

```
#####
# Solution to TPC 1 of Computer Architecture #
# 2024-2025. Peter Stallnga, University of #
# The Algarve #
#####
.data
```

```
intro: .asciiz "Numbers a, b, c, d:\n"
prompt: .asciiz "Give a number: "
answer: .asciiz "The series sum is "
```

```
.text
```

```
# calculate the sum
# a will be in $t5
# b will be in $t6
# c will be in $s5
# d will be in $s6
# i will be in $t0
# j will be in $s0
# sum will be in $t9
```

```
main:
```

```
li $v0, 4
la $a0, intro
syscall
```

```
li $v0, 4
la $a0, prompt
syscall
li $v0, 5
syscall
```

```
move $t5, $v0
li $v0, 4
la $a0, prompt
syscall
li $v0, 5
syscall
move $t6, $v0
li $v0, 4
la $a0, prompt
syscall
li $v0, 5
syscall
move $s5, $v0
```

```

li $v0, 4
la $a0, prompt
syscall
li $v0, 5
syscall
move $s6, $v0

move $t9, $zero      # sum = 0

move $s0, $s5        # j=c
outerloop:
    bgt $s0, $s6, outerready
    move $t0, $t5      # i=a
    innerloop:
        bgt $t0, $t6, innerready
        sub $t3, $t0, $s0    # (i-j)
        mul $t3, $t3, $t3    # (i-j)^2
        div $t4, $t0, 2      # better would be a mask with pattern 0000..001
        mfhi $t4
        beqz $t4, addt3      # if i is odd
        mul $t3, $t3, -1     # then multiply (i-j)^2 with -1
    addt3:
        add $t9, $t9, $t3    # add (-1)^i (i-j)^2 to sum
        addi $t0, $t0, 1     # i++
        j innerloop
    innerready:
        addi $s0, $s0, 1     # j++
        j outerloop
outerready:
    li $v0, 4
    la $a0, answer
    syscall
    li $v0, 1
    move $a0, $t9
    syscall

return:
    li $v0, 10
    syscall

```