

LESSON_18 参考试题

一、编程题

1. 变长编码 (2023年9月)

【问题描述】

小明刚刚学习了三种整数编码方式：原码、反码、补码，并了解到计算机存储整数通常使用补码。但他总是觉得，生活中很少用到 $2^{31}-1$ 这么大的数，生活中常用的 $0\sim 100$ 这种数也同样需要 4 个字节的补码表示，太浪费了些。热爱学习的小明通过搜索，发现了一种正整数的变长编码方式。这种编码方式的规则如下：

1. 对于给定的正整数，首先将其表达为二进制形式。例如， $(0)_{(10)}=(0)_{(2)}$ ， $(926)_{(10)}=(1110011110)_{(2)}$ 。
2. 将二进制数从低位到高位切分成每组 7bit，不足 7bit 的在高位用 0 填补。例如， $(0)_{(2)}$ 变为 0000000 的一组， $(1110011110)_{(2)}$ 变为 0011110 和 0000111 的两组。
3. 由代表低位的组开始，为其加入最高位。如果这组是最后一组，则在最高位填上 0，否则在最高位填上 1。于是，0 的变长编码为 00000000 一个字节，926 的变长编码为 10011110 和 00000111 两个字节。

这种编码方式可以用更少的字节表达比较小的数，也可以用很多的字节表达非常大的数。例如，987654321012345678 的二进制为 $(0001101\ 1011010\ 0110110\ 1001011\ 1110100\ 0100110\ 1001000\ 0010110\ 1001110)_{(2)}$ ，于是它的变长编码为（十六进制表示）CE 96 C8 A6 F4 CB B6 DA 0D，共 9 个字节。

你能通过编写程序，找到一个正整数的变长编码吗？

【输入描述】

输入第一行，包含一个正整数 N 。约定 $0\leq N\leq 10^{18}$ 。

【输出描述】

输出一行，输出 N 对应的变长编码的每个字节，每个字节均以 2 位十六进制表示（其中 A-F 使用大写字母表示），两个字节间以空格分隔。

【样例输入 1】

0

【样例输出 1】

00

【样例输入 2】

926

【样例输出 2】

9E 07

【样例输入 3】

987654321012345678

【样例输出 3】

CE 96 C8 A6 F4 CB B6 DA 0D

【参考代码】

```
#include <bits/stdc++.h>
using namespace std;
void output_digit(int d) {
    if (d >= 10)
        cout << (char)('A' + d - 10);
    else
        cout << (char)('0' + d);
}
void output_code(int s) {
    output_digit(s >> 4);
    output_digit(s & 0x0f);
}
int main() {
    long long n = 0;
    cin >> n;
    int split[10];
    int l = 0;
    while (n > 0) {
        split[l] = (int)(n & 0x7f);
        n >>= 7;
        l++;
    }
    for (int i = 0; i < l - 1; i++) split[i] |= 0x80;
    output_code(split[0]);
    for (int i = 1; i < l; i++) {
        cout << " ";
        output_code(split[i]);
    }
    return 0;
}
```