## 1]

```
    AREA MULT, CODE, READONLY
    ENTRY
    LDR     R1, =ARR
    LDRH    R2, [R1], #2      ; Load first 16-bit value, then auto-increment R1
    LDRH    R3, [R1]          ; Load second 16-bit value
    MUL     R4, R2, R3
    LDR     R5, =RESULT
    STR     R4, [R5]
STOP    B       STOP
    AREA DATA, DATA, READWRITE
ARR     DCW     2_1010, 2_10      ; 10 * 2
RESULT  DCD     0
    END
```

## 2]

```
        AREA ADDITION, CODE, READONLY
ENTRY
START
   MOV R0, #10     ; R0 = 10 ? acts as a counter
   MOV R1, #0      ; R1 = 0  ? accumulator (final sum)
   MOV R2, #1      ; R2 = 1  ? current number to add
NEXT
        ADD R1, R1, R2 ; R1 = R1 + R2 (add current number to sum)
   ADD R2, #1     ; R2 = R2 + 1 (increment current number)
   SUBS R0, #1    ; R0 = R0 - 1 (decrement counter), sets flags
   BNE NEXT       ; If R0 != 0 ? repeat the loop
STOP
   NOP            ; Do nothing (acts as placeholder)
   END            ; End of program
```

## 3]

```
    AREA FACT, CODE, READONLY
    ENTRY

    MOV R0, #5      ; Number to find factorial of
    MOV R1, #1      ; R1 will hold the result

LOOP
    MUL R2, R1, R0  ; Use R2 to avoid unpredictable behavior
    MOV R1, R2
    SUBS R0, R0, #1
    BNE LOOP        ; Repeat if R0 != 0

    NOP
    END
```

## 4]

```
        AREA ADDITION, CODE, READONLY
        ENTRY

        LDR R0, =ARRAY     ; R0 ? base of array
        LDR R5, =SUM       ; R5 ? address to store result
        MOV R1, #10        ; R1 = counter (10 elements)
        MOV R2, #0         ; R2 = accumulator (32-bit sum)

LOOP
        LDRH R3, [R0], #2  ; Load 16-bit value from array and increment R0
        ADD R2, R2, R3     ; Add to accumulator
        SUBS R1, R1, #1    ; Decrement counter, update flags
        BNE LOOP           ; If not zero, continue

        STR R2, [R5]       ; Store final 32-bit sum
STOP    B STOP             ; Infinite loop (halt)
```

```
        AREA DATASEG1, DATA, READONLY
ARRAY   DCW 0x0011,0x0022,0x0033,0x0044,0x0055,0x0066,0x0077,0x0088,0x0099,0x00AA

        AREA DATASEG2, DATA, READWRITE
SUM     DCD 0

        END
```

## 5]

```
        AREA SQUARE, CODE, READONLY
        ENTRY

START
        LDR R0, =TABLE1      ; R0 = base address of square table
        MOV R1, #7           ; R1 = index (7)
        MOV R1, R1, LSL #2   ; R1 = offset in bytes (7 × 4 = 28) because each DCD is 4 bytes
        ADD R0, R0, R1       ; R0 = address of TABLE1[7]
        LDR R3, [R0]         ; R3 = square of 7 (0x00000031 = 49)

XSS     B XSS                ; Infinite loop

; Square lookup table (0² to 10²)
TABLE1
        DCD 0x00000000       ; 0²
        DCD 0x00000001       ; 1²
        DCD 0x00000004       ; 2²
        DCD 0x00000009       ; 3²
        DCD 0x00000010       ; 4²
        DCD 0x00000019       ; 5²
        DCD 0x00000024       ; 6²
        DCD 0x00000031       ; 7²
        DCD 0x00000040       ; 8²
```

```
    DCD 0x00000051      ; 9²
    DCD 0x00000064      ; 10²

    AREA DATA1, DATA, READWRITE
RESULT  DCD 0x00000000

    END
```

## 6]

```
    AREA LARGEST, CODE, READONLY
    ENTRY

START
    MOV R5, #6          ; Number of elements in the array
    LDR R1, =VALUE1     ; Load base address of array
    LDR R2, [R1], #4    ; Load first element into R2, increment pointer by 4 bytes

LOOP
    LDR R4, [R1], #4    ; Load next element into R4, increment pointer
    CMP R2, R4          ; Compare current max (R2) with new element (R4)
    BHI SKIP            ; If R2 > R4, skip updating max
    MOV R2, R4          ; Else, update max to new element (R4)

SKIP
    SUBS R5, R5, #1     ; Decrement loop counter
    CMP R5, #0          ; Check if end of array
    BNE LOOP            ; If not zero, continue loop

    LDR R4, =RESULT     ; Load address of RESULT
    STR R2, [R4]        ; Store the largest number in RESULT

XSS   B XSS             ; Infinite loop to stop program
```

```
; Array of 6 numbers to check
VALUE1
        DCD 0x44444444
        DCD 0x22222222
        DCD 0x11111111
        DCD 0x33333333
        DCD 0xAAAAAAAA
        DCD 0x88888888
        ; (You can add more if needed)

        AREA DATA1, DATA, READWRITE
RESULT  DCD 0x00000000        ; Result storage

        END
```