1.

a. uint32_t green;
   green = green+1
   uint64_t violet;
   violet = 0+green
b. uint6_t red;
   uint8_t blue;
   uint8_t purple;
   red = blue + (purple * 2)
c. uint8_t black;
   int8_t white
   black = black + (R1 * 4)
   white = 0;
   white = black;

2.

a. LDR R0, =0
   STRB R0, u8
b. LDR R0, =0
   STRH R0, u16
c. LDR R0, =0
   STR R0, u32
d. LDR R0, =0
   STRD R0,R1, u64
e. LDRB R0, u8
   STRH R0,u16
f. LDRB R0, u8
   STR R0,u32
g. LDRH R0, u16
   STR R0,u32
h. LDR R0, u32
   LDR R1, =0

STRD R0, R1, u64

i. LDR R0, u32
   STRB R0,u8
j. LDRH R0, u16
   STRB R0,u8
k. LDR R0, u32
   STRH R0,u16

5.

a. LDR R0, =a64;
   LDR R1, =k;
   LDR R1, [R1];
   LDR R2, [R0, R1, LSL 2];
   LDR R3, =k32;
   LDR R3, [R3];
   LDR R4, =p32;
   LDR R5, [R4];
   LDR R4, [R4];
   LDR R4, [R2+1];
b. LDR R0, =pp16;
   LDR R0, [R0];
   LDR R0, [R0];
   LDR R0, [R0];
   LDR R0, [R0, R0, LSL 0];
   MOV R0, 0;
c. LDR Ro, =k32;
   LDR R0, [R0];
   LDR R1, =a32;
   LDR R2, =k;
   LDR R2, [R2];
   LDR R3, [R0, R2, LSL 2];

LDR R4, =p32;

LDr R5, [R4];

SUB R0, R1, R4;

d. LDR R0, =a16;

LDR R1, =k;

LDR R1, [R1];

LDR R2, [R0, R1, LSL 2];

LDR R3, =k32;

LDR R3, [R3];

SUB R4, R3, 1;

LDR R2, [R2, R4, LSL 0];

e. LDR R0, =u32;

LDR R0, [R0];

LDR R1, =u64;

LDRD R2, R3, [R1];

LDR R2, R0;

f. LDR R0, =s8;

LDR R0, [R0];

LDR R1, =s64;

LDRD R2, R3, [R1];

LDRB R1, R0;

7.

swap32:

LDR R2, R0;

LDR R0, R1;

LDR R1, R2;