

1. int32_t a, b ; a = 100 ; LDR R0,=100 STR R0,a	2. int32_t a, b ; a = b ; LDR R0, b STR R0,a
3. int32_t a, b ; a = b + 5 ; LDR R0,b ADD R0,R0,#5 STR R0,a	4. int32_t a ; int16_t b ; a = b ; LDRSH R0,b STR R0,a
5. int8_t a, *p ; p = &a ; ADR R0,a STR R0,p	6. int8_t a[100] ; int32_t k ; a[k] = 0 ; LDR R0,=0 ADR R1,a LDR R2,k STRB R0,[R1,R2]
7. int32_t a[100] ; int32_t k ; a[k] = 0 ; LDR R0,=0 ADR R1,a LDR R2,k STR R0,[R1,R2,LSL #2]	8. int32_t a, b, c ; c = a / b ; LDR R0,a LDR R1,b SDIV R0,R0,R1 STR R0,c
9. int32_t a, b, c ; c = a % b ; LDR R0,a LDR R1,b SDIV R2,R0,R1 MLS R0,R2,R1,R0 STR R0,c	10. uint32_t a, b, c ; c = a / b ; LDR R0,a LDR R1,b UDIV R0,R0,R1 STR R0,c

<p>11. int32_t a, b, c ;</p> <p>c = a * b ;</p> <pre> LDR R0,a LDR R1,b MUL R0,R0,R1 STR R0,c </pre>	<p>12. uint32_t a, b, c ;</p> <p>c = a * b ;</p> <pre> LDR R0,a LDR R1,b MUL R0,R0,R1 STR R0,c </pre>
<p>13. int16_t a ;</p> <p>a = a << 4 ;</p> <pre> LDRSH R0,a LSL R0,R0,#4 STRH R0,a </pre>	<p>14. int64_t a ;</p> <p>a = a << 4 ;</p> <pre> LDRD R0,R1,a LSL R1,R1,#4 ORR R1,R1,R0,LSR #28 LSL R0,R0,#4 STRD R0,R1,a </pre>
<p>15. int64_t a, b ;</p> <p>a = a + b ;</p> <pre> LDRD R0,R1,a LDRD R2,R3,b ADDS R0,R0,R2 ADC R1,R1,R3 STRD R0,R1,a </pre>	<p>16. int64_t a ;</p> <p>a -= 5 ;</p> <pre> LDRD R0,R1,a SUBS R0,R0,#5 SBC R1,R1,#0 STRD R0,R1,a </pre>
<p>17. int32_t a ; int16_t b ;</p> <p>b = a ;</p> <pre> LDR r0,a STRH r0,b </pre>	<p>18. int32_t a ; int16_t b ;</p> <p>a = b ;</p> <pre> LDRSH r0,b STR r0,a </pre>

<p>19. int32_t a ; int32_t k ;</p> <p>*(&a + k) = 0 ;</p> <pre> LDR r0,=0 ADR r1,a LDR r2,k STR r0,[r1,r2,LSL #2] </pre>	<p>20. int16_t *p ; int32_t k ;</p> <p>p[k] = 0 ;</p> <pre> LDR r0,=0 LDR r1,p LDR r2,k STRH r0,[r1,r2,LSL #1] </pre>
<p>21. int32_t a, k ;</p> <p>((int8_t *) &a)[k] = 0 ;</p> <pre> LDR r0,=0 ADR r1,a LDR r2,k STRB r0,[r1,r2] </pre>	<p>22. int16_t a, *p ;</p> <p>*p = a ;</p> <pre> LDRH r0,a LDR r1,p STRH r0,[r1] </pre>
<p>23. int8_t c, **p ;</p> <p>**p = c ;</p> <pre> LDRB r0,c LDR r1,p LDR r1,[r1] STRB r0,[r1] </pre>	<p>24. int32_t a, b ; int64_t c ;</p> <p>c = a * b ;</p> <pre> LDR r0,a LDR r1,b MUL r0,r0,r1 MOV r1,r0,ASR #31 STRD r0,r1,c </pre>
<p>25. uint16_t *pu16, u16 ;</p> <p>*pu16 = u16 ;</p> <pre> LDRH R0,u16 LDR R1,pu16 STRH R0,[R1] </pre>	<p>26. uint8_t **ppu8, u8 ;</p> <p>**ppu8 = u8 ;</p> <pre> LDRB R0,u8 LDR R1,ppu8 LDR R1,[R1] STRB R0,[R1] </pre>

<p>27. uint16_t **ppu16, u16 ;</p> <p>*(*ppu16 + 1) = u16 ;</p> <pre>LDRH R0,a16 LDR R1,pu16 LDR R1,[R1] STRH R0,[R1,#2]</pre>	<p>28. int32_t s32 ;</p> <p>int64_t s64 ;</p> <p>s64 = s32 ;</p> <pre>LDR R0,s32 ASR R1,R0,#31 STRD R0,R1,s64</pre>
<p>29. int32_t a32, b32 ;</p> <p>int64_t c64 ;</p> <p>c64 = a32*b32 ;</p> <pre>LDR R0,a32 LDR R1,b32 MUL R0,R0,R1 ASR R1,R0,#31 STRD R0,R1,c64</pre>	<p>30. int32_t a32, b32 ;</p> <p>a32 = *&b32 ;</p> <pre>LDR R0,b32 STR R0,a32</pre>
<p>31. int32_t a32[100], k32, *p32 ;</p> <p>p32 = &a32[k32] ;</p> <pre>ADR R0,a32 LDR R1,k32 ADD R0,R0,R1,LSL #2 STR R0,p32</pre>	<p>32. int32_t a32[100], k32, j32 ;</p> <p>a32[j32+1] = k32 ;</p> <pre>LDR R0,k32 ADR R1,a32 LDR R2,j32 ADD R2,R2,#1 STR R0,[R1,R2,LSL #2]</pre>

33. int32_t a32[100], k32 ;
int16_t b16[100] ;

a32[k32] = b16[k32] ;

```
LDR    R0,k32
ADR    R1,b16
LDRSH  R2,[R1,R0,LSL #1]
ADR    R1,a32
STR    R2,[R1,R0,LSL #2]
```

34. int32_t a32[100], k32 ;
int16_t b16[100] ;

a32[b16[k32]] = 0 ;

```
LDR    R0,=0
ADR    R1,b16
LDR    R2,k32
LDRSH  R2,[R1,R2,LSL #1]
ADR    R1,a32
STR    R0,[R1,R2,LSL #2]
```