```
#include <MKL25Z4.H>

__asm int my_sqrt(int x)
{
  ;r0 argument
  MOVS r1, #0x0      ;low
  MOV r2, r0         ;high
  LDR r3, =0xFFFFF ;previous mid int

LOOP

    MOVS r4, r1
    ADD r4, r2
    LSRS r4, #0x1    ;DIV BY 2
    MOV  r5, r4
    MULS r5, r5      ;square
    CMP r5,r0        ;compare to argument
    BNE NOTEQUAL
    B RETURN

NOTEQUAL
    CMP r4, r3       ;if previous and current mid point int are equal
    BEQ RETURN       ;if equal then return
    MOV r3, r4       ;update previous mid int
    CMP r5,r0        ;compare to argument
    BGE GREATER      ;if greater
    B SMALLER        ;else

GREATER
    MOV r2, r4       ;if greater then change high
    B LOOP

SMALLER
    MOV r1, r4       ;if greater then change low
    B LOOP

RETURN
    MOV r0, r4
    BX lr            ;return
}

int c_sqrt(int x){
  int low,high,mid,prevmid;
  low =0;
  high=x;
  prevmid=-1;
  while(1){
    mid=(low+high)/2;
    if(mid*mid==x){
      return mid;
    }else{

      if(prevmid==mid){
          return mid;
      }
      prevmid=mid;

      if(mid*mid>x){
        high = mid;
      }else{
        low = mid;
      }
    }
  }
}

int main(void)
{
    volatile int r, j=0;
    r = my_sqrt(0);   // should be 0
    r = c_sqrt(0);
    r = my_sqrt(25);  // should be 5
```

```
73        r = c_sqrt(25);
74        r = my_sqrt(133); // should be 11
75        r = c_sqrt(133);
76        while (1);
77    }
78
```