**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: ТИПЫ ДАННЫХ И ИХ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ В ПАМЯТИ

|  |  |  |
| --- | --- | --- |
| Студент гр. 4373 |  | Матвеев А.В. |
| Преподаватель |  | Глущенко |

Санкт-Петербург

2024

**Цель работы.**

Знакомство с внутренним представлением различных типов данных, используемых компьютером при их обработке.

**Задание на лабораторную работу.**

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. (\*)

**Текст программы**

#include <ctime>

#include <iostream>

using namespace std;

int main()

{

srand(time(0));

setlocale(0, "");

//идз 10

cout << "ИДЗ 10" << '\n';

int a(12);

int per;

const int size = 2;

int array[size] = { 0,1 };

unsigned int orderi = 32; // Количество разрядов

unsigned int maski = 1 << orderi - 1; // Маска побитового сравнения

for (int ii = 1; ii <= orderi; ii++)

{

if (a & maski)

{

per = 1;

if (a > 0)

{

if (ii % 2 != 0)

{

per = array[rand() % size];

}

}

else

{

if (ii % 2 == 0)

{

per == array[rand() % size];

}

}

cout << per;

}

else

{

per = 0;

if (a > 0)

{

if (ii % 2 != 0)

{

per = array[rand() % size];

}

}

else

{

if (ii % 2 == 0)

{

per = array[rand() % size];

}

}

cout << per;

}

a <<= 1; // Побитовый сдвиг числа

if (ii % 8 == 0)

{

putchar(' ');

}

if (ii % orderi - 1 == 0)

{

putchar(' ');

}

}

cout << "" << '\n';

//task1

cout << "task 1" << '\n';

cout << " тип int занимает в памяти " << sizeof(int) << " бита" << '\n';

cout << " тип short int занимает в памяти " << sizeof(short int) << " бита" << '\n';

cout << " тип long int занимает в памяти " << sizeof(long int) << " бита" << '\n';

cout << " тип float занимает в памяти " << sizeof(float) << " бита" << '\n';

cout << " тип double занимает в памяти " << sizeof(double) << " бит" << '\n';

cout << " тип long double занимает в памяти " << sizeof(long double) << " бит" << '\n';

cout << " тип char занимает в памяти " << sizeof(char) << " бит" << '\n';

cout << " тип bool занимает в памяти " << sizeof(bool) << " бит" << '\n';

//task 2

cout << "task 2" << '\n';

int value = 127; // 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(' ');

}

}

//task 3

cout << '\n' << "task 3" << '\n';

union {

float b = 3.14;

int ib;

};

cout << &b << ' ' << &ib << '\n';

cout << b << ' ' << ib << '\n';

mask = 1 << sizeof(float) \* 8 - 1;

for (int i = 0; i <= (sizeof(float) \* 8 - 1); i++) {

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

mask >>= 1;

if (i == 0)

{

putchar(' ');

}

if (i == 8)

{

putchar(' ');

}

}

//task 4

cout << '\n' << "task 4" << '\n';

unsigned long long newMask = 1LL << (sizeof(double) \* 8 - 1);

union

{

double s = 3.14;

int is;

};

cout << &s << ' ' << &is << '\n';

cout << s << ' ' << is << '\n';

//mask = 1 << sizeof(double) \* 8 - 1;

for (int j = 0; j <= (sizeof(double) \* 8 - 1); j++) {

putchar(newMask & is ? '1' : '0');

newMask >>= 1;

if (j == 0)

{

putchar(' ');

}

if (j == 11)

{

putchar(' ');

}

}

}

**Примеры запуска программы**

На рисунке демонстрирована работа программы,

