

**Problem 1:**

Convert the following c-code to MIPS assembly code.

```
int x=10;
int y=5;
int z=0;
if (x>y)
    z = x-y;
else
    z = x+y;
```

**Solution:**

```
.globl main

.text
main:
    li $s0, 10
    li $s1, 5
    bgt $s0, $s1, ifpart
    add $s2, $s0, $s1
    j Exit

ifpart:
    sub $s2, $s0, $s1

Exit:
    li $v0, 10
    syscall
```

**Problem 2:**

Convert the following c-code to MIPS assembly code.

```
int a=5;
int b=6;
int max=0;
if (a<b)
    max = b;
else if (a>b)
    max = a;
else
    max =100;
```

## Solution

```
.globl main

.text
main:
    li $s0, 6
    li $s1, 6
    blt $s0, $s1, part1
    bgt $s0, $s1, part2
    li $s2, 100
    j Exit
part1:
    move $s2, $s1
    j Exit
part2:
    move $s2, $s0
Exit:
    li $v0, 10
    syscall
```

## Problem 3:

Write a MIPS Assembly program that finds the maximum value within 3 values, given the following c-code.

```
int a =5;
int b =6;
int c =8;
int max =0;
if (a>b)
    if (a>c)
        max =a;
    else
        max = c;
else
    if (b>c)
        max = b;
    else
        max =c;
```

**Solution:**

```
.globl main
.text
main:
    li $s0, 5
    li $s1, 6
    li $s2, 8
    bgt $s0, $s1, outerif
    blt $s1, $s2, innerif2
    move $s3, $s1

outerif:
    bgt $s0, $s2, innerif1
    j innerif2

innerif1:
    move $s3, $s0
    j Exit

innerif2:
    move $s3, $s2
    j Exit

Exit:
    li $v0, 10
    syscall
```

**Problem 4:**

Write a MIPS assembly program that is equivalent to the given high level programming language code below.

```
int sum =0;
int avg =0;
for (int i=1; i<=20; i++)
{
    sum +=i;
}
avg = sum/20;
```

**Solution:**

```
.globl main
.text
main:
    li $s1, 1
    li $t1, 20

loop: slti $t0, $s1, 21
      beq $t0, $zero, Exit
      add $s2, $s2, $s1
      addi $s1, $s1, 1
      j loop

Exit:
    div $s3, $s2, $t1
    li $v0, 10
    syscall
```

**Problem 5:**

Write a MIPS assembly program that calculates the value of  $5^3$ , given the high-level programming language code below.

```
int a=1;
for (int i=0; i<3; i++)
{
    a*=5;
}
```

**Solution**

```
.globl main
.text
main:
    li $s0, 1
    li $t3, 5

loop: slti $t0, $s1, 3
      beq $t0, $zero, Exit
      mul $s0, $s0, $t3
      addi $s1, 1
      j loop

Exit:
    li $v0, 10
    syscall
```

### Problem 6:

Write a MIPS assembly program that sums up the numbers from 1 to 10 starting from number 10, given the high-level programming language code below.

```
int sum =0;
for (int i=10; i>0; i--)
{
    sum +=i;
}
```

### Solution

```
.globl main
.text
main:
    li $s1, 10
    li $t0, 1

loop: slt $t1,$zero,$s1
      beq $t1, $zero, Exit
      add $s2, $s2, $s1
      sub $s1, $s1, $t0
      j loop

Exit:
    li $v0, 10
    syscall
```

### Problem 7:

Write a MIPS assembly program to find the greatest common divisor (GCD) of 2 integers, where the GCD is the largest positive integer that divides the numbers without a remainder. For example, the GCD of 8 and 12 is 4, the GCD of 54 and 24 is 6. The below high level programming code is a simple way to find the greatest common divisor (GCD) of a and b.

```
int a = 54;
int b = 24;
while(a != b)
{
    if(a > b)
        a = a - b;
    else
        b = b - a;
}
```

## Solution

```
.globl main
.text
main:
    li $s0, 54
    li $s1, 24

loop: bgt $s0, $s1, ifpart
      blt $s0, $s1, elsepart
      beq $s0, $s1, Exit

ifpart:
    sub $s0, $s0, $s1
    j continue

elsepart:
    sub $s1, $s1, $s0

continue:
    j loop

Exit:
    li $v0, 10
    syscall
```