МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный

электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра «Информационные системы»

отчет

по лабораторной работе №1

по дисциплине «Программирование»

Тема: "Типы данных и их внутреннее представление в памяти"

|  |  |  |
| --- | --- | --- |
| Студент гр. 0323 |  | Землянский Д.А. |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2020

Цель работы.

Ознакомиться со стандартными типами данных, операций и выражений. А также узнать, что такое “время жизни”, “область видимости”, ввод-вывод типов данных.

Основные теоретические положения.

Бит – двоичная ячейка в памяти компьютера, которая может находиться в

двух состояниях, условно обозначаемых 0 и 1

Байт - единица хранения и обработки цифровой информации,

совокупность 8 битов, обрабатываемая компьютером одновременно,

минимальный независимо адресуемый набор данных.

Тип данных – класс данных, характеризуемый членами класса и

операциями, которые могут быть к ним применены (ISO/IEC/IEEE 24765-2017).

Основные (предопределенные) типы данных определены следующими

ключевыми словами:

int (целый);

float (вещественный);

double (вещественный тип с двойной точностью);

bool (логический);

char (символьный).

Существует четыре спецификатора типа, уточняющих внутреннее

представление и диапазон значений стандартных типов

short (короткий)

long (длинный);

signed (знаковый);

unsigned (беззнаковый).

Типы int, bool и char относят к группе целочисленных (целых) типов,

представлены в памяти целым числом в двоичном коде. При этом для знаковых

типов (спецификатор signed) первый бит выделен для хранения информации о

знаке.

Число записывается в память в двоичной системе исчисления в

дополнительном коде. Дополнительный код положительного числа равен числу

в прямом коде. Дополнительный код отрицательного числа получается путём

прибавления единицы к младшему разряду отрицательного числа, записанного

в обратном коде. Обратный код отрицательного числа, в свою очередь,

получается путём инвертирования всех его разрядов.

Внутреннее представление вещественного числа состоит из двух частей

— мантиссы и порядка. Под тип float в памяти отводится 32 бита (первый бит –

знаковый, 8 бит – порядок, 23 – мантисса). Аналогично представляется и тип

double, за тем исключением, что под порядок отводится 11 бит, а под мантиссу

– 52 бита.

Для работы с побитовый представлением данных в языке

присутствуют операции:

~a - побитового отрицания

a&b – поразрядной конъюнкции

a|b – поразрядной дизъюнкции

a^b – поразрядной исключающей дизъюнкции

a<<b – побитового сдвига влево

a>>b - побитового сдвига вправо

Область видимости определяет, где можно использовать переменную.

Продолжительность жизни определяет, где переменная создается и где

уничтожается.

Постановка задачи.

Разработать алгоритм и написать программу, которая позволяет:

Вывести, сколько памяти (в байтах) на вашем компьютере отводится под различные типы данных со спецификаторами и без: int, short int, long int, float, double, long double, char и bool.

Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды отступами или цветом.

Вывести на экран двоичное представление в памяти (все разряды) типа float. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

Вывести на экран двоичное представление в памяти (все разряды) типа double. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок (\*)

Выполнение работы.

Для решения поставленной задачи была написана программа в

Visual Studio 2017 на языке C++. Итоговый код программы представлен в

приложении А.

Выводы.

Поскольку стандартом C++ не установлен объём памяти отводимый под

различные типы данных, при написании программ, предназначенных для

выполнения на различных ЭВМ предпочтительно использование для

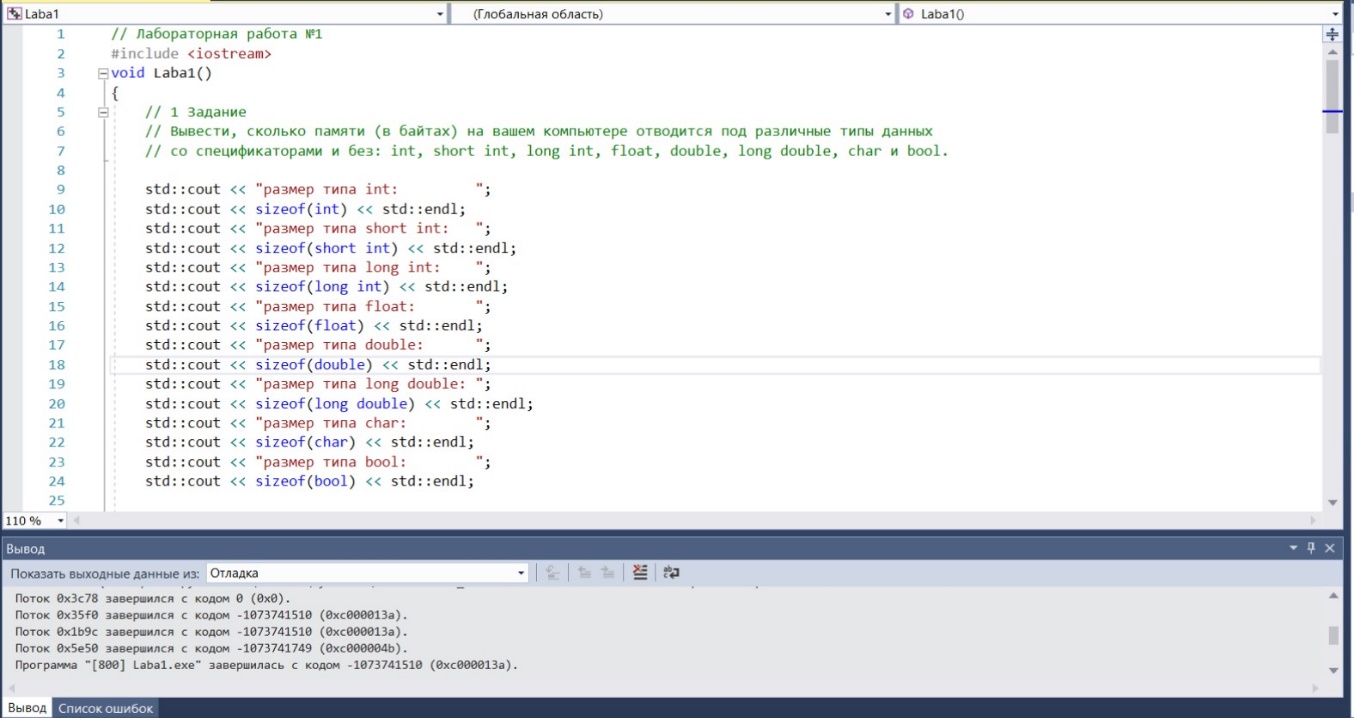
представления целых чисел либо типа char, либо типов с фиксированной

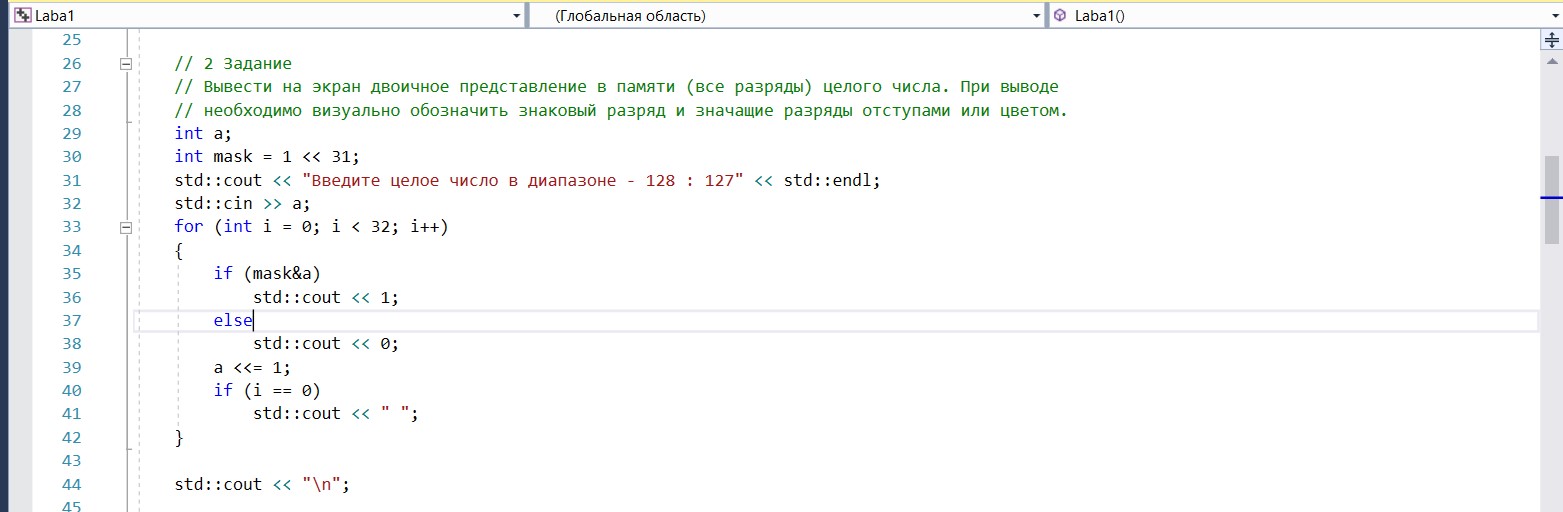
длиной, определённых в заголовочном файле cstdint, введённые стандартом C+

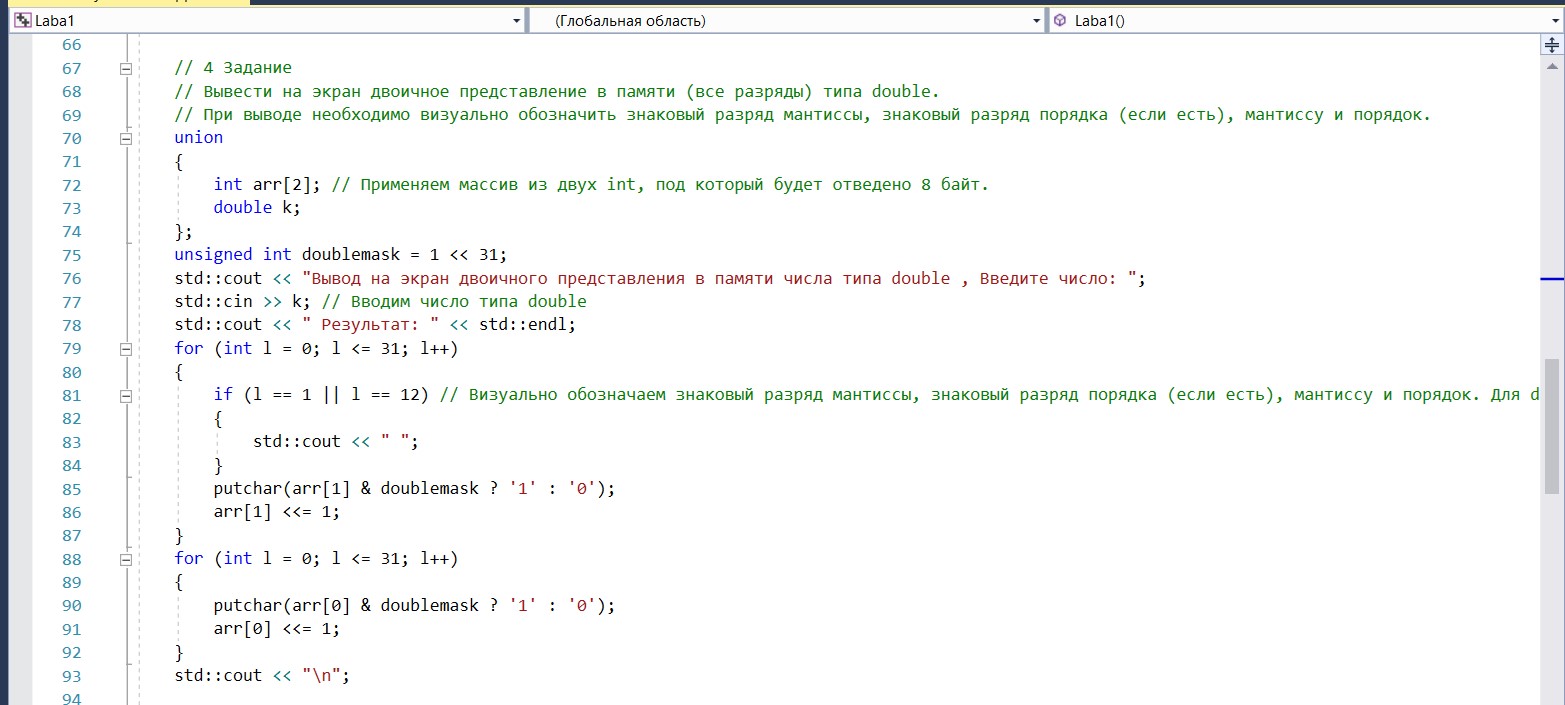
+11, во всех случаях, когда важен размер памяти, отводимый под данный тип.

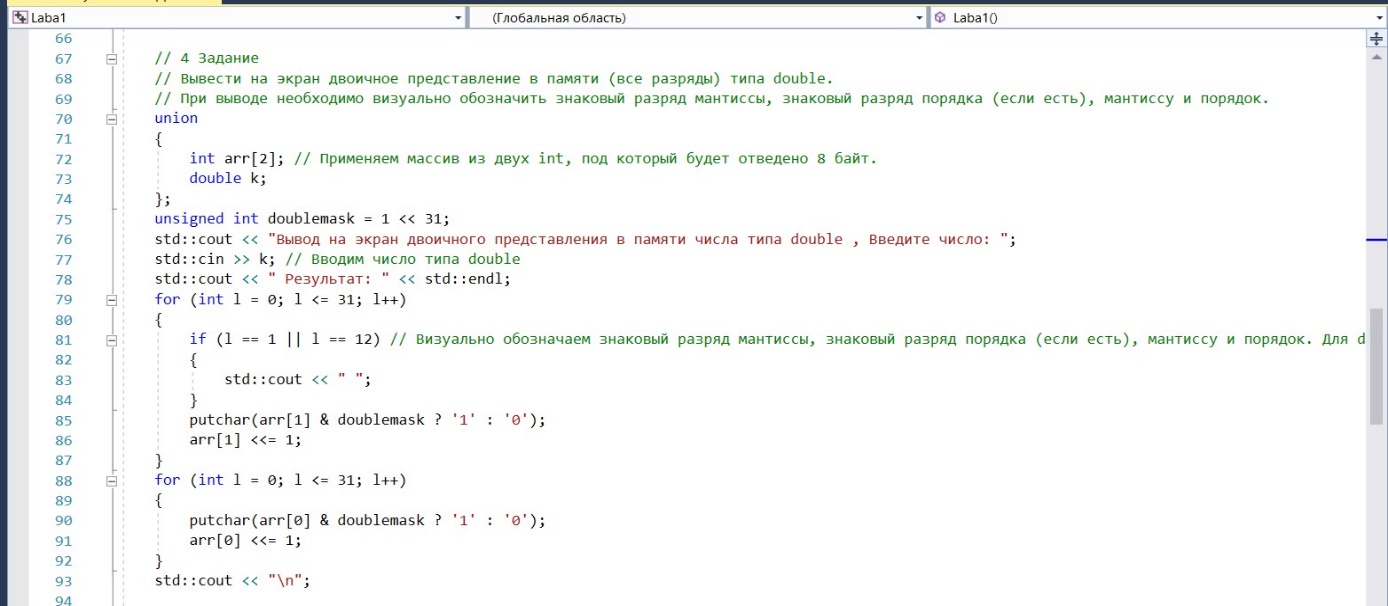
Приложение А

Полный код программы

****

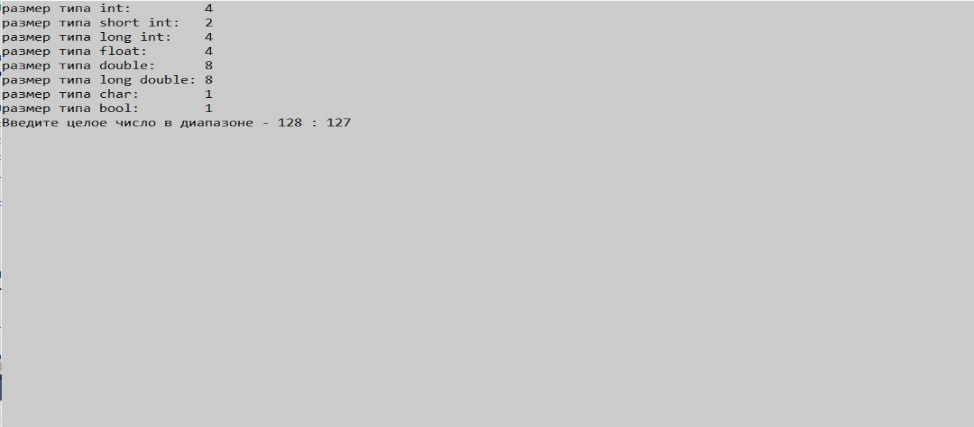
****

****

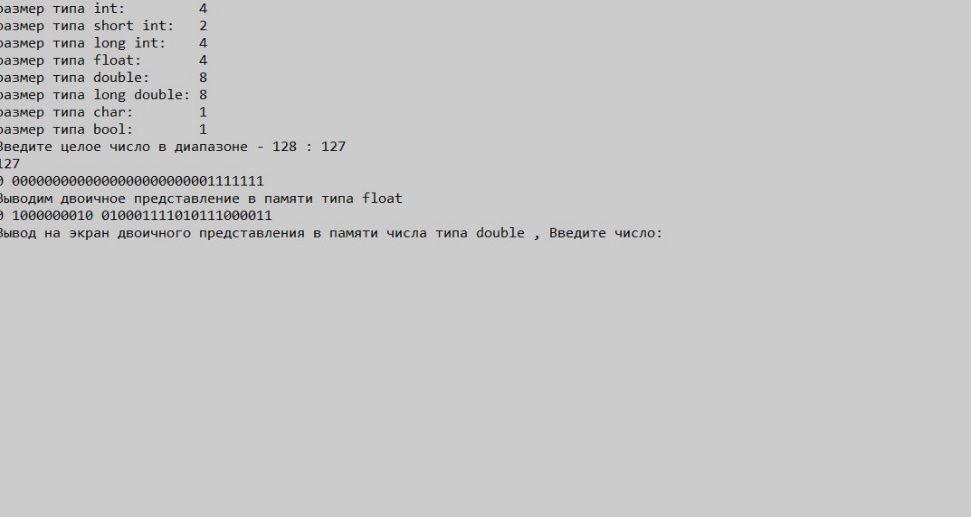
****

Приложение Б

ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ



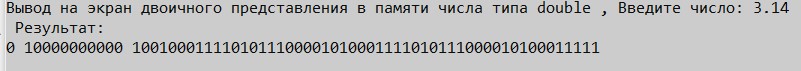
Результат 1 задания



Результат 2 задания



Результат 3 задания



Результат 4 задания