**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Информационных систем**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: Типы данных и их внутреннее представление в памяти

|  |  |  |
| --- | --- | --- |
| Студент гр. 0323 |  | Кольцов К.Э |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2020

**Цель работы.**

Разработать алгоритм и написать программу, которая позволяет:

1) 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.

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

3) 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.

4) 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.

**Основные теоретические положения.**

Числа в памяти компьютера представлены в виде целого двоичного числа. Для знаковых чисел старший разряд числа используется для обозначения знака: ноль для неотрицательных чисел и единица для отрицательных. Для простоты внутренней арифметики последние представлены в дополненном коде. Он получается прибавлением единицы к обратному коду числа (в котором все разряды модуля числа инвертированы).

Вещественные типы бывают только знаковыми. Для них отличается представление в памяти. Они переводятся в нормализованную форму двоичной системы счисления. Так как нормализованная мантисса в двоичной системе счисления больше либо равна единице и меньше двух, значит она всегда начинается с единицы и её можно не записывать в память, освободив место для еще одного разряда порядка или мантиссы. У порядка знакового бита нет, вместо этого к нему прибавляется половина диапазона. Например, порядок равный нулю будет начинаться нулём, а все остальные разряды будут единицами.

Для нахождения размера занимаемой памяти используется функция sizeof, возвращающая количество байт для соответствующего типа данных. Для вывода представления чисел в памяти используется & (оператор побитового И) вместе с числом с единственной единицей в двоичной записи (два в неотрицательной степени), а для вещественных типов данных union (объединение), для получения доступа в области памяти через целочисленные типы (int для float и long long для double).

**Экспериментальные результаты.**

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

Затем по порядку считываются, обрабатываются и выводятся двоичные представления в памяти типов int, float, double.

После этого программа завершает работу (см. рис. 1)

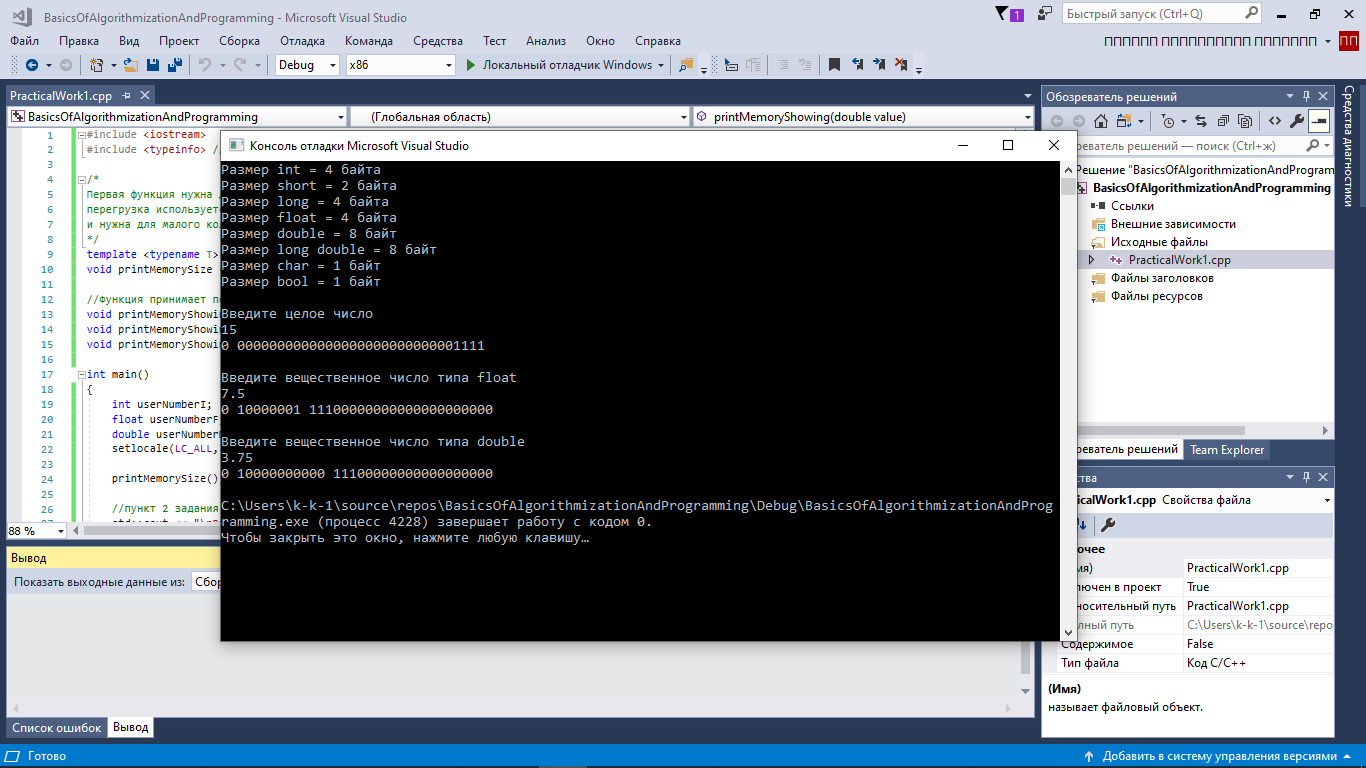


Рисунок 1 – Пример работы программы

**Обработка результатов эксперимента.**

Программа выводит корректные размеры выделяемой памяти для разных типов данных и представление чисел в памяти компьютера.

**Выводы.**

В ходе данной лабораторной работы я изучил внутреннее представление различных типов данных в памяти компьютера.

ПРИЛОЖЕНИЕ А

Листинг программы

#include <iostream>

#include <typeinfo>

template <typename T> void printMemorySize();

void printMemorySize ();

void printMemoryShowing(double);

void printMemoryShowing(float);

void printMemoryShowing(int);

int main()

{

int userNumberI;

float userNumberF;

double userNumberD;

setlocale(LC\_ALL, "Russian"); //Коректный вывод кириллицы

printMemorySize(); //пункт 1 задания

std::cout << "\nВведите целое число\n";

std::cin >> userNumberI;

printMemoryShowing(userNumberI);

std::cout << "\nВведите вещественное число типа float\n";

std::cin >> userNumberF;

printMemoryShowing(userNumberF);

std::cout << "\nВведите вещественное число типа double\n";

std::cin >> userNumberD;

printMemoryShowing(userNumberD);

return 0;

}

void printMemorySize()

{

printMemorySize<int>();

printMemorySize<short int>();

printMemorySize<long int>();

printMemorySize<float>();

printMemorySize<double>();

printMemorySize<long double>();

printMemorySize<char>();

printMemorySize<bool>();

}

template<typename T>

void printMemorySize()

{

T t;

switch (sizeof(T)%10)

{

case 2:

case 3:

case 4:

std::cout << "Размер "<< typeid(t).name() <<" = " << sizeof(T) << " байта\n";

break;

default:

std::cout << "Размер "<< typeid(t).name() <<" = " << sizeof(T) << " байт\n";

}

}

void printMemoryShowing(int value)

{

unsigned int mask = 2147483648;

for (int i = 0;i < 32;i++)

{

std::cout << (value & mask ? '1' : '0');

mask >>= 1;

if (i == 0)

std::cout << ' ';

}

std::cout << '\n';

}

void printMemoryShowing(float value)

{

union {

float valueF;

int valueI;

};

valueF = value;

unsigned int mask = 2147483648;

for (int i = 0;i < 32;i++)

{

std::cout << (valueI & mask ? '1' : '0');

mask >>= 1;

if ((i == 0)||(i==8))

std::cout << ' ';

}

std::cout << '\n';

}

void printMemoryShowing(double value)

{

union {

double valueD;

long long valueLL;

};

valueD = value;

unsigned long long mask = 9223372036854775808;

for (int i = 0;i < 32;i++)

{

std::cout << (valueLL & mask ? '1' : '0');

mask >>= 1;

if ((i == 0) || (i == 11))

std::cout << ' ';

}

std::cout << '\n';

}