**算法分析与设计实验报告**

**第 3 次实验**

**10149 Bit Strings**

|  |
| --- |
| **Problem description** |
| Allow me to state the obvious: sequences of 1s and 0s are rather common in computer science. With that said, sometimes you want a particular sequence of 1s and 0s. For example, if we were looking for all sequences of length 4 that contained exactly 1 pair of consecutive 1s, we would have:   * 1100 * 1101 * 0110 * 1011 * 0011 |
| **Input** |
| Each line of input will be an integer, N, such that 2 <= N <= 12. |
| **Output** |
| For each line of input, your program should print out all the bit strings of length N that have exactly 1 consecutive pairs of 1s. Strings should be ordered in ascending order according to their decimal equivalents. For ordering purposes, the leftmost bit will be considered the least significant bit. |
| **Sample Input** |
| 2  3 |
| **Sample Output** |
| 11  011  110 |

**附录：完整代码**

*//Bit Strings*

#include <stdio.h>

int a[744] = {3, 6, 11, 12, 13, 19, 22, 24, 25, 26, 35, 38, 43, 44, 45, 48, 49, 50, 52, 53, 67, 70, 75, 76, 77,

              83, 86, 88, 89, 90, 96, 97, 98, 100, 101, 104, 105, 106, 131, 134, 139, 140, 141, 147, 150, 152,

              153, 154, 163, 166, 171, 172, 173, 176, 177, 178, 180, 181, 192, 193, 194, 196, 197, 200, 201,

              202, 208, 209, 210, 212, 213, 259, 262, 267, 268, 269, 275, 278, 280, 281, 282, 291, 294, 299,

              300, 301, 304, 305, 306, 308, 309, 323, 326, 331, 332, 333, 339, 342, 344, 345, 346, 352, 353,

              354, 356, 357, 360, 361, 362, 384, 385, 386, 388, 389, 392, 393, 394, 400, 401, 402, 404, 405,

              416, 417, 418, 420, 421, 424, 425, 426, 515, 518, 523, 524, 525, 531, 534, 536, 537, 538, 547,

              550, 555, 556, 557, 560, 561, 562, 564, 565, 579, 582, 587, 588, 589, 595, 598, 600, 601, 602,

              608, 609, 610, 612, 613, 616, 617, 618, 643, 646, 651, 652, 653, 659, 662, 664, 665, 666, 675,

              678, 683, 684, 685, 688, 689, 690, 692, 693, 704, 705, 706, 708, 709, 712, 713, 714, 720, 721,

              722, 724, 725, 768, 769, 770, 772, 773, 776, 777, 778, 784, 785, 786, 788, 789, 800, 801, 802,

              804, 805, 808, 809, 810, 832, 833, 834, 836, 837, 840, 841, 842, 848, 849, 850, 852, 853, 1027,

              1030, 1035, 1036, 1037, 1043, 1046, 1048, 1049, 1050, 1059, 1062, 1067, 1068, 1069, 1072, 1073,

              1074, 1076, 1077, 1091, 1094, 1099, 1100, 1101, 1107, 1110, 1112, 1113, 1114, 1120, 1121, 1122,

              1124, 1125, 1128, 1129, 1130, 1155, 1158, 1163, 1164, 1165, 1171, 1174, 1176, 1177, 1178, 1187,

              1190, 1195, 1196, 1197, 1200, 1201, 1202, 1204, 1205, 1216, 1217, 1218, 1220, 1221, 1224, 1225,

              1226, 1232, 1233, 1234, 1236, 1237, 1283, 1286, 1291, 1292, 1293, 1299, 1302, 1304, 1305, 1306,

              1315, 1318, 1323, 1324, 1325, 1328, 1329, 1330, 1332, 1333, 1347, 1350, 1355, 1356, 1357, 1363,

              1366, 1368, 1369, 1370, 1376, 1377, 1378, 1380, 1381, 1384, 1385, 1386, 1408, 1409, 1410, 1412,

              1413, 1416, 1417, 1418, 1424, 1425, 1426, 1428, 1429, 1440, 1441, 1442, 1444, 1445, 1448, 1449,

              1450, 1536, 1537, 1538, 1540, 1541, 1544, 1545, 1546, 1552, 1553, 1554, 1556, 1557, 1568, 1569,

              1570, 1572, 1573, 1576, 1577, 1578, 1600, 1601, 1602, 1604, 1605, 1608, 1609, 1610, 1616, 1617,

              1618, 1620, 1621, 1664, 1665, 1666, 1668, 1669, 1672, 1673, 1674, 1680, 1681, 1682, 1684, 1685,

              1696, 1697, 1698, 1700, 1701, 1704, 1705, 1706, 2051, 2054, 2059, 2060, 2061, 2067, 2070, 2072,

              2073, 2074, 2083, 2086, 2091, 2092, 2093, 2096, 2097, 2098, 2100, 2101, 2115, 2118, 2123, 2124,

              2125, 2131, 2134, 2136, 2137, 2138, 2144, 2145, 2146, 2148, 2149, 2152, 2153, 2154, 2179, 2182,

              2187, 2188, 2189, 2195, 2198, 2200, 2201, 2202, 2211, 2214, 2219, 2220, 2221, 2224, 2225, 2226,

              2228, 2229, 2240, 2241, 2242, 2244, 2245, 2248, 2249, 2250, 2256, 2257, 2258, 2260, 2261, 2307,

              2310, 2315, 2316, 2317, 2323, 2326, 2328, 2329, 2330, 2339, 2342, 2347, 2348, 2349, 2352, 2353,

              2354, 2356, 2357, 2371, 2374, 2379, 2380, 2381, 2387, 2390, 2392, 2393, 2394, 2400, 2401, 2402,

              2404, 2405, 2408, 2409, 2410, 2432, 2433, 2434, 2436, 2437, 2440, 2441, 2442, 2448, 2449, 2450,

              2452, 2453, 2464, 2465, 2466, 2468, 2469, 2472, 2473, 2474, 2563, 2566, 2571, 2572, 2573, 2579,

              2582, 2584, 2585, 2586, 2595, 2598, 2603, 2604, 2605, 2608, 2609, 2610, 2612, 2613, 2627, 2630,

              2635, 2636, 2637, 2643, 2646, 2648, 2649, 2650, 2656, 2657, 2658, 2660, 2661, 2664, 2665, 2666,

              2691, 2694, 2699, 2700, 2701, 2707, 2710, 2712, 2713, 2714, 2723, 2726, 2731, 2732, 2733, 2736,

              2737, 2738, 2740, 2741, 2752, 2753, 2754, 2756, 2757, 2760, 2761, 2762, 2768, 2769, 2770, 2772,

              2773, 2816, 2817, 2818, 2820, 2821, 2824, 2825, 2826, 2832, 2833, 2834, 2836, 2837, 2848, 2849,

              2850, 2852, 2853, 2856, 2857, 2858, 2880, 2881, 2882, 2884, 2885, 2888, 2889, 2890, 2896, 2897,

              2898, 2900, 2901, 3072, 3073, 3074, 3076, 3077, 3080, 3081, 3082, 3088, 3089, 3090, 3092, 3093,

              3104, 3105, 3106, 3108, 3109, 3112, 3113, 3114, 3136, 3137, 3138, 3140, 3141, 3144, 3145, 3146,

              3152, 3153, 3154, 3156, 3157, 3200, 3201, 3202, 3204, 3205, 3208, 3209, 3210, 3216, 3217, 3218,

              3220, 3221, 3232, 3233, 3234, 3236, 3237, 3240, 3241, 3242, 3328, 3329, 3330, 3332, 3333, 3336,

              3337, 3338, 3344, 3345, 3346, 3348, 3349, 3360, 3361, 3362, 3364, 3365, 3368, 3369, 3370, 3392,

              3393, 3394, 3396, 3397, 3400, 3401, 3402, 3408, 3409, 3410, 3412, 3413};

int l[13] = {0, 0, 1, 2, 5, 10, 20, 38, 71, 130, 235, 20, 744};

int main()

{

    int n, s;

    while (scanf("%d", &n) != EOF)

    {

        s = l[n];*//共有l[n]个符合要求的数*

        for (int i = 0; i < s; i++)

        {

            for (int j = n - 1; j >= 0; j--)

            {

                if ((a[i] & (1 << j)))

                    printf("1");*//从二进制数的第n位开始，第n为和1与，其他位于0与。判断第n位是0还是1*

                else

                    printf("0");

            }

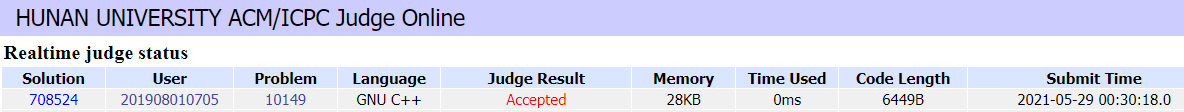
            printf("\n");

        }

    }

}



****

