**Федеральное государственное автономное образовательное учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)**

**(Факультет информационных технологий)**

***(Институт Принтмедиа и информационных технологий)***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 4**

**Дисциплина:** Введение в программирование

**Тема:** Функции

**Выполнил: студент группы 231-338**

Шаура Илья Максимович

**Дата, подпись** 24.10.2023

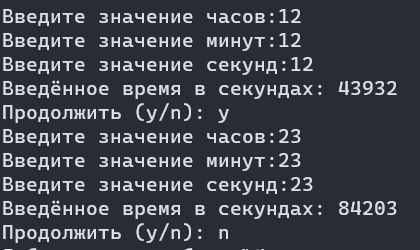


**Москва2023**

Меню выбора лабораторной / пункта

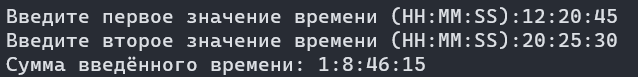


1. Напишите функцию с именем hms\_to\_secs(), имеющую три аргумента типа int: часы, минуты и секунды. Функция должна возвращать эквивалент переданного ей временного значения в секундах (типа long). Создайте программу, которая будет циклически запрашивать у пользователя ввод значения часов, минут и секунд и выводить результат работы функции на экран.



void lab4\_1() {  
 bool is\_loop = true;  
 while (is\_loop) {  
 int h, m, s;  
 char c\_continue;  
 std::cout << "Введите значение часов:";  
 std::cin >> h;  
 std::cout << "Введите значение минут:";  
 std::cin >> m;  
 std::cout << "Введите значение секунд:";  
 std::cin >> s;  
 std::cout << "Введённое время в секундах: " << utl::hms\_to\_secs(h, m, s) << std::endl;  
 def\_p:  
 std::cout << "Продолжить (y/n): ";  
 std::cin >> c\_continue;  
  
 switch (c\_continue) {  
 case 'y':  
 break;  
 case 'n':  
 is\_loop = false;  
 break;  
 default:  
 goto def\_p;  
 }  
 }  
}

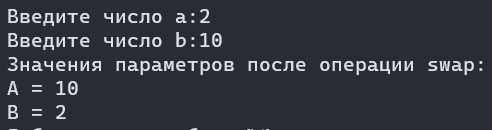
1. Модифицируйте программу, описанную в упражнении 5 предыдущей лабораторной работы «Структуры и перечисления», складывающую два структурных значения типа time. Теперь программа должна включать в себя две функции. Первая, time\_to\_secs(), принимает в качестве аргумента значение типа time и возвращает эквивалентное значение в секундах типа long. Вторая, secs\_to\_time(), в качестве аргумента принимает число секунд, имеющее тип long, а возвращает эквивалентное значение типа time.



void lab4\_2() {  
  
 utl::time first\_time, second\_time, sum\_time;  
  
 std::cout << "Введите первое значение времени (HH:MM:SS):";  
  
 std::cin >> first\_time.hours;  
 std::cin.get();  
 std::cin >> first\_time.minutes;  
 std::cin.get();  
 std::cin >> first\_time.seconds;  
  
 std::cout << "Введите второе значение времени (HH:MM:SS):";  
  
 std::cin >> second\_time.hours;  
 std::cin.get();  
 std::cin >> second\_time.minutes;  
 std::cin.get();  
 std::cin >> second\_time.seconds;  
  
 long sum = time\_to\_secs(first\_time) + time\_to\_secs(second\_time);  
  
 sum\_time = secs\_to\_time(sum);  
  
 std::cout << "Сумма введённого времени: " << sum\_time.days << ':' << sum\_time.hours << ':' << sum\_time.minutes  
 << ':' << sum\_time.seconds << std::endl;  
}



1. Напишите функцию с именем swap(), обменивающую значениями два своих аргумента типа int (обратите внимание, что изменяться должны значения переменных из вызывающей функции, а не локальных переменных функции). Выберите способ передачи аргументов. Напишите вызывающую программу main(), использующую данную функцию

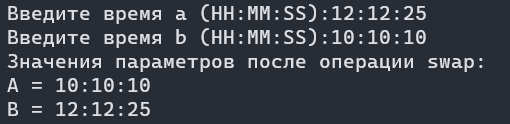




void swap(int& a, int& b) {  
 auto c = a;  
 a = b;  
 b = c;  
}

void lab4\_3() {  
 int a, b;  
 std::cout << "Введите число a:";  
 std::cin >> a;  
 std::cout << "Введите число b:";  
 std::cin >> b;  
  
 swap(a, b);  
  
 std::cout << "Значения параметров после операции swap:" << std::endl  
 << "A = " << a << std::endl  
 << "B = " << b << std::endl;  
}

1. Переработайте программу из предыдущего упражнения 3 так, чтобы функция swap() принимала в качестве аргументов значения типа time (см. упражнение 2).



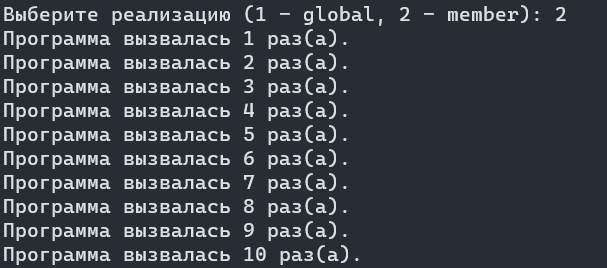


struct time {  
 int days;  
 int hours;  
 int minutes;  
 int seconds;  
  
 time() : days(0), hours(0), minutes(0), seconds(0) {}  
};

void swap(time& a, time& b) {  
 auto c = a;  
 a = b;  
 b = c;  
}

void lab4\_4() {  
 time a, b;  
  
 std::cout << "Введите время a (HH:MM:SS):";  
 std::cin >> a.hours;  
 std::cin.get();  
 std::cin >> a.minutes;  
 std::cin.get();  
 std::cin >> a.seconds;  
 std::cout << "Введите время b (HH:MM:SS):";  
 std::cin >> b.hours;  
 std::cin.get();  
 std::cin >> b.minutes;  
 std::cin.get();  
 std::cin >> b.seconds;  
  
 swap(a, b);  
  
 std::cout << "Значения параметров после операции swap:" << std::endl  
 << "A = " << a.hours << ':' << a.minutes << ':' << a.seconds << std::endl  
 << "B = " << b.hours << ':' << b.minutes << ':' << b.seconds << std::endl;  
}

1. Напишите функцию, которая при каждом вызове будет выводить на экран количество раз, которое она вызывалась ранее. Напишите программу, которая будет вызывать данную функцию не менее 10 раз. Попробуйте реализовать данную функцию двумя различными способами: с использованием глобальной переменной и статической локальной переменной для хранения числа вызовов функции. Какой из способов предпочтительней? Почему для решения задачи нельзя использовать обычную локальную переменную?





uint16\_t g\_counter = 0;  
void f\_g\_counter() {  
 g\_counter++;  
 std::cout << "Программа вызвалась " << g\_counter << " раз(а)." << std::endl;  
}  
void f\_m\_counter() {  
 static uint16\_t m\_counter = 0;  
 m\_counter++;  
 std::cout << "Программа вызвалась " << m\_counter << " раз(а)." << std::endl;  
}

void lab4\_5() {  
 char r;  
impl:  
 std::cout << "Выберите реализацию (1 - global, 2 - member): ";  
 std::cin >> r;  
 switch (r) {  
 case '1':  
 for (int i = 0; i < 10; ++i) {  
 f\_g\_counter();  
 }  
 break;  
 case '2':  
 for (int i = 0; i < 10; ++i) {  
 f\_m\_counter();  
 }  
 break;  
 default:  
 std::cout << "Wrong input" << std::endl;  
 goto impl;  
 }  
}