Câu 1: chuyển đổi thập phân sang nhị phân 10 bits

Số nguyên: 66 -> nhị phân: 0001000010

Số nguyên: - 99 -> nhị phân: 1110011101

Số nguyên: 69 -> nhị phân: 0001000101

Số nguyên: -111 -> nhị phân: 1110010001

Sô nguyên: 22 -> nhị phân: 0000010110

Srce code:

#include <bits/stdc++.h>

using namespace std;

void Exchange(int n, int a[], int k){

int FirstPos = 0;

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

if (a[i] == 1) {FirstPos = i; break;}

}

for (int i = FirstPos + 1; i < k; i++){

if (a[i] == 1) a[i] = 0;

else a[i] = 1;

}

for (int i = k - 1; i >= 0; i--) {

cout << a[i];

}

}

void ToBinary(int n, int a[], int k){

int tmp = n;

n = abs(n);

while (n > 0) {

a[k++] = n%2;

n = n/2;

}

while (k < 10) {

for (int i = 0; i < 10 - k; i++) {

a[k++] = 0;

}

}

if (tmp > 0){

for (int i = k - 1; i >= 0; i--) {

cout << a[i];

}

}

else {

Exchange(n, a, k);

}

}

int main(){

int a[256];

int n; int k = 0;

cout << "nhap n can chuyen sang nhi phan: ";

cin >> n;

ToBinary(n, a, k);

return 0;

}

Câu 2: chuyển đổi nhị phân sang thập phân (xài phép bù 2)

Nhị phân: 0001100011 -> thập phân: 99

Nhị phân: 0000011011 -> thập phân: 27

Nhị phân: 0000100001 -> thập phân: 33

Nhị phân: 1111110100 -> thập phân: -12

Nhị phân: 1110110010 -> thập phân: -78

#include <bits/stdc++.h>

using namespace std;

void ToDemical (string s){

int FirstPos = 0; //lưu vtri 1 dau tien

int tmp = 0; int sum = 0;

int n = 0;

int a[100];

//chuyển string sang mảng int

for (int i = 0; i < s.size(); i++){

a[i] = s[i] - 48;

n++;

}

//kiểm tra và lưu lại số 1 đầu tiên từ phải -> trái

for (int i = n - 1; i >= 0; i--){

if (a[i] == 1) {FirstPos = i; break;}

}

if (a[0] == 1){

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

if (a[i] == 1) a[i] = 0;

else a[i] = 1;

}

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

cout << a[i] << " ";

}

for (int i = n - 1; i >= 0; i--){

sum += a[i]\*pow(2, tmp);

tmp++;

}

cout <<"\n ket qua la: " << -sum;

}

else {

for (int i = n - 1; i >= 0; i--){

sum += a[i]\*pow(2, tmp);

tmp++;

}

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

cout << a[i] << " ";

}

cout << "\nket qua la: ";

cout << sum;

}

}

int main()

{

string s;

cout << "nhap s: ";

cin >> s;

ToDemical(s);

return 0;

}

Câu 3: Mô phỏng thuật toán nhị phân:

Cộng:

11 (3) với 1 + 1 (1)0 ta ghi 0 nhớ 1 cộng dồn lên hàng bên trái

+ 110 (6)

= 1001 (9)

Trừ:

10101 (21)

- 00111 (7) với (1)0 – 1 = 1 (ta mượn 1 để trừ nên nợ được dồn lên bên trái)

=01110 (14)

Nhân:

1110

x 1011 với phép nhân, ta thực hiện như phép nhân thập phân rồi cộng lại

1110

1110

0000

1110

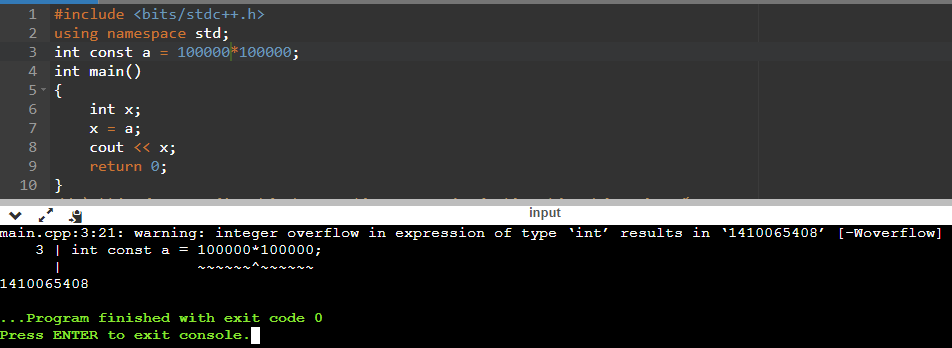
10011010

Chia: với phép chia, ta chia như bình thường rồi thực hiện phép trừ

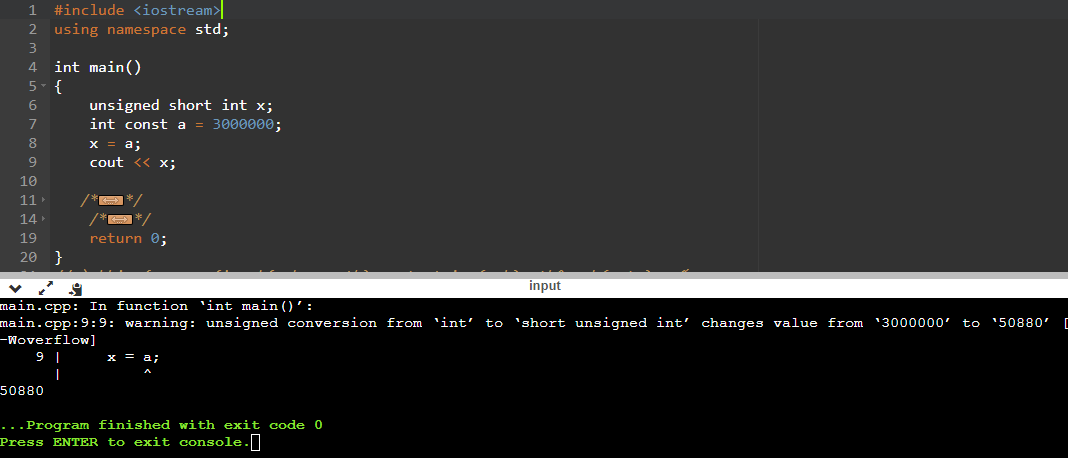
|  |  |
| --- | --- |
| 1001100100 | 11000 |
| -11000 : : : :  011100 : : :  -11000 : : :  00100100  - 11000  01100 (số dư) | 11001 |

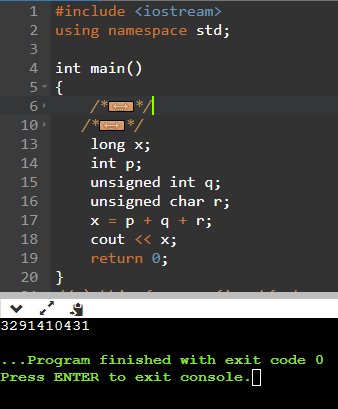
Câu 5:

a) vẫn là giá trị ngẫu nhiên với thông báo tràn số rằng hằng số vượt khỏi giá trị lưu trữ.



b) sẽ xuất ra giá trị ngẫu nhiên kèm thông báo không thể gán số với hằng số vượt ngoài phạm vi kiểu dữ liệu.



c) xuất ra tổng ngẫu nhiên, và phụ thuộc vào kiểu int, thứ tự không ảnh hưởng đến tổng.

Mở rộng:

- để căn chỉnh lề hay format code ta có thể dùng tổ hợp phím ctrl + K, ctrl + D

- để biểu diễn nhiều số nguyên hơn ta sử dụng nhiều bytes hơn, cụ thể là từ 2- 4 bytes để tăng khả năng lưu trữ tương đương việc sử dụng các kiểu dữ liệu như long int hay long long int