**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 и их внутреннее представление в памяти

|  |  |  |
| --- | --- | --- |
| Студент гр. 2372 |  | Соколовский В.Д. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2022

**Цель работы.**

Изучение числовых типов данных и их внутреннего представления в памяти компьютера. Написание программы, показывающей объём памяти, занимаемый разными типами данных и визуализирующей представление чисел в памяти.

**Основные теоретические положения.**

Представление положительных и отрицательных числе в прямом, обратном и дополнительном кодах отличается. В прямом коде в знаковый разряд помещается цифра 1, а в разряды цифровой части числа – двоичный код его абсолютной величины.

Обратный код получается инвертированием всех цифр двоичного кода абсолютной величины, включая разряд знака. Прямой код можно преобразовать в обратный, инвертировав все значения битов.

Дополнительный код получается образованием обратного кода с прибавлением единицы к его младшему разряду.

Увидеть, как тип данных представляется на компьютере, можно при помощи побитового сдвига и поразрядной конъюнкции.

При сдвиге вправо для положительных чисел освобожденные позиции битов, заполняются нулями. Для отрицательных – единицами. Сдвиг влево является логическим сдвигом (биты, сдвигаемые с конца, отбрасываются, включая бит знака).

Внутреннее представление вещественного числа состоит из двух частей – мантиссы и порядка (экспоненты). Для 32-разрядного процессора для float под мантиссу отводится 23 бита, под экспоненту – 8, под знак – 1. Для double под мантиссу отводится 52 бита, под экспоненту – 11, под знак – 1.

**Постановка задачи.**

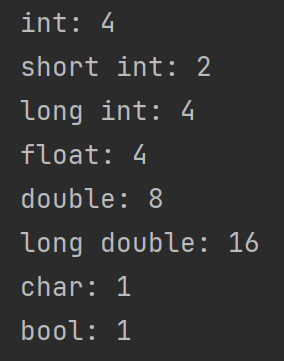
Разработать алгоритм и написать программу, которая позволяет:

1. Вывести, сколько памяти (в байтах) отводится под различные типы данных со спецификатором и без: int, short int, long int, float, double, long double, char, bool.
2. Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды.
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 ест), мантиссу и порядок.

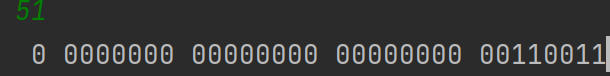
**Выполнение работы.**

Код программы представлен в приложении А.

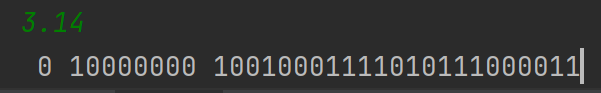
1. При запуске программы выводятся объёмы памяти, отводимые под различные типы данных со спецификатором.



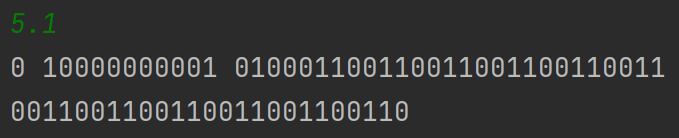
1. Ожидается ввод числа типа int, двоичное представление которого потом будет выведено на экран.



1. Ожидается ввод вещественного числа типа float, двоичное представление которого потом будет выведено на экран.



1. Ожидается ввод вещественного числа типа double, двоичное представление которого потом будет выведено на экран.



**Выводы.**

В ходе работы было изучено представление числовых типов данных в памяти компьютера, побитовые операции сдвига и поразрядной конъюнкции, методы перевода целых и вещественных чисел в двоичный код.

Приложение А

рабочий код

грinclude <iostream>  
using namespace std;  
int main() {  
 // задание 1  
 int a;  
 short int b;  
 long int c;  
 float d;  
 double e;  
 long double f;  
 char g;  
 bool h;  
 cout << "int: " << sizeof(a) << "\n" << "short int: " << sizeof(b) << "\n" << "long int: "  
 << sizeof(c) << "\n" << "float: "<< sizeof(d) << "\n" << "double: " << sizeof(e) <<  
 "\n" << "long double: " << sizeof(f) << "\n" << "char: "<< sizeof(g) << "\n"<< "bool: " << sizeof(h);  
  
  
  
// задание 2  
 int n;  
 cin>>n;  
 int m = sizeof(n)\*8 - 1;  
 int mask = 1 << m;  
 for(int i = 0; i <= m; i ++){  
 if(i == 1 || i % 8 == 0){  
 putchar(' ');  
  
 }  
 putchar(n & mask ? '1' : '0');  
 mask = (unsigned)mask >> 1;  
  
 }  
 cout<< " ";  
  
  
  
// задание 3  
 union {  
 float fl;  
 int nt;  
 };  
 cin>>fl;  
 unsigned int line = 32;  
 mask = 1 << line - 1;  
 for(int p = 1; p <= line; p++){  
 putchar(nt & mask ? '1':'0');  
 nt <<= 1;  
 if(p == 9){  
 putchar(' ');  
 }  
 if(p == 1){  
 putchar(' ');  
  
 }  
  
  
 }  
 cout<< "";  
  
 // задание 4  
 int qa = 0;  
 union{  
 double dbl;  
 int mas[2];  
 };  
 cin>>dbl;  
 unsigned int lines = 32;  
 mask = 1;  
 mask <<= (lines - 1);  
 for(int v = 0; v < 2; v++){  
 if(v == 0){  
 qa += 1;  
 }  
 else{  
 qa -= 1;  
 }  
 for(int po = 0; po < lines; po++){  
 putchar(mas[qa] & mask ? '1':'0');  
 mas[qa] <<= 1;  
 if((po == 0) and (v == 0)){  
 putchar(' ');  
 }  
 if ((po == 11) and (v == 0)){  
 putchar(' ');  
 }  
 }  
 }  
  
 return 0;  
}