Question 1. Translate the following C code segment into an equivalent MIPS assembly code segment. You are allowed to use the following registers: $t0, $t1, $t2 in your code

```
int x = 15;
int y = 7;
int z = x – 4*y;
```

Reference code:

```
li $t0, 15
li $t1, 7
sll $t1, $t1, 2
sub $t2, $t1, $t0
```

Question 2. Convert the following C-style code into equivalent MIPS assembly code:

```
int sum = 0;
for (int i = 1; i <= 10; i++) {
    if (i % 2 == 0) {
        sum += i;
    }
}
```

Provide the MIPS assembly code that initializes the variable sum, uses a loop to iterate through numbers from 1 to 10, checks if each number is even, and accumulates the even numbers in sum. You are allowed to use the following registers: $t0, $t1, $t2 in your code, you can store sum in $t0 and the loop counter in $t1.

Reference code:

```
.text
 li $t0, 0       # $t0 will store the sum
 li $t1, 1       # $t1 will be the loop counter

loop:
    # Check if i is greater than 10
    bgt $t1, 10, exit

    # Check if i is even (i % 2 == 0)
    andi $t2, $t1, 1   # Bitwise AND with 1 to check if the LSB is 1 (odd) or 0 (even)
    bnez $t2, not_even  # If LSB is 1, not even

    # Add i to sum
    add $t0, $t0, $t1

    not_even:
    # Increment i
    addi $t1, $t1, 1
    # Continue the loop
    j loop
exit:
    # Exit the program
    li $v0, 10          # Load syscall code 10 (exit)
    syscall
```

Question 3. Write MIPS assembly code that do the computation: F=|A-B|+C. Suppose the value A is in register $t0, B is in register $t1, C is in register $t2, and the result F should be stored in register $t3.

Reference code:

```
.text
    move $a1, $t0 #A
    move $a2, $t1 #B
    jal calculate       # compute |A-B|

    move $t7, $v0     # store |A-B| in t7
    add $t3, $t2, $t7
    j exit

calculate:
    sub $t5, $a1, $a2   # compute $a1-$a2
    move $t6, $t5
    srl $t6, $t6, 31     # focus on the highest bit
    bnez $t6, negate   # if highest bit is 1, then jump
    move $v0, $t5
    jr $ra

negate:
   sub $v0, $zero, $t5
   jr $ra
exit:
   li $v0, 10        # exit
   syscall
```

Question 4. Split a string by space character and get the number of substrings. Terminators can be '\n' or '\0'. $t0, $t1 and $t2 can be used.

Reference code:

```
# .data segment contains all variables
.data
# string stored in MEM
myMessage:       .asciiz "Hello World!\n"
# .text segment
.text
    la $t0, myMessage       # load address of myMessage to $t0
    li $t2, 0 # store number of substrings.

split:
    lb $t1, ($t0)
    add $t0, $t0, 1
    beq $t1, '\n', exit
    beq $t1, '\0', exit
    beq $t1, ' ', newsubstring
    b split

newsubstring:
    addi $t2, $t2, 1
    b split

exit:
    li $v0, 10
    syscall
```