Национальный исследовательский университет «Московский институт электронной техники»

**Лабораторная работа №1**

**По «Архитектуре вычислительных систем»**

Выполнили: Зиновьева Е.

Ткаченко В.

*Зеленоград*

*2021*

**Задание 1.** Изучите, как интерпретируется одна и та же область памяти, если она рассматривается как знаковое или беззнаковое число, а также как одно и то же число записывается в различных системах счисления.

Необходимо сравнить:

а) беззнаковую интерпретацию переменной в шестнадцатеричной форме;

б) беззнаковую интерпретацию в десятичной форме;

в) знаковую интерпретацию в десятичной форме.

#include <iostream>

#include <bitset>

#include <cmath>

void reckoning(int No) {

    std::cout << "Безнаковая в 16-ричной форме: " << std::hex << (unsigned short)No << std::endl;

    std::cout << "Безнаковая в 10-ричной форме: " << std::dec << (unsigned short)No << std::endl;

    std::cout << "Знаковая в 10-ричной форме: " << std::dec << (short)No << std::endl;

    std::cout << "2-чной форме: " << std::bitset<16>(No) << std::endl;

    }

int main() {

    int x = pow(2, 15);

    std::cout << "x = -2^15:"<< std::endl;

    reckoning(x);

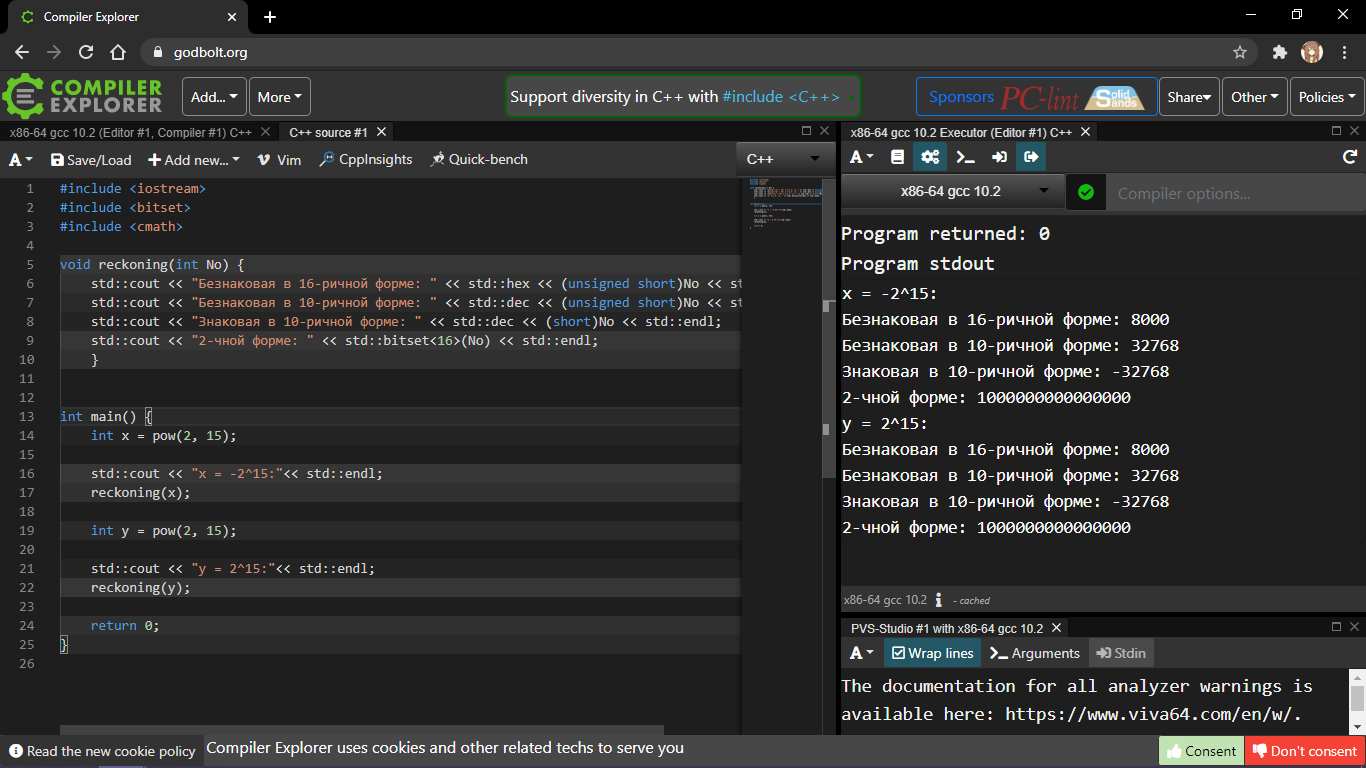
    int y = pow(2, 15);

    std::cout << "y = 2^15:"<< std::endl;

    reckoning(y);

    return 0;

}



**Задание 2.** Найдите и выпишите в отчёт минимальное и максимальное 16-битное число со знаком и без знака в формах представления (а), (б), (в) и в двоичной форме (4 числа, каждое из которых представлено в 4 формах).

#include <iostream>

#include "stdint.h"

#include "math.h"

#include <bitset>

#include <limits>

void reckoning(int No) {

    std::cout << "Безнаковая в 16-ричной форме: " << std::hex << (unsigned short)No << std::endl;

    std::cout << "Безнаковая в 10-ричной форме: " << std::dec << (unsigned short)No << std::endl;

    std::cout << "Знаковая в 10-ричной форме: " << std::dec << (short)No << std::endl;

    std::cout << "2-чной форме: " << std::bitset<16>(No) << std::endl;

    }

int main(){

int16\_t min = std::numeric\_limits<int16\_t>::min();

int16\_t max = std::numeric\_limits<int16\_t>::max();

uint16\_t u\_min = std::numeric\_limits<uint16\_t>::min();

uint16\_t u\_max = std::numeric\_limits<uint16\_t>::max();

std::cout << "Для максимального 16-битного значения со знаком" << std::endl;

reckoning(max);

std::cout << "Для минимального 16-битного значения со знаком" << std::endl;

reckoning(min);

std::cout << "Для максимального 16-битного беззнакового значения" << std::endl;

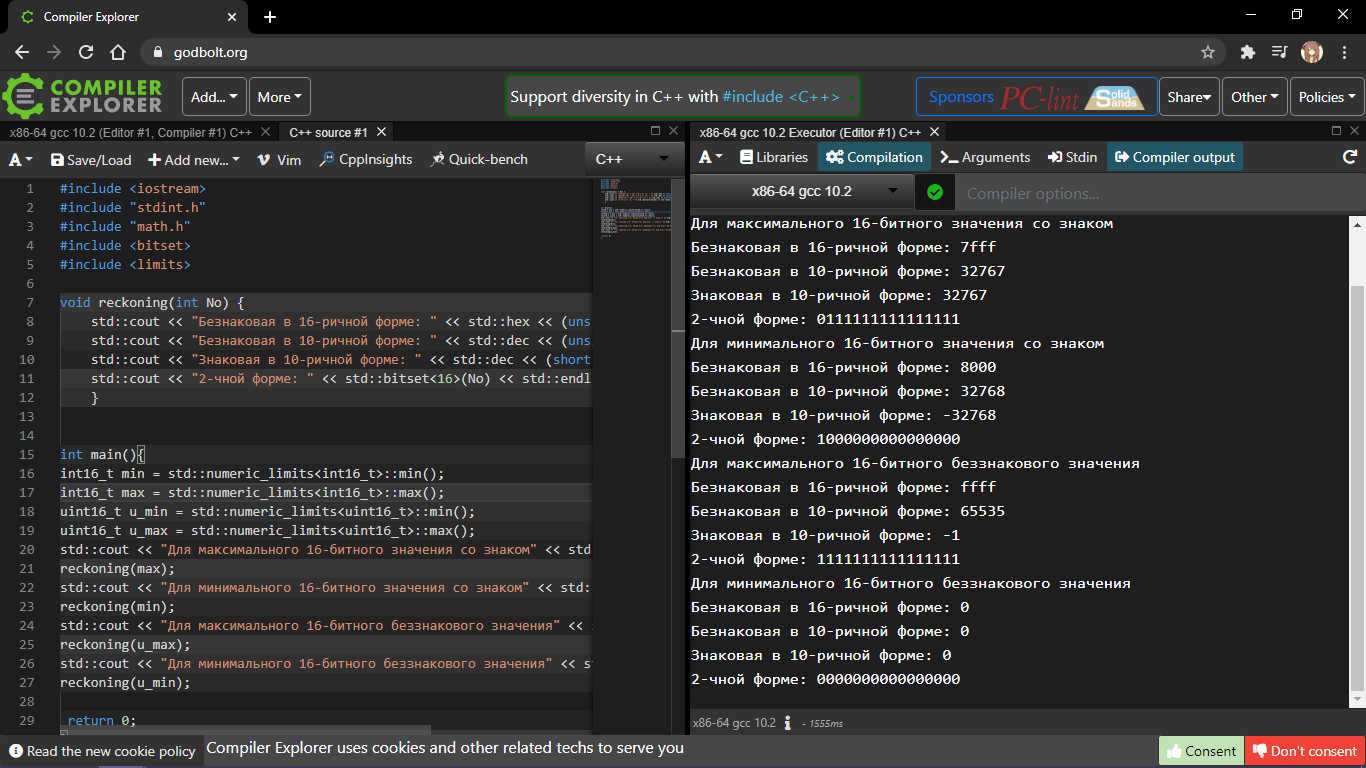
reckoning(u\_max);

std::cout << "Для минимального 16-битного беззнакового значения" << std::endl;

reckoning(u\_min);

 return 0;

}



**Задание 3**. Разработайте программу на языке C++, выполняющую над беззнаковыми шестнадцатибитными целыми числами следующие поразрядные операции (результат должен печататься в десятичной и шестнадцатеричной формах):

бинарные

𝑥 ∧ 𝑦 (конъюнкция),

𝑥 ∨ 𝑦 (дизъюнкция),

𝑥 ⊕ 𝑦 (сложение по модулю два);

унарные

¬𝑥 (отрицание),

neg(𝑥) (дополнение до двух, 𝑥 + neg(𝑥) = 2разрядность 𝑥 );

– 𝑥 ≪ 𝑦 (логический сдвиг влево),

𝑥 ≫ 𝑦 (логический сдвиг вправо).

#include <iostream>

#include <cmath>

int main() {

    uint16\_t x = 0x9211;

    uint16\_t y = 0x0004;

    std::cout<< "X\_16 = "<< std::hex<< x;

    std::cout<< " X\_10 = "<< std::dec<< x << std::endl;

    std::cout<< "y\_16 = "<< std::hex<< y;

    std::cout<< " y\_10 = "<< std::dec<< y << std::endl;

    int i = 0, n=x ;

    while(n>0)

    {

        n = n / 10; ++i;

    }

    //Бинарные

    std::cout << "Коньюнкция :\n10-чной: " << std::dec<< (x&y) << std::endl;

    std::cout << "16-ричной: " << std::hex << (x&y) << std::endl << std::endl;

    std::cout << "Дизъюнкция :\n10-чной: " << std::dec<< (x|y) << std::endl;

    std::cout << "16-ричной: " << std::hex << (x|y) << std::endl << std::endl;

    std::cout << "Сложение по модулю 2 :\n10-чной: " << std::dec<< (x^y) << std::endl;

    std::cout << "16-ричной: " << std::hex << (x^y) << std::endl << std::endl;

    //Унарные

    std::cout << "Отрицание :\n10-чной: " << std::dec<< ~x << std::endl;

    std::cout << "16-ричной: " << std::hex << ~x << std::endl << std::endl;

    std::cout << "Дополнение до двух:\n10-чной: " << std::dec<< (x+pow(2,n)) << std::endl;

    std::cout << "16-ричной: " << std::hex << (x+pow(2,n)) << std::endl << std::endl;

    std::cout << "Сдвиг влево:\n10-чной: " << std::dec<< (x<<y) << std::endl;

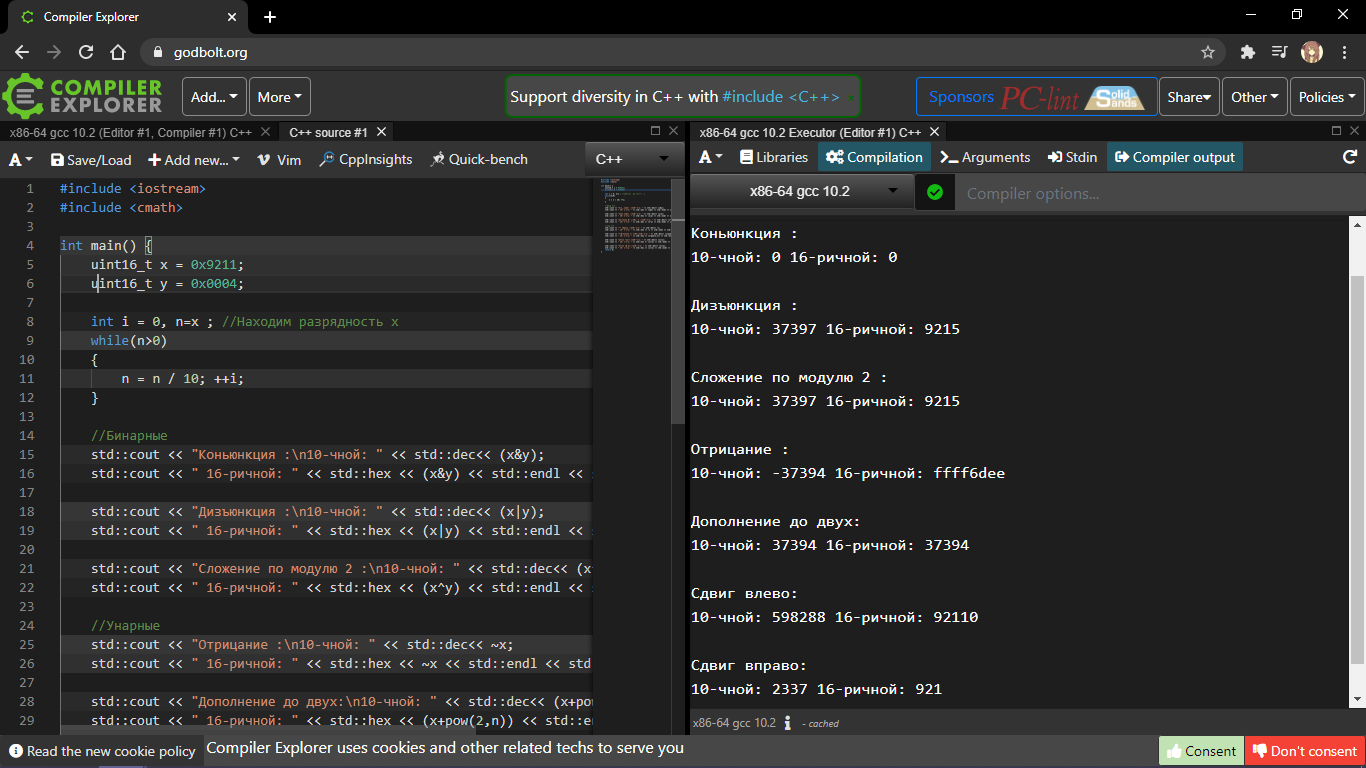
    std::cout << "16-ричной: " << std::hex << (x<<y) << std::endl << std::endl;

    std::cout << "Сдвиг вправо:\n10-чной: " << std::dec<< (x>>y) << std::endl;

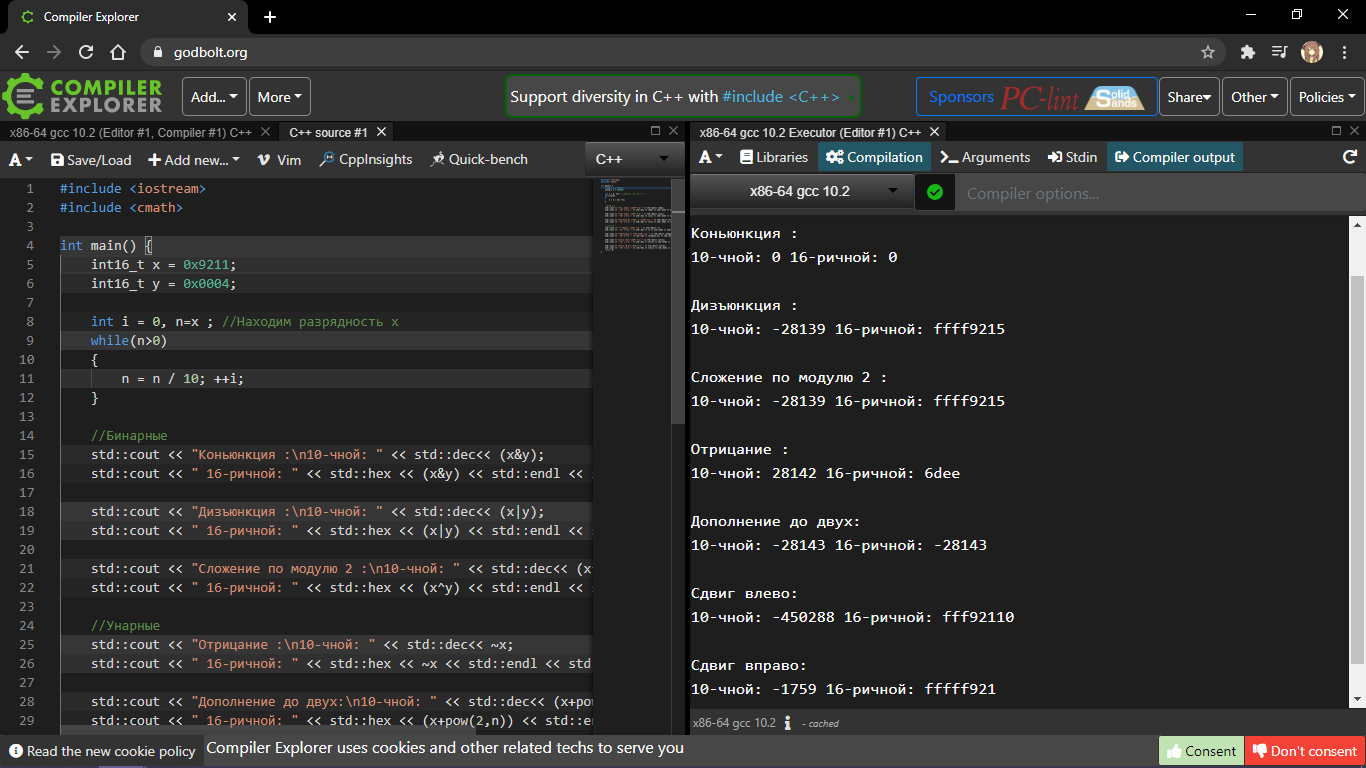
    std::cout << "16-ричной: " << std::hex << (x>>y) << std::endl << std::endl;

    return 0;

}



**Задание 4.** Измените в программе из задания 3 тип переменных на знаковый. Объясните изменение (или неизменность) результата



**Задание 5.** Бонус (+1 балл). Разработайте программу на языке C++ (или дополните программу из задания 3), которая расширяет шестнадцатибитное представление числа 𝑥 до тридцатидвухбитного, рассматривая числа как

– знаковые (signed);

– беззнаковые (unsigned).

#include <iostream>

#include <cmath>

template<typename T>

void print(T x) {

    std::cout << "10-чная: " << std::hex << x;

    std::cout << "   2-чная: " << std::dec << x << std::endl;

}

int main() {

    int16\_t x = 0x8001;

    int32\_t x\_change = x;

    std::cout << "Шестнадцатибитное представление (знаковое)"<< std::endl;

    print(x);

    std::cout << "Тридцатидвухбитное представление (знаковое)"<< std::endl;

    print(x\_change);

    std::cout<< std::endl;

    uint16\_t y = 0x8001;

    uint32\_t y\_change = y;

    std::cout << "Шестнадцатибитное представление (беззнаковое)"<< std::endl;

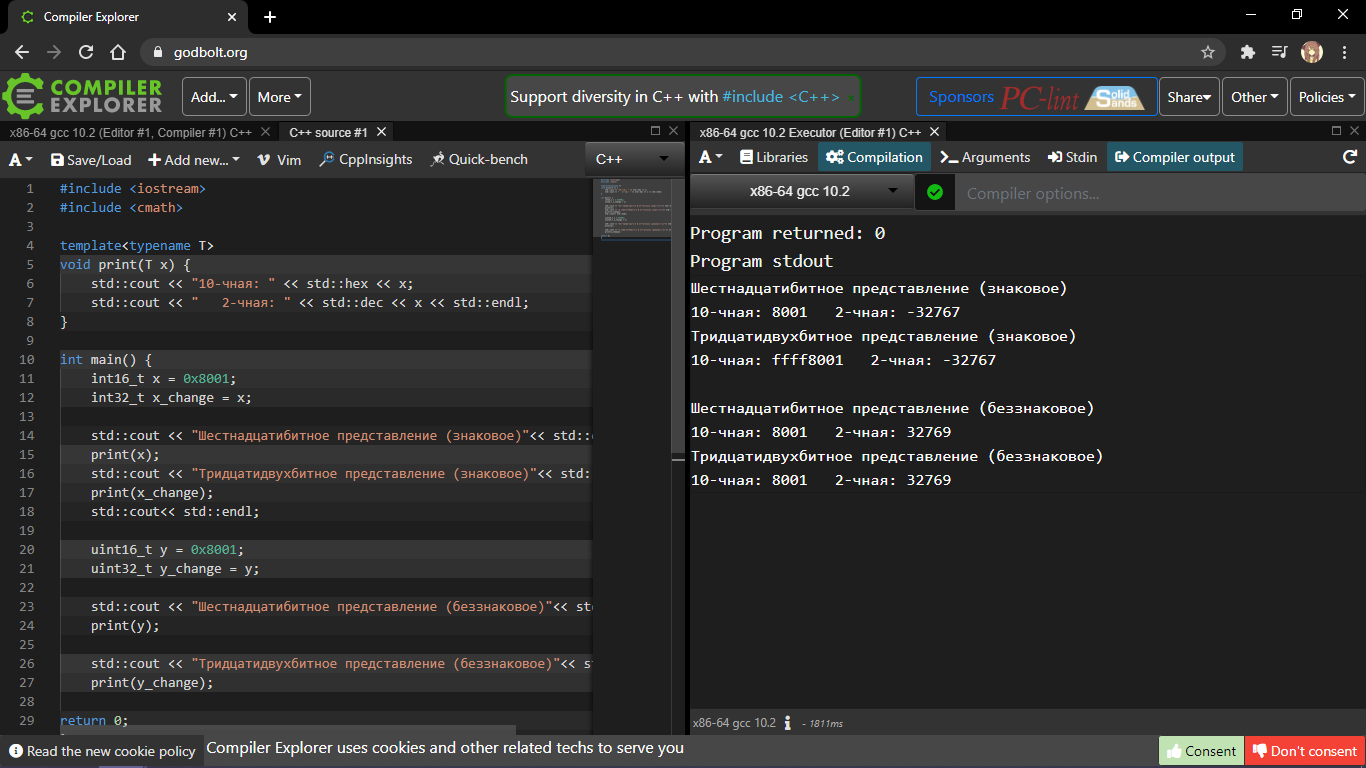
    print(y);

    std::cout << "Тридцатидвухбитное представление (беззнаковое)"<< std::endl;

    print(y\_change);

return 0;

}



**Задание 6.** Определите и выпишите в отчёт, как хранятся в памяти компьютера: – целое число 0x12345678; по результату исследования определите порядок следования байтов в словах для вашего процессора:

– Little-Endian (от младшего к старшему, порядок Intel);

– Big-Endian (от старшего к младшему, порядок Motorola);

320 Приложение А. Лабораторный практикум GNU Assembler

– строки "abcd" и "абвг" (массив из char);

– «широкие» строки L"abcd" и L"абвг" (массив из wchar\_t).

#include <iostream>

#include <bitset>

int main()

{

 int32\_t x = 0x12345678;;

 std::cout << "0x12345678" << std::endl;

 std::cout << "10-чная: "<< std::dec << x << std::endl;

 std::cout << "2-чная: "<<  std::bitset<32>(x) << std::endl;

 char ENG[] = "abcd";

 char RUS[] = "абвг";

 std::cout << std::endl << "abcd:" << std::endl;

 for (int i = 0; i < 8; i++) {

 std::cout << (int)RUS[i] << " ";

 }

 std::cout << std::endl << "абвг" << std::endl;

 for (int i = 0; i < 4; i++) {

 std::cout << (int)ENG[i] << " ";

 }

 std::cout << std::endl;

 wchar\_t expanded\_ENG[] = L"abcd";

 wchar\_t expanded\_RUS[] = L"абвг";

 std::cout << std::endl << "abcd" << std::endl;

 for (int i = 0; i < 4; i++)

 {

 std::cout << (int)expanded\_ENG[i] << " ";

 }

 std::cout << std::endl << "абвг" << std::endl;

 for (int i = 0; i < 4; i++)

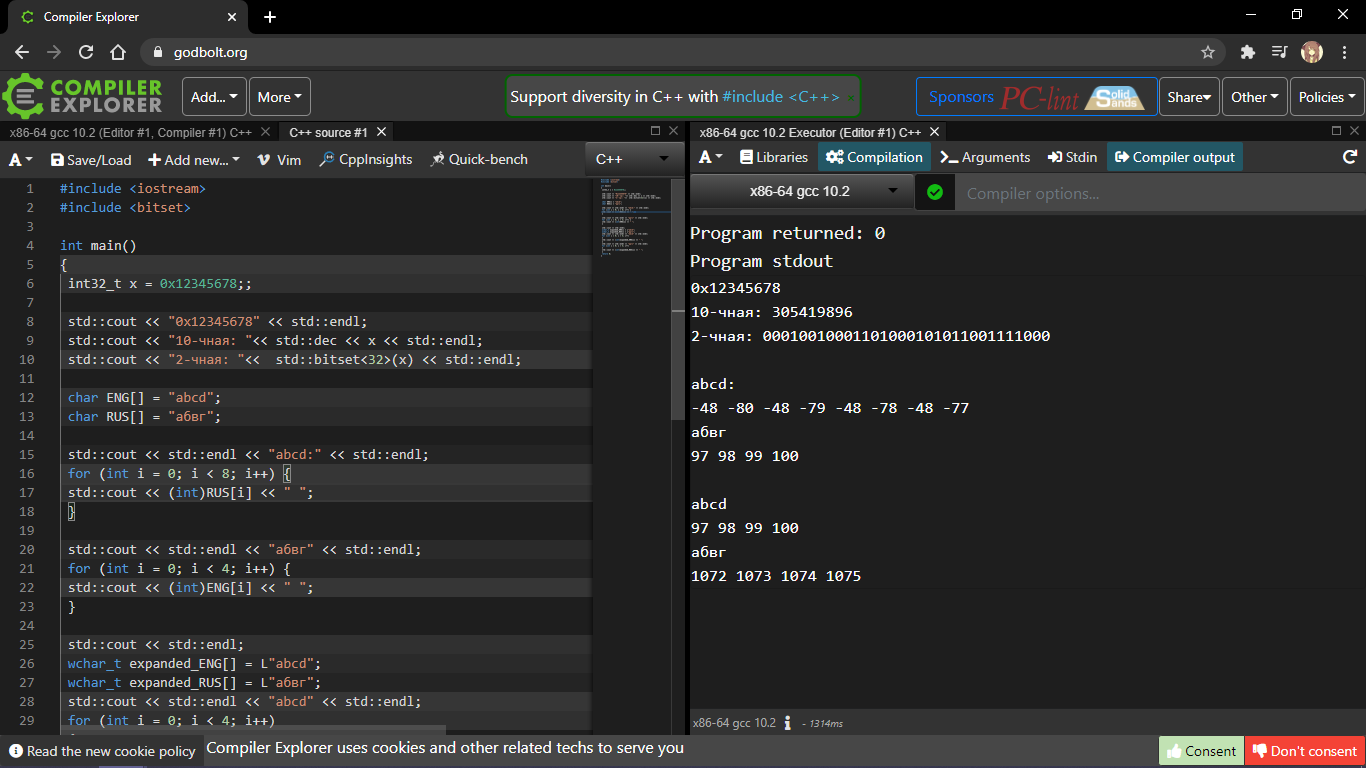
 {

 std::cout << (int)expanded\_RUS[i] << " ";

 }

 return 0;

}



**Задание 7.** При помощи оператора sizeof выясните, сколько байтов занимают переменные следующих типов: char, bool, wchar\_t, short, int, long, long long, float, double, long double, size\_t, ptrdiff\_t, void\*. Результаты оформите в отчёте в виде таблицы, указывая для каждого типа его назначение.

Для выполнения единообразных действий над переменными различных типов используются макросы препроцессора C или шаблоны C++.

Проверьте, соответствуют ли размеры типов современному стандарту C++.

#include <iostream>

#include <string>

template<typename T>

void size\_of(std::string type\_name){

 std::cout << "Размер " << type\_name <<" - " << sizeof(T) << std::endl;

}

int main(){

 size\_of<char>("char");

 size\_of<bool>("bool");

 size\_of<wchar\_t>("wchar\_t");

 size\_of<short>("short");

 size\_of<int>("int");

 size\_of<long>("long");

 size\_of<long long>("long long");

 size\_of<float>("float");

 size\_of<double>("double");

 size\_of<long double>("long double");

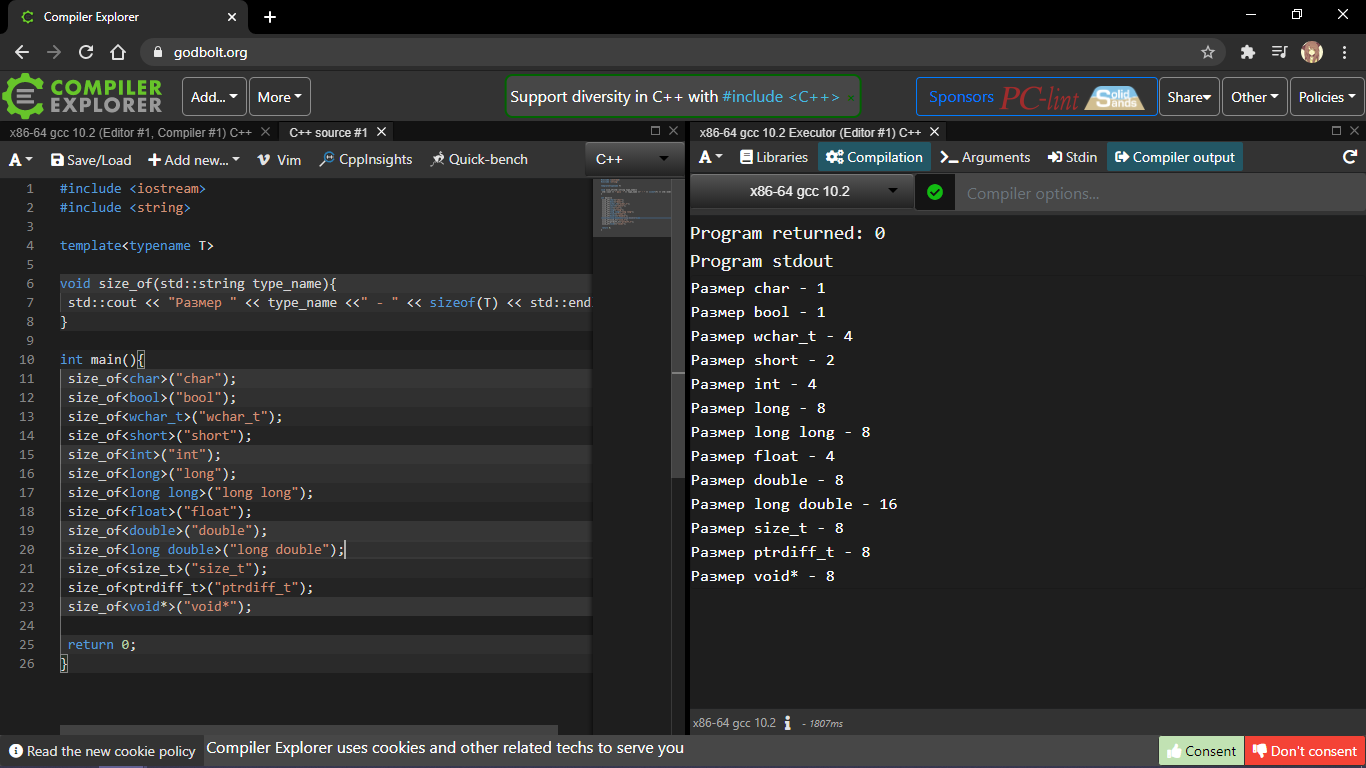
 size\_of<size\_t>("size\_t");

 size\_of<ptrdiff\_t>("ptrdiff\_t");

 size\_of<void\*>("void\*");

 return 0;

}



**Задание 8.** Запустите программу (программы) из заданий 6–7 на двух других платформах, доступных на ВЦ — 32- и 64-разрядной версиях Microsoft Windows и повторите измерения. Для каждого из заданий дополните таблицу результатами новых измерений. Платформы, для которых необходимо провести измерения:

– GNU/Linux Ubuntu, архитектура x86 (компилятор GCC, среда любая);

– 32-разрядная версия Microsoft Windows (ВЦ, среда Microsoft Visual Studio);

– 64-разрядная версия Microsoft Windows (терминал Skylab, среда Microsoft Visual Studio, 64-битная конфигурация).

Результаты однотипных измерений, выполненных на различных платформах, должны быть сгруппированы в таблицу или последовательно описаны в одном разделе.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип данных | Win64 | Win32 | Ubuntu x32 | Ubuntu x64 |
| Char | 1 | 1 | 1 |  |
| Bool | 1 | 1 | 1 |  |
| Wchar\_t | 2 | 2 | 4 |  |
| Short | 2 | 2 | 2 |  |
| Int | 4 | 4 | 4 |  |
| Long | 4 | 4 | 8 |  |
| Long long | 8 | 8 | 8 |  |
| Float | 4 | 4 | 4 |  |
| Double | 8 | 8 | 8 |  |
| Long double | 8 | 8 | 16 |  |
| Size\_t | 8 | 4 | 8 |  |
| Ptrdiff\_t | 8 | 4 | 8 |  |
| Void\* | 8 | 4 | 8 |  |

