**Отчёт по лабораторной работе №1**

**Код программы:**

#include<iostream>

using namespace std;

class Bitstring{

public: unsigned long long a, resa;

unsigned int b, resb;

void AND(unsigned long long x, unsigned int y){

resa = a & x;

resb = b & y;

}

void OR(unsigned long long x, unsigned int y){

resa = a | x;

resb = b | y;

}

void XOR(unsigned long long x, unsigned int y){

resa = a ^ x;

resb = b ^ y;

}

void NOT(unsigned long long x, unsigned int y){

resa = ~x;

resb = ~y;

}

void Shiftleft(int n){

resa = a << n;

resb = b << n;

}

void Shiftright(int n){

resa = a >> n;

resb = b >> n;

}

unsigned long long getResa(){

return resa;

}

unsigned int getResb(){

return resb;

}

int Count\_1 (unsigned long long x, unsigned int y){

int res = 0;

while (x) {

res++;

x &= x-1; // Забираем младшую единичку.

}

while (y) {

res++;

y &= y-1; // Забираем младшую единичку.

}

return res;

}

void compare\_1(unsigned long long x, unsigned int y){

if (Count\_1(a, b) == Count\_1(x, y)){

printf("equal\_1\n");

} else printf("not\_equal\_1\n");

}

void inclusion\_check(unsigned long long x, unsigned int y){

OR(x,y);

if (getResa() == a && getResb() == b){

printf("it`s inclusion\n");

} else printf("it`s not inclusion\n");

}

void print(const string type) {

if(type == "bin") {

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

printf("%llu", (resa >> i)&1);

}

cout << " ";

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

printf("%u", (resb >> i)&1);

}

cout << '\n';

}

}

};

int main() {

Bitstring m;

unsigned long long x;

unsigned int y;

cin >> m.a >> m.b >> x >> y;

m.resa = m.a;

m.resb = m.b;

cout << "A:\n";

m.print("bin");

m.resa = x;

m.resb = y;

cout << "B:\n";

m.print("bin");

m.AND(x, y);

cout << "and:\n";

m.print("bin");

cout << "произведена логическая операция И\n";

m.OR(x,y);

cout << "or:\n";

m.print("bin");

cout << "произведена логическая операция ИЛИ\n";

m.XOR(x,y);

cout << "xor:\n";

m.print("bin");

cout << "произведена логическая операция исключающее ИЛИ\n";

m.NOT(x,y);

cout << "not for B:\n";

m.print("bin");

cout << "произведена логическая операция НЕ для числа B\n";

m.Shiftleft(3);

cout << "left A on 3:\n";

m.print("bin");

cout << "произведен сдвиг влево на 3 бита числа А\n";

m.Shiftright(2);

cout << "right A on 2:\n";

m.print("bin");

cout << "произведен сдвиг вправо на 2 бита числа А\n";

cout << "подсчёт единиц в числе B\n";

printf("count B: %d\n", m.Count\_1(x,y));

cout << "cравнение количества единиц в А и В\n";

m.compare\_1(x,y);

cout << "проверка включения В в А\n";

m.inclusion\_check(x,y);

return 0;

}

**ссылка на git:** <https://github.com/khrengen/oop_exercise_01>

**Тесты:**

120 11 31 5

21111 1234 32321 1290

**Результаты выполнения:**

120 11 31 5

A:

0000000000000000000000000000000000000000000000000000000001111000 00000000000000000000000000001011

B:

0000000000000000000000000000000000000000000000000000000000011111 00000000000000000000000000000101

and:

0000000000000000000000000000000000000000000000000000000000011000 00000000000000000000000000000001

произведена логическая операция И

or:

0000000000000000000000000000000000000000000000000000000001111111 00000000000000000000000000001111

произведена логическая операция ИЛИ

xor:

0000000000000000000000000000000000000000000000000000000001100111 00000000000000000000000000001110

произведена логическая операция исключающее ИЛИ

not for B:

1111111111111111111111111111111111111111111111111111111111100000 11111111111111111111111111111010

произведена логическая операция НЕ для числа B

left A on 3:

0000000000000000000000000000000000000000000000000000001111000000 00000000000000000000000001011000

произведен сдвиг влево на 3 бита числа А

right A on 2:

0000000000000000000000000000000000000000000000000000000000011110 00000000000000000000000000000010

произведен сдвиг вправо на 2 бита числа А

подсчёт единиц в числе B

count B: 7

cравнение количества единиц в А и В

equal\_1

проверка включения В в А

it`s not inclusion

21111 1234 32321 1290

A:

0000000000000000000000000000000000000000000000000101001001110111 00000000000000000000010011010010

B:

0000000000000000000000000000000000000000000000000111111001000001 00000000000000000000010100001010

and:

0000000000000000000000000000000000000000000000000101001001000001 00000000000000000000010000000010

произведена логическая операция И

or:

0000000000000000000000000000000000000000000000000111111001110111 00000000000000000000010111011010

произведена логическая операция ИЛИ

xor:

0000000000000000000000000000000000000000000000000010110000110110 00000000000000000000000111011000

произведена логическая операция исключающее ИЛИ

not for B:

1111111111111111111111111111111111111111111111111000000110111110 11111111111111111111101011110101

произведена логическая операция НЕ для числа B

left A on 3:

0000000000000000000000000000000000000000000000101001001110111000 00000000000000000010011010010000

произведен сдвиг влево на 3 бита числа А

right A on 2:

0000000000000000000000000000000000000000000000000001010010011101 00000000000000000000000100110100

произведен сдвиг вправо на 2 бита числа А

подсчёт единиц в числе B

count B: 12

cравнение количества единиц в А и В

not\_equal\_1

проверка включения В в А

it`s not inclusion

**Объяснение результатов работы программы:**

Программа получает на вход 4 числа, которые составляют 2 битовые строки, представленные двумя полями: unsigned long long(64 бита) и unsigned int(32) бита. Далее программа выполняет побитовые операции с этими строками, записывая ответ в результирующую строку.

Битовые операции в языке:

1) & - and

2) | - or

3) << - shiftleft

4) >> - shiftright

5) ^ - xor

6) ~ - not

Далее программа выполняет подсчёт единиц в выбранной строке, сравнение строк по количеству единиц и проверку на включение.

Для подсчёта количества единиц делаем цикл, который увеличивает счётчик и умножает число на меньшее его на единицу(тем самым обнуляется младшая единица), пока само не станет равным нулю.

Для сравнения по количеству единиц сравниваем результаты выполнения предыдущий функции для обоих чисел.

Для того, чтобы проверить, является ли число Б включением А, нужно проверить не стоит ли у Б единицы в тех местах, где у А - 0. Делаем логическое сложение и сравниваем с А.

После выполнения операций, программа выводит в стандартный вывод либо результирующие строки в бинарном виде(строка 96 бит, поля разделяются пробелом), либо результат выполнения функций.