1. MUL_CONST:

```
        LSL         R1, R0, #1
        LSL         R2, R0, #3
        SUBS        R0, R1, R2
        BX          LR
```

2. asdf

    a. MUL_22:

    ```
        LSL         R1, R0, 4
        ADD         R0, R1, R0
        LSL         R0, R0, 11
        BX          LR
    ```

    b. MUL_23:

    ```
        LSL         R1, R0, 4
        ADD         R0, R1, R0
        LSL         R2, R0, 1
        ADD         R0, R0, R2
        BX          LR
    ```

    c. MUL_25:

    ```
        LSL         R1, R0, 4
        ADD         R2, R0, R1
        ADD         R0, R2, R1
        LSL         R0, R0, 1
        BX          LR
    ```

    d. MUL_26:

    ```
        LSL         R1, R0, 4
        ADD         R2, R0, R1
        ADD         R0, R2, R1
        LSL         R0, R0, 1
        ADD         R0, R0, R2
        BX          LR
    ```

    e. MUL_27:

    ```
        LSL         R1, R0, 4
        ADD         R2, R0, R1
        ADD         R0, R2, R1
        LSL         R0, R0, 1
        ADD         R3, R0, R2
        ADD         R0, R3, R1
        BX          LR
    ```

f. MUL_29:

```
        LSL     R1, R0, 4
        ADD     R2, R0, R1
        ADD     R0, R2, R1
        LSL     R0, R0, 1
        ADD     R3, R0, R2
        ADD     R0, R3, R1
        LSL     R0, R0, 1
        SUBS    R0, R0, R3
        BX      LR
```

4. the reciprocal of 3: ⅓ (0.333…)

6. Assembly

```
    Remainder:
            PUSH    {LR}

            MOV     R2, 0
            CMP     R1, 0
            BEQ     End

            SUB     R1, R1, 11

            CMP     R0, 0
            MOVLT   R3, 1
            ADDLT   R0, R0, R3

            LSRS    R2, R0, R1

            MOV     R3, 1
            LSL     R3, R3, R1

            MUL     R2, R2, R3

            SUB     R0, R0, R2
    End:
            POP     {LR}
            BX      LR
```

C Program

```
#include <stdio.h>
#include <stdint.h>

extern int32_t Remainder(int32_t s32, uint32_t k);

int main() {
   int32_t s32 = -25;
   uint32_t k = 4;
   int32_t remainder = Remainder(s32, k);

   printf("s32: %d\n", s32);
   printf("k: %u\n", k);
   printf("Remainder: %d\n", remainder);

   return 0;
}
```