## 2.3 a)

```
        .data
A:      .word   1, 3, 1, 6, 4
        .word   2, 4, 3, 9, 5
mult:   .word   0

        .code
        daddi   $1, $0, A       ; *A[0]
        daddi   $5, $0, 1       ; $5 = 1 ;; i = 1
        daddi   $6, $0, 10      ; $6 = N ;; N = 10
        lw      $9, 0($1)       ; $9 = A[0]  ;; mult
        daddi   $1, $1, 8       ; Set up for next word (A[1])

loop:   lw      $12, 0($1)      ; $12 = A[i]

        daddi   $5, $5, 1       ; i++
        dmul    $12, $12, $9    ; $12 = $12*$9 ;; $12 = A[i]*mult
        daddi   $1, $1, 8       ; Set up for next word
        dadd    $9, $9, $12     ; $9 = $9 + $12  ;; mult = mult + A[i]*mult

        bne     $6, $5, loop    ; Exit loop if i == N
        sw      $9, mult($0)    ; Store result
        halt                    ; Stop the program execution

;; Expected result: mult = f6180 (hex), 1008000 (dec)
```

## 2.4 a)

```
        .data
A:      .word   1, 3, 1, 6, 4
        .word   2, 4, 3, 9, 5
mult:   .word   0

        .code
        daddi   $1, $0, A       ; *A[0]
        daddi   $5, $0, 1       ; $5 = 1 ;; i = 1
        daddi   $6, $0, 7       ; $6 = 7
        lw      $9, 0($1)       ; $9 = A[0]  ;; mult = A[0]
        lw      $12, 8($1)      ; $12 = A[1]
        lw      $13, 16($1)     ; $13 = A[2]

loop:   dmul    $22, $12, $9    ; $22 = $12*$9 ;; $22 = A[i]*mult
        daddi   $1, $1, 16      ; Set $1 for loading the next two words
        lw      $12, 8($1)      ; $12 = A[i+2] (doesn't interfere with dadd)
        dadd    $9, $9, $22     ; $9 = $9 + $22  ;; mult += A[i]*mult

        dmul    $23, $13, $9    ; $23 = $13*$9 ;; $23 = A[i+1]*mult
        daddi   $5, $5, 2       ; i += 2
        lw      $13, 16($1)     ; $13 = A[i+3] (doesn't interfere with dadd)
        dadd    $9, $9, $23     ; $9 = $9 + $23  ;; mult += A[i+1]*mult

        bne     $6, $5, loop    ; Exit loop if i == 7 (executes only three loops
                                ; to make sure we reduce by a factor of 4)

        ; 9 og iterations, so we are missing 3 (A[7], A[8] and A[9])
        dmul    $22, $12, $9    ; $22 = A[7]*mult
        lw      $14, 24($1)     ; $14 = A[9] (get last word)
        dadd    $9, $9, $22     ; mult += A[7]*mult

        dmul    $23, $13, $9    ; $23 = A[8]*mult
        dadd    $9, $9, $23     ; mult += A[8]*mult

        dmul    $24, $14, $9    ; $24 = A[9]*mult
        dadd    $9, $9, $24     ; mult += A[9]*mult (finally)

        sw      $9, mult($0)    ; Store result
        halt                    ; Stop the program execution

;; Expected result: mult = f6180 (hex), 1008000 (dec)
```

## 2.5 a)

```
        .data
A:      .word   1, 3, 1, 6, 4
        .word   2, 4, 3, 9, 5
mult:   .word   0

        .code
        daddi   $1, $0, A       ; *A[0]
        daddi   $5, $0, 1       ; $5 = 1 ;; i = 1
        daddi   $6, $0, 10      ; $6 = N ;; N = 10
        lw      $9, 0($1)       ; $9 = A[0]  ;; mult
        daddi   $1, $1, 8       ; Set up for next word (A[1])

loop:   lw      $12, 0($1)      ; $12 = A[i]

        daddi   $5, $5, 1       ; i++
        dmul    $12, $12, $9    ; $12 = $12*$9 ;; $12 = A[i]*mult
        daddi   $1, $1, 8       ; Set up for next word

        bne     $6, $5, loop    ; Exit loop if i == N
        dadd    $9, $9, $12     ; $9 = $9 + $12  ;; mult = mult + A[i]*mult
        sw      $9, mult($0)    ; Store result
        halt                    ; Stop the program execution

;; Expected result: mult = f6180 (hex), 1008000 (dec)
```