```
$LL74@main:
a1 00 00 00 00    mov     eax, DWORD PTR _Patternlen
03 c2             add     eax, edx
8b 04 30          mov     eax, DWORD PTR [eax+esi]
25 ff ff ff 00    and     eax, 16777215
0f af c7          imul    eax, edi
8b 1c 30          mov     ebx, DWORD PTR [eax+esi]
89 19             mov     DWORD PTR [ecx], ebx
8b 44 30 04       mov     eax, DWORD PTR [eax+esi+4]
89 41 04          mov     DWORD PTR [ecx+4], eax
83 c2 03          add     edx, 3
03 cf             add     ecx, edi
ff 4c 24 1c       dec     DWORD PTR tv977[esp+376]
75 d7             jne     SHORT $LL74@main
```

Note2: The niftiness lies in readiness for multi-threading and mostly in boosting the search by having the BBs.
The decompressor would upload (at burst speed) the BB data and then read-and-decode one-by-one the triads (BB pool/array indexes), that is a simple copying. The pool houses up to 256*256*256 BBs/elements.

Note7: My benchmark text file OSHO.TXT 206,908,949 bytes where OSHO.TXT.SS 116,871,584 bytes for order 6:
Decompressing OSHO.TXT.SS to RAM without Dumping to DRIVE time: 1704 clocks or 118579 KB/s, an awful result.
For order 4 enforced: 156,174,067 OSHO.TXT.SS is being decompressed at 192804 KB/s.
For order 7 enforced: 122,297,608 OSHO.TXT.SS is being decompressed at 85618 KB/s.
For order 8 enforced: 149,243,106 OSHO.TXT.SS is being decompressed at 115396 KB/s.
Obviously the fastest cache size is crucial, for OSHO.TXT 12MB BB pool vs 1MB L2 cache disbalance is the cause for this badly inferior performance compared to LZ L1 (32KB) cache-friendly variants.
The testing machine is Toshiba Satellite with Intel Merom 2166MHz.

Note9: Major (but still inferior) decompressing tweak since r.1+++, this time Microsoft v16 excels:
For order 7 enforced: 122,297,608 OSHO.TXT.SS is being decompressed at 170083 KB/s.

SIMPLICIUS SIMPLICISSIMUS

A 32/64 BIT BUILDING-BLOCKS TEXT DECOMPRESSOR, REVISION 2

Free download at www.sanmayce.com – on Intel T7500 2200 MHz it decompresses **OSHO.TXT.SS** at 207MB/s.

```
$LL74@main:
a1 00 00 00 00    mov     eax, DWORD PTR _Patternlen
03 c2             add     eax, edx
8b 04 30          mov     eax, DWORD PTR [eax+esi]
25 ff ff ff 00    and     eax, 16777215
0f af c7          imul    eax, edi
8b 1c 30          mov     ebx, DWORD PTR [eax+esi]
89 19             mov     DWORD PTR [ecx], ebx
8b 44 30 04       mov     eax, DWORD PTR [eax+esi+4]
89 41 04          mov     DWORD PTR [ecx+4], eax
83 c2 03          add     edx, 3
03 cf             add     ecx, edi
ff 4c 24 1c       dec     DWORD PTR tv977[esp+376]
75 d7             jne     SHORT $LL74@main
```

Note2: The niftiness lies in readiness for multi-threading and mostly in boosting the search by having the BBs.
The decompressor would upload (at burst speed) the BB data and then read-and-decode one-by-one the triads (BB pool/array indexes), that is a simple copying. The pool houses up to 256*256*256 BBs/elements.

Note7: My benchmark text file OSHO.TXT 206,908,949 bytes where OSHO.TXT.SS 116,871,584 bytes for order 6:
Decompressing OSHO.TXT.SS to RAM without Dumping to DRIVE time: 1704 clocks or 118579 KB/s, an awful result.
For order 4 enforced: 156,174,067 OSHO.TXT.SS is being decompressed at 192804 KB/s.
For order 7 enforced: 122,297,608 OSHO.TXT.SS is being decompressed at 85618 KB/s.
For order 8 enforced: 149,243,106 OSHO.TXT.SS is being decompressed at 115396 KB/s.
Obviously the fastest cache size is crucial, for OSHO.TXT 12MB BB pool vs 1MB L2 cache disbalance is the cause for this badly inferior performance compared to LZ L1 (32KB) cache-friendly variants.
The testing machine is Toshiba Satellite with Intel Merom 2166MHz.

Note9: Major (but still inferior) decompressing tweak since r.1+++, this time Microsoft v16 excels:
For order 7 enforced: 122,297,608 OSHO.TXT.SS is being decompressed at 170083 KB/s.

SIMPLICIUS SIMPLICISSIMUS

A 32/64 BIT BUILDING-BLOCKS TEXT DECOMPRESSOR, REVISION 2

Free download at www.sanmayce.com – on Intel T7500 2200 MHz it decompresses **OSHO.TXT.SS** at 207MB/s.