1 Задание

Написать перегруженные функции (int, double,char) для выполнения следующих задач.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include<iostream>

#include<Windows.h>

#include <iomanip>

#include <string>

#include <conio.h>

using namespace std;

enum userChoice

{

Exit = 0,

Initialize = 1,

Print = 2,

MinMax = 3,

Sort = 4

};

void matrixDoing(int(\*)[10], int, int);

void matrixDoing(int(\*)[10], int);

void matrixDoing(int(\*)[10], int, int, int);

void matrixDoing(int(\*)[10]);

int main()

{

system("chcp 1251");

int const massivDimention = 10;

int massiv[massivDimention][massivDimention] = { 0 };

int limitNumber = 100;

int max = INT\_MIN;

int min = INT\_MAX;

int intUserChoice;

enum userChoice userChoice;

cout << "Эта программа для работы с матрицей" << endl;

do

{

cout << "Выберите то, что вы хотите сделать: " << endl <<

"0 = Выход" << endl << "1 = Инициализировать матрицу" << endl <<

"2 = Ввывод матрицы на экран" << endl << "3 = Найти минимальные и максимальные элементы" << endl <<

"4 = Сортировка строк" << endl;

cin >> intUserChoice;

userChoice = (enum userChoice)intUserChoice;

switch (userChoice)

{

case Exit:

cout << "Вы хотите выйти" << endl;

break;

case Initialize:

matrixDoing(massiv, massivDimention, limitNumber);

break;

case Print:

matrixDoing(massiv, massivDimention);

break;

case MinMax:

matrixDoing(massiv, massivDimention, min, max);

break;

case Sort:

matrixDoing(massiv);

break;

default:

cout << "Ошибка. Неправильный выбор!" << endl;

return 1;

}

} while (userChoice != Exit);

return 0;

}

//Инициализация квадратной матрицы;

void matrixDoing(int arr[][10], int size, int limitNumber)

{

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

{

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

{

arr[i][j] = rand() % limitNumber;

}

}

cout << "Инициализированный массив" << endl;

}

//Вывод матрицы на экран;

void matrixDoing(int arr[][10], int arSize)

{

cout << "Старт массива: " << endl;

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

{

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

{

cout << fixed << setw(3) << arr[i][j] << " ";

}

cout << endl;

}

}

//Определение максимального и минимального элемента на главной диагонали матрицы;

void matrixDoing(int arr[][10], int arSize, int minElement, int maxElement)

{

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

{

if (arr[i][i] > maxElement)

{

maxElement = arr[i][i];

}

if (arr[i][i] < minElement)

{

minElement = arr[i][i];

}

}

cout << "Минимальный элемент = " << minElement << endl <<

"Максимальный элемент = " << maxElement << endl;

}

//Сортировка элементов по возрастанию отдельно для каждой строки матрицы.

void matrixDoing(int arr[][10])

{

int arraySize = 10;

int bufer;

cout << "Сортировка по возрастанию элементов каждой строки отдельно" << endl;

for (int k = 0; k < arraySize; k++)

{

for (int j = 0; j < arraySize - 1; j++)

{

for (int i = 1; i < arraySize - j; i++)

{

if (arr[k][i] < arr[k][i - 1])

{

bufer = arr[k][i - 1];

arr[k][i - 1] = arr[k][i];

arr[k][i] = bufer;

}

}

}

}

}

-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 задание

\_\_ Написать рекурсивную функцию нахождения

наибольшего общего делителя двух целых чисел

-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

#include <iostream>

using namespace std;

int nod(int a, int b);

int main()

{

cout << nod(10, 5);

return 0;

}

int nod(int a, int b)

{

return b ? nod(b, a % b) : a;

}