**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 0324 |  | Жигалова Д. А. |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2020

ОГЛАВЛЕНИЕ

[**ЦЕЛЬ РАБОТЫ** 3](#_Toc54297292)

[**ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ** 3](#_Toc54297293)

[**ПОСТАНОВКА ЗАДАЧИ** 4](#_Toc54297294)

[**ВЫПОЛНЕНИЕ РАБОТЫ** 5](#_Toc54297295)

[**КОД ПРОГРАММЫ** 5](#_Toc54297296)

[**ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ** 11](#_Toc54297297)

[**ВЫВОДЫ** 12](#_Toc54297298)

# **ЦЕЛЬ РАБОТЫ**

Знакомство с внутренним представлением различных типов данных, используемых компьютером при их обработке. Научиться работать с побитовыми операциями.

# **ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ**

Любая программа предназначена для обработки данных. Данные бывают различного типа, причем хранятся и обрабатываются они по-разному. Любые данные хранятся в памяти компьютера в виде двоичных кодов. Память компьютера представляет собой непрерывную последовательность двоичных ячеек, каждая из которых может находиться в двух состояний, которые можно условно обозначить 0 и 1. Каждая такая двоичная ячейка называется битом. Последовательность, условно разбитая по 8 бит, называется байтами. Из этого следует, что 1 байт = 8 бит. Байт в свою очередь является основной единицей измерения объема памяти. В любом алгоритмическом языке каждая константа, переменная, результата вычисления выражения или функции должны иметь определённый тип, который определяет:

o характер данных (знаковое или число без знака, целое или с дробной частью, последовательность символов или одиночный символ и т.д.);

o объем памяти, который занимают в памяти эти данные;

o диапазон или множество возможных значений;

o операции и функции, которые можно применять к величинам этого типа

В языке С++ определено шесть основных типов данных для представления целых, вещественных, символьных и логических величин. К ним относятся массивы, перечисления, функции, структуры, ссылки, указатели, объединения и классы.

Для описания основных типов определены следующие ключевые слова:

o int (целый);

o char (символьный);

o bool (логический);

o float (вещественный);

o double (вещественный тип с двойной точностью).

Существует четыре спецификатора типа, уточняющих внутреннее представление и диапазон значений стандартных типов:

o short (короткий);

o long (длинный);

o signed (знаковый);

o unsigned (беззнаковый)

В разделе может быть приведено описание исследуемых физических явлений (с иллюстрациями), основные теоретические положения (в том числе – математический аппарат, описывающий исследуемые явления), схемы измерений, сведения об используемом при проведении работы лабораторном оборудовании.

# **ПОСТАНОВКА ЗАДАЧИ**

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**

Чтобы узнать, сколько отводиться под тот или иной тип данных или объект памяти, нужно использовать операцию sizeof. Операция sizeof вычисляет размер в байтах.

Вывести то, каким образом тип данных представляется на компьютере, можно при помощи логических операций: побитового сдвига (<<) и поразрядной конъюнкции (&), сравнивая маску с двоичным представлением числа.

Размер типа int зависит от компьютера и компилятора. Для 16-разрядного процессора под величины этого типа отводится 2 байта. Для 32-разрядного – 4 байта.

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.

Каким образом вещественные типы данных представляются в компьютере можно узнать при помощи объединения. Объединения позволяют интерпретировать несколькими различными способами содержимое одной и той же области памяти.

Под вещественное число с двойной точностью отводиться 8 байт. Так как данные любой линейной структуры в память записываются последовательно, нужно использовать массив из двух int, под который будет отведено 8 байт.

# **КОД ПРОГРАММЫ**

#include <iostream>

using namespace std;

int main()

{

int answer;

answer = 0;

while (answer != 5) {

cout << "\nWhat tasks do you wanna check? (Write task's number)\n";

cout << "1. Print how much memory(in bytes) on your computer is allocated for various data types withand without specifiers;\n";

cout << "2. Display the binary representation in memory (all digits) of an integer;\n";

cout << "3. Display a binary representation in memory (all bits) of the float type;\n";

cout << "4. Display a binary representation in memory (all digits) of the double type.\n";

cout << "To exit, enter 5 \n";

cin >> answer;

system("CLS");

switch (answer)

{

case (1):

int a;

short int b;

long int c;

float d;

double e;

long double f;

char g;

bool h;

cout << "int: " << sizeof(a) << "; short int: " << sizeof(b) << "; long int: " << sizeof(c) << "; float: " << sizeof(d) << "; double: " << sizeof(e) << "; long double: " << sizeof(f) << "; char: " << sizeof(g) << "; bool: " << sizeof(h) << "\n";

break;

case (2):

int toBin;

int mask;

mask = 1;

int num[32];

bool answer2;

while (true) {

cout << "\nEnter integer: \n";

cin >> toBin;

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

num[i] = toBin & mask ? 1 : 0;

toBin = toBin >> 1;

}

for (int i = 31; i >= 0; i--) {

if (i == 30)

cout << " ";

cout << num[i];

}

cout << "\nExit? 1/0";

cin >> answer2;

if (answer2) {

system("CLS");

break;

}

}

break;

case (3):

int mask3;

mask3 = 1;

int num3[32];

bool answer3;

union {

int tool;

float toBin2;

};

while (true) {

cout << "\nEnter float: \n";

cin >> toBin2;

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

num3[i] = tool & mask3 ? 1 : 0;

tool = tool >> 1;

}

for (int i = 31; i >= 0; i--) {

cout << num3[i];

if (i == 31 || i == 23) {

cout << " ";

}

}

cout << "\nExit? 1/0";

cin >> answer3;

if (answer3) {

system("CLS");

break;

}

}

break;

case (4):

bool num4[64];

bool answer4;

int index;

union {

double toBin3;

int tool2[2];

};

while (true) {

cout << "\nEnter double: \n";

cin >> toBin3;

index = 0;

for (int k = 0; k < 2; k++) {

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

num4[index] = 1 & tool2[k];

tool2[k] >>= 1;

index++;

}

}

for (int i = 63; i >= 0; i--) {

cout << num4[i];

if (i == 63 || i == 52) {

cout << " ";

}

}

cout << "\nExit? 1/0";

cin >> answer4;

if (answer4) {

system("CLS");

break;

}

}

break;

case (5):

cout << "Have a nice day!\n";

break;

default:

cout << "Oh, no! This task doesn't exist. Let's try again.\n";

break;

};

}

return 0;

}

# **ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ**

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 см. Рисунок 1.

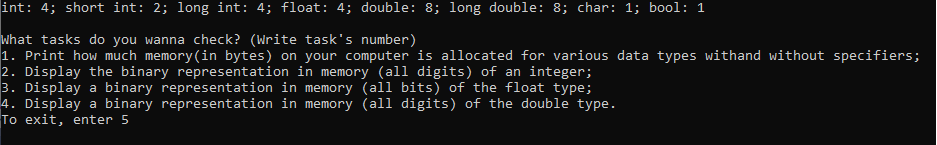


Рисунок 1 - Задание 1

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 см. Рисунок 2.

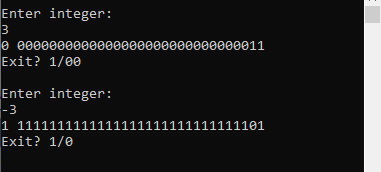


Рисунок 2 - Задание 2

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 см. Рисунок 3.

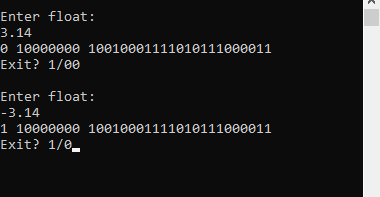


Рисунок 3 - Задание 3

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 см. Рисунок 4.

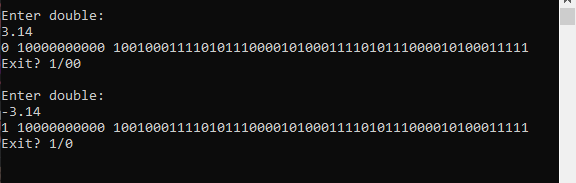
****

Рисунок 4 - Задание 4

# **ВЫВОДЫ**

В ходе проделанной работы были изучены бинарные операции. Познакомились с представлением разных типов данных в памяти.