Introduction
The Balanced Binary Representation (BBR) is a position font for an integer with a base of 2 and the digits -1, 0, 1. The BBR record of a number is a non-adjacent form (NAF) if non-zero digits are next to each other.

Each integer has exactly one BBR-NAF font without leading zeros. For 0 it is empty. For zero x, the least important digit is c:

0 if x is even,

1 if x - 1 is divisible by 4,

-1 if x + 1 is divisible by 4.

More significant x-digits are the spelling of the number (x -c) /2.

The BBR-NAF recording minimizes the Hamming weight of a number in the binary write class, i. e. the number of non-zero digits. For example, Mersenne's prime numbers have weight 2. BBR-NAF is used for elliptic curve cryptography, which is used for the implementation of the Bitcoin cryptocurrency and the Ethereum platform, among other things.

The subject of the task is the representation of the BBR-NAF recording, which allows efficient arithmetic operations on large integers with low Hamming weight.

The representation is based on the number system arithmeticæ localis, described by John Napier in 1617 in the treatise Rabdologiæ. It is a binary system, but not a position system, but an additive system.

In position systems, the value of a number is determined by a series of digits in its recording, taking into account the position in which the digit is placed. In additive systems, the value of a number is the sum of the numerical values.

In the arithmeticæ localis, letters play the role of the numbers, which represent the power of the two. In our representation, which we call Napier-NAF, numbers are integers that represent the powers of the two and the numbers that are opposite to them. A digit with a value of 2 ** n is a non-negative integer n. A number with a value of – (2 ** n) is represented by a negative integer -n-1.

The Napier NAF representation of an integer x is an array of integers whose length is the Hamming weight of the BBR NAF number x.

a negative number n indicates that the number -1 occurs at position -n -1 of the BBR-NAF dataset of number x;

a non-negative number n if the number 1 appears at position n of the BBR-NAF account of the number x;

The numbers are arranged in the order of the least significant numbers and the items are numbered from 0.

Task
In the header file napiernaf. h, the interface of the module for arithmetic operations on integer numbers in the Napier NAF representation is:

Function

void iton (int x, int **a, int *n);
stores the representation of the number x in the dynamically created *n elementary array *a,

Function

int ntoi (int *a, int n);
returns the value of the number represented by the n-elementary array a, or 0, if this value is outside the range of type int;

Function

void nadd (int *a, int an, int *b, int bn, int **c, int *cn);
writes in a dynamically created *cn-elementarray *c the sum of the numbers represented by the an-elementary array a and the bn-elementary array b,

Function

void nsub (int *a, int an, int *b, int bn, int **c, int *cn);
writes in a dynamically created *cn-elementarray *c the difference of the numbers represented by the an-elementary array a and the bn-elementary array b,

Function

void nmul (int *a, int an, int *b, int bn, int **c, int *cn);
writes in a dynamically created *cn-elementarray *c the product of numbers represented by the an-elementary array a and the bn-elementary array b,

Function

void nexp (int *a, int an, int *b, int bn, int **c, int *cn);
writes to a dynamically created *cn-elementarray *c the result of the potentiation, whose base is represented by the an-elementary array a, while the non-negative exponent is represented by the bn-elementary array b,

Function

void ndivmod (int *a, int an, int *b, int bn, int **q, int *qn, int **r, int *rn);
Divides a number represented by the an-elementary array a by a non-zero number represented by the bn-elementary table b. The quotient writes to a dynamically created *qn-elementarray *q. He writes the rest of the division into a dynamically created *rn-elementarray *r.

The function gives the result according to Euclid's division algorithm . The result of dividing an integer a by a non-zero integer b is an integer quotient q and an integer remainder r that satisfy the conditions:

a = b * q + r,

0 <= r < abs(b),

where abs(b) is the absolute value of the number b.

Write a napiernaf. c module that implements the napiernaf. h interface.