#include <stdlib.h>

#include <iostream>

struct TwoInts

{

int a;

int b;

};

struct StructWithArray

{

int arr[4];

int\* someNumber;

};

int main()

{

// Создаем структуру TwoInts и инициализируем значения a и b нулями

TwoInts i2 = { };

// Присваиваем значения переменным a и b

i2.a = 5;

i2.b = 7;

// Выводим значения a и b

std::cout << i2.a << std::endl; // Вывод: 5

std::cout << i2.b << std::endl; // Вывод: 7

// Создаем структуру StructWithArray и инициализируем массив arr нулями

StructWithArray s = { };

// Присваиваем значение элементу массива arr

s.arr[0] = 10;

// Создаем еще одну структуру StructWithArray и присваиваем значение элементу массива arr

StructWithArray s1 = { };

s1.arr[0] = 15;

// Создаем указатель на структуру StructWithArray и присваиваем ему адрес первой структуры s

StructWithArray\* sPointer = &s;

// Изменяем значение элемента массива arr через указатель

sPointer->arr[0] = 20;

// Выводим значение элемента массива arr

std::cout << s.arr[0] << std::endl; // Вывод: 20

// Изменяем значение элемента массива arr

s.arr[0] = 25;

// Выводим значение элемента массива arr

std::cout << s.arr[0] << std::endl; // Вывод: 25

// Изменяем значение элемента массива arr через указатель

sPointer->arr[0] = 30;

// Выводим значение элемента массива arr

std::cout << s.arr[0] << std::endl; // Вывод: 30

// Переопределяем указатель на структуру s1 и изменяем значение элемента массива arr через указатель

sPointer = &s1;

sPointer->arr[0] = 35;

// Выводим значение элемента массива arr для s и s1

std::cout << s.arr[0] << std::endl; // Вывод: 30

std::cout << s1.arr[0] << std::endl; // Вывод: 35

// Создаем массив из двух структур StructWithArray

StructWithArray structArray[2] = { };

// Изменяем значение элемента массива arr для первой структуры

structArray[0].arr[3] = 77;

// Присваиваем указателю значение адреса элемента массива arr второй структуры

structArray[1].someNumber = &structArray[0].arr[3];

// Присваиваем указателю адрес элемента массива arr через указатель sPointer

sPointer = &s;

// Создаем указатель на int и присваиваем ему адрес элемента массива arr через указатель sPointer

int\* pointer = &sPointer->arr[3];

// Изменяем значение элемента массива arr через указатель pointer

s.arr[3] = 72;

// Выводим значение, на которое указывает указатель pointer

std::cout << \*pointer; // Вывод: 72

// Создаем экземпляр структуры StructWithArray и заполняем его нулями

StructWithArray memory;

// Заполняем всю память экземпляра memory нулями

memset(&memory, 0, sizeof(StructWithArray));

return 0;

}

| **Адрес** | **0** | **1** | **2** | **3** |
| --- | --- | --- | --- | --- |
| s.arr | 20 | 0 | 0 | 72 |
| s1.arr | 35 | 0 | 0 | 77 |
| sPointer | Адрес s | - | - | - |
| pointer | Адрес s.arr[3] | - | - | - |
| structArray[0].arr | 0 | 0 | 0 | 77 |
| structArray[1].someNumber | Адрес structArray[0].arr[3] | - | - | - |
| memory | 0 | 0 | 0 | 0 |