**1. Make some example that often using bitwise Operators (and (&), or (|), xor (^))**

#include <iostream>

int main()

{

/\*make some example that often using bitwise

Operators (and (&), or (|), xor (^))\*/

 //x 1111.1111.0000.0000.0001.0001.1000.1000

//1U 0000.0000.0000.0000.0000.0000.0000.0001

unsigned int x = 0xFF001188U;

/\*check bit 7 is 1 or 0\*/

unsigned check = x & (1U << 6);

std::cout << check << std::endl;

/\*if bit 7 is 0 set bit 7\*/

if (check == 0)

{

x = x | (1U << 6);

}

std::cout << std::hex << x << std::endl;

/\*Toggle bit 7\*/

x = x ^ (1U << 6);

std::cout << std::hex << x << std::endl;

system("pause");

return 0;

}

**2. Viết hàm tìm số nguyên tố**

input: số tự nhiên bất kỳ nhỏ hơn 99999

output: số nguyên tố lớn nhất nhỏ hơn hoặc bằng số input

note: nếu input > 999999 hoặc input < 0 thì return -1.

#include <iostream>

int isPrime(int x)

{

if (x < 2) { return 0; }

if (x == 2) { return 1; }

if (x % 2 == 0) { return 0; }

for (int i = 3; i <= sqrt(x); i = i + 2)

{

if (x % i == 0)

{

return 0;

}

}

return 1;

}

int searchPrime(int x)

{

if (x < 0 || x > 999999)

{

return -1;

}

for (int i = x; i >= 2; i--)

{

if (isPrime(i) == 1)

{

return i;

}

}

return -1;

}

int main()

{

int x;

std::cout << "\nEnter x = ";

std::cin >> x;

std::cout << searchPrime(x);

system("pause");

return 0;

}

**3. Viết một hàm thực hiện chức nắng năng in ra kết quả của các phép (cộng, trừ, nhân, chia, tìm số lớn, tìm số bé) 2 số kiểu float**

input: uint cờ đại diện phép toán cần thực hiện

float số thứ 1

float số thứ 2

void Calculator(uint MathType, float num1, float num2)

{

...

}

ps: uint MathType sẽ có dạng như dưới hoặc là tổ hợp phép (|)

#define CONG 0x0001//: cộng

#define TRU 0x0002//: trừ

#define NHAN 0x0004//: nhân

#define CHIA 0x0008//: chia

#define MAX 0x0010//: số lớn

#define MIN 0x0020//: số bé

MathType có thể là kết hợp các phép toán cần thực hiên:

vd: người dùng muốn vừa biết kq phép cộng và phép trừ, người dùng sẽ đưa vào có input như sau:

Calculator(CONG|TRU, a, b){

}

-> expect là in ra kq phép cộng rồi lại tiếp tục in ra kq phép trừ

#include <iostream>

#define CONG 0x0001//: cộng

#define TRU 0x0002//: trừ

#define NHAN 0x0004//: nhân

#define CHIA 0x0008//: chia

#define MAX 0x0010//: số lớn

#define MIN 0x0020//: số bé

using namespace std;

void Calculator(unsigned int MathType, float num1, float num2)

{

if (MathType & CONG)

{

cout << num1 << " + " << num2 << " = " << num1+num2 << endl;

}

if (MathType & TRU)

{

cout << num1 << " - " << num2 << " = " << num1-num2 << endl;

}

if (MathType & NHAN)

{

cout << num1 << " \* " << num2 << " = " << num1\*num2 << endl;

}

if (MathType & CHIA)

{

cout << num1 << " / " << num2 << " = " << num1/num2 << endl;

}

if (MathType & MAX)

{

cout << "MAX("<< num1 <<","<< num2 <<")="<<((num1>num2)?num1:num2) << endl;

}

if (MathType & MIN)

{

cout << "MIN("<< num1 <<","<< num2 <<")="<<((num1<num2)?num1:num2) << endl;

}

}

int main()

{

Calculator(CONG|TRU|NHAN, 1, 2);

Calculator(MAX|MIN, 1, 2);

system("pause");

return 0;

}

**4. Taị sao đảo của int a = 0; lại cho ra giá trị a = -1;**

#include <iostream>

int main()

{

int x = ~0;

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

{

std::cout << ((x<<i)&(1U<<31)?1:0);

}

//Biểu diễn số 0: 0000.0000.0000.0000.0000.0000.0000.0000

//Đảo của số 0: 1111.1111.1111.1111.1111.1111.1111.1111

//Biểu diễn số 1: 0000.0000.0000.0000.0000.0000.0000.0001

//Số bù 1 của số 1: 1111.1111.1111.1111.1111.1111.1111.1110

//Số bù 2 của số 1: 1111.1111.1111.1111.1111.1111.1111.1111

//Biểu diễn số -1 bằng cách sử dụng số bù 2 của số 1

system("pause");

return 0;

}