Двоичный дамп целого числа

Задание. Для заданного короткого целого числа построить его двоичный дамп.

Проблема. Представить описание алгоритма, позволяющего выполнить это задание.

Обсуждение хода решения:

1) ввести число;
2) сформировать символьную последовательность на основе символьных литералов ***'0'*** и ***'1'***, сохраняя в ней ведущие нули;
3) сформировать символьную последовательность на основе символьных литералов ***'0'*** и ***'1'***, “подавляя” в ней ведущие нули.

Перед тем как переходить к разработке алгоритмов, обратимся к иллюстрации механизма построения двоичного дампа целых чисел, например, 13_{10} и -13_{10}. С помощью операций арифметического сдвига можно сформировать символьную последовательность из ***'0'*** и ***'1'***, например, анализируя всякий раз значение младшего (выталкиваемого) разряда числа перед каждым арифметическим сдвигом вправо на один двоичный разряд или, наоборот, значение старшего (знакового) разряда числа перед каждым арифметическим сдвигом влево на один двоичный разряд. Для представления коротких целых чисел будем опираться на 16-разрядную архитектуру.

Обратимся к двоичному дампу короткого целого числа 13_{10}, строя его при помощи арифметического сдвига вправо на один двоичный разряд, анализируя всякий раз значение младшего (выталкиваемого) разряда числа:

Десятичный код	Восьмеричный код	Двоичный код	Двоичный дамп
13	000015	0000000000001101	1
6	000006	0000000000000110	0
3	000003	0000000000000011	1
1	000001	0000000000000001	1
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0
0	000000	0000000000000000	0

Обратимся к двоичному дампу короткого целого числа -13_{10}, строя его так же при помощи арифметического сдвига вправо на один двоичный разряд, анализируя всякий раз значение младшего (выталкиваемого) разряда числа:

Десятичный код	Восьмеричный код	Двоичный код	Двоичный дамп
–13	177763	1111111111110011	1
–7	177771	1111111111111001	1
–4	177774	1111111111111100	0
–2	177776	1111111111111110	0
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1
–1	177777	1111111111111111	1

Обратимся к двоичному дампу короткого целого числа 13_{10}, строя его на этот раз при помощи арифметического сдвига влево на один двоичный разряд, анализируя всякий раз значение старшего (знакового) разряда числа:

Десятичный код	Восьмеричный код	Двоичный код	Двоичный дамп
13	000015	0000000000001101	0
26	000032	0000000000011010	0
52	000064	0000000000110100	0
104	000150	0000000001101000	0
208	000320	0000000011010000	0
416	000640	0000000110100000	0
832	001500	0000001101000000	0
1664	003200	0000011010000000	0
3328	006400	0000110100000000	0
6656	015000	0001101000000000	0
13312	032000	0011010000000000	0
26624	064000	0110100000000000	0
–12228	150000	1101000000000000	1
–24576	120000	1010000000000000	1
16384	040000	0100000000000000	0
–32768	100000	1000000000000000	1

Обратимся к двоичному дампу короткого целого числа -13_{10}, строя его так же при помощи арифметического сдвига влево на один двоичный разряд, анализируя всякий раз значение старшего (знакового) разряда числа:

Десятичный код	Восьмеричный код	Двоичный код	Двоичный дамп
–13	177763	1111111111110011	1
–26	177746	1111111111100110	1
–52	177714	1111111111001100	1
–104	177630	1111111110011000	1
–208	177460	1111111100110000	1
–416	177140	1111111001100000	1
–832	176300	1111110011000000	1
–1664	174600	1111100110000000	1
–3328	171400	1111001100000000	1
–6656	163000	1110011000000000	1
–13312	146000	1100110000000000	1
–26624	114000	1001100000000000	1
12288	30000	0011000000000000	0
24576	60000	0110000000000000	0
–16384	140000	1100000000000000	1
–32768	100000	1000000000000000	1

Описания алгоритмов построения двоичных дампов.

При разработке алгоритмов построения двоичных дампов коротких целых чисел на первых порах будем опираться на арифметический сдвиг вправо. Чтобы символьная последовательность при этом формировалась в прямом порядке, следует использовать циклическую операцию маскирования исходного числа, которая позволит последовательно выявить значения всех его двоичных разрядов, начиная с разряда, следующего за знаковым. Знаковый разряд тем самым следует рассматривать независимо от остальных. Начальное значение 16-разрядной маски можно задать, например, с помощью восьмеричного литерала 040000. Всякий раз после каждой операции маскирования исходного числа маску следует сдвигать вправо на один двоичный разряд.

Представим структурированное описание алгоритма на естественном языке:

1. Начало
2. Повторить
2.1. Вывести значение “N? ”
2.2. Ввести значение N
пока не будет (N >= –32768) и (N <= 32767)
3. Положить M = 040000
4. Если N < 0
то
4.1. Вывести значение ‘1’
иначе
4.2. Вывести значение ‘0’
5. От k = 1 до k = 15 повторить
5.1. Если N and M не равно 0
то
5.1.1. Вывести значение ‘1’
иначе
5.1.2. Вывести значение ‘0’
5.2. Положить M = M asr 1
6. Конец

Перейдём к альтернативной версии алгоритма построения двоичного дампа, где вместо операции маскирования будет использоваться операция проверки знака числа. Циклическая операция проверки знака числа перед каждым арифметическим сдвигом влево на один двоичный разряд позволит выявить значения всех его двоичных разрядов.

Представим структурированное описание алгоритма на естественном языке:

1. Начало
2. Повторить
2.1. Вывести значение “N? ”
2.2. Ввести значение N
пока не будет (N >= –32768) и (N <= 32767)
3. От k = 1 до k = 16 повторить
3.1. Если N < 0
то
3.1.1. Вывести значение ‘1’
иначе
3.1.2. Вывести значение ‘0’
3.2. Положить N = N asl 1
4. Конец

Теперь зададимся целью реализовать механизм подавления ведущих нулей для неотрицательных чисел. В основу этого механизма кладётся флаг – переменная, ненулевое значение которой “просигналит” о появлении первой “ведущей” единицы в символьной последовательности. Нулевое значение флага препятствует появлению в символьной последовательности литералов ***'0'*** , трактуя их как ведущие нули. Очевидно, что сам нуль станет теперь “особым” числом, которое следует рассматривать независимо от остальных неотрицательных чисел.

Представим структурированное описание алгоритма на естественном языке, опираясь на операцию маскирования исходного числа:

1. Начало
2. Повторить
2.1. Вывести значение “N? ”
2.2. Ввести значение N
пока не будет (N >= –32768) и (N <= 32767)
3. Положить M = 040000
4. Положить flag = false
5. Если N не равно 0
то
5.1. Если N < 0
то
5.1.1. Положить flag = true
5.1.2. Вывести значение ‘1’
5.2. От k = 1 до k = 15 повторить
5.2.1. Если N and M не равно 0
то
5.2.1.1. Положить flag = true
5.2.1.2. Вывести значение ‘1’
иначе
5.2.1.3. Если flag = true
то
5.2.1.3.1. Вывести значение ‘0’
5.2.2. Положить M = M asr 1
иначе
5.3. Вывести значение ‘0’
6. Конец

Представим структурированное описание алгоритма на естественном языке, опираясь на операцию проверки знака числа:

1. Начало
2. Повторить
2.1. Вывести значение "N? "
2.2. Ввести значение N
пока не будет (N >= –32768) и (N <= 32767)
3. Положить flag = false
4. Если N не равно 0
то
4.1. От k = 1 до k = 16 повторить
4.1.1. Если N < 0
то
4.1.1.1. Положить flag = true
4.1.1.2. Вывести значение '1'
иначе
4.1.1.3. Если flag = true
то
4.1.1.3.1. Вывести значение '0'
4.1.2. Положить N = N asl 1
иначе
4.2. Вывести значение '0'
5. Конец