**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

Тема: Типы данных, определяемые пользователем.

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. 2373 |  | Суднищиков М.К. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2021

**Цель работы.**

Разработать алгоритм и написать программу, которая позволяет, понять сколько памяти занимают основные типы данных, понять, как работает двоичное представление числа в языке C++.

**Основные теоретические положения.**

Бит информации – символ или сигнал, который может принимать два значения: включено или выключено, да или нет, высокий или низкий, заряженный или незаряженный; в двоичной системе исчисления это 1 (единица) или 0 (ноль).

Тип данных – тип никогда не изменяется (не следует путать со статическими объектами данных). При объявлении переменной в коде необходимо явно указать его тип или использовать ключевое слово, чтобы указать компилятору выводить тип из инициализатора. При объявлении функции в коде необходимо указать тип каждого аргумента и его возвращаемое значение. В отличие от некоторых других языков, в C++ нет универсального базового типа, от которого наследуются все остальные типы. Язык включает множество фундаментальных типов, также известных как встроенные типы. Сюда входят числовые типы, такие как int, double, longbool, а также char типы и wchar\_t типы для символов ASCII и ЮНИКОД соответственно. Большинство целочисленных фундаментальных типов (за исключением bool, doublewchar\_tи связанных типов) имеют unsigned версии, которые изменяют диапазон значений, которые может хранить переменная. Например, int32-разрядное целое число со знаком может представлять значение от -2 147 483 648 до 2 147 483 647. Значение unsigned int, которое также хранится как 32-разрядные, может хранить значение от 0 до 4294 967 295. Общее количество возможных значений в каждом случае одинаково, отличается только диапазон.

**Постановка задачи.**

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.

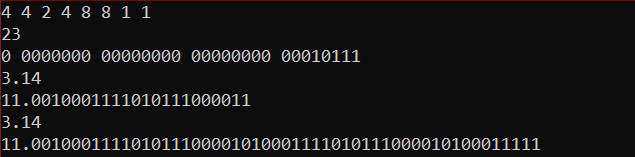
**Выполнение работы.**

Код программы представлен в приложении А.

Блок описания кода и использованных алгоритмов

В моем коде используются циклы и простейшие типы данных.

Блок скриншотов работы программы



**Выводы.**

Мы вывели двоичное значение типа данных int и float. Понял как устроено двоичное представление числа в языке С++.

Приложение А

рабочий код

#include <iostream>

#include <string>

using namespace std;

int main()

{

int a,j; // Задание 1

int long b;

short int c;

float d,x;

double e;

long double f;

char g;

bool k;

cout << sizeof(a) << " " << sizeof(b) << " " << sizeof(c) << " " << sizeof(d)

<< " " << sizeof(e) << " " << sizeof(f) << " " << sizeof(g) << " " << sizeof(k) << " " <<endl;

cin >> j; // Задание 2

int value = j; // Значение числа

unsigned int order = 32; // Количество разрядов

unsigned int mask = 1 << order - 1; // Маска побитового сравнения

for (int i = 1; i <= order; i++)

{

putchar(value & mask ? '1' : '0');

value <<= 1; // Побитовый сдвиг числа

if (i % 8 == 0)

{

putchar(' ');

}

if (i % order - 1 == 0)

{

putchar(' ');

}

}

cout << endl;

cin >> x; //Задание 3

string ret;

unsigned int z = x; //целая часть

double r = x - z; //дробная часть

while (z != 0)

{

ret = (char)(z % 2 + 0x30) + ret;

z = z / 2;

}

int counter = 0;

if (r != 0) ret += ".";

while (r != 0)

{

z = r \* 2;

ret += (char)(z + 0x30);

r = r \* 2 - z;

++counter;

}

cout << ret;

return 0;

}