Shayshu Nahata-Ragubance - 1005035196

## PART 2:

```
/* Program that finds the largest number in a list of integers   */

         .text                   // executable code follows
         .global  _start
_start:
         MOV      R4, #RESULT    // R4 points to result location
         LDR      R2, [R4, #4]   // R2 holds number of elements in the list
         MOV      R3, #NUMBERS   // R3 points to the list of integers
         LDR      R0, [R3]       // R0 holds the largest number so far

LOOP:    SUBS     R2, #1         // decrement the loop counter
         BEQ      DONE           // if result is equal to 0, branch
         ADD      R3, #4
         LDR      R1, [R3]       // get the next number
         CMP      R0, R1         // check if larger number found
         BGE      LOOP
         MOV      R0, R1         // update the largest number
         B        LOOP
DONE:    STR      R0, [R4]       // store largest number into result location

END:     B        END

RESULT:  .word    0
N:       .word    7              // number of entries in the list
NUMBERS: .word    4, 5, 3, 6     // the data
         .word    1, 8, 2

         .end
```

## Part 3:

```
 /* Program that finds the largest number in a list of integers */
.text // executable code follows
.global _start

_start:
        MOV R4, #RESULT  // R4 points to result location
        LDR R2, [R4, #4] // R0 holds the number of elements in the list
        MOV R1, #NUMBERS // R1 points to the start of the list
        MOV R0, #0       //The largest number is now zero
        BL LARGE         // Start the subroutine
        STR R0, [R4] // R0 holds the subroutine return value

END: B END

// Subroutine for finding the largest number from a list of integers
LARGE:
        SUBS   R2, #1   //Subtract 1 from the counter
        MOVEQ PC, LR    //Return back to main is the subtraction returns zero
        LDR   R3, [R1]  //Load the integer from the numbers list
        ADD   R1, #4    //Add 4 to the pointer to the numbers list to get next value
        CMP   R0, R3    //Compare the current largest value and a number from the list
        BGE   LARGE     //If the current largest value is greater restart the loop
        MOV   R0, R3    //Otherwise make the current largest value the list value
        B     LARGE     //Restart the subroutine

```

```

RESULT:     .word 0
N:          .word 7 // number of entries in the list
NUMBERS:    .word 4, 5, 3, 6 // the data
            .word 1, 8, 2
            .end
```

## Part 4:

```
/* Program that converts a binary number to decimal */
           .text                   // executable code follows
           .global _start
_start:
            MOV    R4, #N
            MOV    R5, #Digits  // R5 points to the decimal digits storage location
            LDR    R4, [R4]     // R4 holds N
            MOV    R0, R4       // parameter for DIVIDE goes in R0
            BL     DIVIDE
STRB   R6, [R5, #3]
STRB   R3, [R5, #2]
            STRB   R1, [R5, #1] // Tens digit is now in R1
            STRB   R0, [R5]     // Ones digit is in R0
END:        B      END

/* Subroutine to perform the integer division R0 / 10.
 * Returns: quotient in R1, and remainder in R0 */
DIVIDE:     MOV    R2, #0 //R2 is the tens bit
MOV           R3, #0 //R3 is the hundreds bit
MOV           R6, #0 //R6 is the thousands bit


THOUSANDS:  CMP    R0, #1000
            BLT    HUNDREDS
            SUB    R0, #1000
            ADD    R6, #1
            B      THOUSANDS

HUNDREDS:        CMP           R0, #100
BLT           TENS
SUB           R0, #100
ADD    R3, #1
B           HUNDREDS

TENS:                CMP           R0, #10
BLT           DIV_END
SUB           R0, #10
ADD    R2, #1
B           TENS

DIV_END:    MOV    R1, R2     // quotient in R1 (remainder in R0)
            MOV    PC, LR


N:          .word  9178         // the decimal number to be converted
Digits:     .space 4          // storage space for the decimal digits

            .end
```