On Fast Multiplication in Binary Finite Fields and Optimal Primitive Polynomials over GF(2)

Alexander Maximov
alexander.maximov@ericsson.com
Ericsson Research

Helena Sjoberg
hsjob@kth.se
KTH Royal Institute of Technology

September 13, 2017

Abstract

In this paper we present a number of algorithms and optimization techniques to speedup computations in binary extension fields over GF(2). Particularly, we consider multiplication and modular reduction solutions. Additionally, we provide the table of optimal binary primitive polynomials over GF(2) of degree $2 \leq d < 2048$, and the class of functions for optimal modular reduction algorithms for each of the listed polynomials.

We give implementation examples targeting Intel CPU architectures, but generic results can be applied on other platforms as well.

1 Introduction

Primitive polynomials over GF(2) may be used to construct binary extension fields. The elements of such a field are polynomials of a degree lower than the generating primitive polynomial $f(z)$. Each coefficient of the polynomials is either 0 or 1, and all operations are executed modulo the primitive polynomial.

For efficient implementation of a modular multiplication algorithm we propose the optimal Karatsuba-like multiplication formulas for $n = 2, 3, 4, 5$-way multiplication, where not only the number of multiplications is minimized, but also the number of other operations were taken into consideration. For an effective implementation of symmetric multiplication schemes we propose a generic 4-step methodology, which makes it possible to utilize the intrinsic instruction `PCLMULQDQ` in the most efficient way. The presented schemes may be useful for a general purpose reduction over polynomials or in other fields of applications.

For the purpose of efficient implementation of the Montgomery reduction technique we present Karatsuba-like schemes for partial multiplications, for when we are only interested in the low or high half of the result. Some of the presented schemes are symmetric and the others are non-symmetric.

The performance of many cryptographic algorithms rely on the efficiency of a modular reduction, where the result of a multiplication, or an element squaring, should then be taken in modulo the primitive polynomial.

Montgomery and Barrett reduction techniques may be used for reduction modulo an arbitrary primitive. However, when the modular polynomial has a low Hamming weight then the reduction may work faster.

In this paper we consider polynomials of two types: $f(z) = z^d + z^a + 1$ and $f(z) = z^d + z^c + z^b + z^a + 1$ over GF(2), where $a, b, c, d \in \mathbb{N}$ and $0 < a, b, c < d < 2048$. The first type is the result of the case when $a = b = c$.

For every value of d, we searched and found the first primitive polynomials in lexiographical order of Hamming weights 3 and 5.

For the classical reduction technique we developed a class of optimal reduction algorithms, where the number of CPU instructions is as minimal as possible. Correspondingly, for every value of d we searched for an optimal primitive polynomial with the lowest penalty for the reduction algorithm. The penalty is approximated as a number of cycles, or the count of CPU instructions. This is an upper bound for the real number of cycles, which might be less due to potential CPU and compiler further optimizations.

We have tested all of the optimal primitives we have found and measured the reduction speed against the best possible implementation of the Montgomery reduction technique. The measurements are also given in Appendix C. As d increases the optimal reduction technique starts to outperform the Montgomery reduction.

2 Field Elements Representation and CPU Intrinsics

2.1 Representation of Elements in the Binary Extension Field

An element of the binary extension field generated by the primitive polynomial $f(z)$ over GF(2) of degree d may be represented by a d-bit binary string. The d bits $(x_{d-1}, ..., x_0)$ map one to one to the coefficients in the field element $x(z)$ so that $x(z) = x_{d-1}z^{d-1} + ... + x_0z^0$.

Considering modern CPU architectures we suggest storing the field elements in n 64-bit blocks, as in the following;

```
typedef uint64_t u64;
typedef uint64_t etype[n];
```

where either $n = \lfloor \frac{d+63}{64} \rfloor$ or $n = 1 + \lfloor \frac{d}{64} \rfloor$ depending on whether we want the coefficient for the term z^d to fit into `u64[n]` or not.

Using n 64-bit blocks to represent the elements, $(b_{n-1}, b_{n-2}, ..., b_1, b_0)$ where $b_i = (x_{di+d-1}, x_{di+d-2}, ..., x_{di+1}, x_{di})$, we can see each block as a separate polynomial of degree $d-1$ which also functions as a coefficient. The element in the extension field is then $x(z) = X(T) = b_{n-1}T^{n-1} + b_{n-2}T^{n-2} + ... + b_1T + b_0$ where $T = z^{64}$. Each such block can also be called a *limb*.

When we perform a multiplication $r(z) = x(z) \cdot y(z)$ without performing the following modular reduction, the resulting element $r(z)$ would then fit into `u64[2n]`. More exactly, the number of valid bits in the binary string-representation of $r(z)$ will be at most $2d-1$.

In this paper we will focus mainly on multiplications and reductions, although squaring with reduction is clearly a related subject. However, squaring may be done faster than multiplication when we work in a binary extension field over GF(2), which makes it a usually less critical area to optimize.

2.2 CPU Intrinsics

In this paper we will for simplicity use macros to refer to certain intrinsic functions. We will here define the most common of those:

```
// Note: casting from __m128i to/from __m128 in some of the
//       below macros is skipped for simplicity reasons.

// Basic 128-bit operations
#define XOR(a, b)        _mm_xor_si128(a, b)
#define SHHR(a)          _mm_srli_si128(a, 8)
#define SHHL(a)          _mm_slli_si128(a, 8)
#define SHUFFLE2x64(a, b, im) _mm_shuffle_pd(a, b, im))

// Carryless multiplication of low/high 64-bits of x and
// low/high 64-bit of y
#define CLMUL_LOHI(x, y) _mm_clmulepi64_si128(x, y, 0x10)
#define CLMUL_HILO(x, y) _mm_clmulepi64_si128(x, y, 0x01)

// Returns a new 128-bit register combined from low/high halves
// of x and y
#define MOVELH(a, b)     _mm_movelh_ps(a, b)
#define MOVEHL(a, b)     _mm_movehl_ps(a, b)
```

For a substantially more detailed description of the above intrinsic functions we refer to Intel's Intrinsics Guide [2].

3 Implementation of Multiplication Algorithms

3.1 Introduction

Most modern CPUs have intrinsic functions and `PCLMULQDQ` is the most interesting one for fast multiplication of two binary polynomials represented by two `u64`. That is if the degree d of the primitive modular polynomial is less than or equal to 64. Then the multiplication of the two polynomials $x(z)$ and $y(z)$ can be performed with a single call of the above instruction, excluding the instructions for loading the inputs into one or two 128-bit registers and storing the result.

However, if elements of the field are represented as `u64[n]` blocks with $n > 1$ we need to perform a fast multiplication of two polynomials with $n \times 64$-bit coefficients each. As described in section 2.1 these polynomials can also be viewed as having n coefficients, each coefficient being a sub-polynomial represented by one of the 64-bit blocks. The basic multiplication of two coefficients here is then performed by `PCLMULQDQ` too.

For not so large values for n we believe that a Karatsuba-like multiplication scheme is the most effective. For a small n, when $n < 8$, we could even implement intrinsic functions that perform multiplication the optimal way.

In [3] Peter Montgomery present a class of division-free Karatsuba-like formulas for polynomials with five, six or seven terms. Montgomery states that with these formulas two five-term polynomials can be multiplied with 13 scalar multiplications, two six-term polynomials with 17 scalar multiplications and two seven-term polynomials with 22 scalar multiplications. By mixing these formulas recursively with the classic formulas for multiplication of two-term and three-term he further states these formulas will lead to better performance for many other degrees too. The question on how to implement these schemes with the minimum number of other operations too does however remain, as not only scalar multiplication operations counts when it comes to performance. This leads to our first problems;

Problem 1. The multiplication scheme with the best performance can not just be determined by the minimum number of multiplications. One also needs to take into account other basic operations, such as `XOR`s, and minimize the number of operations of all kinds.

Problem 2. If one implements the Montgomery reduction algorithm then the `redc()` function needs to perform a *partial multiplication* where we are only interested in the low or high half of the result. This may save a few unnecessary multiplications from the full scheme. What would be the optimal Karatsuba-like formulas for partial multiplications, with minimum number of multiplications and the minimum number of other basic operations?

3.2 Symmetric Karatsuba-like Multiplication Schemes

One possible candidate for an optimal multiplication algorithm for two-term polynomials is the following Karatsuba-like multiplication algorithm:

```
w0 = (x0+x1) * (y0+y1)
w1 = (x1) * (y1)
w2 = (x0) * (y0)
c0 = w2
c1 = w0+w1+w2
c2 = w1
```

where the input polynomials $X = x_1T + x_0$ and $Y = y_1T + y_0$, with $T = z^{64}$ and the coefficients x_i, y_i are 64-bit sub-polynomials over GF(2). The result of the multiplication is $C = c_2T^2 + c_1T + c_0$, where each c_i is now a 128-bit object.

The above scheme is *symmetric* since we will evaluate the same additions and multiplications for both $R = X \cdot Y$ and $R = Y \cdot X$. This as the sums $(x_0 + x_1)$ and $(y_0 + y_1)$ are evaluated for the same indices for both X and Y and then the intermediate multiplications are done for similar linear combinations of x_i, y_i-terms.

3.3 Intrinsics API for Extended Multiplications

The mentioned intrinsic function `PCLMULQDQ` takes two 128-bit registers as input and, using the selector, multiplies either the lower and/or the high halves of the input registers and produces a 128-bit result.

Assume we sketch the 2x64-bit multiplication algorithm as a new intrinsic function for $n \leq 2$:

```
inline void _gf128_mul(__m128i & x, __m128i & y)
{...}
```

with x and y being two polynomials over GF(2) each represented by 128 bits. Thus x_0 is the low 64-bits of x, and x_1 is the upper 64-bits of x. The same applied for y.

We then expect the result of multiplication will fit into 2x128 bit registers, so therefore we would need to return a 256-bit result; thus we can use (x, y) as the return registers with low 128 bits of the result in y and high 128 bits of the result in x, to avoid extra transfers between the registers and memory at this moment. Memory transfers may however be done in the caller function if no further intrinsics operations are needed.

For $n \leq 3$ and $n \leq 4$ respectively we could design an intrinsics multiplication API as follows:

```
inline void _gf192_mul(__m128i   xhi, __m128i & xlo,
                       __m128i & yhi, __m128i & ylo)

inline void _gf256_mul(__m128i & xhi, __m128i & xlo,
                       __m128i & yhi, __m128i & ylo)
...
```

with the returned result in registers $[x_hi|x_lo|y_hi|y_lo]$.

Note that in 192-bit case ($n \leq 3$) only the lower 64 bits of the registers x_{hi} and y_{hi} contain valid input and the result then only needs to use 3×128 bits so the register x_{hi} is not used to store the result and is therefore not returned.

3.4 Evaluation Bottleneck

In the example above for $n = 2$, x_1 is stored in the upper 64 bits of the 128-bit register and x_0 is stored in the lower 64-bits of the same register. The same applies to the register storing y. This means that when we need to evaluate the sum of $(x_0 + x_1)$ and $(y_0 + y_1)$ the two parts of each sum is not alligned correctly to be summed easily. How do we solve this?

One possible way would be to rotate the input registers x and y, then `XOR` each input register with its rotated version. This would take 2 instructions for each input register. Note that each of the two `XOR`s will give us a 128-bit register where only 64-bits are useful for further computation.

The situation for $n = 2$ may look simple, but when it comes to multiplication schemes for larger values for n, this problem becomes more serious since many more instructions would be required to align the registers before executing the `XOR` instruction. We would therefore want to remove the aligning problem regardless of the value of n and regardless of the multiplication scheme used. We would also like to fully use each 128-bit `XOR` fully for both the upper and lower part of those 128 bits. If this succeeds we would be able to reduce the number of needed `XOR`s by a factor of 2.

4 A 4-Step Methodology for Efficient Implementation of any Symmetric Karatsuba-like Multiplication Algorithm

We will now present a nice and generic way to implement any given symmetric multiplication scheme with the following 4 steps:

1. Recombination
2. Mul-Evaluation
3. Sum-Evaluation

4. Assembly

For now we will use the following example for multiplication of two 256-bit binary polynomials over GF(2), which means that here $n = 4$:

```
w0 = (x0+x1+x2+x3) * (y0+y1+y2+y3)
w1 = (x2+x3) * (y2+y3)
w2 = (x1+x3) * (y1+y3)
w3 = (x3) * (y3)
w4 = (x0+x2) * (y0+y2)
w5 = (x2) * (y2)
w6 = (x0+x1) * (y0+y1)
w7 = (x1) * (y1)
w8 = (x0) * (y0)
c0 = w8
c1 = w6+w7+w8
c2 = w4+w5+w7+w8
c3 = w0+w1+w2+w3+w4+w5+w6+w7+w8
c4 = w2+w3+w5+w7
c5 = w1+w3+w5
c6 = w3
```

where $X = x_3T^3 + x_2T^2 + x_1T + x_0$ and $Y = y_3T^3 + y_2T^2 + y_1T + y_0$, x_i, y_i are 64-bit blocks and $T = z^{64}$. The result is then $C = X * Y = c_6T^6 + c_5T^5 + c_4T^4 + c_3T^3 + c_2T^2 + c_1T + c_0$ where c_i are 128-bit blocks.

4.1 Recombination

The purpose of the recombination step is to shuffle the input 128-bit registers $x_{hi}|x_{lo}|y_{hi}|y_{lo}$ into another set of 128-bit registers $t_3|t_2|t_1|t_0$. This is done in such a way that t_0 now contains low 64-bits of x_{lo} and low 64-bits of y_{lo}, t_1 contains high halves of x_{lo} and y_{lo}; t_2 contains low halves of x_{hi} and y_{hi}; t_3 contains high halves of x_{hi} and y_{hi}. The shuffeling mixes the input vectors of X and Y into registers t_i such that low parts of each t_i contain 64-bit blocks of X and high halves of each t_i contain 64-bit blocks of Y. Note also that both x_i and y_i are now placed together in t_i, for all values of i.

The benefit of such recombination is that further evaluation of the symmetric sums can now be done in parallel for x_i and y_i terms, using t_is instead.

For the example above, the recombination code looks as follows:

```
__m128i t0 = MOVELH(xlo, ylo);
__m128i t1 = MOVEHL(ylo, xlo);
__m128i t2 = MOVELH(xhi, yhi);
__m128i t3 = MOVEHL(yhi, xhi);
```

4.2 Mul-Evaluation

In this step we compute the intermediate 128-bit variables w_0 to w_8 using t_0 to t_3 for parallel computing of the sums of the x and y coefficients. The optimal evaluation of the sums may involve other intermediate sums (`tmpx`) to reduce the total number of `XOR`s needed. The following is our example of the optimal evaluation of w_0 to w_8:

```
--- Mul-Evaluation ---
tmp1 = t0+t1
tmp2 = t2+t3
e8 = t0         => w8 = CLMUL_LOHI(e8, e8)
e7 = t1         => w7 = CLMUL_LOHI(e7, e7)
e6 = tmp1       => w6 = CLMUL_LOHI(e6, e6)
e5 = t2         => w5 = CLMUL_LOHI(e5, e5)
e4 = t0+t2      => w4 = CLMUL_LOHI(e4, e4)
e3 = t3         => w3 = CLMUL_LOHI(e3, e3)
e2 = t1+t3      => w2 = CLMUL_LOHI(e2, e2)
e1 = tmp2       => w1 = CLMUL_LOHI(e1, e1)
e0 = tmp2+tmp1 => w0 = CLMUL_LOHI(e0, e0)
```

Thus, the evaluation of w_0 to w_8 can be implemented with nine carryless 64-bit multiplications and just five 128-bit `XOR`s as follows:

```
// mul-evaluation, 9mul+5op
__m128i tmp1 = XOR(t0, t1);
__m128i tmp2 = XOR(t2, t3);
__m128i w8 = CLMUL_LOHI(t0, t0);
__m128i w7 = CLMUL_LOHI(t1, t1);
__m128i w6 = CLMUL_LOHI(tmp1, tmp1);
__m128i w5 = CLMUL_LOHI(t2, t2);
__m128i w4 = XOR(t0, t2);
        w4 = CLMUL_LOHI(w4, w4);
__m128i w3 = CLMUL_LOHI(t3, t3);
__m128i w2 = XOR(t1, t3);
        w2 = CLMUL_LOHI(w2, w2);
__m128i w1 = CLMUL_LOHI(tmp2, tmp2);
__m128i w0 = XOR(tmp2, tmp1);
        w0 = CLMUL_LOHI(w0, w0);
```

4.3 Sum-Evaluation

When w_0 to w_8 are all computed, we can evaluate the c-values. This part is straight-forward and may be implemented with 13 `XOR`s:

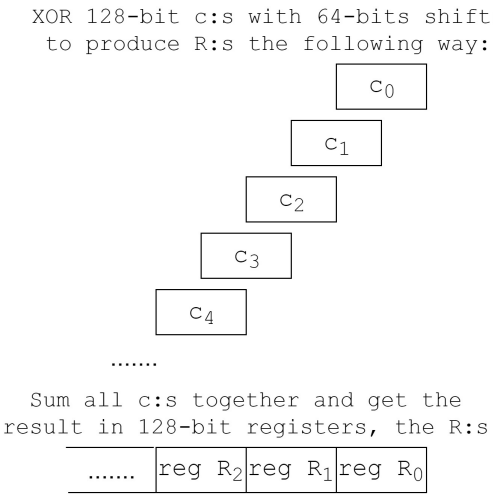
```
--- Sum-Evaluation ---
tmp1 = w2+w3
tmp2 = w7+w8
t9 = w4+w5+tmp2
c6 = w3
c5 = w1+w3+w5
c4 = w5+w7+tmp1
c3 = w0+w1+w6+t9+tmp1
c2 = t9
c1 = w6+tmp2
c0 = w8
```

4.4 Assembly

Now we have the result $C = X * Y = c_6T^6 + c_5T^5 + c_4T^4 + c_3T^3 + c_2T^2 + c_1T + c_0$ where the coefficients of c_i are 128-bit overlapping blocks but we need to translate this back to the format originally used for X and Y, thus we need to remove the overlapping.

The purpose of the final step is to combine the 128-bit c-coefficients into the sequence of 64-bit values and place them into the returned `__m128i` registers.

Visually what we need to do is the following summation:



Each resulting register R_i may be of one of three types. It might be the first register, the last register or one of the middle register, and in each case the summation needs to be performed slightly different. Thus we define three intrinsic help functions to perform this assembly effectively. Each function has two basic CPU instructions:

```
// assembly the low element, 2op
```

```
inline __m128i _gfn_assemble_lo(__m128i t, __m128i tp)
{       return XOR(t, SHHL(tp));
}

// assembly the high element, 2op
inline __m128i _gfn_assemble_hi(__m128i tm, __m128i t)
{       return XOR(t, SHHR(tm));
}

// assembly an element in the middle, 2op
inline __m128i _gfn_assemble_mi(__m128i tm, __m128i t, __m128i tp)
{       return XOR(t, SHUFFLE2x64(tm, tp, 1));
}
```

In our example above for the 256-bit multiplication algorithm the assembly part can thus be implemented as:

```
// assembly, 8op
ylo = _gfn_assemble_lo(    c0, c1);
yhi = _gfn_assemble_mi(c1, c2, c3);
xlo = _gfn_assemble_mi(c3, c4, c5);
xhi = _gfn_assemble_hi(c5, c6);
```

4.5 Montgomery Reduction and Non-Symmetric Partial Multiplications

In the Montgomery reduction algorithm we need to have *partial multiplication* schemes, where we are only interested either in the low or upper half of the result.

Symmetric schemes for partial multiplications require less number of multiplications than in full-multiplication schemes. The advantage with a symmetric scheme is also that the four generic steps above can be applied.

The disadvantage of a symmetric scheme here would be that it might not be optimal in terms of the number of multiplications. *Non-symmetric* partial-multiplication schemes might require less multiplication operations. The disadvantages for non-symmetric schemes are the difficulty of making an optimal implementation and the time it takes to search for a non-symmetric scheme is much longer than for a symmetric one.

4.6 Our Results

We have written the simulation program that is searching for the best multiplication schemes with the lowest number of multiplications, **and** the lowest penalty (count) of other basic operations, such as `XOR`, rotations, shifts, recombinations, etc.

The resulting optimal multiplication schemes can be found in Appendix A.

5 Optimal Binary Primitive Polynomials

Performing the reduction is computationally expensive. By selecting a suitable primitive polynomial for the construction of the extension field a signiticant speedup may be achieved.

Montgomery and Barrett reductions can improve the performance a lot and can be used with an arbitraty primitive polynomial. This is especially true when using the `PCLMULQDQ` instruction. However, for certain cases the classical reduction may perform better, especially when n is large, which is when we have many limbs. The reason behind this is that the Montgomery reduction algorithm needs two partial multiplications performed on each half of the input, which is the result of multiplication that we want to reduce. At the same time for the classical algorithm we could instead do a few shifts and `XOR`s.

We will further on in this section assume that elements are represented in data structures `u64[n]` (see section 2.1), where the upper bits which are not used are zeroized, meaning they are all zero. We will also assume that after multiplication $r(z) = x(z) \cdot y(z)$ the resulting polynomial $r(z)$ has size `u64[2n]` with $2 \cdot d - 1$ valid bits and zeroized upper bits.

5.1 Class of Optimal Reduction Algorithms

Assume we have $r(z)$, the result of a single multiplication or squaring where the first $2 \cdot d - 1$ bits are valid and upper bits are zeroized. We want to compute $r(z)$ modulo $f(z)$, where $f(z)$ is the primitive binary polynomial of degree d.

The classical reduction algorithm consists of a number of rounds of *bit-reductions* until all bits $\geq d$ become zero. One bit-reduction is performed as follows; we take the bit-shift `w=r>>d` (all bits of $r(z)$ with indices $\geq d$) and zeroize the upper bits of r starting from the d's bit. We then `XOR` w with r at all bit-positions of $f(z)$ where $f(z)$ is one, except for the bit position d. Then we repeat the same round if there are still ones at positions $\geq d$ of $r(z)$.

We have developed a class of optimal reduction algorithms for degrees $2 \leq d < 2048$, and for optimal primitive polynomials that have the form of either $f(z) = z^d + z^a + z^b + z^c + 1$ or $f(z) = z^d + z^a + 1$, where the second type appears if we set $a = b = c$.

The class of reduction algorithms is presented in Appendix B. The API is as follows:

```
template<u32 algorithm, u64 data> inline void reduction_templ(u64 * r);
```

which means that the reduction algorithm for a certain primitive is constructed by the compiler given the constant parameters: a 32-bit value of `algorithm` and a 64-bit value of `data`. The complete definition of that function is listed in Appendix B.

Although the function `reduction_templ<>` has many `if-else` branches, those branches are constant-valued, and, therefore, the compiler will generate the reduction code without branches, given the constant parameters to the template.

For example, if we take the primitive $f(z) = z^{127} + z^7 + 1$ the corresponding reduction of `u64 r[4]` is then called as

```
reduction_templ<0xf00017c0ul, 0x7f0102000000000full>(r);
```

where `<0xf00017c0ul, 0x7f0102000000000full>` are the constant parameters associated with the selected primitive $f(z)$. The compiler will produce *approximately* the following **branchless reduction code** for the selected primitive polynomial $f(z)$. Approximately, because of there are other optimization tricks that the compiler does which we currently do not take into account. The code below is what the compiler produces if we place that reduction into a `for(;;)` loop.

```
$LL15@wmain: ; for(;;) loop
vmovdqu xmm0, XMMWORD PTR
↪  r$[rbp+15]
vmovdqu XMMWORD PTR w$[rbp], xmm0
movzx   eax, BYTE PTR r$[rbp+31]
mov     QWORD PTR w$[rbp+16], rax
vmovdqu XMMWORD PTR r$[rbp+16],
↪  xmm5
and     cl, 127
mov     BYTE PTR r$[rbp+15], cl
and     QWORD PTR w$[rbp], -128
vmovdqu xmm1, XMMWORD PTR w$[rbp]
vmovdqu xmm0, XMMWORD PTR r$[rbp]
vpxor   xmm4, xmm0, xmm1
xor     BYTE PTR r$[rbp+16], al
vpsrlq  xmm2, xmm1, 7
vmovdqu xmm0, XMMWORD PTR
↪  w$[rbp+8]
vpsllq  xmm1, xmm0, 57
vpxor   xmm3, xmm1, xmm2
vpxor   xmm0, xmm4, xmm3
vmovdqu XMMWORD PTR r$[rbp], xmm0
mov     rax, QWORD PTR r$[rbp+15]
mov     QWORD PTR r$[rbp+16], rbx
movzx   ecx, BYTE PTR r$[rbp+15]
and     cl, 127
mov     BYTE PTR r$[rbp+15], cl
and     rax, -128
mov     rdx, QWORD PTR r$[rbp]
xor     rdx, rax
shr     rax, 7
xor     rdx, rax
mov     QWORD PTR r$[rbp], rdx
dec     r8
jne     SHORT $LL15@wmain
```

This is without branches and with all constant values from the input parameters `algorithm` and `data` extracted and encoded in the assembly code at compile time.

5.2 The Ideas Behind the Optimal Reduction Algorithm

Minimize the Hamming Weight To make an optimal reduction algorithm efficient we should aim at a Hamming weight of the primitive polynomial which is as low as possible. The less the Hamming weight, the less number of bit-reductions (per round) required. Due to this we have only considered primitives of the forms $f(z) = z^d + z^a + z^b + z^c + 1$ and $f(z) = z^d + z^a + 1$.

Minimize the Exponents Another idea to minimize the number of bit reductions is to let the exponents a, b, c be at most $d/2$. If this holds we will only need two rounds of bit-reductions. As a counter example, if we take $f(z) = z^{128} + z^{127} + 1$, a random polynomial fulfilling the requirement in question, as our primitive polynomial we will need to perform about 127 bit-reductions until all bits at positions ≥ 128 become zero. But if we take $f(z) = z^{128} + z + 1$ then at most 2 round are needed.

Note also that although the number of rounds needed for $f(z) = z^{128} + z + 1$ and $f(z) = z^{128} + z^{64} + 1$ are the same, the difference is that in the second round of reduction we only need to deal with 1 bit in the first case but 63 bits in the second. Selecting a, b, c so that they are as small as possible is yet another approach to reduce the number of CPU instructions in the resulting reduction algorithm.

When the reduction algorithm performs the second round we can reduce the number of `u64`-blocks by at least 2. If the selected a, b, c happen to be much smaller than $\frac{d}{2}$ then the number of limbs n of `w=r>>d` in the second round may be very small, much smaller than the upper bound $n/2$, and that reduces the cost of basic operations even further.

Move the Pointer *Bit-shifting* of a long $r(z)$, when we have many $m = 2 \times n$ limbs, is an expensive operation. We should therefore try to avoid it as much as possible. If $r(z)$ has m limbs then $r(z)$ is represented by `u64 r[m]`. Left or right bit-shifting would then cost us at least $1 + 3 \times (m - 1)$ basic CPU instructions.

On the other hand, if we want to shift by $8t$ bits, instead, then just change the pointer of the source and/or destination by t bytes by which we then eliminate the need of expensive bit-shifting. This way we, essentially, get our new input or output data almost for free compared to actually performing the bit-shifting operations.

Set $a \equiv d$ or $0 \pmod 8$ if possible When $a \equiv d \pmod 8$, it means that we can just `XOR` the upper bits with r shifted by $\frac{d-a}{8}$ bytes to the right, without having to do an actual bit-shifting of the long value.

Since we anyways will need to make at least one shift by d bits to the right, if d is not aligned by 8 bits, in order to `XOR` at $...+z^0...$ position of the primitive $f(z)$, we can thus try to search for such indexes a, b, c that, perhaps, some of them may be equal to 0 modulo 8. In this case the bit-reduction at such a bit-position is again just a byte shifting, instead of a bit-shifting.

Set $b \equiv d$, 0 or $a \pmod 8$ if possible When a is not equivalent to d or $0 \pmod 8$ we have to make a bit-shifting by $(a \pmod 8)$ bits. In this case, for the next bit-location b we can then try to find such b that

$b \equiv a \pmod 8$, so that for b we will not need to make bit-shifting since we have already done that for the bit-position a.

Summarizing the above, we can avoid bit-shifting for the position b if it is $\equiv d, \equiv 0$, or $\equiv a \pmod 8$. It seems like we have now a larger choice for an optimal b. Same strategy works for the last index c, that may be congruent to either or some of $0, a, b, d$ modulo 8.

Example Previously we presented the assembly code for reduction modulo the primitive polynomial $f(z) = z^{127} + z^7 + 1$. In many publications we have seen people choose $f(z) = z^{127} + z + 1$. That polynomial does however have a worse reduction scheme performance-wise since it requires two bit-shifts in one round of bit-reduction. This in order to `XOR` of `w=r>>127` at offsets 0 and 1 bits.

If we replace $+z^1$ by $+z^7$ in $f(z)$, then we would need to make only one bit-shifting in one bit-reduction round since the position $a = 7$ is aligned with $d = 127$. We simple do: (a) read w from r at byte offset $d/8$; (b) set 7 low bits of w to zero; (c) `XOR` of w with r at **byte offset** 0 (which corresponds to bit-reduction at position 7); (d) then right shift w by 7 bits (here is one bit-shifting needed); (e) `XOR` w with r at **byte offset** 0, which corresponds to bit-reduction at position 0.

5.3 Searching for Optimal Primitives

When selecting a, b, c, such that $1 < a, b, c < d$, consider the binary polynomial $f(z) = z^d + z^a + z^b + z^c + 1$ of degree d. The polynomial is said to be primitive if it is irreducible and the order of z is $2^d - 1$, that is it is primal. The test for irreducibility is quite simple, we need to raise z to $2^d - 1$ and check that $z^{2^d-1} \pmod{f(z)} \equiv 1$.

The test for primality is more complicated. We have to test that if we take any non trivial factor of $2^d - 1$, say k, then $z^{(2^d-1)/k} \not\equiv 1 \pmod{f(z)}$ must hold. For this test we need to have factorization of $2^d - 1$ for all $2 \leq d < 2048$. We have borrowed the factorizations from the Cunningham project [1] and parsed it to extract the factors. Then, having all the factorizations of $(2^d - 1)$ we implemented the primality test.

An efficient searching of a primitive polynomial $f(z)$ with the best reduction algorithm requires the computation of some penalty, in order to determine which of any two choices, $f_1(z)$ and $f_2(z)$, is preferable.

Thus, we first start with computing the penalty of each new candidate polynomial. If it's penalty is larger than the best found candidate so far we drop this candidate and proceed to check the next one. Otherwise we continue by checking if the candidate is an irreducible polynomial, and then we test it for primality. If all tests are passed then that new candidate becomes the now currently best optimal primitive polynomial with the smallest possible penalty.

5.4 Simulation of Reduction Algorithm and Penalty Calculations

For a given candidate polynomial the generation program constructs and simulates the best reduction algorithm for the candidate polynomial. It keeps track on-the-fly the number of bits the reduction algorithm needs to take care of and also performs virtual bit and byte shifts in the reduction steps where it is applicable.

While virtually shifting to the right or to the left, the searching program also counts the number of bits that we need to take care of during the reduction. Thus the byte overflow bit is set on or off per each step per each shift and per each `XOR` operation. Such a byte overflow is a rare situation and it may happen only in the first round of bit-reduction.

For the second round the algorithm just selects a large enough number of limbs to operate on in order to avoid byte overflow handlings. This may not be done in the first round.

If we return to our exampled primitive polynomial $f(z) = z^{127} + z^7 + 1$, then in the first round we should read 17 bytes into w, starting from the offset 15 ($= 127 >> 3$), in order to load all valid bits up to $2 * d - 1$. This means that it is not enough to load only `2*u64` words. Here we have a situation with an extra 17^{th} byte to care of as well. We call this situation for *byte overflow*.

Alongside, the simulation of the reduction algorithm generates needed values of the `algorithm` and `data` parameters and also computes the total penalty of the invisioned reduction algorithm of the given polynomial.

The penalty is basically the maximum number of basic CPU instructions to be exectued. The smaller the penalty the faster the reduction algorithm will work, projectively. We want to have a penalty as small as possible. The primitive with the smallest penalty is selected.

Note: The penalty is only computed as the *maximum* number of CPU instructions, because of there are other optimization tricks that the compiler does in cooperation with CPU, but these optimizations are harder to count. For example, the compiler may keep some variables in registers for connected or related computations, thus avoiding at least two instructions to transfer a value between a register and a memory location. These we did not count.

5.5 The Construction of 32-bit `algorithm` and 64-bit `data`

The constant 32-bit parameter `algorithm` encodes the constant branching logic for bit-reduction at the four points $0, a, b, c$ (or two points $0, a$ in case $a = b = c$). It is constructed as follows: There are 4 blocks of 7 bits each. The first 3 bits encode the number of bit shifts, the 4th bit encode if we need to shift to the right, and the 5th bit encode if we need to shift to the left. The 6th and 7th bit flags says if the shift or `XOR` needs to be performed

with handling of byte overflow.

The last 4 bits at positions 28 to 31 encode general information of the algorithm for reduction with the corresponding primitive:

- **bit28:** is a byte overflow flag. It says that in the first round copying only 64-bit blocks is not enough, and a single highest 8-bit byte should be considered as well in a byte-manner, since 64-bit manner would be outside the input buffer r. This does not apply to the second round since there we will not have 64-bit overflows over the buffer due to the already done partial reduction in the first round.
- **bit29:** is true if the degree d is not aligned, if $d \pmod 8 \neq 0$. This means that when taking r modulo z^d, when zeroizing the upper bits, it is not enough to make only certain bytes zero, but a mask to the last valid byte near the position d should be applied as well.
- **bit30:** is true if after loading high bytes of the input r, we should set the low bits to zero by applying a byte mask. This happens when d is not aligned, and when a, b, or c is equal to d in modulo 8.
- **bit31:** is true when $a = b = c$, that is when two rounds of reduction may be skipped.

The constant 64-bit parameter `data` contains eight bytes of the algorithm's constant parameters, and they are well described by the comments in the function body of `reduction_templ<>` itself. For further information please see Appendix B.

5.6 Lexicographically First Binary Primitive Polynomials

In [4] the authors provided the table of primitive polynomials up to $d < 5000$. The authors also used the results of the Cunningham project, just as we do in this paper. However, the primitive polynomials they provided for larger values of d are random. This despite the importance of keeping the middle exponents as low as possible for an efficient reduction algorithm.

We have done a similar search. However, in our result we present a table of lexicographically ordered primitive polynomials for each $2 \leq d < 2048$ of the two types: $f(z) = z^d + z^a + z^b + z^c + 1$ and $f(z) = z^d + z^a + 1$, if such polynomials exists.

References

[1] Machine-readable cunningham tables. http://cage.ugent.be/~jdemeyer/cunningham.

[2] Intel intrinsics guide. https://software.intel.com/sites/landingpage/IntrinsicsGuide/.

[3] P. L. Montgomery. Five, six, and seven-term karatsuba-like formulae. *IEEE Transactions on Computers*, 54(3):362–369, 2005.

[4] M. Živković. A table of primitive binary polynomials. *Mathematics of Computation*, 62(205):385–386, 1994.

Appendices

A Karatsuba-like Formulas for Multiplication $n = 2..5$

In the below you will find for each $n = 2, 3, 4, 5$, the best found multiplication scheme `MUL`, and truncated multiplication schemes `MULLO, MULHI` as the support for Montgomery reduction routines.

In each scheme we give the total cost in terms of the number of `PCLMULQDQ` multiplications and the number of basic operations, such as `XOR`s.

If a scheme is missing sum- or mul- evaluation, then those evaluations are trivial and straight-forward to implement.

```
// === n=2 MUL ===
// Cost = 3mul + 9op
w0 = (x0+x1) * (y0+y1)
w1 = (x1) * (y1)
w2 = (x0) * (y0)
c0 = w2
c1 = w0+w1+w2
c2 = w1

// === n=2 MULLO ===
// Cost = 3mul + 3op
w0 = (x1) * (y0)
w1 = (x0) * (y1)
w2 = (x0) * (y0)
c0 = w2
c1 = w0+w1

// === n=2 MULHI ===
// Cost = 3mul + 3op
w0 = (x1) * (y1)
w1 = (x1) * (y0)
w2 = (x0) * (y1)
c1 = w1+w2
c2 = w0

// === n=3 MUL ===
// Cost = 6mul+18op
w0 = (x1+x2) * (y1+y2)
w1 = (x0+x2) * (y0+y2)
w2 = (x2) * (y2)
w3 = (x0+x1) * (y0+y1)
w4 = (x1) * (y1)
w5 = (x0) * (y0)
c0 = w5
c1 = w3+w4+w5
c2 = w1+w2+w4+w5
c3 = w0+w2+w4
c4 = w2
--- Sum-Evaluation ---
tmp1 = w4+w5
c4 = w2
c3 = w0+w2+w4
c2 = w1+w2+tmp1
c1 = w3+tmp1
```

```
c0 = w5

// === n=3 MULLO ===
// Cost = 5mul+12op
w0 = (x2) * (y0)
w1 = (x0+x1) * (y0+y1)
w2 = (x1) * (y1)
w3 = (x0) * (y2)
w4 = (x0) * (y0)
c0 = w4
c1 = w1+w2+w4
c2 = w0+w2+w3

// === n=3 MULHI ===
// Cost = 5mul+10op
w0 = (x1+x2) * (y1+y2)
w1 = (x2) * (y2)
w2 = (x2) * (y0)
w3 = (x1) * (y1)
w4 = (x0) * (y2)
c2 = w2+w3+w4
c3 = w0+w1+w3
c4 = w1

// === n=4 MUL ===
// Cost = 9mul+30op
w0 = (x0+x1+x2+x3) *
     (y0+y1+y2+y3)
w1 = (x2+x3) * (y2+y3)
w2 = (x1+x3) * (y1+y3)
w3 = (x3) * (y3)
w4 = (x0+x2) * (y0+y2)
w5 = (x2) * (y2)
w6 = (x0+x1) * (y0+y1)
w7 = (x1) * (y1)
w8 = (x0) * (y0)
c0 = w8
c1 = w6+w7+w8
c2 = w4+w5+w7+w8
c3 = w0+w1+w2+w3+w4+w5+w6+w7+w8
c4 = w2+w3+w5+w7
c5 = w1+w3+w5
c6 = w3
--- Mul-Evaluation ---
tmp1 = t0+t1
tmp2 = t2+t3
e8 = t0
e7 = t1
e6 = tmp1
e5 = t2
e4 = t0+t2
e3 = t3
e2 = t1+t3
e1 = tmp2
e0 = tmp2+tmp1
--- Sum-Evaluation ---
tmp1 = w2+w3
tmp2 = w7+w8
t9 = w4+w5+tmp2
c6 = w3
c5 = w1+w3+w5
c4 = w5+w7+tmp1
c3 = w0+w1+w6+t9+tmp1
c2 = t9
c1 = w6+tmp2
c0 = w8

// === n=4 MULLO ===
// Cost = 8mul+19op
w0 = (x0+x3) * (y0+y3)
w1 = (x3) * (y3)
w2 = (x1+x2) * (y1+y2)
w3 = (x0+x2) * (y0+y2)
w4 = (x2) * (y2)
w5 = (x0+x1) * (y0+y1)
w6 = (x1) * (y1)
w7 = (x0) * (y0)
c0 = w7
c1 = w5+w6+w7
c2 = w3+w4+w6+w7
c3 = w0+w1+w2+w4+w6+w7
--- Sum-Evaluation ---
tmp1 = w6+w7
tmp2 = w4+tmp1
c3 = w0+w1+w2+tmp2
c2 = w3+tmp2
c1 = w5+tmp1
c0 = w7

// === n=4 MULHI ===
// Cost = 8mul+19op
w0 = (x2+x3) * (y2+y3)
w1 = (x1+x3) * (y1+y3)
w2 = (x0+x3) * (y0+y3)
w3 = (x3) * (y3)
w4 = (x1+x2) * (y1+y2)
w5 = (x2) * (y2)
w6 = (x1) * (y1)
```

```
w7 = (x0) * (y0)
c3 = w2+w3+w4+w5+w6+w7
c4 = w1+w3+w5+w6
c5 = w0+w3+w5
c6 = w3
--- Sum-Evaluation ---
tmp1 = w3+w5
tmp2 = w6+tmp1
c6 = w3
c5 = w0+tmp1
c4 = w1+tmp2
c3 = w2+w4+w7+tmp2

// === n=5 MUL ===
// Cost = 13mul+39op
w0 = (x0+x1+x2+x3+x4) *
     (y0+y1+y2+y3+y4)
w1 = (x0+x1+x3+x4) * (y0+y1+y3+y4)
w2 = (x3+x4) * (y3+y4)
w3 = (x1+x2+x4) * (y1+y2+y4)
w4 = (x2+x4) * (y2+y4)
w5 = (x4) * (y4)
w6 = (x0+x2+x3) * (y0+y2+y3)
w7 = (x3) * (y3)
w8 = (x0+x2) * (y0+y2)
w9 = (x2) * (y2)
w10 = (x0+x1) * (y0+y1)
w11 = (x1) * (y1)
w12 = (x0) * (y0)
c0 = w12
c1 = w10+w11+w12
c2 = w8+w9+w11+w12
c3 = w0+w1+w4+w5+w6+w7+w9+w12
c4 = w0+w2+w3+w5+w6+w7+w10+w11+w12
c5 = w0+w1+w3+w5+w8+w9+w11+w12
c6 = w4+w5+w7+w9
c7 = w2+w5+w7
c8 = w5
--- Mul-Evaluation ---
tmp1 = t0+t1
tmp2 = t0+t2
tmp3 = t2+t4
tmp4 = t3+t4
tmp5 = tmp4+tmp1
e12 = t0
e11 = t1
e10 = tmp1
e9 = t2
e8 = tmp2
e7 = t3
e6 = t3+tmp2
e5 = t4
e4 = tmp3
e3 = t1+tmp3
e2 = tmp4
e1 = tmp5
e0 = t2+tmp5
--- Sum-Evaluation ---
tmp1 = w4+w9
tmp2 = w10+w11
tmp3 = w5+w7
tmp4 = w8+w9+w11+w12
tmp5 = w0+w6+w12+tmp3
c8 = w5
c7 = w2+tmp3
c6 = tmp3+tmp1
c5 = w0+w1+w3+w5+tmp4
c4 = w2+w3+tmp5+tmp2
c3 = w1+tmp5+tmp1
c2 = tmp4
c1 = w12+tmp2
c0 = w12

// === n=5 MULLO ===
// Cost = 11mul+28op
w0 = (x0+x4) * (y0+y4)
w1 = (x4) * (y4)
w2 = (x1+x3) * (y1+y3)
w3 = (x0+x3) * (y0+y3)
w4 = (x3) * (y3)
w5 = (x1+x2) * (y1+y2)
w6 = (x0+x2) * (y0+y2)
w7 = (x2) * (y2)
w8 = (x0+x1) * (y0+y1)
w9 = (x1) * (y1)
w10 = (x0) * (y0)
c0 = w10
c1 = w8+w9+w10
c2 = w6+w7+w9+w10
c3 = w3+w4+w5+w7+w9+w10
c4 = w0+w1+w2+w4+w7+w9+w10
--- Sum-Evaluation ---
tmp1 = w9+w10
tmp2 = w7+tmp1
tmp3 = w4+tmp2
c4 = w0+w1+w2+tmp3
c3 = w3+w5+tmp3
c2 = w6+tmp2
```

```
c1 = w8+tmp1
c0 = w10

// === n=5 MULHI ===
// Cost = 11mul+26op
w0 = (x3+x4) * (y3+y4)
w1 = (x2+x4) * (y2+y4)
w2 = (x1+x4) * (y1+y4)
w3 = (x0+x4) * (y0+y4)
w4 = (x4) * (y4)
w5 = (x2+x3) * (y2+y3)
w6 = (x1+x3) * (y1+y3)
w7 = (x3) * (y3)
w8 = (x2) * (y2)
w9 = (x1) * (y1)
```

```
w10 = (x0) * (y0)
c4 = w3+w4+w6+w7+w8+w9+w10
c5 = w2+w4+w5+w7+w8+w9
c6 = w1+w4+w7+w8
c7 = w0+w4+w7
c8 = w4
--- Sum-Evaluation ---
tmp1 = w4+w7
tmp2 = w8+tmp1
tmp3 = w9+tmp2
c8 = w4
c7 = w0+tmp1
c6 = w1+tmp2
c5 = w2+w5+tmp3
c4 = w3+w6+w10+tmp3
```

B Class of Optimal Reduction Algorithms

```
1  #define ATST(i) (algorithm & (1UL<<(i))) /* test bit i of algorithm */
2
3  template<u32 algorithm, u64 data> inline void reduction_templ(u64 * r)
4  {
5      const int n1   = (data>>40)&0xff;  // size u64[n1] for the 1st round
6      const int n2   = (data>>48)&0xff;  // size u64[n2] for the 2nd round
7      const int doff = (data>> 0)&0xff;  // is d>>3, byte offset
8      const int dmsk = (data>>56)&0xff;  // is d&7, bit mask
9      const int off0 = (data>> 8)&0xff;  // offset for 1st reduction bit
10     const int off1 = (data>>16)&0xff;  // offset for 2nd reduction bit
11     const int off2 = (data>>24)&0xff;  // offset for 3rd reduction bit
12     const int off3 = (data>>32)&0xff;  // offset for 4th reduction bit
13     const int sh0  = (algorithm>> 0)&7;// shift for 1st reduction bit
14     const int sh1  = (algorithm>> 7)&7;// shift for 2nd reduction bit
15     const int sh2  = (algorithm>>14)&7;// shift for 3rd reduction bit
16     const int sh3  = (algorithm>>21)&7;// shift for 4th reduction bit
17
18     // ==== ROUND 1 ====
19     u64 w[n1+1];
20     gf2_copy<n1>(w, (u64*)(((u8*)r) + doff)); // copy upper bits
21
22     if(ATST(28)) // care about byte overflow
23         w[n1] = ((u8*)r)[(n1<<4)-1];
24     else
25         w[n1] = 0;
26
27     // zeroizing upper bits that has been transferred to w[]
28     if(ATST(29)) // if d is not aligned (d ≠ 0~mod8)
29     {    gf2_set_zero<n1>((u64*)(((u8*)r) + doff + 1));
30         ((u8*)r)[doff] &= dmsk;
31     }
32     else
33         gf2_set_zero<n1>((u64*)(((u8*)r) + doff));
```

```
34
35      // should I apply the mask after loading?
36      if(ATST(30))
37          w[0] &= ~(u64)dmsk;
38
39  #define AROUND1(k)\
40          if(ATST(k*7+3)) /* shift right 1..63 */\
41          {    if(ATST(k*7+5)) gf2_shr1const<n1+1, sh##k>(w);\
42              else gf2_shr1const<n1, sh##k>(w);\
43              gf2_xoradd<n1>(r, w);\
44          }\
45          else /* conditional shift left 0..63 */\
46          {    if(ATST(k*7+4))\
47                  if(ATST(k*7+5)) gf2_shl1const<n1+1, sh##k>(w);\
48                  else gf2_shl1const<n1, sh##k>(w);\
49              gf2_xoradd<n1>((u64*)(((u8*)r) + off##k), w);\
50              if(ATST(k*7+6)) /* take care of byte overflow, if any */\
51                  ((u8*)r)[off##k + (n1<<3)] ^= ((u8*)w)[n1<<3];\
52          }
53
54      AROUND1(0);
55      AROUND1(1);
56      if(!ATST(31)) /* stop if hw=3, continue if hw=5 */
57      {  AROUND1(2);
58         AROUND1(3);
59      }
60
61      // ==== ROUND 2 ====
62      gf2_copy<n2>(w, (u64*)(((u8*)r) + doff));
63
64      // zeroizing upper X
65      if(ATST(29)) // if d is not aligned
66      {    gf2_set_zero<n2>((u64*)(((u8*)r) + doff + 1));
67          ((u8*)r)[doff] &= dmsk;
68      }
69      else
70          gf2_set_zero<n2>((u64*)(((u8*)r) + doff));
71
72      // should I apply the mask after loading?
73      if(ATST(30))
74          w[0] &= ~(u64)dmsk;
75
76  #define AROUND2(k)\
77          if(ATST(k*7+3)) /* shift right 1..63 */\
78          {    gf2_shr1const<n2, sh##k>(w);\
79              gf2_xoradd<n2>(r, w);\
80          }\
81          else /* conditional shift left 0..63 */\
82          {    if(ATST(k*7+4))\
83                  gf2_shl1const<n2, sh##k>(w);\
84              gf2_xoradd<n2>((u64*)(((u8*)r) + off##k ), w);\
85          }
86
87      AROUND2(0);
```

```
88      AROUND2(1);
89      if(!ATST(31)) /* stop if hw=3, continue if hw=5 */
90      {  AROUND2(2);
91         AROUND2(3);
92      }
93  }
94
95  #undef ATST
96  #undef AROUND1
97  #undef AROUND2
98
99  // ------------------------------------------------------------------
100 // Small helper template functions (only API)
101 // In the below x[limbs] only means that x consists of limbs number of
102 // 64-bit blocks and the operation to be done on the whole x as a vector
103 // or a big value
104 // ------------------------------------------------------------------
105
106 // set x[limbs] to zero
107 template<int limbs>inline void gf2_set_zero(u64 * x)
108 {...}
109
110 // copy destination[limbs] = source[limbs]
111 template<int limbs>inline void gf2_copy(u64 * destination, u64 * source)
112 {...}
113
114 // computes to[limbs] = to[limbs] XOR from[limbs]
115 template<int limbs>inline void gf2_xoradd(u64 * to, u64 * from)
116 {...}
117
118 // shifts the long value x[limbs] to the RIGHT by 0<shift<64 bits
119 template<int limbs, int shift>inline void gf2_shr1const(u64 * x)
120 {...}
121
122 // shifts the long value x[limbs] to the LEFT by 0<shift<64 bits
123 template<int limbs, int shift>inline void gf2_shl1const(u64 * x)
124 {...}
```

C Table of Binary Primitive Polynomials

In the following table we give 3 types of primitive binary polynomials over GF(2). The first Type I corresponds to the first existing primitive polynomial in lexicographical order of the form $f(z) = z^d + z^a + 1$. Type II polynomials have the form $f(z) = z^d + z^a + z^b + z^c + 1$.

Type III polynomials are *optimal primitives* for which we have an optimal reduction algorithm. For each an optimal primitive polynomial we provide the constant parameters of `<algorithm, data>` as the configuration of the reduction function given in Appendix B. Also, in the column with "Performance" we give: (a) our own heuristic counts of 'penalty' for the corresponding reduction algorithm; (b) the measured number of CPU cycles

for the corresponding reduction with the respective optimal primitive; (c) the number of CPU cycles for a similar reduction algorithm but performed by an optimized Montgomery's `redc()` function, for comparison purposes (conversions to/from the Montgomery representation are not included into the measurements).

Simulations were carried out on Intel Core i5-4300U CPU @ 1.90/2.50GHz.

Degree	Type I	Type II	Optimal Primitives. Type III			Performance		
d	a	a, b, c	Algorithm	Data	a or a, b, c	penalty approx.	Opt. cycles	Mont. cycles
2	1	-	a000088a	0301010000000000	1	41	22	48
3	1	-	a000088b	0701010000000000	1	41	22	47
4	1	-	a000088c	0f01010000000000	1	41	23	48
5	2	1,2,3	a000090d	1f01010000000000	2	41	23	49
6	1	1,3,4	a000088e	3f01010000000000	1	41	22	48
7	1	1,2,3	a000088f	7f01010000000000	1	41	22	48
8	-	2,3,4	02244900	0001010000000001	2,3,4	57	26	50
9	4	1,3,4	a0000a09	0101010000000001	4	41	23	48
10	3	1,3,4	a000098a	0301010000000001	3	41	23	51
11	2	1,2,4	a000090b	0701010000000001	2	41	23	49
12	-	1,4,6	62a44600	0f01010000000001	4,1,6	69	24	50
13	-	1,3,4	2224888d	1f01010000000001	1,3,4	69	24	49
14	-	1,3,5	2244888e	3f01010000000001	1,3,5	69	25	51
15	1	1,2,4	a000088f	7f01010000000001	1	41	23	49
16	-	2,3,5	02454000	0001010000000102	8,5,7	51	26	50
17	3	1,2,3	a0000989	0101010000000002	3	41	23	49
18	7	1,2,5	a0000b8a	0301010000000002	7	41	23	51
19	-	1,2,5	22a4400b	0701010000010002	8,1,6	63	23	50
20	3	1,4,6	a000098c	0f01010000000002	3	41	23	51
21	2	1,2,5	a000090d	1f01010000000002	2	41	23	49
22	1	3,4,5	a000088e	3f01010000000002	1	41	23	50
23	5	1,3,5	a0000a8f	7f01010000000002	5	41	23	48
24	-	1,3,4	02648000	0001010000000103	8,2,5	51	26	51
25	3	1,2,3	a0000989	0101010000000003	3	41	23	49
26	-	1,2,6	2004c00a	0301010001010003	8,11,3	57	25	52
27	-	1,2,5	62a00580	0701010001000003	3,8,5	63	24	49
28	3	1,4,6	a000098c	0f01010000000003	3	41	23	51
29	2	1,2,4	a000090d	1f01010000000003	2	41	23	49
30	-	1,4,6	2264400e	3f01010000010003	8,1,4	63	25	51
31	3	1,2,3	a000098f	7f01010000000003	3	41	23	49
32	-	2,6,7	02648000	0001010000000104	8,2,5	51	26	48
33	13	1,4,6	a0000a89	0101010000010004	13	41	23	49
34	-	3,4,8	2224c00a	0301010000010004	8,3,4	63	25	49
35	2	1,7,8	a000090b	0701010000000004	2	41	23	50
36	11	1,7,8	a000098c	0f01010000010004	11	41	23	50
37	-	1,4,6	2284c00d	1f01010000010004	8,3,7	63	23	49
38	-	1,5,6	62200700	3f01010001000004	6,8,1	63	24	49
39	4	1,4,7	a000000f	7f01010000010004	8	35	22	49
40	-	3,4,5	02454000	0001010000000105	8,5,7	51	26	51
41	3	1,2,3	a0000989	0101010000000005	3	41	23	51
42	-	3,4,7	2005c00a	0301010001020005	16,15,7	57	27	52
43	-	3,4,6	2004400b	0701010001020005	16,9,1	57	26	51
44	-	2,5,6	2005c00c	0f01010001010005	8,15,7	57	27	49
45	-	1,3,4	2260000d	1f01010101020005	16,8,11	57	24	49
46	-	6,7,8	2004400e	3f01010001020005	16,9,1	57	26	48

47	5	1,4,5	a0000a8f	7f01010000000005	5	41	23	51
48	-	4,7,9	02a44000	0001010001000106	8,9,6	51	24	48
49	9	4,5,6	e0000480	0101010000000106	9	41	22	48
50	-	2,3,4	2004c00a	0301010002020006	16,19,3	57	26	49
51	-	1,3,6	2240000b	0701010001020006	16,8,2	57	24	47
52	3	1,3,6	a000098c	0f01010000000006	3	41	23	51
53	-	1,2,6	2005800d	1f01010001010006	8,14,6	57	25	51
54	-	3,6,8	2000088e	3f01010001030006	25,9,1	57	27	50
55	24	1,2,6	a000000f	7f01010000030006	24	35	22	51
56	-	2,4,7	02248000	0001010000000107	8,2,3	51	27	49
57	7	2,3,5	a0000b89	0101010000000007	7	41	23	50
58	19	1,5,6	a000098a	0301010000020007	19	41	23	52
59	-	2,4,7	2004800b	0701010102020007	16,18,10	57	25	50
60	1	2,4,5	a000088c	0f01010000000007	1	41	23	48
61	-	1,2,5	3004802d	1f01010002010007	8,18,2	64	24	48
62	-	3,5,6	300448ae	3f01010001010007	9,10,2	70	25	50
63	1	1,4,5	b00008af	7f01010000000007	1	48	25	50
64	-	1,3,4	0e444000	0001010200000208	16,1,19	61	33	35
65	18	1,3,4	a0000009	0101020000040008	32	42	34	35
66	-	6,8,9	22e0000a	0301020102030008	24,16,15	70	40	36
67	-	1,2,5	22a0000b	0701020001020008	16,8,5	70	40	37
68	9	1,5,7	a000088c	0f01020000010008	9	54	35	35
69	-	2,5,6	22e0000d	1f01020101030008	24,8,15	70	39	36
70	-	1,3,5	22a0000e	3f01020001020008	16,8,5	70	40	36
71	6	1,3,5	a0000b0f	7f01020000000008	6	54	34	40
72	-	3,9,10	0004c900	0001020001010009	10,13,5	64	38	35
73	25	2,3,4	e0000480	0101020000000309	25	48	34	37
74	-	3,4,7	2004c00a	0301020102020009	16,19,11	70	39	35
75	-	1,3,6	2240000b	0701020001030009	24,8,2	70	39	37
76	-	2,4,5	2005c00c	0f01020102020009	16,23,15	70	39	35
77	-	2,5,6	2000090d	1f01020001020009	18,10,2	70	40	37
78	-	1,2,7	2005400e	3f01020001010009	8,13,5	70	39	35
79	9	2,3,4	a000088f	7f01020000010009	9	54	36	36
80	-	2,4,9	02248000	000102010000010a	8,2,11	64	37	34
81	4	2,3,6	a0000009	010102000002000a	16	42	33	35
82	-	4,6,9	2005c00a	030102020101000a	8,15,23	70	39	35
83	-	2,4,7	2005800b	070102000101000a	8,14,6	70	39	35
84	13	1,7,9	a0000a8c	0f0102000001000a	13	54	35	35
85	-	1,2,8	2240000d	1f0102000102000a	16,8,2	70	39	35
86	-	2,5,6	22a0000e	3f0102010102000a	16,8,13	70	39	35
87	13	1,5,7	a0000a8f	7f0102000001000a	13	54	35	35
88	-	8,9,11	02444000	000102010100010b	8,9,11	64	37	35
89	38	3,5,6	a0000b09	010102000004000b	38	54	35	35
90	-	2,3,5	2004c00a	030102000101000b	8,11,3	70	39	36
91	-	1,5,8	22a0000b	070102030102000b	16,8,29	70	41	36
92	-	2,5,6	2004c00c	0f0102000101000b	8,11,3	70	39	35
93	2	2,3,6	a000090d	1f0102000000000b	2	54	34	38
94	21	1,5,6	a0000a8e	3f0102000002000b	21	54	35	34
95	11	1,5,7	a000098f	7f0102000001000b	11	54	35	35
96	-	6,9,10	02444000	000102000000020c	16,1,3	64	37	37
97	6	2,4,6	e0000480	010102000000040c	33	48	34	37
98	11	1,7,8	a000098a	030102000001000c	11	54	34	34
99	-	4,5,7	2004800b	070102000203000c	24,18,2	70	41	36
100	37	2,7,8	a0000a8c	0f0102000004000c	37	54	34	35
101	-	1,6,7	20000b0d	1f0102000102000c	22,14,6	70	40	37
102	-	3,5,6	2004400e	3f0102000102000c	16,9,1	70	42	36
103	9	1,4,9	e0000780	7f0102000000030c	31	48	34	37
104	-	1,10,11	00044b00	000102000101000d	14,15,7	64	36	34
105	16	2,5,7	a0000009	010102000002000d	16	42	33	34
106	15	1,5,6	a0000b8a	030102000001000d	15	54	35	35
107	-	4,7,9	2004800b	070102000103000d	24,10,2	70	40	36

108	31	5,11,12	a0000b8c	0f0102000003000d	31	54	34	34
109	-	2,4,5	2004800d	1f0102000201000d	8,18,2	70	42	35
110	-	1,4,6	2005400e	3f0102030201000d	8,21,29	70	42	34
111	10	2,4,7	a000090f	7f0102000001000d	10	54	34	34
112	-	4,6,11	02254000	000102000100010e	8,13,6	64	38	35
113	9	2,3,5	e0000480	010102000000010e	9	48	34	36
114	-	1,2,11	2005c00a	030102020101000e	8,15,23	70	40	35
115	-	5,7,8	2004800b	070102000201000e	8,18,2	70	42	35
116	-	2,5,6	2000088c	0f0102000102000e	17,9,1	70	41	36
117	-	1,2,5	2004800d	1f0102000102000e	16,10,2	70	42	36
118	33	2,5,6	a000088e	3f0102000004000e	33	54	35	34
119	8	4,7,9	a000000f	7f0102000001000e	8	42	33	33
120	-	2,6,9	02454000	000102020200010f	8,21,23	64	37	36
121	18	1,5,8	a0000909	010102000002000f	18	54	35	35
122	-	1,2,6	2005c00a	030102000102000f	16,15,7	70	41	36
123	2	1,4,8	a000090b	070102000000000f	2	54	34	37
124	37	5,6,7	a0000a8c	0f0102000004000f	37	54	34	35
125	-	5,6,7	3004802d	1f0102010202000f	16,18,10	71	37	35
126	-	2,4,7	3004402e	3f0102000403000f	24,33,1	71	39	35
127	1	1,3,7	f00017c0	7f0102000000000f	7	53	34	39
128	-	1,2,7	0ec00880	0001020001020010	17,9,7	68	40	59
129	5	1,4,5	a0000a89	0101030000000010	5	65	39	59
130	3	1,2,5	a000098a	0301030000000010	3	65	36	59
131	-	2,3,8	22a0000b	0701030102030010	24,16,13	89	45	58
132	29	2,5,9	a0000a8c	0f01030000030010	29	65	34	59
133	-	2,8,9	2240000d	1f01030001020010	16,8,2	89	45	57
134	57	1,5,7	a000088e	3f01030000070010	57	65	35	59
135	11	3,4,6	a000000f	7f01030000020010	16	53	31	58
136	-	2,3,8	02248000	0001030000000111	8,2,3	83	44	59
137	21	1,4,11	e0000480	0101030000000711	57	59	34	59
138	-	1,7,8	22e0000a	0301030302030011	24,16,31	89	47	58
139	-	3,5,8	22c0000b	0701030001020011	16,8,6	89	45	58
140	29	1,4,8	a0000a8c	0f01030000030011	29	65	36	55
141	-	1,6,13	2005c00d	1f01030001020011	16,15,7	89	47	58
142	21	1,3,10	a0000a8e	3f01030000020011	21	65	35	56
143	-	2,3,5	2000000f	7f01030104060011	48,32,8	77	39	58
144	-	2,4,7	02c44000	0001030102000112	8,17,15	83	44	58
145	52	1,5,6	a0000a09	0101030000060012	52	65	37	58
146	-	2,3,5	2005c00a	0301030002020012	16,23,7	89	48	59
147	-	2,4,11	2005800b	0701030201010012	8,14,22	89	48	59
148	27	3,5,7	a000098c	0f01030000030012	27	65	37	58
149	-	7,9,10	22c0000d	1f01030101030012	24,8,14	89	47	59
150	53	2,3,8	a0000a8e	3f01030000060012	53	65	36	56
151	3	1,2,3	e0000780	7f01030000000112	15	59	32	58
152	-	2,3,6	02444000	0001030100000213	16,1,11	83	44	58
153	1	4,5,8	a0000009	0101030000010013	8	53	31	59
154	-	1,5,9	22e0000a	0301030101020013	16,8,15	89	46	57
155	-	4,5,7	2220000b	0701030101020013	16,8,9	89	47	58
156	-	3,5,9	22a0000c	0f01030102030013	24,16,13	89	47	57
157	-	2,5,6	2260000d	1f01030101020013	16,8,11	89	45	58
158	-	5,6,8	22a0000e	3f01030101020013	16,8,13	89	46	58
159	31	3,6,11	a000000f	7f01030000050013	40	53	30	60
160	-	2,3,5	02444000	0001030000000214	16,1,3	83	44	58
161	18	2,3,6	a0000909	0101030000020014	18	65	37	59
162	-	4,7,8	2260000a	0301030202030014	24,16,19	89	46	59
163	-	3,6,7	2004800b	0701030201010014	8,10,18	89	47	57
164	-	5,6,12	22e0000c	0f01030001030014	24,8,7	89	47	58
165	-	3,8,9	22e0000d	1f01030101030014	24,8,15	89	45	58
166	-	2,3,10	2220000e	3f01030002030014	24,16,1	89	46	57
167	6	2,4,6	a0000b0f	7f01030000000014	6	65	37	59
168	-	6,9,16	02a44000	0001030001000215	16,9,6	83	44	58

169	34	5,6,8	e0000480	0101030000000715	57	59	34	59
170	23	1,4,9	a0000b8a	0301030000020015	23	65	36	58
171	-	2,5,6	2005400b	0701030002030015	24,21,5	89	47	58
172	7	3,7,11	a0000b8c	0f01030000000015	7	65	36	58
173	-	2,5,8	2004800d	1f01030001030015	24,10,2	89	46	57
174	13	5,8,9	a0000a8e	3f01030000010015	13	65	36	58
175	6	2,4,6	a000000f	7f01030000020015	16	53	30	59
176	-	9,11,12	02444000	0001030202000216	16,17,19	83	44	58
177	8	2,3,5	a0000009	0101030000010016	8	53	31	58
178	87	2,7,8	a0000b8a	03020300000a0016	87	78	46	58
179	-	1,2,4	2220000b	0701030203040016	32,24,17	89	46	58
180	-	7,10,12	2000088c	0f01030001030016	25,9,1	89	47	58
181	-	1,6,7	2005800d	1f01030002020016	16,22,6	89	47	58
182	-	1,6,8	2005400e	3f01030201010016	8,13,21	89	47	58
183	56	4,7,8	a000000f	7f01030000070016	56	53	30	58
184	-	7,8,9	02c44000	0001030001000117	8,9,7	83	45	58
185	24	1,3,8	a0000009	0101030000030017	24	53	30	58
186	-	6,8,9	20000a8a	0301030001030017	29,13,5	89	46	58
187	-	5,6,7	22c0000b	0701030001020017	16,8,6	89	46	58
188	-	2,5,6	22a0000c	0f01030002030017	24,16,5	89	46	58
189	-	2,5,6	3240002d	1f01030503040017	32,24,42	96	49	58
190	-	2,6,13	3220002e	3f01030401020017	16,8,33	96	49	57
191	9	4,6,7	b00008af	7f01030000010017	9	72	39	59
192	-	5,11,15	0ec44000	0001030202000118	8,17,23	93	44	62
193	15	4,7,9	e0000480	0102040000000918	73	61	36	62
194	87	2,3,4	a0000b8a	03020400000a0018	87	79	45	62
195	-	2,3,8	2004800b	0701040002030018	24,18,2	82	41	62
196	-	2,9,11	2005400c	0f01040201010018	8,13,21	82	41	62
197	-	2,4,9	2240000d	1f01040102030018	24,16,10	82	41	61
198	65	5,8,15	a000088e	3f02040000080018	65	79	45	61
199	34	1,4,9	2000000f	7f01040203050018	40,24,16	64	35	62
200	-	2,3,5	0004c900	0001040001020019	18,13,5	76	38	62
201	14	2,3,6	e0000480	0101040000000219	17	54	28	56
202	55	4,6,7	a0000b8a	0301040000060019	55	66	37	62
203	-	1,7,8	22a0000b	0701040001030019	24,8,5	82	41	61
204	-	3,4,10	22a0000c	0f01040202030019	24,16,21	82	41	62
205	-	2,5,9	2005800d	1f01040001020019	16,14,6	82	42	62
206	-	5,9,10	2220000e	3f01040002030019	24,16,1	82	41	63
207	43	1,6,9	2000000f	7f01040205060019	48,40,16	64	34	62
208	-	1,3,9	02400880	000104000001001a	9,1,3	76	39	61
209	6	2,3,5	a0000009	0101040000001001a	8	48	29	61
210	-	3,4,12	2005c00a	0301040102030001a	24,23,15	82	41	63
211	-	8,10,11	2220000b	070104010102001a	16,8,9	82	42	62
212	105	3,4,7	a000088c	0f020400000d001a	105	79	44	62
213	-	2,5,6	2005c00d	1f0104000102001a	16,15,7	82	41	62
214	-	1,3,5	2004400e	3f0104020303001a	24,25,17	82	41	63
215	23	3,5,6	e0000780	7f0104000000021a	23	54	30	56
216	-	1,3,7	02444000	000104020200021b	16,17,19	76	38	62
217	45	4,5,6	a0000009	010104000008001b	64	48	30	61
218	11	1,7,8	a000098a	030104000001001b	11	66	36	61
219	-	1,4,8	2005800b	070104000103001b	24,14,6	82	41	62
220	-	9,10,12	2005c00c	0f0104000102001b	16,15,7	82	41	62
221	-	2,6,8	2005800d	1f0104000101001b	8,14,6	82	42	62
222	-	2,5,8	2004400e	3f0104010204001b	32,17,9	82	41	62
223	33	2,4,5	a000000f	7f0204000008001b	64	55	32	62
224	-	2,7,12	02200880	000104000102001c	17,9,2	76	39	62
225	32	1,5,10	a0000009	010104000004001c	32	48	28	58
226	-	3,7,10	22e0000a	030104030203001c	24,16,31	82	41	62
227	-	4,9,10	2004800b	070104030202001c	16,18,26	82	41	62
228	-	2,11,12	22a0000c	0f0104010102001c	16,8,13	82	44	62
229	-	1,4,10	22c0000d	1f0104000102001c	16,8,6	82	41	62

230	-	6,7,8	2000088e	3f0104000203001c	25,17,1	82	41	62
231	26	2,4,7	2000000f	7f0104030607001c	56,48,24	64	35	61
232	-	4,9,11	02a44000	000104000200011d	8,17,6	76	39	61
233	74	1,4,9	20000009	010104010206001d	48,16,8	64	38	61
234	31	4,7,11	a0000b8a	030104000003001d	31	66	36	61
235	-	1,6,9	22c0000b	070104000203001d	24,16,6	82	41	63
236	5	7,8,10	a0000a8c	0f0104000000001d	5	66	36	62
237	-	1,4,7	2004800d	1f0104000102001d	16,10,2	82	42	61
238	-	1,2,5	2004400e	3f0104020303001d	24,25,17	82	42	62
239	36	1,7,12	2000000f	7f0104020507001d	56,40,16	64	34	62
240	-	3,5,8	0244c000	000104000000011e	8,3,5	76	41	61
241	70	4,8,9	20000009	010104030407001e	56,32,24	64	35	62
242	-	1,6,11	2005c00a	030104010203001e	24,23,15	82	45	62
243	-	1,5,8	2005800b	070104000202001e	16,22,6	82	41	61
244	-	1,4,9	20000a8c	0f0104000102001e	21,13,5	82	43	61
245	-	1,4,6	2005800d	1f0104030101001e	8,14,30	82	42	61
246	-	1,2,11	20000a8e	3f0104000304001e	37,29,5	82	41	62
247	82	2,4,9	60000780	7f0104030400031e	31,32,24	70	36	62
248	-	10,14,15	02454000	000104010100021f	16,13,15	76	39	62
249	86	1,4,7	20000009	010104030406001f	48,32,24	64	34	61
250	103	3,5,10	a0000b8a	03020400000c001f	103	79	44	64
251	-	2,4,7	2220000b	070104000304001f	32,24,1	82	41	61
252	67	1,5,11	a000098c	0f0204000008001f	67	79	44	61
253	-	2,3,7	3000092d	1f0104000102001f	18,10,2	83	43	63
254	-	1,2,7	3004402e	3f0104020306001f	48,25,17	83	42	62
255	52	2,3,5	b000002f	7f0104000007001f	56	49	27	61
256	-	2,5,10	0e444000	0001040000000220	16,1,3	80	42	70
257	12	2,6,7	a0000009	0101050000060020	48	59	25	74
258	83	4,6,9	a000098a	03020500000a0020	83	90	50	73
259	-	2,6,10	2005800b	0701050002030020	24,22,6	101	51	71
260	-	7,8,10	2005400c	0f01050002030020	24,21,5	101	53	73
261	-	4,6,7	2005c00d	1f01050001030020	24,15,7	101	52	73
262	-	4,8,9	2220000e	3f01050301030020	24,8,25	101	54	73
263	93	2,5,11	2000000f	7f01050104060020	48,32,8	83	47	74
264	-	1,9,10	02200880	0001050100010021	9,1,10	95	46	73
265	42	2,3,5	a0000909	0101050000050021	42	77	39	74
266	47	1,6,7	a0000b8a	0301050000050021	47	77	38	73
267	-	3,6,8	22c0000b	0701050101030021	24,8,14	101	53	73
268	25	1,4,10	a000088c	0f01050000030021	25	77	40	73
269	-	1,6,7	2000090d	1f01050001020021	18,10,2	101	53	73
270	53	3,7,10	a0000a8e	3f01050000060021	53	77	39	73
271	58	6,7,11	a000090f	7f01050000070021	58	77	39	71
272	-	2,6,9	00058880	0001050001020022	17,15,7	95	45	72
273	23	1,2,7	a0000009	01020500000b0022	88	66	37	73
274	67	2,7,9	a000098a	0302050000080022	67	90	51	73
275	-	9,10,11	2004400b	0701050001020022	16,9,1	101	54	74
276	-	1,3,6	2004400c	0f01050102040022	32,17,9	101	53	72
277	-	3,6,12	2004800d	1f01050002010022	8,18,2	101	54	73
278	5	1,4,5	a0000a8e	3f01050000000022	5	77	38	73
279	5	1,4,5	a000000f	7f01050000050022	40	59	24	74
280	-	2,5,9	02a44000	0001050103000123	8,25,14	95	45	72
281	93	1,4,9	60000480	0101050103000323	25,24,8	89	39	73
282	35	4,5,10	a000098a	0301050000040023	35	77	39	73
283	-	5,7,12	2004800b	0701050201010023	8,10,18	101	59	74
284	119	5,6,8	a0000b8c	0f020500000e0023	119	90	57	73
285	-	5,7,10	22e0000d	1f01050001020023	16,8,7	101	74	74
286	69	1,10,15	a0000a8e	3f02050000080023	69	90	76	74
287	71	2,5,6	e0000780	7f02050000000823	71	72	83	83
288	-	1,10,11	02248000	0001050400000224	16,2,35	95	57	79
289	21	3,4,12	a0000a89	0101050000020024	21	77	65	92
290	-	2,3,5	2004c00a	0301050002020024	16,19,3	101	53	73

291	-	5,11,12	2005800b	0701050001010024	8,14,6	101	53	73
292	97	1,3,7	a000088c	0f020500000c0024	97	90	50	72
293	-	1,6,11	22c0000d	1f01050102030024	24,16,14	101	53	73
294	61	2,3,9	a0000a8e	3f02050000070024	61	90	50	73
295	48	2,4,5	a000000f	7f01050000060024	48	59	24	73
296	-	4,9,11	02c44000	0001050100000125	8,1,15	95	46	73
297	5	1,4,5	a0000a89	0101050000000025	5	77	38	74
298	-	4,8,11	2000088a	0301050001030025	25,9,1	101	54	72
299	-	4,6,11	2220000b	0701050302040025	32,16,25	101	53	73
300	7	10,12,13	a0000b8c	0f01050000000025	7	77	39	74
301	-	2,5,9	2005800d	1f01050002010025	8,22,6	101	54	72
302	41	5,9,12	a000088e	3f01050000050025	41	77	39	73
303	-	6,12,13	6003c000	7f01050200000325	31,7,16	89	40	74
304	-	1,2,11	02a48000	0001050101000226	16,10,15	95	46	73
305	102	2,6,7	60024000	0101050300010226	17,9,24	89	41	73
306	-	1,3,7	2260000a	0301050302030026	24,16,27	101	54	73
307	-	2,4,8	22a0000b	0701050001030026	24,8,5	101	55	74
308	-	2,9,15	2260000c	0f01050003040026	32,24,3	101	53	72
309	-	4,6,10	2005800d	1f01050103030026	24,30,14	101	54	72
310	-	1,5,8	22a0000e	3f01050101040026	32,8,13	101	53	73
311	-	3,5,7	2000000f	7f01050405060026	48,40,32	83	42	75
312	-	5,10,11	00054900	0001050002010027	10,23,7	95	46	72
313	79	1,3,7	e0000480	0102050000000f27	121	72	37	74
314	15	3,9,14	a0000b8a	0301050000010027	15	77	38	74
315	-	1,9,10	22c0000b	0701050201020027	16,8,22	101	53	72
316	135	7,11,12	22a0000c	0f01050102030027	24,16,13	101	57	72
317	-	2,4,7	3004802d	1f01050002010027	8,18,2	108	50	73
318	-	5,6,8	3004402e	3f01050102040027	32,17,9	108	50	73
319	36	1,2,11	b0003a2f	7f01050000040027	36	94	44	74
320	-	1,3,4	0ea44000	0001050003000228	16,25,6	105	59	151
321	31	2,5,7	a0000009	0101060000070028	56	66	40	150
322	67	1,2,17	a000098a	0302060000080028	67	103	67	152
323	-	1,3,10	2004800b	0701060102020028	16,18,10	114	68	152
324	-	3,4,6	2004c00c	0f01060001010028	8,11,3	114	68	151
325	-	2,5,10	22c0000d	1f01060101020028	16,8,14	114	68	152
326	-	1,3,10	2004400e	3f01060001020028	16,9,1	114	68	151
327	34	2,5,8	a000000f	7f03060000130028	152	84	48	151
328	-	5,7,9	02444000	0001060200000229	16,1,19	108	60	152
329	50	3,6,8	a0000909	0101060000060029	50	90	52	152
330	-	2,7,8	2004c00a	0301060003010029	8,27,3	114	68	152
331	-	2,6,10	2004400b	0701060001020029	16,9,1	114	68	152
332	123	7,11,12	a000098c	0f020600000f0029	123	103	66	151
333	2	2,4,8	a000090d	1f01060000000029	2	90	52	151
334	-	1,4,7	2005400e	3f01060002020029	16,21,5	114	68	152
335	-	2,7,10	2000000f	7f01060103040029	32,24,8	90	49	152
336	-	1,4,7	02c44000	000106000000022a	16,1,7	108	60	151
337	55	1,6,10	e0000480	010106000000072a	57	72	40	152
338	-	2,3,6	22e0000a	030106000102002a	16,8,7	114	69	152
339	-	7,10,16	22a0000b	070106010102002a	16,8,13	114	68	152
340	-	3,4,11	2004400c	0f0106000103002a	24,9,1	114	68	152
341	-	5,11,14	22e0000d	1f0106000102002a	16,8,7	114	68	152
342	125	1,2,11	a0000a8e	3f030600000f002a	125	114	76	152
343	75	5,8,10	e0000780	7f03060000000102a	135	90	50	153
344	-	6,10,11	0004c900	0001060302010002b	10,21,29	108	60	152
345	22	2,4,8	a0000b09	010106000002002b	22	90	52	152
346	-	2,7,11	2260000a	030106000102002b	16,8,3	114	69	151
347	-	3,10,11	2220000b	070106030102002b	16,8,25	114	69	151
348	-	4,7,8	2004400c	0f0106010203002b	24,17,9	114	69	151
349	-	2,5,6	2260000d	1f0106010103002b	24,8,11	114	70	151
350	53	10,13,14	a0000a8e	3f0106000006002b	53	90	54	152
351	34	3,6,8	e0000780	7f0106000000062b	55	72	41	152

352	-	6,11,13	02400980	000106000102002c	19,11,5	108	61	152
353	69	4,7,9	e0000480	010306000000132c	153	90	50	151
354	-	5,13,14	2004c00a	030106000202002c	16,19,3	114	70	153
355	-	1,5,6	22a0000b	070106040104002c	32,8,37	114	69	151
356	-	7,9,10	2005400c	0f0106000301002c	8,29,5	114	68	152
357	-	2,10,11	2260000d	1f0106030103002c	24,8,27	114	69	152
358	-	7,8,14	2220000e	3f0106030103002c	24,8,25	114	68	152
359	68	1,7,9	60000780	7f0106040300002c	7,24,32	96	53	150
360	-	1,25,26	02200880	000106030003002d	25,1,26	108	62	152
361	-	1,4,7	60024000	010106020000012d	9,1,16	96	52	151
362	63	7,8,18	a0000b8a	030206000007002d	63	103	67	151
363	-	3,5,8	2005800b	070106010402002d	16,38,14	114	68	151
364	67	1,5,12	a000098c	0f0206000008002d	67	103	67	152
365	-	5,6,9	2004800d	1f0106010202002d	16,18,10	114	69	151
366	29	4,7,14	a0000a8e	3f0106000003002d	29	90	51	152
367	21	2,4,9	a0000a8f	7f0106000002002d	21	90	53	152
368	-	7,9,17	02c00880	000106000102002e	17,9,7	108	62	151
369	91	2,10,11	60024000	010106030001032e	25,9,24	96	53	152
370	139	2,3,5	a000098a	030306000011002e	139	114	75	152
371	-	2,3,8	2005800b	070106000101002e	8,14,6	114	68	152
372	-	3,7,15	2005400c	0f0106000104002e	32,13,5	114	69	152
373	-	2,7,8	2000090d	1f0106000103002e	26,10,2	114	69	152
374	-	5,6,8	22a0000e	3f0106030206002e	48,16,29	114	68	152
375	16	1,7,8	a000000f	7f0106000002002e	16	66	39	152
376	-	5,7,8	02454000	000106000000012f	8,5,7	108	61	152
377	41	1,3,8	e0000480	010106000000052f	41	72	41	152
378	43	4,13,15	a000098a	030106000005002f	43	90	53	152
379	-	5,8,10	2004800b	070106000305002f	40,26,2	114	68	151
380	47	3,6,14	a0003b8c	0f0106000005002f	47	94	55	152
381	-	1,2,5	3260002d	1f0106020204002f	32,16,19	115	74	152
382	81	3,7,18	b00008ae	3f020600000a002f	81	104	72	152
383	90	1,5,9	f00017c0	7f0306000000102f	135	95	49	151
384	-	6,15,16	0ec44000	0001060303000230	16,25,31	112	126	289
385	6	2,4,6	a0000009	0101070000030030	24	77	101	286
386	83	5,6,10	a000098a	03020700000a0030	83	114	139	290
387	-	2,8,9	22a0000b	0701070202050030	40,16,21	133	148	288
388	-	1,3,14	2004400c	0f01070104030030	24,33,9	133	150	288
389	-	5,9,10	2240000d	1f01070001030030	24,8,2	133	151	291
390	89	2,10,13	a000088e	3f020700000b0030	89	114	144	290
391	28	1,2,6	e0000780	7f01070000000330	31	83	104	288
392	-	6,10,13	0244c000	0001070201000331	24,11,21	127	137	290
393	7	1,2,7	a0000b89	0101070000000031	7	101	108	292
394	135	2,7,8	a0000b8a	0303070000100031	135	125	146	289
395	-	5,6,11	20000b0b	0701070001030031	30,14,6	133	148	289
396	25	4,6,7	a000088c	0f01070000030031	25	101	112	291
397	-	6,7,12	2240000d	1f01070102030031	24,16,10	133	146	289
398	-	5,6,14	2220000e	3f01070101020031	16,8,9	133	150	289
399	86	2,9,11	a0000b0f	7f020700000a0031	86	114	144	289
400	-	2,3,5	00044a80	0001070102020032	21,22,14	127	135	290
401	152	5,8,12	a0000009	0103070000130032	152	95	107	289
402	-	3,4,9	2004c00a	0301070403030032	24,27,35	133	144	289
403	-	5,8,9	2005400b	0701070102050032	40,21,13	133	148	290
404	189	4,6,7	a0000a8c	0f04070000170032	189	126	159	293
405	-	7,8,17	2004800d	1f01070002030032	24,18,2	133	143	292
406	157	4,9,13	a0000a8e	3f03070000130032	157	125	153	289
407	71	5,7,9	e0000780	7f02070000000832	71	90	109	290
408	-	1,5,7	02254000	0001070002000233	16,21,6	127	137	288
409	87	3,5,7	a0000b89	01020700000a0033	87	114	145	289
410	-	3,4,10	2000088a	0301070001030033	25,9,1	133	145	292
411	-	3,10,12	2240000b	0701070202030033	24,16,18	133	151	290
412	147	3,8,11	a000098c	0f03070000120033	147	125	152	289

413	-	6,7,10	2005800d	1f01070002040033	32,22,6	133	148	290
414	-	9,13,16	22a0000e	3f01070301020033	16,8,29	133	149	289
415	102	2,4,9	2000000f	7f01070103040033	32,24,8	109	111	288
416	-	2,5,9	00044880	0001070001030034	25,10,2	127	138	290
417	107	1,3,10	e0000480	0102070000000e34	113	90	107	291
418	-	1,3,15	2260000a	0301070102030034	24,16,11	133	146	291
419	-	4,5,15	22a0000b	0701070002030034	24,16,5	133	145	289
420	-	8,10,13	2220000c	0f01070001040034	32,8,1	133	150	291
421	-	2,4,5	2260000d	1f01070101030034	24,8,11	133	147	289
422	149	1,6,10	a0000a8e	3f03070000120034	149	125	147	292
423	25	3,5,9	a000088f	7f01070000030034	25	101	110	288
424	-	2,7,9	02c44000	0001070102000135	8,17,15	127	134	288
425	12	3,4,7	a0000a09	0101070000010035	12	101	109	290
426	-	11,12,14	22e0000a	0301070201030035	24,8,23	133	150	291
427	-	5,6,11	22a0000b	0701070302030035	24,16,29	133	151	292
428	105	2,3,11	a000088c	0f020700000d0035	105	114	146	291
429	-	7,8,10	2004c00d	1f01070206050035	40,51,19	133	146	291
430	-	11,13,15	22a0000e	3f01070002040035	32,16,5	133	144	290
431	120	1,3,5	a000000f	7f020700000f0035	120	84	108	289
432	-	3,4,13	02454000	0001070002000236	16,21,7	127	139	286
433	33	3,5,11	e0000480	0101070000000436	33	83	104	291
434	-	5,11,12	2004c00a	0301070402020036	16,19,35	133	147	289
435	-	5,9,12	2240000b	0701070402040036	32,16,34	133	147	291
436	165	4,5,6	a0000a8c	0f03070000140036	165	125	153	289
437	-	1,2,6	22e0000d	1f01070201040036	32,8,23	133	148	289
438	65	2,6,17	a000088e	3f02070000080036	65	114	136	288
439	49	2,3,8	a000088f	7f01070000060036	49	101	110	288
440	-	1,3,4	02248000	0001070000000137	8,2,3	127	138	292
441	31	1,8,11	a0000b89	0101070000030037	31	101	114	289
442	-	2,5,7	2260000a	0301070101020037	16,8,11	133	146	290
443	-	1,6,10	2220000b	0701070101020037	16,8,9	133	150	289
444	-	9,12,13	2004c00c	0f01070004040037	32,35,3	133	146	290
445	-	4,6,7	3260002d	1f01070504060037	48,32,43	140	164	290
446	105	4,7,15	b00008ae	3f020700000d0037	105	121	157	288
447	73	1,6,9	b00008af	7f02070000090037	73	121	156	290
448	-	4,6,11	0e244000	0001070004000138	8,33,2	137	75	230
449	134	4,6,11	61200000	0101080001040538	41,33,9	101	63	230
450	79	7,12,16	a0000b8a	0302080000090038	79	114	71	227
451	-	1,10,16	2005800b	0701080005050038	40,46,6	125	85	227
452	-	4,5,6	2005400c	0f01080403020038	16,29,37	125	77	218
453	-	4,6,15	2005c00d	1f01080102030038	24,23,15	125	78	225
454	-	5,9,10	2220000e	3f01080001030038	24,8,1	125	81	227
455	38	2,6,11	a0000b0f	7f01080000040038	38	101	63	228
456	-	2,11,23	02400880	0001080100030039	25,1,11	119	70	227
457	16	3,8,11	a0000009	0101080000020039	16	71	54	226
458	203	5,10,13	2260000a	0301080001020039	16,8,3	125	82	225
459	-	2,5,12	22a0000b	0701080301020039	16,8,29	125	81	228
460	61	1,5,9	a0000a8c	0f02080000070039	61	114	73	228
461	-	1,6,7	2260000d	1f01080102030039	24,16,11	125	83	227
462	73	5,11,12	a000088e	3f02080000090039	73	114	72	228
463	93	7,8,11	a000000f	7f03080000150039	168	89	57	228
464	-	4,9,23	02c44000	000108030100043a	32,9,31	119	68	231
465	59	2,3,8	a0000989	0101080000070003a	59	101	62	226
466	-	6,11,14	2004c00a	030108030704003a	32,59,27	125	77	227
467	-	1,6,11	2004800b	070108030202003a	16,18,26	125	79	229
468	-	4,9,15	2005400c	0f0108030405003a	40,37,29	125	77	224
469	-	2,5,9	22e0000d	1f0108010104003a	32,8,15	125	80	227
470	149	2,8,9	a0000a8e	3f0308000012003a	149	125	79	227
471	1	2,3,6	e0000780	7f02080000000e3a	119	84	60	228
472	-	2,3,11	00044900	000108010000003b	2,3,11	119	69	227
473	-	3,6,8	20000009	010108060708003b	64,56,48	95	66	226

474	191	9,11,18	2004c00a	030108010204003b	32,19,11	125	79	226
475	-	4,8,9	2004800b	070108030101003b	8,10,26	125	78	226
476	15	1,8,10	a0000b8c	0f0108000001003b	15	101	65	226
477	-	7,15,16	2005c00d	1f0108000102003b	16,15,7	125	80	230
478	121	1,4,6	a000088e	3f020800000f003b	121	114	75	228
479	104	4,7,9	a000000f	7f020800000d003b	104	78	60	229
480	-	7,13,16	02454000	000108000100023c	16,13,7	119	70	229
481	138	1,9,10	e0000480	010408000000193c	201	90	60	229
482	-	5,6,9	2260000a	030108040203003c	24,16,35	125	81	226
483	-	4,6,9	2240000b	070108010105003c	40,8,10	125	78	228
484	105	1,2,14	a000088c	0f020800000d003c	105	114	78	227
485	-	6,16,17	2004800d	1f0108010205003c	40,18,10	125	79	226
486	-	5,8,14	22a0000e	3f0108040104003c	32,8,37	125	82	228
487	94	2,4,9	e0000780	7f03080000000f3c	127	95	59	227
488	-	1,3,4	02444000	000108000000013d	8,1,3	119	70	227
489	83	5,6,9	20000009	010108010607003d	56,48,8	95	69	227
490	219	5,7,9	2260000a	030108000102003d	16,8,3	125	81	227
491	-	3,6,11	2005400b	070108010204003d	32,21,13	125	78	226
492	-	1,7,8	2005400c	0f0108000104003d	32,13,5	125	80	226
493	-	3,5,10	2004800d	1f0108000201003d	8,18,2	125	81	227
494	137	1,5,13	a000088e	3f0308000011003d	137	125	89	224
495	76	1,9,15	60000780	7f0108010200033d	31,16,8	101	62	224
496	-	2,5,16	02648000	000108000000023e	16,2,5	119	68	228
497	78	4,9,11	a0000009	01040800001b003e	216	84	57	228
498	-	3,9,11	2004c00a	030108010202003e	16,19,11	125	87	224
499	-	5,6,11	2240000b	070108010304003e	32,24,10	125	81	230
500	-	1,6,10	2220000c	0f0108010103003e	24,8,9	125	81	234
501	-	2,4,5	2005c00d	1f0108000204003e	32,23,7	125	87	227
502	-	4,5,8	2005400e	3f0108000101003e	8,13,5	125	81	228
503	3	1,2,3	a000000f	7f040800001f003e	248	84	60	226
504	-	2,14,21	02244000	000108010300023f	16,25,10	119	73	227
505	156	5,8,12	20000009	010108030507003f	56,40,24	95	71	228
506	95	5,12,15	a0000b8a	03020800000b003f	95	114	75	229
507	-	3,6,13	2240000b	070108040204003f	32,16,34	125	85	226
508	109	3,8,13	a0000a8c	0f020800000d003f	109	114	74	227
509	-	3,7,8	3000092d	1f0108010204003f	34,18,10	126	76	226
510	-	9,10,12	300009ae	3f0108000203003f	27,19,3	126	76	227
511	10	2,8,10	f00017c0	7f0108000000013f	15	82	57	226
512	-	2,5,8	0e244000	0001080004000140	8,33,2	123	174	488
513	85	8,10,13	60024000	0101090200000140	9,1,16	120	127	490
514	-	3,5,7	2005c00a	0301090001020040	16,15,7	144	175	486
515	-	4,7,14	2240000b	0701090002030040	24,16,2	144	178	484
516	-	2,5,7	2260000c	0f01090402050040	40,16,35	144	173	485
517	-	2,10,12	2004800d	1f01090302020040	16,18,26	144	169	484
518	33	2,3,11	a000088e	3f01090000040040	33	112	155	483
519	79	2,8,12	e0000780	7f02090000000940	79	95	135	485
520	-	11,13,17	02a00900	0001090201020041	18,10,23	138	182	484
521	32	2,7,9	a0000009	0101090000040041	32	82	134	484
522	-	4,13,15	22e0000a	0301090102030041	24,16,15	144	173	484
523	-	2,6,13	22a0000b	0701090002040041	32,16,5	144	173	487
524	167	1,5,9	a0000b8c	0f03090000140041	167	136	165	483
525	-	1,4,6	22e0000d	1f01090002040041	32,16,7	144	173	487
526	-	1,5,9	2220000e	3f01090201050041	40,8,17	144	175	485
527	47	1,7,9	e0000780	7f01090000000541	47	88	139	485
528	-	2,6,11	02254000	0001090002000242	16,21,6	138	193	486
529	42	1,4,7	a0000909	0101090000050042	42	112	155	483
530	-	3,7,10	2260000a	0301090401030042	24,8,35	144	169	486
531	-	2,6,12	2000090b	0701090104050042	42,34,10	144	171	482
532	1	3,4,10	a000088c	0f01090000000042	1	112	149	486
533	-	2,3,4	2004800d	1f01090003040042	32,26,2	144	178	485
534	-	1,5,7	22a0000e	3f01090203040042	32,24,21	144	174	484

535	-	2,6,8	6003c000	7f01090500000142	15,7,40	120	160	492
536	-	3,5,7	02454000	0001090000000343	24,5,7	138	190	483
537	94	1,2,10	60000480	0101090105000343	25,40,8	120	154	485
538	-	1,2,5	2005c00a	0301090102020043	16,23,15	144	174	479
539	-	4,5,10	2220000b	0701090101020043	16,8,9	144	171	483
540	179	3,6,11	a000098c	0f03090000160043	179	136	169	483
541	-	4,10,13	2005800d	1f01090103020043	16,30,14	144	167	484
542	-	2,3,9	2005400e	3f01090003040043	32,29,5	144	173	485
543	16	5,7,11	a000000f	7f01090000020043	16	82	136	480
544	-	6,9,13	02444000	0001090200000244	16,1,19	138	191	484
545	122	6,8,13	20000009	0101090205060044	48,40,16	114	157	483
546	-	1,2,8	2005c00a	0301090504040044	32,39,47	144	168	479
547	-	4,7,13	2240000b	0701090301020044	16,8,26	144	173	475
548	-	3,5,10	2220000c	0f01090101020044	16,8,9	144	173	472
549	-	3,4,16	2005c00d	1f01090403020044	16,31,39	144	169	474
550	193	4,17,21	a000088e	3f04090000180044	193	137	170	473
551	135	1,4,9	a000000f	7f040900001e0044	240	95	137	471
552	-	2,5,20	02248000	0001090400000345	24,2,35	138	187	470
553	39	3,4,11	e0000480	0101090000000745	57	88	134	473
554	-	3,8,11	2004c00a	0301090001010045	8,11,3	144	175	477
555	-	4,9,10	2004800b	0701090002010045	8,18,2	144	175	473
556	153	7,10,16	a000088c	0f03090000130045	153	136	163	478
557	-	5,6,7	2004c00d	1f01090204050045	40,35,19	144	166	480
558	-	5,9,14	2005400e	3f01090104030045	24,37,13	144	168	474
559	34	2,7,9	a000090f	7f01090000040045	34	112	153	475
560	-	6,9,11	00044880	0001090102030046	25,18,10	138	177	474
561	71	4,7,11	20000009	0101090102040046	32,16,8	114	154	476
562	-	2,4,11	2260000a	0301090401050046	40,8,35	144	172	478
563	-	3,7,14	22a0000b	0701090102040046	32,16,13	144	171	474
564	163	1,3,6	a000098c	0f03090000140046	163	136	163	467
565	-	1,6,11	2260000d	1f01090002040046	32,16,3	144	172	473
566	153	2,5,6	a000088e	3f03090000130046	153	136	160	475
567	143	6,9,11	e0000780	7f03090000001146	143	106	143	474
568	-	10,11,17	02600980	0001090001020047	19,11,6	138	182	475
569	77	1,10,12	60024000	0101090400000347	25,1,32	120	155	474
570	67	7,12,18	a000098a	0302090000080047	67	125	158	475
571	-	2,5,10	2000090b	0701090002030047	26,18,2	144	174	476
572	-	1,8,12	2220000c	0f01090301020047	16,8,25	144	171	476
573	-	4,6,10	3260002d	1f01090402040047	32,16,35	151	174	472
574	13	5,9,14	b0003aae	3f01090000010047	13	129	159	472
575	146	3,5,6	3000002f	7f020902090d0047	104,72,16	128	171	475
576	-	3,4,13	0ec44000	0001090201000748	56,9,23	148	101	409
577	25	2,3,8	e0000480	01010a0000000348	25	95	63	406
578	-	16,22,23	2005c00a	03010a0004040048	32,39,7	157	96	399
579	-	7,9,12	22c0000b	07010a0202030048	24,16,22	157	105	405
580	-	1,4,6	2005c00c	0f010a0103040048	32,31,15	157	97	401
581	-	6,7,13	22c0000d	1f010a0301030048	24,8,30	157	103	402
582	85	3,6,11	a0000a8e	3f020a00000a0048	85	138	83	403
583	130	2,6,8	60000780	7f010a0203000348	31,24,16	127	86	400
584	-	3,13,14	00054880	00010a0001020049	17,14,6	151	104	405
585	121	2,3,8	e0000480	01020a0000000f49	121	102	65	404
586	-	2,5,7	2260000a	03010a0401020049	16,8,35	157	94	403
587	-	1,6,11	2004800b	07010a0104040049	32,34,10	157	103	403
588	151	11,16,17	a0000b8c	0f030a0000120049	151	149	93	402
589	-	3,4,10	2005c00d	1f010a0102020049	16,23,15	157	94	403
590	93	2,3,12	a0000a8e	3f020a00000b0049	93	138	93	401
591	-	4,6,9	6003c000	7f010a0100020649	55,23,8	127	84	403
592	-	1,19,24	02444000	00010a020000034a	24,1,19	151	104	399
593	86	5,8,9	e0000480	01030a000000164a	177	113	66	404
594	19	8,10,11	a000098a	03010a000002004a	19	125	80	399
595	-	1,2,9	2000090b	07010a010203004a	26,18,10	157	98	406

596	-	4,5,6	2260000c	0f010a000205004a	40,16,3	157	100	402
597	-	9,12,14	2260000d	1f010a030203004a	24,16,27	157	98	403
598	-	1,6,7	2005400e	3f010a030202004a	16,21,29	157	100	403
599	30	6,8,9	a0000b0f	7f010a000003004a	30	125	85	406
600	-	1,10,11	02600980	00010a010001004b	11,3,14	151	106	401
601	201	1,4,12	e0000480	01040a000000194b	201	108	68	405
602	-	6,8,11	20000a8a	03010a000102004b	21,13,5	157	100	402
603	-	3,4,6	22a0000b	07010a050104004b	32,8,45	157	100	402
604	-	4,6,15	2000088c	0f010a000103004b	25,9,1	157	106	403
605	-	5,7,10	2005800d	1f010a010202004b	16,22,14	157	100	406
606	-	4,7,15	2220000e	3f010a020102004b	16,8,17	157	104	400
607	105	7,9,12	6003c000	7f010a010001024b	23,15,8	127	84	405
608	-	2,6,23	02454000	00010a030000024c	16,5,31	151	122	399
609	31	8,9,12	a0000009	01020a000010004c	128	96	64	402
610	127	8,10,11	a0000b8a	03030a00000f004c	127	149	91	402
611	-	2,4,10	2220000b	07010a040103004c	24,8,33	157	111	404
612	-	5,10,14	22e0000c	0f010a000204004c	32,16,7	157	103	404
613	-	4,10,19	2240000d	1f010a000103004c	24,8,2	157	101	402
614	-	1,2,7	2004c00e	3f010a010404004c	32,35,11	157	108	400
615	211	1,6,7	a000000f	7f040a00001d004c	232	102	71	401
616	-	3,10,19	00044900	00010a000201004d	10,19,3	151	106	404
617	200	5,9,10	a0000009	01040a000019004d	200	102	63	405
618	-	5,13,20	22e0000a	03010a030203004d	24,16,31	157	103	403
619	-	5,8,9	2005400b	07010a000104004d	32,13,5	157	102	401
620	-	1,2,9	2005400c	0f010a020101004d	8,13,21	157	95	405
621	-	5,6,12	22c0000d	1f010a000103004d	24,8,6	157	103	403
622	297	10,12,17	22a0000e	3f010a030103004d	24,8,29	157	98	400
623	68	9,10,11	e0000780	7f020a0000000a4d	87	102	68	405
624	-	7,9,12	02c44000	00010a010000024e	16,1,15	151	107	401
625	133	5,8,12	20000009	01010a050607004e	56,48,40	121	79	405
626	-	3,5,13	2005c00a	03010a050302004e	16,31,47	157	98	401
627	-	5,10,14	2000090b	07010a000304004e	34,26,2	157	103	402
628	223	4,9,12	a0000b8c	0f040a00001b004e	223	150	89	399
629	-	2,5,6	22e0000d	1f010a020205004e	40,16,23	157	104	405
630	-	2,4,7	2004400e	3f010a010605004e	40,49,9	157	103	404
631	307	6,8,14	2000000f	7f010a020306004e	48,24,16	121	84	403
632	-	3,13,19	02400980	00010a010002004f	19,3,13	151	111	402
633	101	1,2,7	20000009	01010a020407004f	56,32,16	121	85	404
634	315	3,5,7	2005c00a	03010a010202004f	16,23,15	157	96	403
635	-	4,10,14	22a0000b	07010a020102004f	16,8,21	157	98	404
636	-	4,8,13	2260000c	0f010a010103004f	24,8,11	157	102	400
637	-	1,9,14	3240002d	1f010a000103004f	24,8,2	158	95	403
638	-	1,5,6	3004402e	3f010a000304004f	32,25,1	158	102	401
639	16	3,4,10	b000002f	7f010a000002004f	16	90	65	402
640	-	2,3,14	0ec00880	00010a0001020050	17,9,7	155	271	522
641	11	1,5,19	a0000989	01010b0000010050	11	136	224	524
642	119	6,9,10	a0000b8a	03020b00000e0050	119	149	233	520
643	-	2,3,11	2004400b	07010b0304040050	32,33,25	176	277	523
644	-	10,11,12	2004c00c	0f010b0104030050	24,35,11	176	297	523
645	-	4,8,11	2005800d	1f010b0502020050	16,22,46	176	291	521
646	249	11,12,13	a000088e	3f040b00001f0050	249	161	247	523
647	5	1,4,5	e0000780	7f040b0000001a50	215	119	209	523
648	-	1,22,23	02c44000	00010b0401000451	32,9,39	170	325	518
649	37	1,5,11	e0000480	01020b0000000951	73	113	186	521
650	3	6,15,18	a000098a	03010b0000000051	3	136	225	523
651	-	5,13,14	2220000b	07010b0102030051	24,16,9	176	299	525
652	93	5,9,11	a0000a8c	0f020b00000b0051	93	149	236	522
653	-	7,8,10	2004800d	1f010b0004030051	24,34,2	176	282	524
654	-	5,11,14	2220000e	3f010b0202040051	32,16,17	176	277	525
655	88	5,15,17	a000000f	7f020b00000b0051	88	107	188	523
656	-	10,18,19	02200900	00010b0201020052	18,10,19	170	289	522

657	38	1,7,8	a0000b09	01010b0000040052	38	136	214	522
658	55	7,10,12	a0000b8a	03010b0000060052	55	136	225	523
659	-	2,4,15	2005800b	07010b0103030052	24,30,14	176	289	523
660	-	3,4,12	2220000c	0f010b0403040052	32,24,33	176	282	523
661	-	4,11,12	22e0000d	1f010b0202030052	24,16,23	176	280	521
662	297	3,6,12	a000088e	3f050b0000250052	297	172	272	523
663	257	8,11,14	6003c000	7f010b0200020652	55,23,16	146	215	523
664	-	2,4,15	02a00880	00010b0100020053	17,1,14	170	291	520
665	33	4,6,11	e0000480	01010b0000000453	33	106	190	524
666	-	2,7,10	2004c00a	03010b0106060053	48,51,11	176	278	520
667	-	3,7,18	2005400b	07010b0605050053	40,45,53	176	276	522
668	-	10,12,17	2260000c	0f010b0101030053	24,8,11	176	280	521
669	-	2,4,5	22e0000d	1f010b0304050053	40,32,31	176	297	523
670	153	1,5,6	a000088e	3f030b0000130053	153	160	246	522
671	15	2,6,9	e0000780	7f010b0000000153	15	106	184	525
672	-	5,6,11	02400a80	00010b0000010054	13,5,7	170	284	521
673	28	7,9,10	a0000a09	01010b0000030054	28	136	221	522
674	-	3,9,14	22e0000a	03010b0001030054	24,8,7	176	280	522
675	-	1,3,6	2004800b	07010b0002030054	24,18,2	176	282	524
676	241	1,5,12	a000088c	0f040b00001e0054	241	161	244	522
677	-	3,4,8	22c0000d	1f010b0101040054	32,8,14	176	280	524
678	-	3,5,15	2005400e	3f010b0003050054	40,29,5	176	272	523
679	66	3,12,18	a000000f	7f040b00001b0054	216	113	197	525
680	-	1,30,35	00044a80	00010b0203040055	37,30,22	170	289	524
681	-	3,9,11	20000009	01010b0102080055	64,16,8	140	230	517
682	-	1,3,7	20000a8a	03010b0001030055	29,13,5	176	278	525
683	-	1,6,11	2005400b	07010b0001030055	24,13,5	176	279	525
684	-	3,13,18	2260000c	0f010b0201040055	32,8,19	176	291	520
685	-	1,3,4	2240000d	1f010b0003040055	32,24,2	176	277	525
686	197	2,12,13	a0000a8e	3f040b0000180055	197	161	242	523
687	13	5,12,14	a0000a8f	7f010b0000010055	13	136	224	526
688	-	6,14,19	0004c980	00010b0001020056	19,14,6	170	293	520
689	14	3,6,8	a0000009	01060b00002a0056	336	131	229	524
690	-	3,7,10	2005400a	03010b0205040056	32,45,21	176	294	520
691	-	2,6,13	2005800b	07010b0004010056	8,38,6	176	278	522
692	299	5,6,14	a000098c	0f050b0000250056	299	172	268	521
693	-	2,8,15	2005800d	1f010b0304040056	32,38,30	176	276	523
694	-	3,13,17	2004400e	3f010b0104050056	40,33,9	176	301	523
695	212	1,4,9	2000000f	7f010b0405060056	48,40,32	140	230	522
696	-	2,10,23	02a00900	00010b0200010057	10,2,23	170	290	520
697	267	8,12,16	60024000	01010b0400040557	41,33,32	146	199	522
698	215	8,9,10	a0000b8a	03040b00001a0057	215	161	246	523
699	-	1,10,15	22c0000b	07010b0401030057	24,8,38	176	297	522
700	-	2,5,6	22a0000c	0f010b0101030057	24,8,13	176	278	522
701	-	2,4,16	3004802d	1f010b0003040057	32,26,2	183	276	524
702	37	1,3,7	b0003aae	3f010b0000040057	37	153	233	525
703	-	1,7,12	3000002f	7f020b03040a0057	80,32,24	154	244	525
704	-	3,5,12	0e444000	00010b0101000558	40,9,11	180	113	418
705	19	5,6,8	e0000480	01030c0000001458	161	118	72	401
706	-	9,11,14	22e0000a	03010c0102040058	32,16,15	168	125	411
707	-	5,8,15	2004400b	07010c0204030058	24,33,17	168	111	407
708	287	2,4,5	22e0000c	0f010c0001040058	32,8,7	168	111	414
709	-	1,3,4	22c0000d	1f010c0205060058	48,40,22	168	124	410
710	-	1,14,15	2220000e	3f010c0301030058	24,8,25	168	115	413
711	92	7,8,11	6003c000	7f010c0100000558	47,7,8	132	101	405
712	-	3,4,5	02400980	00010c0000020059	19,3,5	162	107	410
713	41	7,10,17	e0000480	01010c0000000559	41	100	80	407
714	23	5,7,13	a0000b8a	03010c0000020059	23	136	100	409
715	-	1,4,7	2005800b	07010c0102030059	24,22,14	168	117	409
716	183	10,11,12	a0000b8c	0f030c0000160059	183	160	113	401
717	-	1,7,16	22c0000d	1f010c0201050059	40,8,22	168	120	410

718	-	1,2,5	2005400e	3f010c0002040059	32,21,5	168	119	410
719	150	8,9,12	6003c000	7f010c0100000159	15,7,8	132	91	411
720	-	2,8,11	02248000	00010c010000015a	8,2,11	162	122	411
721	9	3,6,9	e0000480	01010c000000015a	9	100	66	402
722	231	1,4,15	a0000b8a	03040c00001c005a	231	161	101	411
723	-	6,13,16	2004800b	07010c020403005a	24,34,18	168	124	409
724	-	5,8,13	2005400c	0f010c000101005a	8,13,5	168	110	409
725	-	5,6,9	22e0000d	1f010c000203005a	24,16,7	168	115	411
726	5	1,4,5	a0000a8e	3f010c000000005a	5	136	98	411
727	180	6,8,11	2000000f	7f010c050607005a	56,48,40	126	100	412
728	-	2,3,4	02200880	00010c010204005b	33,17,10	162	125	410
729	58	3,5,11	60024000	01010c010002035b	25,17,8	132	101	410
730	147	4,15,19	a000098a	03030c000012005b	147	160	112	387
731	-	2,6,8	2240000b	07010c020103005b	24,8,18	168	122	407
732	-	3,4,7	2005c00c	0f010c000202005b	16,23,7	168	116	411
733	-	2,7,8	2005c00d	1f010c000105005b	40,15,7	168	111	406
734	-	10,13,14	2005400e	3f010c000204005b	32,21,5	168	119	413
735	44	2,7,8	6003c000	7f010c020001035b	31,15,16	132	93	410
736	-	6,8,13	02254000	00010c000100015c	8,13,6	162	120	412
737	5	1,4,5	20000009	01010c040508005c	64,40,32	126	100	409
738	347	8,9,11	22e0000a	03010c000204005c	32,16,7	168	132	410
739	-	8,16,18	2240000b	07010c020102005c	16,8,18	168	115	414
740	153	3,12,24	a000088c	0f030c000013005c	153	160	95	392
741	-	3,8,9	2004c00d	1f010c000104005c	32,11,3	168	118	409
742	-	1,4,12	22a0000e	3f010c060106005c	48,8,53	168	124	410
743	90	1,12,13	a000000f	7f030c000012005c	144	112	67	409
744	-	1,11,13	02c44000	00010c010300035d	24,25,15	162	113	410
745	258	5,7,8	a0000009	01060c00002a005d	336	125	83	402
746	351	8,13,18	2005c00a	03010c000104005d	32,15,7	168	111	410
747	-	4,6,10	22c0000b	07010c010103005d	24,8,14	168	113	409
748	-	4,5,15	2260000c	0f010c010203005d	24,16,11	168	113	407
749	-	1,6,7	2000090d	1f010c000305005d	42,26,2	168	122	412
750	-	4,9,16	2005400e	3f010c010503005d	24,45,13	168	113	412
751	18	1,3,11	6003c000	7f010c020000035d	31,7,16	132	104	407
752	-	3,20,21	00048a80	00010c020100005e	5,15,23	162	123	407
753	158	5,8,13	60000480	01010c010300075e	57,24,8	132	95	407
754	19	12,14,19	a000098a	03010c000002005e	19	136	89	407
755	-	1,10,12	2004400b	07010c010202005e	16,17,9	168	115	410
756	349	1,9,16	2004400c	0f010c000102005e	16,9,1	168	121	406
757	-	1,6,7	2005c00d	1f010c010202005e	16,23,15	168	111	412
758	-	1,12,17	22a0000e	3f010c050304005e	32,24,45	168	179	412
759	98	2,3,9	61e00000	7f010c000102045e	39,23,15	132	100	411
760	-	3,13,26	02600900	00010c000004005f	34,2,5	162	151	407
761	3	1,2,3	e0000480	01010c000000045f	33	100	71	410
762	83	1,7,17	a000098a	03020c00000a005f	83	149	90	407
763	-	9,14,16	2004400b	07010c010303005f	24,25,9	168	118	411
764	-	3,5,6	2004400c	0f010c000304005f	32,25,1	168	109	402
765	-	5,10,11	3260002d	1f010c050306005f	48,24,43	169	116	414
766	-	9,19,22	3004402e	3f010c030406005f	48,33,25	169	115	410
767	168	4,7,8	b000002f	7f030c000015005f	168	113	80	412
768	-	4,17,19	0e200880	00010c0501060060	49,9,42	166	235	750
769	120	6,7,9	a0000009	01020d00000f0060	120	112	185	751
770	-	2,5,14	2004400a	03010d0206050060	40,49,17	187	248	753
771	-	6,15,17	2220000b	07010d0301020060	16,8,25	187	255	754
772	7	5,6,8	a0000b8c	0f010d0000000060	7	147	216	757
773	-	6,8,10	22c0000d	1f010d0301060060	48,8,30	187	254	756
774	185	7,14,16	a000088e	3f030d0000170060	185	171	234	753
775	367	4,6,7	e0000780	7f060d0000002d60	367	142	215	753
776	-	3,12,17	00044880	00010d0102030061	25,18,10	181	252	753
777	29	1,10,16	20000009	01010d0104070061	56,32,8	145	221	757
778	375	3,16,19	2004c00a	03010d0002020061	16,19,3	187	244	752

779	-	3,8,10	2004400b	07010d0203030061	24,25,17	187	248	753
780	-	5,8,16	22a0000c	0f010d0001020061	16,8,5	187	255	755
781	-	2,16,17	2260000d	1f010d0203050061	40,24,19	187	258	754
782	329	2,3,9	22a0000e	3f010d0003050061	40,24,5	187	250	750
783	68	1,7,10	e0000780	7f020d0000000861	71	118	194	755
784	-	6,9,13	00058880	00010d0103020062	17,31,15	181	249	749
785	92	5,9,10	60000480	01010d0102000162	9,16,8	151	222	752
786	-	4,6,15	2004c00a	03010d0001060062	48,11,3	187	249	753
787	-	3,6,7	2005400b	07010d0002050062	40,21,5	187	249	751
788	-	3,10,17	2220000c	0f010d0401030062	24,8,33	187	257	750
789	-	1,2,5	22e0000d	1f010d0403070062	56,24,39	187	257	756
790	-	3,7,9	2005400e	3f010d0504020062	16,37,45	187	249	752
791	30	1,4,6	a0000b0f	7f010d0000030062	30	147	220	754
792	-	13,17,23	02200880	00010d0302030063	25,17,26	181	256	755
793	253	1,4,9	60024000	01010d0100040563	41,33,8	151	226	756
794	143	5,12,13	a0000b8a	03030d0000110063	143	171	228	753
795	-	14,15,20	2004400b	07010d0304050063	40,33,25	187	248	753
796	-	1,4,9	2260000c	0f010d0301050063	40,8,27	187	257	753
797	-	4,10,12	2240000d	1f010d0201030063	24,8,18	187	255	750
798	-	3,6,7	22a0000e	3f010d0102060063	48,16,13	187	259	752
799	25	4,6,9	a000088f	7f010d0000030063	25	147	216	754
800	-	6,9,14	00054880	00010d0001010064	9,14,6	181	252	758
801	217	1,7,9	e0000480	01040d0000001b64	217	124	195	753
802	-	12,13,15	2005c00a	03010d0102020064	16,23,15	187	243	752
803	-	2,9,14	2240000b	07010d0201020064	16,8,18	187	253	752
804	295	7,9,10	a0000b8c	0f050d0000240064	295	183	236	753
805	-	2,7,8	20000b0d	1f010d0002050064	46,22,6	187	247	753
806	141	5,9,10	a0000a8e	3f030d0000110064	141	171	230	751
807	7	3,6,7	e0000780	7f010d0000000064	7	111	189	754
808	-	2,3,22	02200b00	00010d0100020065	22,6,15	181	261	749
809	15	1,3,8	e0000480	01040d0000001d65	233	124	197	754
810	299	7,12,13	a000098a	03050d0000250065	299	183	240	751
811	-	8,10,12	2240000b	07010d0302060065	48,16,26	187	258	749
812	167	2,6,17	a0000b8c	0f030d0000140065	167	171	227	756
813	-	1,3,10	20000b0d	1f010d0102040065	38,22,14	187	240	746
814	145	1,4,10	a000088e	3f030d0000120065	145	171	231	751
815	333	7,10,12	a000000f	7f060d00002a0065	336	136	201	751
816	-	5,15,23	00048a80	00010d0201000066	5,15,23	181	251	752
817	52	5,8,13	a0000a09	01010d0000060066	52	147	217	753
818	119	1,7,15	a0000b8a	03020d00000e0066	119	160	225	752
819	-	7,9,16	2005800b	07010d0203030066	24,30,22	187	247	754
820	-	3,7,12	2220000c	0f010d0202040066	32,16,17	187	247	747
821	-	2,11,15	22c0000d	1f010d0403050066	40,24,38	187	254	750
822	-	5,16,22	2004400e	3f010d0302010066	8,17,25	187	251	754
823	9	3,6,9	a000000f	7f050d0000230066	280	129	199	755
824	-	3,11,14	02600980	00010d0100010067	11,3,14	181	254	751
825	38	3,7,13	a0000b09	01010d0000040067	38	147	225	752
826	255	6,8,13	a0000b8a	03050d00001f0067	255	183	237	755
827	-	7,10,12	2240000b	07010d0401030067	24,8,34	187	258	754
828	205	10,13,14	a0000a8c	0f040d0000190067	205	172	231	755
829	-	1,3,4	3260002d	1f010d0403040067	32,24,35	194	257	755
830	-	7,10,17	70045740	3f010d0604000267	22,33,49	204	266	752
831	49	6,8,11	b00008af	7f010d0000060067	49	154	245	749
832	-	2,5,13	0ea00880	00010d0201040068	33,9,22	191	120	628
833	149	2,3,8	20000009	01010e0607080068	64,56,48	152	100	626
834	-	4,7,12	2004c00a	03010e0104070068	56,35,11	200	136	626
835	-	5,7,14	2220000b	07010e0404050068	40,32,33	200	135	626
836	-	2,9,10	2004400c	0f010e0301010068	8,9,25	200	148	631
837	-	5,6,8	2000090d	1f010e0003040068	34,26,2	200	131	626
838	61	1,8,12	a0000a8e	3f020e0000070068	61	173	110	626
839	54	2,6,14	e0000780	7f060e0000002868	327	149	93	626

840	-	1,5,11	02400a80	00010e0301020069	21,13,31	194	133	629
841	144	4,5,6	a0000009	01030e0000120069	144	130	112	626
842	47	7,8,12	a0000b8a	03010e0000050069	47	160	107	623
843	-	7,10,11	22a0000b	07010e0202030069	24,16,21	200	132	627
844	-	11,16,18	2260000c	0f010e0102040069	32,16,11	200	141	626
845	2	7,8,10	a000090d	1f010e0000000069	2	160	106	626
846	-	10,12,13	2005400e	3f010e0102030069	24,21,13	200	129	629
847	136	3,13,17	a000000f	7f030e0000110069	136	130	82	629
848	-	1,4,11	02400a80	00010e000102006a	21,13,7	194	135	626
849	253	3,9,11	60024000	01010e010002036a	25,17,8	158	106	629
850	111	6,7,16	a0000b8a	03020e00000d006a	111	173	120	623
851	-	5,10,13	20000b0b	07010e010305006a	46,30,14	200	136	629
852	-	4,5,8	22e0000c	0f010e040205006a	40,16,39	200	134	625
853	-	1,7,10	22c0000d	1f010e020305006a	40,24,22	200	141	626
854	-	3,5,7	2004400e	3f010e000306006a	48,25,1	200	133	627
855	29	2,3,17	e0000780	7f030e000000126a	151	136	106	627
856	-	3,10,19	00044900	00010e000201006b	10,19,3	194	126	626
857	119	3,5,6	60000480	01010e030800056b	41,64,24	158	127	629
858	-	7,10,16	22e0000a	03010e010103006b	24,8,15	200	132	626
859	-	4,15,17	2220000b	07010e000203006b	24,16,1	200	135	631
860	-	2,11,14	2220000c	0f010e010105006b	40,8,9	200	134	627
861	-	2,5,9	22c0000d	1f010e010107006b	56,8,14	200	136	626
862	349	1,6,7	2005400e	3f010e000303006b	24,29,5	200	138	630
863	-	2,3,6	61e00000	7f010e000002036b	31,23,7	158	116	629
864	-	6,10,21	02454000	00010e020200016c	8,21,23	194	133	626
865	1	1,4,6	e0000480	01010e000000006c	1	118	77	627
866	75	6,13,14	a000098a	03020e000009006c	75	173	117	623
867	-	2,5,9	2000090b	07010e010204006c	34,18,10	200	137	627
868	145	5,12,17	a000088c	0f030e000012006c	145	184	115	626
869	-	6,7,11	2004800d	1f010e010403006c	24,34,10	200	144	627
870	-	11,16,17	22a0000e	3f010e000304006c	32,24,5	200	130	628
871	378	1,4,12	60000780	7f010e050300026c	23,24,40	158	109	627
872	-	2,7,16	02a48000	00010e000000026d	16,2,7	194	131	624
873	-	1,3,7	60024000	01010e040003066d	49,25,32	158	108	623
874	-	4,7,12	22e0000a	03010e010205006d	40,16,15	200	129	629
875	-	1,8,12	22a0000b	07010e010204006d	32,16,13	200	135	628
876	-	5,8,14	22e0000c	0f010e040203006d	24,16,39	200	131	627
877	-	4,5,6	2005800d	1f010e000501006d	8,46,6	200	136	624
878	-	7,9,20	2004c00e	3f010e000301006d	8,27,3	200	127	627
879	11	3,5,9	a000000f	7f020e00000a006d	80	119	103	622
880	-	5,7,15	00048a80	00010e010000006e	5,7,15	194	133	627
881	78	5,8,11	a0000009	01070e000031006e	392	154	104	627
882	-	1,5,11	2260000a	03010e000103006e	24,8,3	200	131	624
883	-	12,16,17	2000090b	07010e000102006e	18,10,2	200	129	627
884	173	8,11,15	a0000a8c	0f030e000015006e	173	184	127	628
885	-	1,7,8	2260000d	1f010e020104006e	32,8,19	200	144	626
886	-	8,9,13	2005400e	3f010e010204006e	32,21,13	200	124	626
887	147	4,6,7	a000000f	7f060e00002a006e	336	143	105	628
888	-	10,18,19	02200900	00010e020102006f	18,10,19	194	134	625
889	169	6,8,10	e0000480	01030e000000156f	169	136	92	623
890	-	5,13,18	2260000a	03010e030103006f	24,8,27	200	136	628
891	-	3,10,12	22c0000b	07010e020102006f	16,8,22	200	138	623
892	31	6,7,14	a0003b8c	0f010e000003006f	31	164	124	628
893	-	6,8,11	3004c02d	1f010e040506006f	48,43,35	201	123	628
894	173	1,7,19	b0003aae	3f030e000015006f	173	189	117	625
895	12	4,8,12	3000002f	7f020e06070d006f	104,56,48	160	102	627
896	-	16,21,23	0ec44000	00010e0505000670	48,41,47	198	261	895
897	113	7,8,16	e0000480	01020f0000000e70	113	136	204	900
898	207	2,5,12	a0000b8a	03040f0000190070	207	196	239	902
899	-	5,15,18	2004800b	07010f0102040070	32,18,10	219	251	890
900	1	5,7,10	a000088c	0f010f0000000070	1	171	232	803

901	-	6,7,13	2240000d	1f010f0201050070	40,8,18	219	259	893
902	-	9,16,20	6004c700	3f010f0403000270	22,27,35	225	259	815
903	160	1,7,8	a000000f	7f030f0000140070	160	141	186	893
904	-	1,11,15	02600980	00010f0202030071	27,19,22	213	271	824
905	117	2,9,15	20000009	01010f0207080071	64,56,16	171	229	899
906	187	2,11,13	a000098a	03030f0000170071	187	195	241	901
907	-	2,10,12	2220000b	07010f0001070071	56,8,1	219	259	843
908	143	4,7,8	a0000b8c	0f030f0000110071	143	195	240	903
909	-	1,4,14	2005c00d	1f010f0004060071	48,39,7	219	255	903
910	-	7,9,15	2220000e	3f010f0002050071	40,16,1	219	277	901
911	204	1,6,8	60000780	7f010f0102000471	39,16,8	177	241	898
912	-	2,5,20	02454000	00010f0103000172	8,29,15	213	272	904
913	91	9,12,18	e0000480	01030f0000001072	129	147	205	895
914	-	1,2,4	2005c00a	03010f0203030072	24,31,23	219	258	899
915	-	3,6,8	2240000b	07010f0403060072	48,24,34	219	270	900
916	-	8,12,17	2004400c	0f010f0104030072	24,33,9	219	257	891
917	-	7,10,12	2005800d	1f010f0001030072	24,14,6	219	255	900
918	77	10,12,13	a0000a8e	3f020f0000090072	77	184	232	886
919	36	5,7,9	a000000f	7f060f00002a0072	336	154	209	891
920	-	4,7,21	00058880	00010f0102030073	25,23,15	213	260	899
921	221	6,8,9	60024000	01010f0700020473	33,17,56	177	225	900
922	-	5,6,7	2260000a	03010f0202050073	40,16,19	219	265	891
923	-	13,14,16	2005800b	07010f0403020073	16,30,38	219	259	906
924	-	8,10,13	2004c00c	0f010f0706060073	48,51,59	219	248	896
925	-	7,15,16	2005c00d	1f010f0001020073	16,15,7	219	262	889
926	365	5,6,13	2220000e	3f010f0201040073	32,8,17	219	263	905
927	403	3,7,9	e0000780	7f080f0000003873	455	165	238	891
928	-	3,11,17	00048880	00010f0001020074	17,11,3	213	259	891
929	-	3,4,11	60024000	01010f0300020574	41,17,24	177	239	902
930	-	11,14,18	2004c00a	03010f0205050074	40,43,19	219	252	897
931	-	4,9,10	2000090b	07010f0102030074	26,18,10	219	257	895
932	275	1,2,9	a000098c	0f050f0000220074	275	207	235	893
933	-	1,6,16	2004800d	1f010f0203030074	24,26,18	219	255	898
934	-	5,6,22	22a0000e	3f010f0104060074	48,32,13	219	266	903
935	417	7,11,12	2000000f	7f010f0406070074	56,48,32	171	227	896
936	-	7,13,16	02454000	00010f0001000275	16,13,7	213	264	896
937	217	4,5,9	e0000480	01040f0000001b75	217	142	201	905
938	207	3,6,15	a0000b8a	03040f0000190075	207	196	236	890
939	-	4,5,7	22a0000b	07010f0003060075	48,24,5	219	248	899
940	-	3,16,17	20000a8c	0f010f0001020075	21,13,5	219	254	895
941	-	1,6,11	22c0000d	1f010f0103050075	40,24,14	219	260	898
942	-	13,14,19	2220000e	3f010f0401030075	24,8,33	219	263	888
943	24	4,14,17	a000000f	7f010f0000030075	24	123	195	901
944	-	2,3,14	02a00900	00010f0201020076	18,10,23	213	264	889
945	79	1,4,21	60000480	01010f0605000476	33,40,48	177	237	896
946	-	10,11,23	2260000a	03010f0503050076	40,24,43	219	259	905
947	-	5,6,9	22c0000b	07010f0001050076	40,8,6	219	269	897
948	-	15,16,18	2260000c	0f010f0403040076	32,24,35	219	258	895
949	-	2,3,8	22e0000d	1f010f0303040076	32,24,31	219	269	900
950	-	9,14,16	2005c00e	3f010f0205040076	32,47,23	219	262	903
951	260	1,6,7	e0000780	7f070f0000003076	391	171	223	903
952	-	7,9,16	02c44000	00010f0001000277	16,9,7	213	260	894
953	168	2,7,10	a0000009	01040f00001c0077	224	136	199	895
954	-	6,8,11	2000098a	03010f0002030077	27,19,3	219	256	901
955	-	3,6,7	2005400b	07010f0106060077	48,53,13	219	247	909
956	305	4,5,6	a000088c	0f050f0000260077	305	207	247	897
957	-	6,9,10	3240002d	1f010f0102040077	32,16,10	226	259	895
958	-	5,9,14	3220002e	3f010f0403060077	48,24,33	226	266	891
959	143	8,10,13	b000002f	7f050f0000270077	312	154	202	892
960	-	6,9,13	0ec44000	00010f0303000578	40,25,31	223	145	659
961	18	8,9,14	20000009	0101100107080078	64,56,8	157	111	663

962	-	5,8,15	22e0000a	0301100002030078	24,16,7	211	129	678
963	-	6,9,20	2004800b	0701100204030078	24,34,18	211	139	691
964	103	1,4,6	a0000b8c	0f021000000c0078	103	184	119	688
965	-	2,4,15	2004800d	1f01100001020078	16,10,2	211	132	690
966	-	7,9,12	2005400e	3f01100001050078	40,13,5	211	150	694
967	36	3,8,20	2000000f	7f01100105070078	56,40,8	157	104	688
968	-	13,18,19	0244c000	0001100201000479	32,11,21	205	141	689
969	74	4,10,11	20000009	0101100107080079	64,56,8	157	99	687
970	-	2,5,12	2000098a	0301100104050079	43,35,11	211	153	691
971	-	1,2,6	2004400b	0701100403020079	16,25,33	211	143	691
972	115	5,13,20	a000098c	0f021000000e0079	115	184	113	678
973	-	4,6,13	2004800d	1f01100007060079	48,58,2	211	134	691
974	-	1,10,21	2220000e	3f01100403040079	32,24,33	211	140	686
975	19	1,4,13	60000780	7f01100107000379	31,56,8	163	113	693
976	-	2,4,21	02c44000	000110000300037a	24,25,7	205	141	692
977	15	2,3,6	a0000009	01081000003c007a	480	153	103	690
978	-	5,7,11	2005c00a	030110050201007a	8,23,47	211	137	690
979	-	14,16,18	2000090b	070110010405007a	42,34,10	211	141	695
980	-	6,7,8	2004c00c	0f0110030202007a	16,19,27	211	129	681
981	-	5,6,12	62600680	1f0110030300037a	29,24,27	217	150	691
982	277	2,6,7	a0000a8e	3f0510000022007a	277	207	132	691
983	230	3,4,10	2000000f	7f0110040506007a	48,40,32	157	110	689
984	-	10,23,41	00048880	000110010505007b	41,43,11	205	129	690
985	222	2,4,8	60024000	010110020000037b	25,1,16	163	114	691
986	-	10,12,19	2004c00a	030110000504007b	32,43,3	211	140	694
987	-	12,13,16	2004800b	070110040302007b	16,26,34	211	138	689
988	121	1,3,17	a000088c	0f021000000f007b	121	184	134	689
989	-	2,4,10	2240000d	1f0110010206007b	48,16,10	211	142	693
990	-	11,16,17	62a00700	3f0110000200047b	38,16,5	217	149	691
991	39	3,9,10	e0000780	7f0110000000047b	39	123	89	694
992	-	7,14,27	02a44000	000110000400017c	8,33,6	205	149	689
993	62	1,8,10	60024000	010110050000027c	17,1,40	163	111	693
994	223	6,9,19	a0000b8a	03041000001b007c	223	196	126	694
995	-	1,11,16	20000b0b	070110030405007c	46,38,30	211	136	691
996	-	1,8,11	2000088c	0f0110000205007c	41,17,1	211	137	695
997	-	3,6,12	22c0000d	1f0110030205007c	40,16,30	211	150	690
998	101	14,15,19	a0000a8e	3f021000000c007c	101	184	119	687
999	59	5,11,14	2000000f	7f0210040508007c	64,40,32	164	114	692
1000	-	2,3,16	02248000	000110000000027d	16,2,3	205	138	689
1001	17	1,3,5	e0000480	010110000000027d	17	123	82	696
1002	-	2,3,5	62a28000	030110000000027d	18,2,5	217	144	689
1003	-	3,8,13	2004400b	070110010303007d	24,25,9	211	148	689
1004	-	12,14,17	2004c00c	0f0110000207007d	56,19,3	211	132	691
1005	-	7,16,18	2260000d	1f0110020102007d	16,8,19	211	143	693
1006	-	3,4,5	2220000e	3f0110040207007d	56,16,33	211	142	695
1007	75	1,3,8	a000000f	7f021000000c007d	96	124	88	699
1008	-	10,13,27	02600900	000110040104007e	34,10,37	205	143	687
1009	55	2,4,11	60024000	010110040001027e	17,9,32	163	117	692
1010	-	2,7,10	2004400a	030110020504007e	32,41,17	211	154	693
1011	-	4,7,10	2000090b	070110000205007e	42,18,2	211	132	689
1012	-	1,2,9	2004c00c	0f0110020303007e	24,27,19	211	138	693
1013	-	6,8,9	2240000d	1f0110040305007e	40,24,34	211	140	689
1014	385	1,11,17	2004400e	3f0110030606007e	48,49,25	211	141	691
1015	186	1,3,12	e0000780	7f0810000000377e	447	159	112	694
1016	-	3,6,15	0004c900	000110010402007f	18,37,13	205	138	693
1017	-	1,4,9	60000480	010110010400027f	17,32,8	163	110	692
1018	-	5,10,12	2005c00a	030110000101007f	8,15,7	211	138	683
1019	-	1,8,10	2005400b	070110020304007f	32,29,21	211	135	687
1020	461	7,12,15	2260000c	0f0110020506007f	48,40,19	211	144	690
1021	-	1,2,5	3240002d	1f0110030506007f	48,40,26	212	141	692
1022	317	7,12,14	300009ae	3f0110020304007f	35,27,19	212	137	691

1023	7	3,8,14	f00017c0	7f0110000000007f	7	128	81	688
1024	-	9,22,23	0e400880	0001100203040080	33,25,19	209	280	1061
1025	294	1,5,9	60024000	0101110300020480	33,17,24	182	247	1066
1026	35	4,14,19	a000098a	0301110000040080	35	182	241	1048
1027	-	6,12,13	22c0000b	0701110002030080	24,16,6	230	281	1057
1028	203	1,5,9	a000098c	0f04110000190080	203	207	247	1055
1029	-	3,8,9	2005800d	1f01110102020080	16,22,14	230	272	1062
1030	93	1,5,17	a0000a8e	3f021100000b0080	93	195	247	1051
1031	68	2,3,12	e0000780	7f05110000002380	287	158	219	1065
1032	-	10,18,25	00044880	0001110102030081	25,18,10	224	285	1052
1033	108	2,5,12	20000009	0101110102040081	32,16,8	176	235	1069
1034	75	6,10,13	a000098a	0302110000090081	75	195	240	1058
1035	-	5,6,12	22a0000b	0701110001040081	32,8,5	230	191	1049
1036	411	6,9,13	2004c00c	0f01110005050081	40,43,3	230	270	1066
1037	-	2,7,12	2000090d	1f01110203050081	42,26,18	230	264	1045
1038	-	3,7,15	22a0000e	3f01110204060081	48,32,21	230	283	1059
1039	21	2,4,5	a000000f	7f021100000b0081	88	135	198	1059
1040	-	1,19,26	02200900	0001110503040082	34,26,43	224	285	1053
1041	412	1,5,8	20000009	0101110207080082	64,56,16	176	233	1047
1042	439	1,3,16	2260000a	0301110202030082	24,16,19	230	275	1054
1043	-	6,7,10	2240000b	0701110304060082	48,32,26	230	284	1055
1044	41	1,8,20	a000088c	0f01110000050082	41	182	243	1059
1045	-	6,9,13	22e0000d	1f01110302030082	24,16,31	230	280	1060
1046	-	13,14,16	2005c00e	3f01110501010082	8,15,47	230	275	1054
1047	10	2,8,10	a000090f	7f01110000010082	10	182	248	1061
1048	-	3,21,24	0244c000	0001110200000383	24,3,21	224	286	1052
1049	141	3,5,11	20000009	0101110103070083	56,24,8	176	232	1048
1050	-	2,9,17	2005400a	0301110003010083	8,29,5	230	269	1057
1051	-	10,12,13	22c0000b	0701110101020083	16,8,14	230	280	1058
1052	291	1,7,8	a000098c	0f05110000240083	291	218	257	1056
1053	-	1,9,10	2260000d	1f01110203050083	40,24,19	230	283	1059
1054	105	4,5,14	a000088e	3f021100000d0083	105	195	242	1048
1055	24	5,6,15	a000000f	7f01110000030083	24	128	192	1049
1056	-	6,9,14	00054880	0001110001010084	9,14,6	224	272	1059
1057	198	4,9,14	60000480	0101110307000684	49,56,24	182	242	1058
1058	27	7,13,14	a000098a	0301110000030084	27	182	238	1055
1059	-	1,3,6	22a0000b	0701110502030084	24,16,45	230	282	1061
1060	-	8,13,17	2220000c	0f01110201020084	16,8,17	230	278	1049
1061	-	1,3,10	2260000d	1f01110106070084	56,48,11	230	282	1060
1062	-	1,2,5	2004400e	3f01110703020084	16,25,57	230	270	1050
1063	168	2,6,11	a000000f	7f041100001a0084	208	141	207	1060
1064	-	6,9,18	02454000	0001110203000385	24,29,23	224	283	1060
1065	463	5,7,8	61200000	0101110000020385	25,17,1	182	252	1049
1066	-	3,9,10	2260000a	0301110103070085	56,24,11	230	286	1063
1067	-	8,9,13	2005800b	0701110102020085	16,22,14	230	273	1057
1068	-	3,8,15	2220000c	0f01110201030085	24,8,17	230	284	1058
1069	-	8,16,18	2240000d	1f01110201020085	16,8,18	230	286	1064
1070	-	11,14,15	2005400e	3f01110506060085	48,53,45	230	267	1061
1071	50	5,7,10	a000090f	7f01110000060085	50	182	239	1063
1072	-	8,9,19	02444000	0001110201000186	8,9,19	224	286	1060
1073	-	3,6,12	20000009	0102110204090086	72,32,16	183	240	1058
1074	-	10,14,17	2005c00a	0301110403030086	24,31,39	230	268	1056
1075	-	5,14,15	2220000b	0701110102030086	24,16,9	230	286	1065
1076	-	1,4,17	2260000c	0f01110201020086	16,8,19	230	284	1050
1077	-	7,9,10	22c0000d	1f01110101040086	32,8,14	230	278	1062
1078	445	4,14,17	20000a8e	3f01110102050086	45,21,13	230	261	1052
1079	230	5,14,17	60000780	7f01110104000486	39,32,8	182	243	1056
1080	-	3,12,29	0244c000	0001110501000187	8,11,45	224	286	1058
1081	24	2,14,16	a0000009	0101110000030087	24	128	196	1063
1082	407	6,11,14	2004c00a	0301110102030087	24,19,11	230	270	1058
1083	-	2,7,16	2220000b	0701110701070087	56,8,57	230	284	1052

1084	189	5,9,17	a0000a8c	0f04110000170087	189	207	262	1061
1085	-	2,4,10	3004802d	1f01110001020087	16,10,2	237	280	1055
1086	-	4,9,10	3220002e	3f01110204060087	48,32,17	237	295	1050
1087	112	3,7,8	b000002f	7f021100000e0087	112	142	219	1062
1088	-	10,21,22	0ec44000	0001110401000688	48,9,39	234	164	925
1089	91	1,8,16	a0000009	0107120000320088	400	177	110	921
1090	79	1,2,8	a0000b8a	0302120000090088	79	208	137	925
1091	-	5,10,12	2004800b	0701120103020088	16,26,10	243	149	923
1092	23	15,20,21	a0000b8c	0f01120000020088	23	195	143	924
1093	-	1,6,7	2000090d	1f01120104050088	42,34,10	243	149	923
1094	261	3,4,5	a0000a8e	3f05120000200088	261	231	135	925
1095	139	2,4,5	e0000780	7f07120000002f88	383	183	111	923
1096	-	6,15,24	02258000	0001120100000389	24,6,15	237	157	923
1097	14	3,11,13	60000480	0101120104000189	9,32,8	189	127	930
1098	83	7,15,18	a000098a	03021200000a0089	83	208	129	925
1099	-	1,9,16	2004400b	0701120001020089	16,9,1	243	175	930
1100	-	14,16,19	2004c00c	0f01120004010089	8,35,3	243	163	930
1101	-	4,7,9	22c0000d	1f01120304070089	56,32,30	243	192	924
1102	117	8,10,15	a0000a8e	3f021200000e0089	117	208	130	924
1103	65	4,6,13	e0000780	7f02120000000889	71	148	106	930
1104	-	7,12,21	00044880	000112010203008a	25,18,10	237	149	923
1105	21	6,9,16	e0000480	010112000000078a	57	141	103	926
1106	195	3,11,16	a000098a	030412000018008a	195	220	142	921
1107	-	10,11,23	2004800b	070112030101008a	8,10,26	243	182	926
1108	327	6,9,13	2220000c	0f0112040203008a	24,16,33	243	185	921
1109	-	3,14,17	22e0000d	1f0112020203008a	24,16,23	243	200	927
1110	-	1,6,16	2220000e	3f0112020205008a	40,16,17	243	168	920
1111	13	4,6,7	a000000f	7f0112000005008a	40	135	99	925
1112	-	6,8,15	02258000	000112010000018b	8,6,15	237	173	924
1113	107	5,9,12	60024000	0101120400000038b	25,1,32	189	138	922
1114	-	6,10,19	22e0000a	030112060206008b	48,16,55	243	194	923
1115	-	3,15,18	2240000b	070112050304008b	32,24,42	243	166	924
1116	479	2,4,5	2000098c	0f0112010204008b	35,19,11	243	190	926
1117	-	4,10,12	20000b0d	1f0112000203008b	30,22,6	243	162	921
1118	-	5,7,9	2004400e	3f0112010304008b	32,25,9	243	168	926
1119	283	2,5,11	6003c000	7f0112040002058b	47,23,32	189	127	926
1120	-	5,11,13	00048980	000112000101008c	11,13,5	237	157	924
1121	62	5,9,12	e0000480	0102120000000e8c	113	148	125	924
1122	-	1,13,16	22e0000a	030112020304008c	32,24,23	243	153	924
1123	-	3,7,14	2005400b	070112000103008c	24,13,5	243	157	926
1124	-	5,12,22	2004400c	0f0112000203008c	24,17,1	243	161	927
1125	-	3,8,15	22e0000d	1f0112010107008c	56,8,15	243	184	924
1126	309	13,17,21	a0000a8e	3f0512000026008c	309	231	140	926
1127	27	2,3,9	a000000f	7f0912000040008c	512	182	142	922
1128	-	1,3,7	02c44000	000112030000048d	32,1,31	237	170	927
1129	103	5,6,8	a0000009	01041200001a008d	208	148	109	926
1130	551	3,10,11	2004c00a	030112000102008d	16,11,3	243	176	926
1131	-	1,6,10	22a0000b	070112030205008d	40,16,29	243	186	925
1132	-	1,2,20	2005c00c	0f0112000105008d	40,15,7	243	182	921
1133	-	4,6,11	2240000d	1f0112050104008d	32,8,42	243	166	919
1134	-	4,13,16	2004400e	3f0112000301008d	8,25,1	243	183	928
1135	9	2,4,9	6003c000	7f0112030000028d	23,7,24	189	120	922
1136	-	2,4,9	02c44000	000112020000028e	16,1,23	237	159	923
1137	277	5,12,17	e0000480	010512000000278e	313	165	124	927
1138	31	3,4,7	a0000b8a	030112000003008e	31	195	144	928
1139	-	5,12,13	2240000b	070112040102008e	16,8,34	243	172	928
1140	539	4,16,17	2005c00c	0f0112000302008e	16,31,7	243	171	924
1141	-	3,7,12	2005c00d	1f0112000104008e	32,15,7	243	171	919
1142	357	6,9,12	2220000e	3f0112030204008e	32,16,25	243	165	918
1143	-	1,2,7	60000780	7f0112010300058e	47,24,8	189	136	931
1144	-	1,10,15	02400a80	000112020001008f	13,5,23	237	158	924

1145	227	8,12,16	20000009	0101120105060 08f	48,40,8	183	137	920
1146	131	3,9,16	a000098a	0303120000100 8f	131	219	141	918
1147	-	3,6,7	22a0000b	0701120003050 8f	40,24,5	243	168	924
1148	23	4,5,25	a0003b8c	0f01120000020 8f	23	199	143	926
1149	-	3,17,20	3004802d	1f01120207060 8f	48,58,18	244	158	928
1150	-	1,4,13	3220002e	3f01120203040 8f	32,24,17	244	174	921
1151	90	10,13,15	3000002f	7f02120507080 8f	64,56,40	191	123	929
1152	-	10,11,17	0ea44000	0001120104000 390	24,33,14	241	294	1082
1153	241	2,12,14	e0000480	0104130000001 e90	241	165	234	1084
1154	75	1,2,9	a000098a	0302130000090 090	75	219	258	1074
1155	-	1,6,13	2004800b	0701130005040 090	32,42,2	262	286	1079
1156	307	11,13,17	a000098c	0f05130000260 090	307	242	264	1075
1157	-	3,7,8	2005800d	1f01130001040 090	32,14,6	262	282	1075
1158	245	1,4,8	a0000a8e	3f041300001e0 090	245	231	261	1059
1159	66	5,10,14	a000000f	7f07130000350 090	424	188	233	1077
1160	-	6,15,30	02200b00	0001130100030 091	30,6,15	256	304	1084
1161	365	2,11,13	e0000480	0107130000003 391	409	194	248	1075
1162	-	11,16,18	22e0000a	0301130003050 091	40,24,7	262	290	1075
1163	-	1,10,11	2005800b	0701130002010 091	8,22,6	262	285	1079
1164	19	2,14,17	a000098c	0f01130000020 091	19	206	255	1073
1165	-	1,6,8	2240000d	1f01130003040 091	32,24,2	262	287	1077
1166	189	7,12,17	a0000a8e	3f04130000170 091	189	231	265	1078
1167	133	5,13,17	60000780	7f01130104000 591	47,32,8	208	257	1072
1168	-	2,7,12	00048980	0001130203050 092	43,29,21	256	296	1079
1169	114	4,8,13	20000009	0101130203040 092	32,24,16	202	246	1072
1170	-	9,11,12	2005c00a	0301130605050 092	40,47,55	262	285	1074
1171	-	1,5,6	22c0000b	0701130001050 092	40,8,6	262	293	1074
1172	-	5,13,15	22e0000c	0f01130002040 092	32,16,7	262	293	1082
1173	-	5,14,17	22e0000d	1f01130201030 092	24,8,23	262	273	1079
1174	133	2,9,10	a0000a8e	3f03130000100 092	133	230	262	1082
1175	476	4,11,12	60000780	7f01130305000 392	31,40,24	208	269	1074
1176	-	1,2,19	02444000	0001130302000 193	8,17,27	256	299	1082
1177	16	1,3,11	a0000009	0101130000020 093	16	146	211	1077
1178	375	1,10,11	a0000b8a	03061300002e0 093	375	255	284	1069
1179	-	6,8,15	2004800b	0701130302010 093	8,18,26	262	281	1064
1180	-	3,4,6	2220000c	0f01130102030 093	24,16,9	262	299	1079
1181	-	6,11,17	2000090d	1f01130001020 093	18,10,2	262	283	1070
1182	-	3,11,15	22a0000e	3f01130401020 093	16,8,37	262	293	1066
1183	87	5,10,11	e0000780	7f02130000000 a93	87	159	231	1090
1184	-	2,3,5	02400a80	0001130300010 094	13,5,31	256	300	1083
1185	134	2,3,12	60000480	0101130503000 194	9,24,40	208	264	1071
1186	171	5,6,10	a000098a	0303130000150 094	171	230	258	1066
1187	-	4,8,13	2004800b	0701130602020 094	16,18,50	262	286	1078
1188	413	5,16,25	22a0000c	0f01130606070 094	56,48,53	262	300	1079
1189	-	1,3,8	2004800d	1f01130603020 094	16,26,50	262	289	1078
1190	233	1,10,20	a000088e	3f041300001d0 094	233	231	264	1075
1191	196	1,6,7	2000000f	7f021308090e0 094	112,72,64	209	253	1078
1192	-	7,8,9	02c44000	0001130001000 195	8,9,7	256	297	1070
1193	173	2,9,10	60024000	0101130200020 395	25,17,16	208	268	1079
1194	-	12,14,15	62600500	0301130104000 195	10,32,11	268	309	1084
1195	-	5,6,13	22a0000b	0701130301020 095	16,8,29	262	298	1074
1196	519	11,15,17	2004c00c	0f01130203050 095	40,27,19	262	283	1074
1197	-	2,8,9	22c0000d	1f01130001040 095	32,8,6	262	288	1070
1198	-	5,9,10	62e00700	3f01130102000 195	14,16,15	268	310	1082
1199	114	2,7,16	6003c000	7f01130400010 295	23,15,32	208	265	1068
1200	-	5,8,23	02454000	0001130200000 196	8,5,23	256	300	1082
1202	287	2,4,13	a0000b8a	0305130000230 096	287	242	272	1071
1204	-	2,6,17	2005c00c	0f01130503030 096	24,31,47	262	283	1076
1206	-	12,13,22	2005400e	3f01130004050 096	40,37,5	262	284	1078
1208	-	6,10,11	02400880	0001130201030 097	25,9,19	256	306	1083
1210	243	5,7,9	a000098a	03041300001e0 097	243	231	266	1078

1212	203	2,11,15	a000098c	0f04130000190097	203	231	265	1074
1214	257	4,5,9	b00008ae	3f05130000200097	257	249	281	1083
1216	-	11,12,29	0ec44000	0001130401000398	24,9,39	266	169	951
1218	-	10,14,17	22e0000a	0301140202050098	40,16,23	254	169	938
1220	413	5,6,18	62200600	0f01140101000398	28,8,9	260	174	935
1222	-	1,16,18	2005400e	3f01140201010098	8,13,21	254	173	937
1224	-	7,9,12	02444000	0001140002000199	8,17,3	248	165	941
1226	167	9,17,20	a0000b8a	0303140000140099	167	230	168	933
1228	27	1,4,9	a000098c	0f01140000030099	27	206	155	938
1230	-	1,3,15	2004400e	3f01140006020099	16,49,1	254	183	933
1232	-	2,10,25	00044880	000114000103009a	25,10,2	248	168	939
1234	427	6,11,14	2005c00a	030114050404009a	32,39,47	254	154	935
1236	151	4,7,11	a0000b8c	0f0314000012009a	151	230	152	940
1238	153	1,10,12	a000088e	3f0314000013009a	153	230	158	933
1240	-	6,13,23	02a48000	000114000000039b	24,2,7	248	162	933
1242	395	10,13,15	2004c00a	030114000301009b	8,27,3	254	165	931
1244	-	9,19,22	2005c00c	0f0114000404009b	32,39,7	254	156	932
1246	25	5,10,11	a000088e	3f0114000003009b	25	206	141	933
1248	-	3,13,16	0244c000	000114010000029c	16,3,13	248	174	930
1250	-	10,13,15	22e0000a	030114000105009c	40,8,7	254	180	936
1252	-	1,5,9	22e0000c	0f0114000103009c	24,8,7	254	173	935
1254	-	5,16,20	2004400e	3f0114020404009c	32,33,17	254	189	934
1256	-	2,30,31	02200900	000114040104009d	34,10,35	248	160	933
1258	-	4,6,9	2004c00a	030114000104009d	32,11,3	254	158	931
1260	-	3,8,15	62200600	0f0114020400019d	12,32,17	260	166	941
1262	-	2,4,7	22a0000e	3f0114000306009d	48,24,5	254	168	939
1264	-	6,11,17	02600980	000114020002009e	19,3,22	248	163	937
1266	-	7,10,13	2260000a	030114040104009e	32,8,35	254	156	933
1268	-	1,2,12	22e0000c	0f0114050206009e	48,16,47	254	177	931
1270	-	5,15,16	62200700	3f0114060500069e	54,40,49	260	189	931
1272	-	3,9,25	02400880	000114000103009f	25,9,3	248	175	935
1274	-	4,13,15	2260000a	030114000102009f	16,8,3	254	176	938
1276	427	2,11,16	2004400c	0f0114010303009f	24,25,9	254	181	934
1278	385	13,14,22	3ea0002e	3f0114000607009f	56,48,5	259	162	936
1280	-	7,15,17	0e444000	00011403000001a0	8,1,27	252	301	1388
1282	231	6,17,22	a0000b8a	03041500001c00a0	231	242	284	1397
1284	223	3,13,15	a0000b8c	0f041500001b00a0	223	242	278	1398
1286	153	5,8,18	a000088e	3f031500001300a0	153	241	288	1394
1288	-	7,24,33	02c44000	00011500040003a1	24,33,7	267	310	1393
1290	-	3,4,10	22e0000a	0301150104060 0a1	48,32,15	273	300	1396
1292	-	1,14,15	2000098c	0f011502040500a1	43,35,19	273	300	1402
1294	-	2,8,9	2005400e	3f011501050400a1	32,45,13	273	298	1396
1296	-	14,15,21	00048980	00011501020300a2	27,21,13	267	304	1400
1298	399	13,19,20	22e0000a	03011502010300a2	24,8,23	273	311	1328
1300	217	7,11,17	a000088c	0f041500001b00a2	217	242	276	1398
1302	325	1,6,13	a0000a8e	3f061500002800a2	325	266	221	1334
1304	-	15,17,20	02454000	00011503030003a3	24,29,31	267	311	1397
1306	39	7,8,11	a0000b8a	03011500000400a3	39	217	268	1403
1308	-	1,11,16	2260000c	0f011503030400a3	32,24,27	273	304	1398
1310	333	3,11,13	a0000a8e	3f061500002900a3	333	266	297	1398
1312	-	2,21,25	02200900	00011503000300a4	26,2,27	267	311	1393
1314	-	6,11,18	22e0000a	03011504010300a4	24,8,39	273	301	1401
1316	-	11,12,19	2004c00c	0f011501040400a4	32,35,11	273	297	1390
1318	-	7,12,13	20000a8e	3f011501020500a4	45,21,13	273	297	1401
1320	-	8,11,25	02444000	00011501030001a5	8,25,11	267	310	1399
1322	-	8,13,15	20000b8a	03011501030500a5	47,31,15	273	300	1398
1324	337	1,5,12	a000088c	0f061500002a00a5	337	266	297	1396
1326	-	2,9,12	2004c00e	3f011505010100a5	8,11,43	273	294	1396
1328	-	5,17,26	02400880	00011502020400a6	33,17,19	267	312	1395
1330	-	1,6,11	2260000a	03011501020700a6	56,16,11	273	303	1401
1332	95	8,13,16	a0000b8c	0f021500000b00a6	95	230	269	1393

1334	-	5,8,20	2005c00e	3f011500050600a6	48,47,7	273	297	1398
1336	-	2,3,8	02248000	00011500000001a7	8,2,3	267	313	1398
1338	511	4,17,19	2005c00a	03011500010600a7	48,15,7	273	301	1400
1340	189	9,10,16	a0000a8c	0f041500001700a7	189	242	274	1397
1342	-	7,9,15	3220002e	3f011501040500a7	40,32,9	280	308	1399
1344	-	1,6,15	0e444000	00011501060007a8	56,49,11	277	179	1261
1346	-	2,3,6	2260000a	03011602010300a8	24,8,19	286	177	1265
1348	553	11,12,16	2260000c	0f011606020300a8	24,16,51	286	184	1260
1350	-	14,15,17	6004c700	3f011601020001a8	14,19,11	292	182	1265
1352	-	17,27,29	02454000	00011601040005a9	40,37,15	280	177	1257
1354	255	3,4,7	a0000b8a	03051600001f00a9	255	266	155	1262
1356	275	3,6,8	a000098c	0f051600002200a9	275	266	161	1257
1358	-	4,13,20	22a0000e	3f011603010300a9	24,8,29	286	174	1262
1360	-	1,21,23	02200880	00011602010300aa	25,9,18	280	179	1264
1362	655	6,11,24	62e28000	03011603000004aa	34,2,31	292	187	1258
1364	-	3,7,8	22e0000c	0f011602030700aa	56,24,23	286	172	1264
1366	1	21,22,27	a000088e	3f011600000000aa	1	230	147	1250
1368	-	5,9,15	00058880	00011603020000ab	1,23,31	280	182	1260
1370	-	3,5,10	2005c00a	03011601020600ab	48,23,15	286	180	1264
1372	181	5,13,14	a0000a8c	0f031600001600ab	181	254	151	1262
1374	-	2,5,19	2005400e	3f011601060500ab	40,53,13	286	184	1260
1376	-	11,14,23	02444000	00011603040001ac	8,33,27	280	184	1262
1378	-	3,6,7	2260000a	03011605010600ac	48,8,43	286	180	1251
1380	-	11,13,14	2260000c	0f011605010200ac	16,8,43	286	189	1255
1382	-	2,9,10	2220000e	3f011604040500ac	40,32,33	286	182	1256
1384	-	1,5,11	00044900	00011602040300ad	26,35,19	280	193	1259
1386	-	3,8,12	62200500	03011601020002ad	18,16,9	292	188	1258
1388	-	5,8,11	2260000c	0f011604020600ad	48,16,35	286	193	1258
1390	129	11,17,19	a000088e	3f031600001000ad	129	254	159	1256
1392	-	6,17,21	02444000	00011601040001ae	8,33,11	280	180	1259
1394	-	9,13,17	2260000a	03011604020300ae	24,16,35	286	176	1259
1396	339	8,24,29	a000098c	0f061600002a00ae	339	279	168	1258
1398	-	11,12,21	2004400e	3f011605030200ae	16,25,41	286	168	1260
1400	-	6,10,13	00044900	00011601030200af	18,27,11	280	183	1257
1402	127	6,12,15	a0000b8a	03031600000f00af	127	254	162	1258
1404	661	3,7,19	2220000c	0f011604030700af	56,24,33	286	185	1261
1406	-	2,20,23	3004402e	3f011607060400af	32,49,57	287	175	1260
1408	-	10,12,15	0e400880	00011600010400b0	33,9,3	284	318	1391
1410	383	3,8,18	a0000b8a	03071700002f00b0	383	301	320	1401
1412	153	4,7,12	a000088c	0f031700001300b0	153	265	302	1397
1414	429	1,3,13	a0000a8e	3f071700003500b0	429	301	310	1401
1416	-	6,9,10	02c44000	00011700020001b1	8,17,7	299	325	1398
1418	-	2,3,5	2005c00a	03011701040700b1	56,39,15	305	318	1399
1420	-	3,6,12	22e0000c	0f011704010400b1	32,8,39	305	323	1398
1422	-	4,7,13	2220000e	3f011700040700b1	56,32,1	305	318	1406
1424	-	2,6,11	00044900	00011704010000b2	2,11,35	299	321	1405
1426	103	3,5,11	a0000b8a	03021700000c00b2	103	254	282	1404
1428	557	8,12,15	2000098c	0f011700010400b2	35,11,3	305	318	1403
1430	-	1,9,13	2004400e	3f011700050200b2	16,41,1	305	313	1401
1432	-	6,10,11	02200880	00011701010200b3	17,9,10	299	328	1407
1434	-	6,8,11	20000a8a	03011700020600b3	53,21,5	305	318	1406
1436	-	13,17,21	2260000c	0f011702020400b3	32,16,19	305	309	1400
1438	357	22,23,24	a0000a8e	3f061700002c00b3	357	290	301	1404
1440	-	7,13,14	02444000	00011700020003b4	24,17,3	299	324	1403
1442	-	2,4,9	2004c00a	03011703050400b4	32,43,27	305	311	1403
1444	-	10,13,16	2005400c	0f011700020100b4	8,21,5	305	322	1398
1446	-	2,3,17	2004400e	3f011703040500b4	40,33,25	305	316	1401
1448	-	6,22,25	00054880	00011700020300b5	25,22,6	299	323	1404
1450	-	3,12,16	20000b8a	03011700010400b5	39,15,7	305	325	1410
1452	-	5,14,17	20000b8c	0f011701040500b5	47,39,15	305	318	1404
1454	297	2,6,11	a000088e	3f051700002500b5	297	277	217	1400

1456	-	2,4,9	02400880	00011700000400b6	33,1,3	299	330	1405
1458	-	13,19,21	62e00500	03011701030004b6	34,24,15	311	331	1406
1460	-	10,17,18	22e0000c	0f011703010300b6	24,8,31	305	340	1408
1462	-	2,5,13	6004c700	3f011701040003b6	30,35,11	311	338	1404
1464	-	1,14,23	00044b00	00011700020100b7	14,23,7	299	324	1407
1466	311	3,10,11	a0000b8a	03051700002600b7	311	277	299	1401
1468	181	8,11,16	a0000a8c	0f031700001600b7	181	265	286	1406
1470	569	1,5,8	b00008ae	3f091700004700b7	569	319	342	1410
1472	-	1,4,11	0ec00880	00011701040500b8	41,33,15	309	189	1272
1474	-	10,11,17	2005c00a	03011800040400b8	32,39,7	297	179	1272
1476	265	3,8,11	a000088c	0f051800002100b8	265	277	180	1276
1478	69	5,12,25	a0000a8e	3f021800000800b8	69	254	155	1276
1480	-	3,4,17	02c00880	00011802020400b9	33,17,23	291	185	1273
1482	-	1,12,15	22e0000a	03011800010400b9	32,8,7	297	190	1274
1484	-	7,11,15	2004400c	0f011804030100b9	8,25,33	297	186	1272
1486	85	9,10,13	a0000a8e	3f021800000a00b9	85	254	177	1277
1488	-	14,15,21	02c00880	00011801020400ba	33,17,15	291	186	1273
1490	-	8,13,15	2260000a	03011800010500ba	40,8,3	297	214	1280
1492	-	8,13,14	2005400c	0f011805020200ba	16,21,45	297	199	1272
1494	-	9,12,13	62a38000	3f011800000104ba	38,14,5	303	202	1283
1496	-	4,11,13	02c00880	00011801000200bb	17,1,15	291	191	1281
1498	-	10,11,17	22e0000a	03011802010200bb	16,8,23	297	193	1278
1500	-	1,2,14	2004400c	0f011801030300bb	24,25,9	297	204	1278
1502	-	1,5,12	2005400e	3f011800020400bb	32,21,5	297	194	1285
1504	-	2,3,8	02248000	00011800000001bc	8,2,3	291	183	1279
1506	-	10,16,19	2004400a	03011800030100bc	8,25,1	297	200	1267
1508	599	9,15,18	2005c00c	0f011804010100bc	8,15,39	297	183	1279
1510	-	4,6,17	22a0000e	3f011800030400bc	32,24,5	297	207	1278
1512	-	3,25,40	02444000	00011800030005bd	40,25,3	291	186	1288
1514	299	6,7,16	a000098a	03051800002500bd	299	277	207	1273
1516	-	2,16,17	2220000c	0f011801020300bd	24,16,9	297	193	1281
1518	-	1,8,14	2220000e	3f011803010200bd	16,8,25	297	231	1278
1520	-	3,10,13	02c00880	00011800030500be	41,25,7	291	189	1279
1522	-	10,11,18	2005c00a	03011802050600be	48,47,23	297	225	1277
1524	293	3,4,7	a0000a8c	0f051800002400be	293	277	184	1276
1526	473	5,8,18	2005400e	3f011804020200be	16,21,37	297	224	1274
1528	-	7,17,18	02200900	00011803020500bf	42,18,27	291	186	1282
1530	-	3,19,28	22e0000a	03011803050700bf	56,40,31	297	229	1276
1532	-	13,20,22	2005400c	0f011801020500bf	40,21,13	297	184	1278
1534	225	4,5,20	b00008ae	3f041800001c00bf	225	267	191	1277
1536	-	2,6,21	0ec44000	00011801030002c0	16,25,15	295	329	1779
1538	-	1,2,6	2005c00a	03011900010100c0	8,15,7	316	277	1701
1540	-	1,3,22	22a0000c	0f011904020500c0	40,16,37	316	266	1777
1542	365	2,14,15	a0000a8e	3f061900002d00c0	365	301	308	1780
1544	-	1,7,11	02600980	00011901000100c1	11,3,14	310	262	1775
1546	-	5,8,10	22e0000a	03011906030400c1	32,24,55	316	227	1775
1548	505	3,20,21	a000088c	0f081900003f00c1	505	312	283	1726
1550	189	3,8,15	a0000a8e	3f041900001700c1	189	277	301	1770
1552	-	2,12,19	00048a80	00011901020300c2	29,23,15	310	342	1765
1554	-	1,7,14	2005c00a	03011901050300c2	24,47,15	316	318	1775
1556	-	5,9,17	2260000c	0f011905020700c2	56,16,43	316	337	1778
1558	-	2,7,11	2004c00e	3f011901060200c2	16,51,11	316	313	1768
1560	-	10,12,31	02c44000	00011902000004c3	32,1,23	310	335	1768
1562	95	2,7,10	a0000b8a	03021900000b00c3	95	265	243	1771
1564	7	4,13,18	a0000b8c	0f011900000000c3	7	252	218	1776
1566	497	4,13,18	a000088e	3f081900003e00c3	497	312	273	1771
1568	-	7,10,21	02400880	00011901020400c4	33,17,11	310	334	1777
1570	-	3,10,16	62600500	03011900020001c4	10,16,3	322	334	1778
1572	-	2,8,9	22a0000c	0f011901010400c4	32,8,13	316	328	1788
1574	-	5,8,9	2005400e	3f011902030300c4	24,29,21	316	327	1770
1576	-	6,15,24	02258000	00011901000003c5	24,6,15	310	351	1775

1578	731	6,9,16	20000b8a	03011900010300c5	31,15,7	316	323	1773
1580	647	3,7,8	2004c00c	0f011901020200c5	16,19,11	316	316	1772
1582	121	13,20,24	a000088e	3f021900000f00c5	121	265	284	1771
1584	-	10,14,25	02c44000	00011904050006c6	48,41,39	310	330	1771
1586	-	5,9,11	62628000	03011903000305c6	42,26,27	322	329	1775
1588	159	7,13,18	a0000b8c	0f031900001300c6	159	276	306	1782
1590	169	7,8,13	a000088e	3f031900001500c6	169	276	294	1775
1592	-	5,7,12	02400a80	00011902010200c7	21,13,23	310	340	1770
1594	-	1,7,10	2005c00a	03011901020500c7	40,23,15	316	328	1783
1596	697	9,24,25	2004400c	0f011901030300c7	24,25,9	316	330	1772
1598	-	3,7,13	70045740	3f011903040005c7	46,33,25	333	333	1774
1600	-	5,7,20	0e444000	00011900010005c8	40,9,3	320	200	1637
1602	-	1,2,10	22e0000a	03011a00040500c8	40,32,7	329	206	1630
1604	-	8,12,17	2005400c	0f011a00020100c8	8,21,5	329	198	1631
1606	577	2,14,15	2220000e	3f011a06030400c8	32,24,49	329	224	1632
1608	-	8,29,31	02454000	00011a03030001c9	8,29,31	323	206	1626
1610	-	3,5,11	2004400a	03011a07020100c9	8,17,57	329	206	1633
1612	771	1,6,11	2000088c	0f011a00010300c9	25,9,1	329	196	1628
1614	-	3,9,17	22a0000e	3f011a04020400c9	32,16,37	329	203	1626
1616	-	2,3,15	00048980	00011a03020100ca	11,21,29	323	218	1627
1618	211	1,5,6	a000098a	03041a00001a00ca	211	290	172	1629
1620	227	3,8,9	a000098c	0f041a00001c00ca	227	290	182	1640
1622	-	10,14,15	2000098e	3f011a01030500ca	43,27,11	329	213	1631
1624	-	5,11,13	00048980	00011a00010100cb	11,13,5	323	218	1633
1626	-	1,5,11	20000a8a	03011a00020300cb	29,21,5	329	189	1626
1628	603	2,19,25	2005c00c	0f011a00050300cb	24,47,7	329	196	1624
1630	741	17,20,24	62a38000	3f011a00000304cb	38,30,5	335	204	1640
1632	-	7,17,20	02a48000	00011a00000005cc	40,2,7	323	205	1631
1634	683	12,14,21	22e0000a	03011a04040500cc	40,32,39	329	183	1620
1636	37	2,16,17	a0000a8c	0f011a00000400cc	37	265	185	1638
1638	397	8,13,17	a0000a8e	3f071a00003100cc	397	325	198	1633
1640	-	9,17,18	02200880	00011a02010200cd	17,9,18	323	206	1635
1642	231	1,11,18	a0000b8a	03041a00001c00cd	231	290	175	1630
1644	-	1,9,18	2000088c	0f011a00020300cd	25,17,1	329	205	1632
1646	-	1,5,16	2220000e	3f011a01020400cd	32,16,9	329	229	1631
1648	-	1,13,18	00048880	00011a00020100ce	9,19,3	323	200	1628
1650	727	14,19,25	62628000	03011a03000005ce	42,2,27	335	212	1634
1652	-	3,10,19	22a0000c	0f011a03020300ce	24,16,29	329	199	1631
1654	-	5,9,16	2004400e	3f011a02050500ce	40,41,17	329	228	1630
1656	-	17,35,38	0244c000	00011a02020005cf	40,19,21	323	205	1640
1658	-	5,6,17	62600500	03011a03010000cf	2,8,27	335	219	1643
1660	37	1,4,18	a0000a8c	0f011a00000400cf	37	265	165	1642
1662	-	7,13,21	3220002e	3f011a02060700cf	56,48,17	330	214	1632
1664	-	6,9,17	0ea00880	00011a00010200d0	17,9,6	327	349	1783
1666	-	4,12,15	2005c00a	03011b00020400d0	32,23,7	348	339	1779
1668	-	3,8,20	2260000c	0f011b03030400d0	32,24,27	348	348	1782
1670	-	2,5,6	6004c700	3f011b00030000d0	6,27,3	354	355	1780
1672	-	1,4,19	02444000	00011b02010002d1	16,9,19	342	353	1781
1674	755	8,11,15	2000088a	03011b02030500d1	41,25,17	348	358	1786
1676	363	5,12,14	a000098c	0f061b00002d00d1	363	325	328	1780
1678	129	3,15,18	a000088e	3f031b00001000d1	129	300	316	1782
1680	-	2,6,11	02254000	00011b01040001d2	8,37,14	342	366	1785
1682	-	8,12,19	2004400a	03011b07040400d2	32,33,57	348	344	1775
1684	-	3,4,10	2005c00c	0f011b00020300d2	24,23,7	348	343	1777
1686	-	5,22,23	2220000e	3f011b07020500d2	40,16,57	348	345	1780
1688	-	6,8,15	02258000	00011b01000001d3	8,6,15	342	313	1781
1690	-	5,11,16	2005c00a	03011b03060500d3	40,55,31	348	347	1785
1692	-	15,16,22	2220000c	0f011b02030700d3	56,24,17	348	340	1780
1694	-	2,5,8	60044700	3f011b04020001d3	14,17,33	354	357	1781
1696	-	2,5,31	02444000	00011b01000004d4	32,1,11	342	367	1783
1698	767	4,10,11	2004400a	03011b01060600d4	48,49,9	348	346	1783

1700	311	8,9,17	a0000b8c	0f051b00002600d4	311	312	318	1780
1702	-	5,22,24	2004c00e	3f011b05010100d4	8,11,43	348	336	1779
1704	-	3,11,14	02600980	00011b01000100d5	11,3,14	342	355	1786
1706	-	6,17,19	22e0000a	03011b04010600d5	48,8,39	348	345	1781
1708	93	4,12,15	a0000a8c	0f021b00000b00d5	93	289	305	1785
1710	-	2,4,17	2000098e	3f011b00030500d5	43,27,3	348	357	1788
1712	-	11,19,21	02400980	00011b02010200d6	19,11,21	342	355	1779
1714	-	4,6,7	2000088a	03011b01030600d6	49,25,9	348	340	1783
1716	-	1,5,13	22e0000c	0f011b00030500d6	40,24,7	348	350	1781
1718	-	9,14,17	2005400e	3f011b01040400d6	32,37,13	348	340	1780
1720	-	10,14,17	02454000	00011b04000001d7	8,5,39	342	351	1783
1722	839	9,17,20	2005c00a	03011b01050300d7	24,47,15	348	336	1786
1724	-	1,7,17	2260000c	0f011b00010500d7	40,8,3	348	341	1780
1726	-	2,5,13	3220002e	3f011b06030700d7	56,24,49	355	355	1784
1728	-	5,10,11	0ec44000	00011b02020001d8	8,17,23	352	222	1683
1730	-	1,13,15	2004c00a	03011c01060500d8	40,51,11	340	215	1642
1732	-	5,17,21	22a0000c	0f011c02040500d8	40,32,21	340	207	1646
1734	-	1,2,19	2005400e	3f011c02030400d8	32,29,21	340	209	1653
1736	-	20,21,27	02c00880	00011c03000400d9	33,1,31	334	224	1643
1738	-	1,2,8	22e0000a	03011c01030600d9	48,24,15	340	211	1645
1740	-	5,11,16	2260000c	0f011c04010400d9	32,8,35	340	223	1643
1742	869	5,9,15	22a0000e	3f011c04010600d9	48,8,37	340	212	1646
1744	-	9,12,31	00044b00	00011c03010100da	14,15,31	334	221	1639
1746	-	20,24,27	2004c00a	03011c04030300da	24,27,35	340	195	1639
1748	-	1,7,8	2004400c	0f011c05040100da	8,33,41	340	215	1640
1750	457	2,14,15	a000088e	3f081c00003900da	457	336	214	1636
1752	-	8,18,27	02248000	00011c03020001db	8,18,27	334	211	1649
1754	-	6,17,19	2260000a	03011c01030500db	40,24,11	340	217	1653
1756	99	2,5,6	a000098c	0f021c00000c00db	99	289	185	1652
1758	-	12,15,17	2005400e	3f011c00030600db	48,29,5	340	232	1650
1760	-	28,33,35	02444000	00011c02070001dc	8,57,19	334	214	1643
1762	-	17,21,25	2260000a	03011c00010700dc	56,8,3	340	214	1641
1764	-	8,19,22	2004400c	0f011c04020100dc	8,17,33	340	207	1643
1766	-	2,3,10	62a00700	3f011c03040003dc	30,32,29	346	219	1642
1768	-	10,24,25	02244000	00011c01030003dd	24,25,10	334	218	1640
1770	607	8,19,26	22e0000a	03011c01020600dd	48,16,15	340	214	1655
1772	-	8,16,21	22a0000c	0f011c02010200dd	16,8,21	340	207	1644
1774	-	4,9,13	2220000e	3f011c00010300dd	24,8,1	340	213	1642
1776	-	13,15,24	02454000	00011c01010003de	24,13,15	334	215	1646
1778	815	8,15,16	22e0000a	03011c01010200de	16,8,15	340	217	1663
1780	457	5,6,29	a000088c	0f081c00003900de	457	336	232	1642
1782	-	3,6,20	60044700	3f011c00030001de	14,25,1	346	205	1649
1784	-	6,21,23	02444000	00011c00040001df	8,33,3	334	227	1659
1786	-	6,13,20	2260000a	03011c05020500df	40,16,43	340	198	1639
1788	-	3,15,22	2004c00c	0f011c01020500df	40,19,11	340	209	1647
1790	-	13,16,17	3220002e	3f011c01040600df	48,32,9	341	217	1651
1792	-	21,28,31	0e3c8000	00011c04010004e0	32,10,35	342	360	2203
1794	-	2,8,25	22e0000a	03011d05020500e0	40,16,47	359	378	2234
1796	-	5,9,12	22a0000c	0f011d00030400e0	32,24,5	359	372	2236
1798	685	18,21,22	2005c00e	3f011d00050700e0	56,47,7	359	392	2236
1800	-	35,49,50	02a00900	00011d06030700e1	58,26,55	353	396	2228
1802	843	7,17,18	2260000a	03011d05050600e1	48,40,43	359	381	2237
1804	747	2,8,13	20000a8c	0f011d00010200e1	21,13,5	359	378	2241
1806	101	8,10,23	a0000a8e	3f021d00000c00e1	101	300	333	2234
1808	-	13,23,27	02c44000	00011d01000004e2	32,1,15	353	393	2191
1810	171	3,5,14	a000098a	03031d00001500e2	171	311	343	2237
1812	-	8,12,23	2260000c	0f011d03010600e2	48,8,27	359	382	2232
1814	545	2,5,6	a000088e	3f091d00004400e2	545	358	406	2240
1816	-	1,18,23	02c44000	00011d00010003e3	24,9,7	353	390	2233
1818	871	13,15,16	2004c00a	03011d01030400e3	32,27,11	359	380	2233
1820	359	9,19,20	a0000b8c	0f061d00002c00e3	359	336	350	2232

1822	-	14,15,19	2005400e	3f011d01020500e3	40,21,13	359	366	2238
1824	-	13,19,28	02a00900	00011d05000400e4	34,2,47	353	397	2238
1826	-	6,18,23	22e0000a	03011d00020600e4	48,16,7	359	372	2234
1828	343	10,13,19	a0000b8c	0f061d00002a00e4	343	336	366	2240
1830	-	1,11,20	2000098e	3f011d04060700e4	59,51,35	359	368	2237
1832	-	7,13,18	02444000	00011d02020005e5	40,17,19	353	390	2236
1834	-	7,11,15	2260000a	03011d01020300e5	24,16,11	359	370	2238
1836	-	8,19,20	2000098c	0f011d00020500e5	43,19,3	359	373	2242
1838	-	2,5,6	2004400e	3f011d01040600e5	48,33,9	359	394	2242
1840	-	2,10,21	02600900	00011d02000100e6	10,2,21	353	392	2230
1842	-	7,10,12	2260000a	03011d02030600e6	48,24,19	359	371	2246
1844	375	4,24,25	a0000b8c	0f061d00002e00e6	375	336	352	2234
1846	-	2,9,10	2005400e	3f011d03020100e6	8,21,29	359	372	2243
1848	-	22,23,29	00048a80	00011d00060300e7	29,55,7	353	397	2236
1850	-	3,5,18	22e0000a	03011d04020400e7	32,16,39	359	379	2232
1852	-	6,13,16	2000098c	0f011d02030500e7	43,27,19	359	375	2238
1854	-	5,7,14	3220002e	3f011d01020600e7	48,16,9	366	390	2240
1856	-	5,15,19	0ec44000	00011d02040003e8	24,33,23	363	240	2064
1858	-	7,15,16	2005c00a	03011e00010200e8	16,15,7	372	248	2078
1860	761	6,19,22	2000088c	0f011e03040600e8	49,33,25	372	291	2065
1862	149	10,18,23	a0000a8e	3f031e00001200e8	149	324	215	2063
1864	-	1,9,11	02400880	00011e01000100e9	9,1,11	366	251	2071
1866	-	3,5,18	2004c00a	03011e05040100e9	8,35,43	372	234	2074
1868	-	5,11,14	2220000c	0f011e00010600e9	48,8,1	372	255	2069
1870	-	3,4,6	62e00700	3f011e02040005e9	46,32,23	378	275	2071
1872	-	6,21,22	00044a80	00011e00020200ea	21,22,6	366	244	2067
1874	-	3,4,20	2005c00a	03011e01020700ea	56,23,15	372	247	2083
1876	-	10,19,24	22a0000c	0f011e01020500ea	40,16,13	372	242	2067
1878	253	2,5,6	a0000a8e	3f051e00001f00ea	253	336	227	2074
1880	-	2,4,15	02454000	00011e03010004eb	32,13,31	366	240	2073
1882	-	1,6,9	60054500	03011e05030001eb	10,29,45	378	253	2065
1884	-	10,19,25	2004400c	0f011e01020400eb	32,17,9	372	245	2067
1886	-	2,8,17	2220000e	3f011e04030600eb	48,24,33	372	245	2074
1888	-	10,15,21	0004c900	00011e00030200ec	18,29,5	366	245	2075
1890	-	2,7,19	22e0000a	03011e00050600ec	48,40,7	372	243	2063
1892	-	2,14,17	22e0000c	0f011e00010300ec	24,8,7	372	261	2082
1894	-	9,10,14	2220000e	3f011e05010300ec	24,8,41	372	252	2062
1896	-	2,11,26	02200900	00011e01000300ed	26,2,11	366	245	2063
1898	-	1,15,23	2005c00a	03011e00010400ed	32,15,7	372	244	2082
1900	297	11,13,18	a000088c	0f051e00002500ed	297	336	214	2067
1902	733	11,22,23	22a0000e	3f011e02030400ed	32,24,21	372	247	2071
1904	-	10,18,19	02200900	00011e02010200ee	18,10,19	366	237	2067
1906	931	6,9,10	22e0000a	03011e03030700ee	56,24,31	372	269	2085
1908	-	4,7,8	2220000c	0f011e00040700ee	56,32,1	372	261	2068
1910	-	11,13,18	20000b8e	3f011e00010300ee	31,15,7	372	250	2070
1912	-	5,11,30	02444000	00011e04030001ef	8,25,35	366	247	2067
1914	-	5,10,14	2005400a	03011e00030700ef	56,29,5	372	268	2071
1916	-	12,16,19	62600600	0f011e02020001ef	12,16,19	378	237	2083
1918	705	5,6,19	3004402e	3f011e06040200ef	16,33,49	373	235	2083
1920	-	1,20,23	0e400880	00011e00060700f0	57,49,3	370	357	2198
1922	-	15,16,23	2005c00a	03011f01020200f0	16,23,15	391	349	2211
1924	-	1,9,20	2000098c	0f011f02030400f0	35,27,19	391	361	2214
1926	-	8,13,35	60044700	3f011f00020004f0	38,17,1	397	387	2210
1928	-	7,16,17	02c44000	00011f00020002f1	16,17,7	385	373	2208
1930	-	17,20,25	60054500	03011f00030002f1	18,29,5	397	375	2207
1932	277	3,6,8	a0000a8c	0f051f00002200f1	277	347	342	2208
1934	413	1,4,12	a0000a8e	3f071f00003300f1	413	371	360	2207
1936	-	10,18,19	02200900	00011f02010200f2	18,10,19	385	371	2207
1938	-	6,7,22	2004c00a	03011f04020200f2	16,19,35	391	353	2211
1940	113	3,11,27	a000088c	0f021f00000e00f2	113	324	323	2204
1942	-	5,10,17	2000098e	3f011f00010500f2	43,11,3	391	369	2213

1944	-	18,22,27	02200880	00011f03030600f3	49,25,26	385	362	2216
1946	51	13,18,23	a000098a	03011f00000600f3	51	311	322	2208
1948	603	9,13,24	2004400c	0f011f03010100f3	8,9,25	391	353	2217
1950	-	9,13,21	2220000e	3f011f04030400f3	32,24,33	391	358	2223
1952	-	6,14,15	02200b00	00011f01000100f4	14,6,15	385	391	2228
1954	-	2,11,16	2005c00a	03011f06040200f4	16,39,55	391	291	2210
1956	-	5,8,13	2005400c	0f011f00010100f4	8,13,5	391	304	2207
1958	-	1,5,7	2220000e	3f011f02030600f4	48,24,17	391	360	2205
1960	-	6,22,25	00054880	00011f00020300f5	25,22,6	385	366	2211
1962	-	3,9,10	2004c00a	03011f00030500f5	40,27,3	391	358	2210
1964	21	10,16,17	a0000a8c	0f011f00000200f5	21	311	322	2214
1966	-	10,15,24	2220000e	3f011f00020300f5	24,16,1	391	378	2211
1968	-	2,15,30	02248000	00011f03020005f6	40,18,27	385	383	2212
1970	-	2,7,10	2260000a	03011f02020300f6	24,16,19	391	369	2209
1972	-	1,11,12	62e30000	0f011f02000203f6	28,20,23	397	392	2216
1974	745	7,9,12	22a0000e	3f011f00050700f6	56,40,5	391	363	2214
1976	-	17,19,25	02400880	00011f02020300f7	25,17,19	385	365	2209
1978	-	6,8,11	2004400a	03011f06030200f7	16,25,49	391	350	2211
1980	-	5,7,26	22a0000c	0f011f03010600f7	48,8,29	391	300	2211
1982	-	2,9,14	7004d740	3f011f00010002f7	22,11,3	408	385	2206
1984	-	5,11,13	0ec44000	00011f04050002f8	16,41,39	395	241	2081
1986	-	4,10,21	2004c00a	03012000030200f8	16,27,3	383	258	2078
1988	-	1,2,10	2220000c	0f012000010500f8	40,8,1	383	237	2088
1990	217	1,6,9	a000088e	3f042000001b00f8	217	336	254	2075
1992	-	2,3,6	02200900	00012001000300f9	26,2,11	377	252	2089
1994	15	3,12,15	a0000b8a	03012000000100f9	15	311	205	2079
1996	-	15,18,21	20000a8c	0f012000030400f9	37,29,5	383	237	2081
1998	-	2,12,15	62a00700	3f012003020004f9	38,16,29	389	262	2109
2000	-	12,19,41	0004c900	00012006030000fa	2,29,53	377	236	2080
2002	-	11,21,22	22e0000a	03012001010500fa	40,8,15	383	240	2080
2004	-	7,8,14	2220000c	0f012000010200fa	16,8,1	383	240	2100
2006	917	5,7,13	62a00700	3f012004020004fa	38,16,37	389	285	2104
2008	-	2,15,32	02a48000	00012001000004fb	32,2,15	377	236	2089
2010	-	9,14,17	2005400a	03012001060200fb	16,53,13	383	267	2079
2012	-	3,19,23	2220000c	0f012001040500fb	40,32,9	383	252	2083
2014	-	6,7,9	20000b8e	3f012000030400fb	39,31,7	383	256	2105
2016	-	2,22,23	02a44000	00022004080002fc	16,65,38	390	244	2083
2018	-	3,7,20	2260000a	03012002020300fc	24,16,19	383	272	2099
2020	-	5,9,19	2004400c	0f012003050600fc	48,41,25	383	248	2086
2022	349	4,17,20	a0000a8e	3f062000002b00fc	349	360	238	2077
2024	-	15,34,41	0244c000	00012005020004fd	32,19,45	377	251	2088
2026	-	3,10,18	2260000a	03012004010300fd	24,8,35	383	255	2079
2028	301	8,9,17	a0000a8c	0f052000002500fd	301	347	216	2080
2030	-	3,7,19	60054700	3f012000010003fd	30,13,5	389	280	2088
2032	-	2,6,7	02454000	00012003000004fe	32,5,31	377	242	2084
2034	-	1,11,16	2005c00a	03012003020200fe	16,23,31	383	279	2085
2036	71	4,19,24	a0000b8c	0f022000000800fe	71	324	191	2087
2038	-	6,11,16	2220000e	3f012002040600fe	48,32,17	383	280	2082
2040	-	17,30,33	02a00880	00012003020400ff	33,17,30	377	240	2084
2042	-	7,8,16	22e0000a	03012000010200ff	16,8,7	383	269	2083
2044	45	7,13,15	a0000a8c	0f012000000500ff	45	311	209	2101
2046	-	1,15,16	3220002e	3f022007060a00ff	80,48,57	397	244	2096