**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 № 5**

**Дисциплина:** Введение в программирование

**Тема:** Массивы и строки

**Выполнил: студент группы 231-338**

Шаура Илья Максимович

**Дата, подпись** 04.11.2023



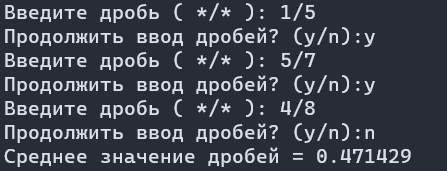
**Москва2023**

Меню выбора лабораторной / пункта



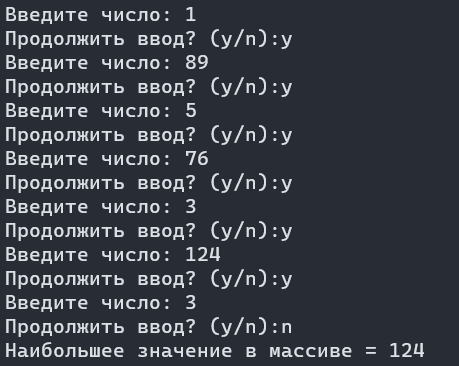
1. Напишите функцию main(), которая получает случайные дробные числа от пользователя, сохраняет их в массиве типа fraction, вычисляет среднее значение и выводит результат.





void lab5\_1() {  
 std::vector<fraction> fractions {};  
 bool is\_loop = true;  
 while (is\_loop) {  
 char y\_n;  
 fraction f;  
  
 std::cout << "Введите дробь ( \*/\* ): ";  
 std::cin >> f.full\_fraction\_s;  
 fractions.push\_back(f);  
  
 def\_p:  
 std::cout << "Продолжить ввод дробей? (y/n):";  
 std::cin >> y\_n;  
 switch (y\_n) {  
 case 'y':  
 break;  
 case 'n':  
 is\_loop = false;  
 break;  
 default:  
 goto def\_p;  
 }  
 }  
 double sum = 0;  
 for (auto & fraction : fractions) sum += fraction.value();  
  
 std::cout << "Среднее значение дробей = " << sum / static\_cast<double>(fractions.size()) << std::endl;  
}

1. Начните с программы (содержимое функции main()), которая позволяет пользователю вводить целые числа, а затем сохранять их в массиве типа int. Напишите функцию maxint(), которая, обрабатывая элементы массива один за другим, находит наибольший. Функция должна принимать в качестве аргумента адрес массива и количество элементов в нем, а возвращать индекс наибольшего элемента. Программа должна вызвать эту функцию, а затем вывести наибольший элемент и его индекс



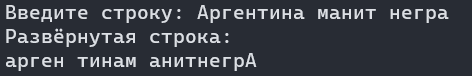


void lab5\_2() {  
 std::vector<int> nums {};  
 bool is\_loop = true;  
 while (is\_loop) {  
 char y\_n;  
 int i;  
  
 std::cout << "Введите число: ";  
 std::cin >> i;  
 nums.push\_back(i);  
  
 def\_p:  
 std::cout << "Продолжить ввод? (y/n):";  
 std::cin >> y\_n;  
 switch (y\_n) {  
 case 'y':  
 break;  
 case 'n':  
 is\_loop = false;  
 break;  
 default:  
 goto def\_p;  
 }  
 }  
  
 maxint(nums, nums.size());  
}

void maxint(std::vector<int>& nums, int amount) {  
 int max = 0;  
 for (int i = 0; i < amount; ++i)  
 if (nums[i] > max)  
 max = nums[i];  
  
 std::cout << "Наибольшее значение в массиве = " << max << std::endl;  
}

1. Напишите функцию reversit(), которая переворачивает строку (массив типа char). Используйте цикл for, который меняет местами первый и последний символы, затем следующие и т. д. до предпоследнего. Строка должна передаваться в функцию reversit() как аргумент.

Напишите программу для выполнения функции reversit(). Программа должна принимать строку от пользователя, вызывать функцию reversit(), а затем выводить полученный результат. Используйте метод ввода, который позволяет использовать внутренние пробелы. Протестируйте программу на примере фразы «Аргентина манит негра».

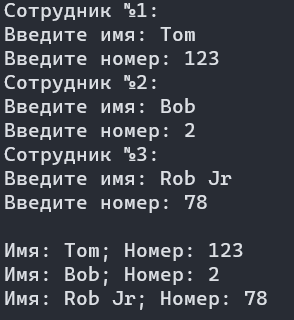


void lab5\_3() {  
 int size = 256;  
 char\* input = new char[size];  
  
 std::cout << "Введите строку: ";  
 std::cin.ignore();  
 std::cin.getline(input, size, '\n');  
  
 auto real\_size = 0;  
 int i = 0;  
 while (input[i] != '\0') {  
 real\_size++;  
 i++;  
 }  
  
 reversit(input, real\_size);  
}

void reversit(char\* input, int size) {  
 for (int i = 0; i < size / 2; ++i) {  
 auto c = input[i];  
 input[i] = input[size - i - 1];  
 input[size - i - 1] = c;  
 }  
  
 std::cout << "Развёрнутая строка:" << std::endl;  
 std::cout << input << std::endl;  
}



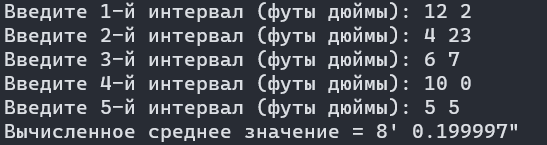
1. Создайте структуру employee, которая содержит имя (массив типа char) и номер (типа long) служащего. Напишите функцию getdata(), предназначенную для получения данных от пользователя и помещения их в структурную переменную, и функцию putdata(), для вывода данных. Предполагаем, что имя может иметь внутренние пробелы. Напишите функцию main(), использующую эту структуру. Вам нужно будет создать массив типа employee, а затем предложить пользователю ввести данные на трёх служащих. Наконец, вам нужно будет вывести данные всех служащих.





void lab5\_4() {  
 struct employee {  
 employee() {  
 name = new char[256];  
 number = 0;  
 }  
  
 char\* name;  
 long number;  
  
 void getdata() {  
 std::cout << "Введите имя: ";  
 std::cin.ignore();  
 std::cin.getline(name, 256, '\n');  
  
 std::cout << "Введите номер: ";  
 std::cin >> number;  
 }  
  
 void putdata() {  
 std::cout << "Имя: " << name << "; Номер: " << number << std::endl;  
 }  
 } e1 {}, e2 {}, e3 {};  
  
  
 std::cout << "Сотрудник №1:" << std::endl;  
 e1.getdata();  
 std::cout << "Сотрудник №2:" << std::endl;  
 e2.getdata();  
 std::cout << "Сотрудник №3:" << std::endl;  
 e3.getdata();  
 std::cout << std::endl;  
  
 e1.putdata();  
 e2.putdata();  
 e3.putdata();  
 std::cout << std::endl;  
}

1. Напишите программу, вычисляющую среднее значение 5-ти интервалов, введенных пользователем. Создайте структуру Distance, содержащую поля футов и дюймов. Создайте массив типа Distance. Напишите функцию вычисления среднего значения интервалов, хранящихся в массиве. Функция должна получать в качестве параметров массив и его размер. В качестве результата функция должна возвращать вычисленное среднее значение.



struct Distance {  
 int feet;  
 int inches;  
  
 Distance() {  
 feet = 0;  
 inches = 0;  
 }  
};

void max\_distance(Distance\* d, int size) {  
 float inches = 0, feet = 0;  
  
 for (int i = 0; i < size; ++i)  
 inches += (float)d[i].feet \* 12 + (float)d[i].inches;  
  
 inches /= (float)size;  
  
 while (inches >= 12) {  
 feet++;  
 inches -= 12;  
 }  
  
 std::cout << "Вычисленное среднее значение = " << feet << "\' " << inches << "\"" << std::endl;  
}

void lab5\_5() {  
 Distance d[5] {{}, {}, {}, {}, {}};  
  
 std::cout << "Введите 1-й интервал (футы дюймы): ";  
 std::cin >> d[0].feet >> d[0].inches;  
 std::cout << "Введите 2-й интервал (футы дюймы): ";  
 std::cin >> d[1].feet >> d[1].inches;  
 std::cout << "Введите 3-й интервал (футы дюймы): ";  
 std::cin >> d[2].feet >> d[2].inches;  
 std::cout << "Введите 4-й интервал (футы дюймы): ";  
 std::cin >> d[3].feet >> d[3].inches;  
 std::cout << "Введите 5-й интервал (футы дюймы): ";  
 std::cin >> d[4].feet >> d[4].inches;  
  
 max\_distance(d, 5);  
}

