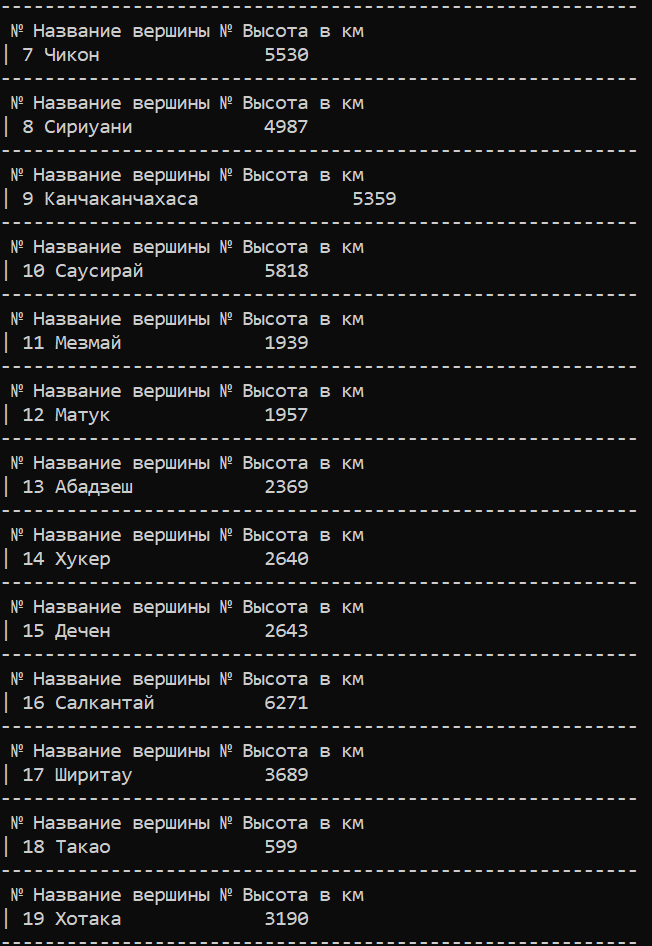
}

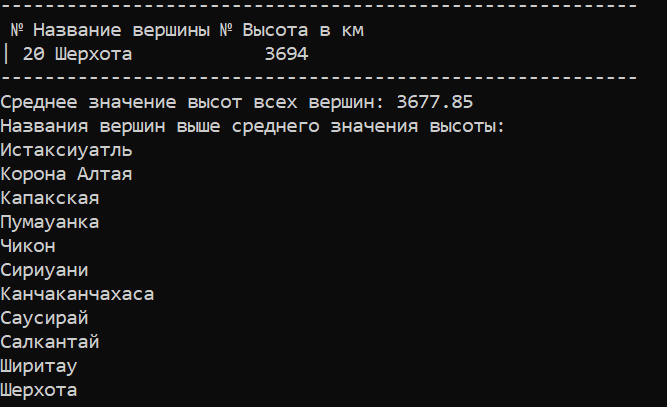
}











1. Определить комбинированный (структурный) тип для представления анкеты студента, состоящей из его фамилии, дня рождения и пола. «День рождения» состоит из полей: «число», «месяц», «год». Ввести информацию по 25 студентам из группы. Вывести все фамилии мальчиков, родившихся в мае 1986 года.

#include <iostream>

using namespace std;

struct Birthday {

int day;

int month;

int year;

};

struct Student {

char surname[50];

Birthday birthdate;

char gender[10];

};

Student\* AddStudent(Student\* students, int amount);

void SetData(Student\* students, int amount);

void ShowData(const Student\* students, int amount);

void ShowMay1986MaleStudents(const Student\* students, int amount);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Student\* students = nullptr;

int studentAmount = 0;

int continueInput = 0;

do {

students = AddStudent(students, studentAmount);

SetData(students, studentAmount);

studentAmount++;

cout << "Для продолжения ввода нажмите 1, для завершения ввода 0: ";

cin >> continueInput;

cin.get();

} while (continueInput != 0);

ShowData(students, studentAmount);

ShowMay1986MaleStudents(students, studentAmount);

delete[] students;

return 0;

}

Student\* AddStudent(Student\* students, int amount) {

if (amount == 0) {

students = new Student[amount + 1];

}

else {

Student\* tempStudents = new Student[amount + 1];

for (int i = 0; i < amount; i++) {

tempStudents[i] = students[i];

}

delete[] students;

students = tempStudents;

}

return students;

}

void SetData(Student\* students, int amount) {

cout << amount + 1 << "-й студент:\n";

cout << "Фамилия: ";

cin.getline(students[amount].surname, 50);

cout << "Введите дату рождения (число месяц год): ";

cin >> students[amount].birthdate.day >> students[amount].birthdate.month >> students[amount].birthdate.year;

cin.get();

cout << "Пол: ";

cin.getline(students[amount].gender, 10);

cout << endl;

}

void ShowData(const Student\* students, int amount) {

cout << "Информация по студентам:\n";

for (int i = 0; i < amount; i++) {

cout << "Студент №" << i + 1 << ":" << endl;

cout << "Фамилия: " << students[i].surname << endl;

cout << "Дата рождения: " << students[i].birthdate.day << "/" << students[i].birthdate.month << "/" << students[i].birthdate.year << endl;

cout << "Пол: " << students[i].gender << endl << endl;

}

}

void ShowMay1986MaleStudents(const Student\* students, int amount) {

cout << "Фамилии мальчиков, родившихся в мае 1986 года:\n";

for (int i = 0; i < amount; i++) {

if (students[i].birthdate.month == 5

&& students[i].birthdate.year == 1986

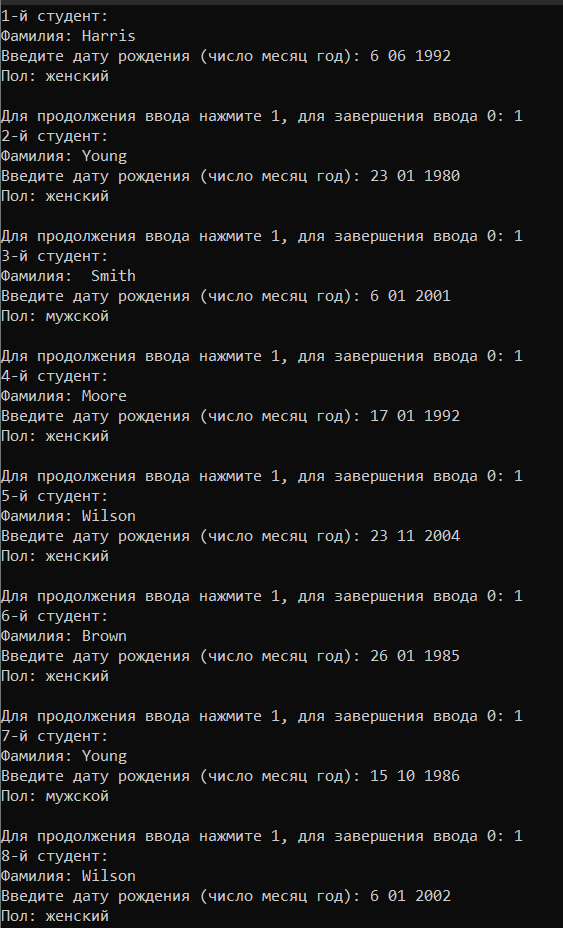
&& strcmp(students[i].gender, "мужской") == 0) {

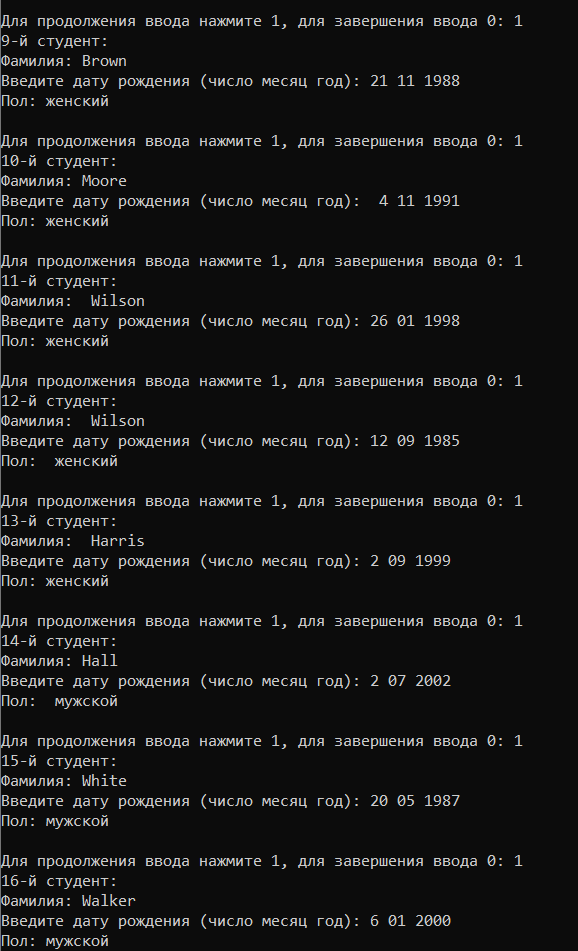
cout << students[i].surname << endl;

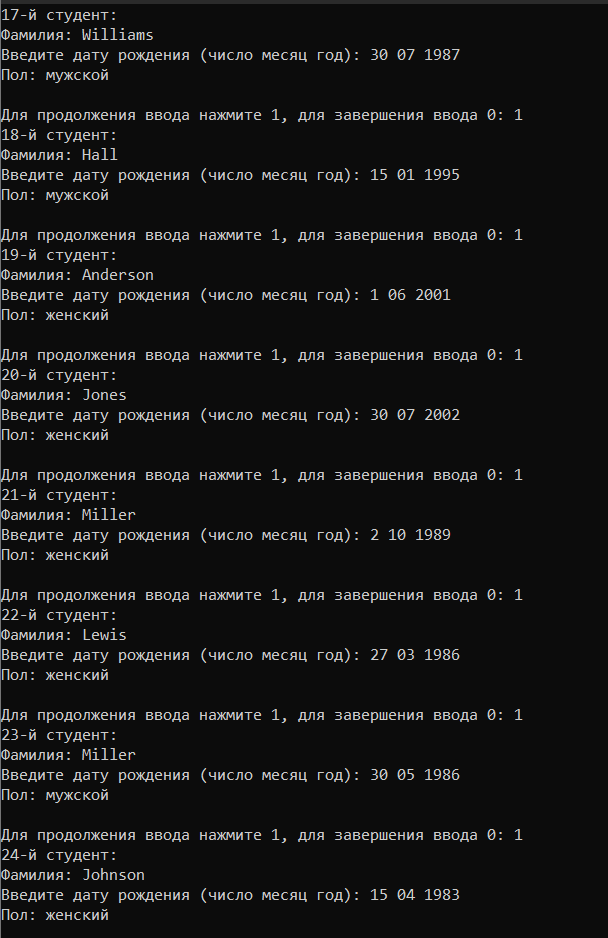
}

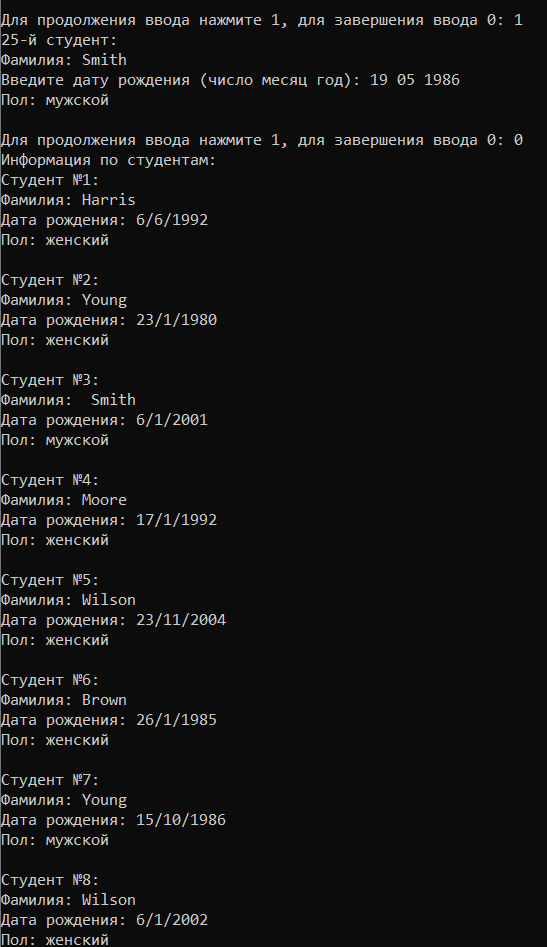
}

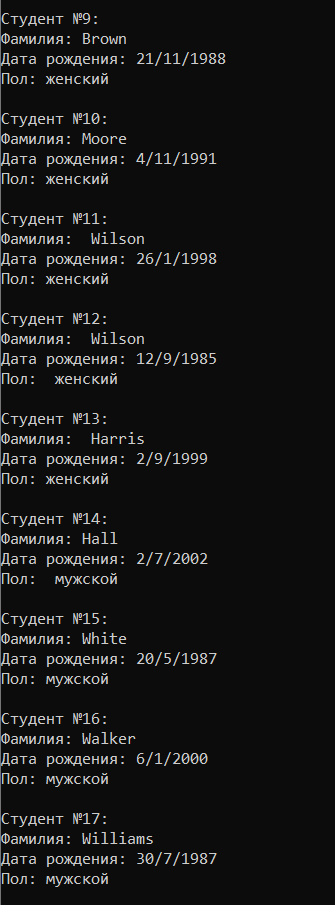
}

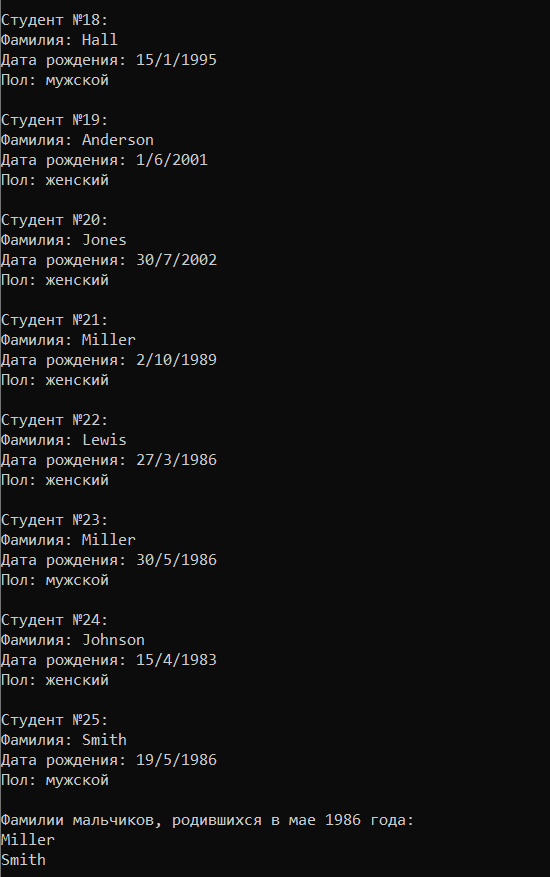












1. Определить комбинированный (структурный) тип, описывающий окружность и состоящий из двух полей: «радиус» и «центр». Поле «центр» в свою очередь состоит еще из двух полей: «координата Х» и «координата Y». Ввести информацию по 10 окружностям. Переписать из исходного массива в другой массив, информацию только о тех окружностях, центр которых лежит в 33-ой четверти.

#include <iostream>

using namespace std;

struct Center {

double x;

double y;

};

struct Circle {

double radius;

Center center;

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int n = 10;

Circle circles[n];

Circle selectedCircles[n];

int selectedCount = 0;

// Ввод информации по окружностям

for (int i = 0; i < n; i++) {

cout << "Введите радиус окружности " << i + 1 << ": ";

cin >> circles[i].radius;

cout << "Введите координату X центра окружности " << i + 1 << ": ";

cin >> circles[i].center.x;

cout << "Введите координату Y центра окружности " << i + 1 << ": ";

cin >> circles[i].center.y;

}

// Выборка окружностей, центр которых лежит в 3-й четверти

for (int i = 0; i < n; i++) {

if (circles[i].center.x < 0 && circles[i].center.y < 0) {

selectedCircles[selectedCount] = circles[i];

selectedCount++;

}

}

// Вывод информации о выбранных окружностях

cout << "\nИнформация о выбранных окружностях:\n";

for (int i = 0; i < selectedCount; i++) {

cout << "Окружность " << i + 1 << ":\n";

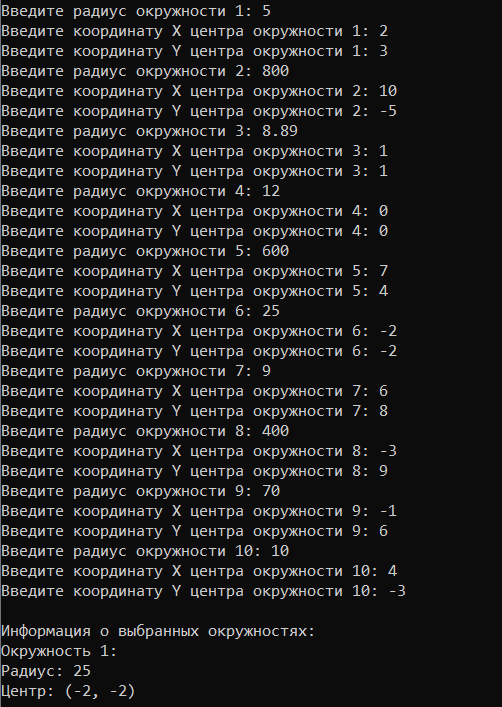
cout << "Радиус: " << selectedCircles[i].radius << "\n";

cout << "Центр: (" << selectedCircles[i].center.x << ", "

<< selectedCircles[i].center.y << ")\n\n";

}

return 0;



1. Определить комбинированный (структурный) тип, описывающий окружность и состоящий из двух полей: «радиус» и «центр». Поле «центр» в свою очередь состоит еще из двух полей: «координата Х» и «координата Y». Ввести информацию по 10 окружностям. Переписать из исходного массива в другой массив, информацию только о тех окружностях, центр которых лежит в 33-ой четверти.

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

struct HotelRoom {

string hotelName;

int roomNumber;

string comfortLevel;

int capacity;

double cost;

};

bool compareByCost(const HotelRoom& room1, const HotelRoom& room2) {

return room1.cost < room2.cost;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int n = 10;

HotelRoom rooms[n];

HotelRoom selectedRooms[n];

int selectedCount = 0;

// Заполнение информации о гостиничных номерах

for (int i = 0; i < n; i++) {

cout << "Введите название гостиницы для номера " << i + 1 << ": ";

cin >> rooms[i].hotelName;

cout << "Введите номер гостиничного номера " << i + 1 << ": ";

cin >> rooms[i].roomNumber;

cout << "Введите уровень комфортности для номера " << i + 1 << " (люкс, полулюкс, стандарт, эконом): ";

cin >> rooms[i].comfortLevel;

cout << "Введите количество человек, размещаемых в номере " << i + 1 << ": ";

cin >> rooms[i].capacity;

cout << "Введите стоимость гостиничного номера " << i + 1 << ": ";

cin >> rooms[i].cost;

}

// Выборка гостиничных номеров с названием, оканчивающимся на "plaza"

for (int i = 0; i < n; i++) {

std::string hotelName = rooms[i].hotelName;

if (hotelName.length() >= 5 && hotelName.substr(hotelName.length() - 5) == "плаза") {

selectedRooms[selectedCount] = rooms[i];

selectedCount++;

}

}

// Сортировка нового массива по возрастанию стоимости

sort(selectedRooms, selectedRooms + selectedCount, compareByCost);

// Вывод информации о выбранных гостиничных номерах

std::cout << "\nИнформация о гостиничных номерах с названием, оканчивающимся на 'plaza', отсортированная по возрастанию стоимости:\n";

for (int i = 0; i < selectedCount; i++) {

cout << "Гостиница: " << selectedRooms[i].hotelName << "\n";

cout << "Номер: " << selectedRooms[i].roomNumber << "\n";

cout << "Уровень комфортности: " << selectedRooms[i].comfortLevel << "\n";

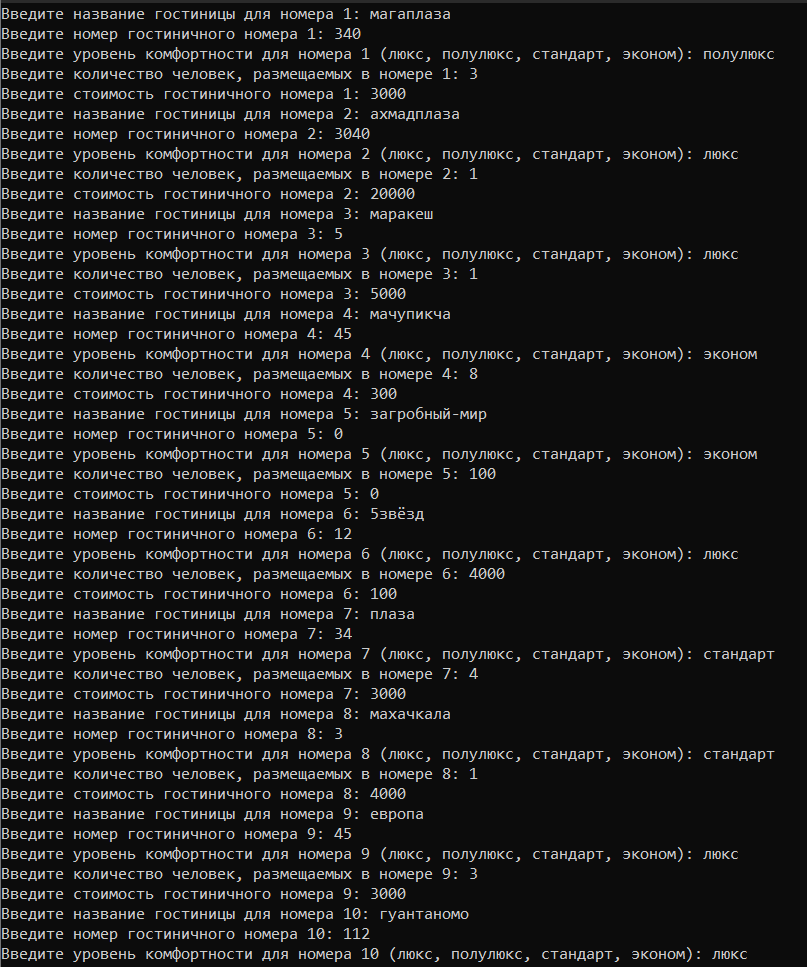
cout << "Количество человек: " << selectedRooms[i].capacity << "\n";

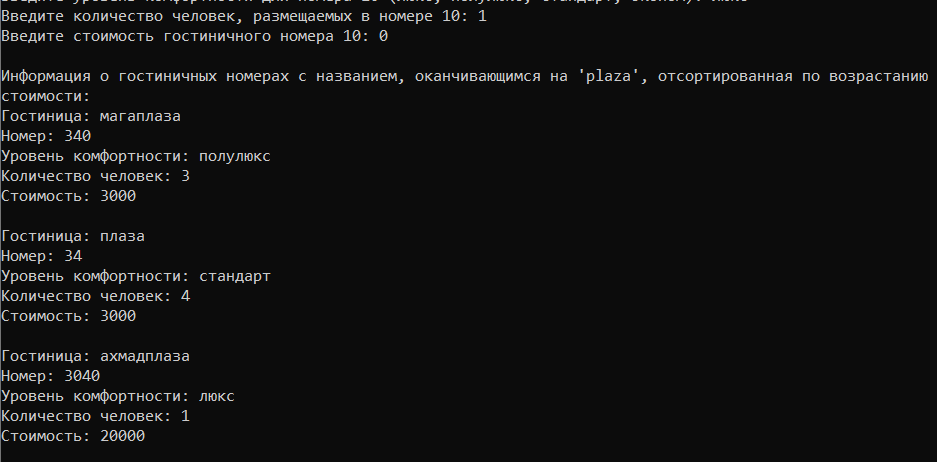
cout << "Стоимость: " << selectedRooms[i].cost << "\n\n";

}

return 0;

}





5.Определить комбинированный (структурный) тип для

представления анкеты ребенка, состоящей из его имени, пола и

роста. Ввести информацию по 20 детям. Вывести средний рост

девочек. Затем вывести информацию, отсортированную по

имени по алфавиту.

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

struct ChildRecord {

string name;

string gender;

double height;

};

bool compareByName(const ChildRecord& record1, const ChildRecord& record2) {

return record1.name < record2.name;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int childrenCount = 20;

ChildRecord children[childrenCount];

// Ввод информации о детях

for (int i = 0; i < childrenCount; ++i) {

cout << "Введите имя ребенка " << i + 1 << ": ";

cin >> children[i].name;

cout << "Введите пол ребенка " << i + 1 << " (м/ж): ";

cin >> children[i].gender;

cout << "Введите рост ребенка " << i + 1 << ": ";

cin >> children[i].height;

}

// Вычисление среднего роста девочек

double totalHeightGirls = 0;

int girlsCount = 0;

for (int i = 0; i < childrenCount; ++i) {

if (children[i].gender == "ж") {

totalHeightGirls += children[i].height;

girlsCount++;

}

}

double averageHeightGirls = girlsCount > 0 ? totalHeightGirls / girlsCount : 0;

// Вывод среднего роста девочек

cout << "Средний рост девочек: " << averageHeightGirls << "\n";

// Сортировка информации по имени

sort(children, children + childrenCount, compareByName);

// Вывод отсортированной информации

cout << "Информация, отсортированная по имени:\n";

for (int i = 0; i < childrenCount; ++i) {

cout << "Имя: " << children[i].name << "\n";

cout << "Пол: " << children[i].gender << "\n";

cout << "Рост: " << children[i].height << "\n\n";

}

return 0;

}

