МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 3**

по дисциплине:” Системное программирование”

на тему: “Компиляция и отладка простейшего приложения в Linux”

Выполнил**:** студент группы 10701321

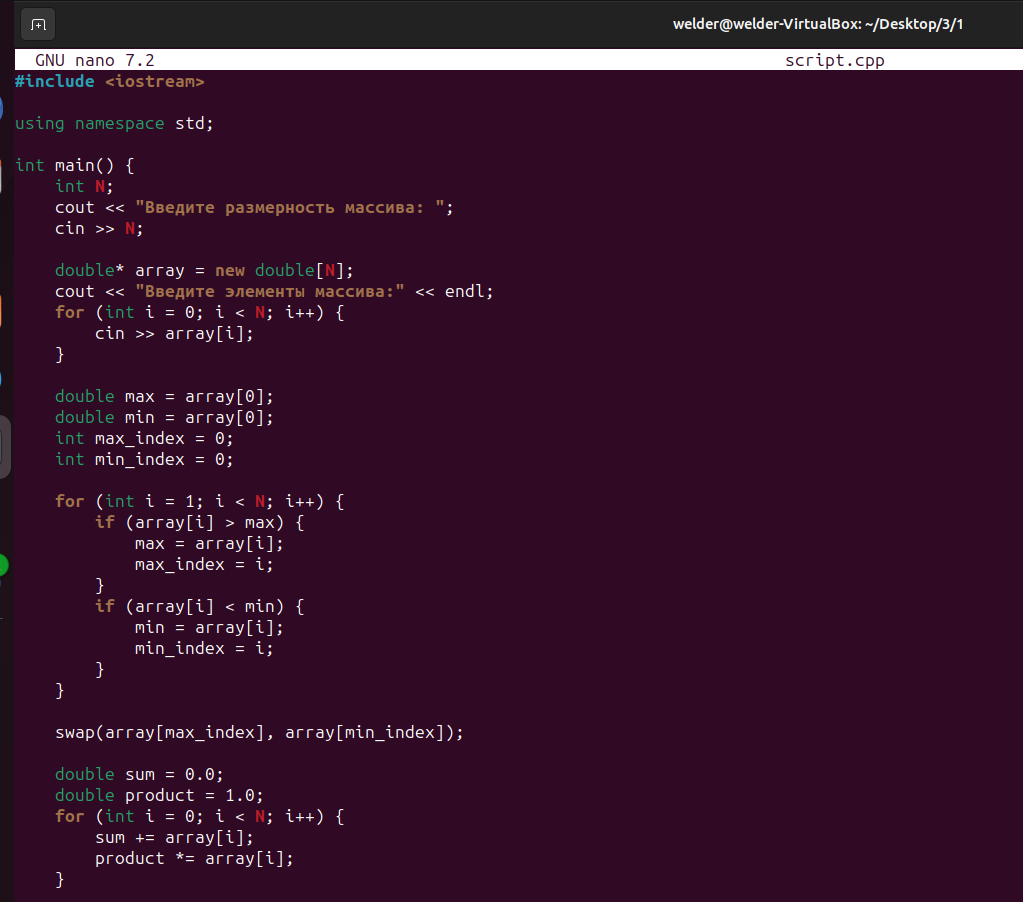
Рабштина Ю.В.

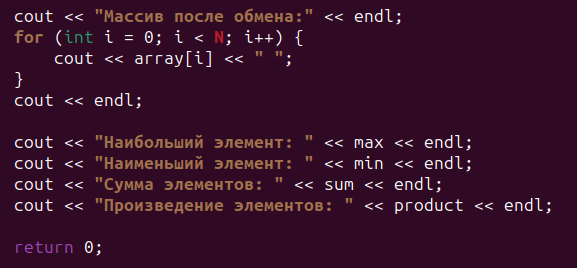
Принял**:** Давыденко Н.В.

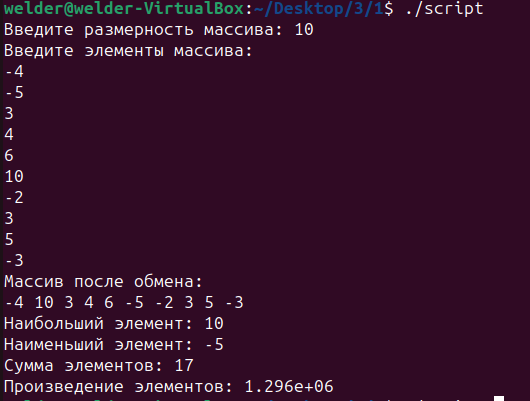
Минск 2023

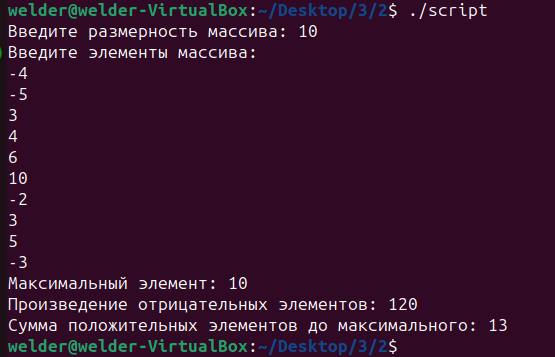
# Лабораторная работа № 3. Компиляция и отладка простейшего приложения в Linux

# Цель работы: Изучить встроенный инструментарий для разработки приложений под семейство ОС Linux и фундаментальные основы системного программирования с использованием компиляторов gcc/g++, отладчика gdb и других для проектирования, компиляции, отладки и запуска приложений на языке программирования С/C++.

Основное задание  
Ввести массив вещественных чисел размером N. Найти его наибольший и наименьший элементы и поменять их местами. Найти сумму и произведение всех элементов массива.





Индивидуальное задание9) В одномерном массиве, состоящем из n вещественных элементов, вычислить: произведение отрицательных элементов массива и сумму положительных элементов массива, расположенных до максимального элемента.  
  
  
  
  
  
Вывод*:* В ходе выполнения лабораторной работы научились компилировать программы с помощью g++  
  
Контрольные вопросы:  
1. Опишите основные команды отладчика gdb (выполнение программы, просмотр переменных, вывод листинга программы, установка точек останова, вставка исправлений с помощью отладчика).

1. Запуск программы:
   * run (или r): Запускает программу с начала выполнения. Можно указать аргументы командной строки после команды run.
2. Остановка выполнения:
   * break (или b): Устанавливает точку останова в коде программы. Например, break main установит точку останова в функции main.
   * tbreak: Устанавливает временную точку останова, которая срабатывает только один раз.
3. Выполнение кода:
   * next (или n): Выполняет следующую строку кода и переходит к следующей строке на текущем уровне вызова функции.
   * step (или s): Входит в функцию, если она есть, и выполняет первую строку функции.
   * continue (или c): Продолжает выполнение до следующей точки останова.
4. Трассировка стека:
   * backtrace (или bt): Выводит стек вызовов, отображая, какие функции вызываются и в каком порядке.
5. Просмотр переменных:
   * print (или p): Выводит значение переменной. Например, print x выведет значение переменной x.
6. Установка условных точек останова:
   * watch: Устанавливает точку останова на изменение значения переменной. Например, watch x остановит выполнение программы, если значение переменной x изменится.
7. Просмотр исходного кода:
   * list (или l): Выводит исходный код программы с текущей позиции. Можно указать номер строки, чтобы просмотреть конкретную часть кода. Например, list 10 покажет код с 10-й строки.
8. Изменение значений переменных:
   * set variable (или set var): Позволяет изменить значение переменной. Например, set variable x = 42 установит значение переменной x равным 42.
9. Выход из GDB:
   * quit (или q): Выходит из GDB.