**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. Структуры

|  |  |  |
| --- | --- | --- |
| Студентка гр. 2373 |  | Маркова З.А. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2022

**Цель работы:**

Изучение типов данных и их внутреннего представления в памяти; получение практических навыков работы с типами данных; определить, каким образом типы данных представляются на компьютере.

**Основные теоретические положения:**

Программе необходимо точно представлять какие данные хранятся в этом байте памяти.

Для разрешения подобных коллизий в языках программирования введено **понятие типов данных**.

Тип данных для каждого программного объекта, представляющего данные, определяет:

* характер данных (число, со знаком или без знака, целое или с дробной частью, одиночный символ или текст, представляющий последовательность символов и т.д.);
* объем памяти, который занимают в памяти эти данные;
* диапазон или множество возможных значений;
* правила обработки этих данных (например, допустимые операции)

В разных языках программирования определены разные наборы типов данных, но, в целом, типы данных можно разделить на две группы: простые и структурированные типы. Простые типы данных представляют неразделимые данные, не имеющие внутренней структуры (это, например, числа, символы и т.д.). Структурированные типы данных, как это вытекает из их названия, имеют внутреннюю структуру (иногда достаточно сложную). Структурированные типы строятся на основе простых типов данных.

Другой уровень классификации разделяет все типы данных на предопределенные (изначально встроенные в язык программирования) и пользовательские (типы данных, определяемые программистом) типы данных.

Основные (предопределенные) типы данных часто называют арифметическими, поскольку их можно использовать в арифметических операциях.

Типы **int**,**bool**и**char** относят к группе целочисленных (целых) типов, а **float**и**double** - к группе вещественных типов - типов с плавающей точкой. Код, который формирует компилятор для обработки целых величин, отличается от кода для величин с плавающей точкой.

Существует четыре спецификатора типа, уточняющих внутреннее представление и диапазон значений стандартных типов: **short**(короткий); **long** (длинный); **signed** (знаковый); **unsigned** (без знаковый).

**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.

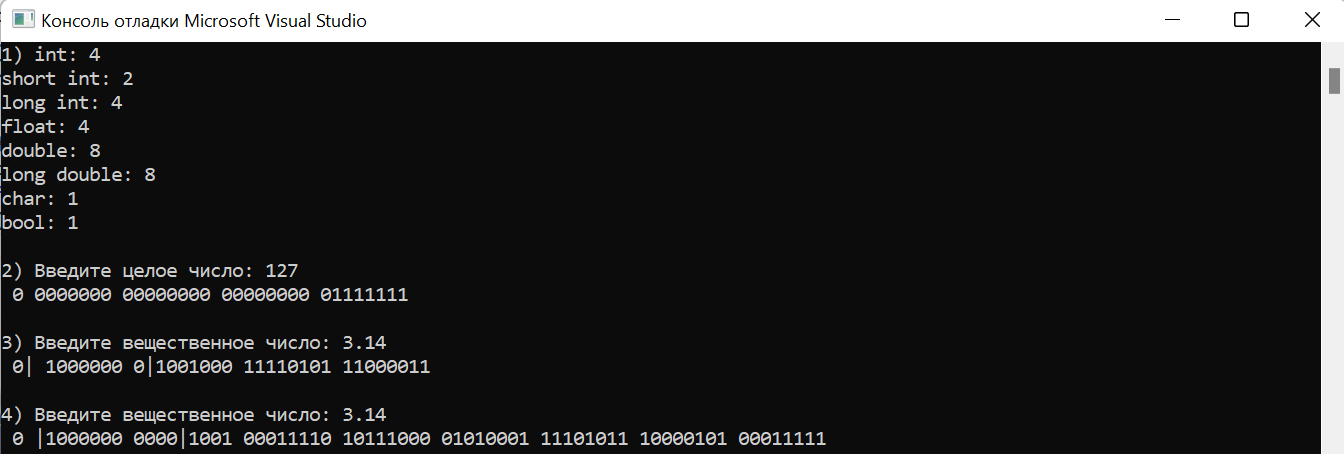
3) Вывести на экран двоичное представление в памяти (все разряды) типа float. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

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.

**Выполнение работы:**

Код программы представлен в приложении А.

1. На экран выводится, сколько памяти (в байтах) на компьютере отводится под различные типы данных со спецификаторами и без в таком порядке: int, short int, long int, float, double, long double, char, bool.
2. Программа просит ввести целое число и считывает введенное путём функции cin.
3. После определения маски и количества разрядов, программа проходит через цикл for, сравнивая биты и разделяя их
4. Объединение в памяти переменных формата float и int с помощью union, что поможет обойти ограничение с невозможностью применения операций сдвига к типу float.
5. Программа просит ввести вещественное число и считывает введенное.
6. Программа выводит число, записанное в типе данных float.
7. Программа просит ввести вещественное число и считывает введенное.
8. Программа выводит число, записанное в типе данных double.

**Результаты работы программы:**

**Выводы:**

В этой практической работе мы изучили типы данных и их внутреннее представления в памяти; получили практические навыки работы с типами данных; определили, каким образом типы данных представляются на компьютере.

**ПРИЛОЖЕНИЕ А**

**РАБОЧИЙ КОД**

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

//1)

cout << "1) " << "int: " << sizeof(int) << '\n';

cout << "short int: " << sizeof(short) << '\n';

cout << "long int: " << sizeof(long int) << '\n';

cout << "float: " << sizeof(float) << '\n';

cout << "double: " << sizeof(double) << '\n';

cout << "long double: " << sizeof(long double);

cout << '\n' << "char: " << sizeof(char);

cout << '\n' << "bool: " << sizeof(bool) << '\n';

cout << "\n";

//2)

cout <<"2) "<< "Введите целое число: ";

int a;

cin >> a;

int s = sizeof(a) \* 8;

int mask = 1 << s - 1;

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

{

if (i % 8 == 0 || i == 1)

{

putchar(' ');

}

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

a <<= 1;

}

cout << "\n";

cout << "\n";

//3)

cout <<"3) "<< "Введите вещественное число: ";

union

{

int b;

float fl;

};

cin >> fl;

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

{

if ( i % 8 == 0 || i == 1 )

{

putchar(' ');

}

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

b <<= 1;

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

{

putchar('|');

//'|'- разделение знакового разряда, порядка и мантисы

}

}

cout << "\n";

cout << "\n";

//4)

cout <<"4) "<< "Введите вещественное число: ";

union

{

int c[2];

double db;

};

cin >> db;

int s4 = sizeof(db) \* 8;

unsigned int mask4 = 1 << s4 - 1;

for (int i = 0; i <s4/2; i++)

{

if (i % 8 == 0 || i == 1)

{

putchar(' ');

}

if (i == 1 || i == 12)

{

putchar('|');

//'|'- разделение знакового разряда, порядка и мантисы

}

putchar(c[1] & mask4 ? '1' : '0');

c[1] <<= 1;

}

for (int j = 0; j < s4/2; j++)

{

if (j % 8 == 0)

{

putchar(' ');

}

putchar(c[0] & mask4 ? '1' : '0');

c[0] <<= 1;

}

}