

1.

```

addi a1,x0,52 //a1 is the upper border
addi a2,x0,-20 //a2 is the lower border
andi a3,a2,1
bnez a3,cal // a3!=0 jump to cal
addi a2,a2,1 //if a2 is a even number,add 1 to become a odd number
cal:bgt a2,a1,done //determine when the program is done
mul a4,a2,a2 //get the square
mul a4,a4,a2 //get the cube
add s1,s1,a4
addi a2,a2,2 //get the next odd number
jal x0,cal // loop
done:

```

When the upper border is 52 and lower border is -20.The result is 893376.

The screenshot shows a MIPS assembly simulator interface. On the left, there is a table with three columns: Machine Code, Basic Code, and Original Code. The Basic Code column contains the assembly instructions for the first program. On the right, there is a 'Registers' panel showing the values of various registers. The register a1 (x9) is highlighted with a red box, and its value is 893376. The 'Display Settings' dropdown is set to 'Decimal'.

Machine Code	Basic Code	Original Code
0x03400593	addi x11 x0 52	addi a1,x0,52
0xfec00613	addi x12 x0 -20	addi a2,x0,-20
0x001e7693	andi x13 x12 1	andi a3,a2,1
0x00069463	bne x13 x0 8	bnez a3,cal
0x001e0613	addi x12 x12 1	addi a2,a2,1
0x00c5cc63	blt x11 x12 24	cal:bgt a2,a1,done
0x02c60733	mul x14 x12 x12	mul a4,a2,a2
0x02c70733	mul x14 x14 x12	mul a4,a4,a2
0x00e4e4b3	add x9 x9 x14	add s1,s1,a4
0x00260613	addi x12 x12 2	addi a2,a2,2
0xfedff06f	jal x0 -20	jal x0,cal

Registers:

- zero: 0
- ra (x1): 0
- sp (x2): 2147483632
- gp (x3): 269439456
- tp (x4): 0
- t0 (x5): 0
- t1 (x6): 0
- t2 (x7): 0
- s0 (x8): 0
- a1 (x9): 893376**
- a0 (x10): 0
- a1 (x11): 52
- a2 (x12): 53
- a3 (x13): 0
- a4 (x14): 132651
- a5 (x15): 0
- a6 (x16): 0

Display Settings: Decimal

2.

```

addi a1,x0,10 //a1 is the upper border
addi a2,x0,1 //a2 is the start
addi s1,x0,1 // s1 is the result
loop:bge a2,a1,done // if a2>=a1, the loop is done
addi a2,a2,1 // a2=a2+1
mul s1,s1,a2 //s1=s1*a2
jal x0,loop // jump to loop
done:

```

When the upper border is 10, the result 10!=3628800

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00a00593	addi x11 x0 10	addi a1,x0,10
0x00100613	addi x12 x0 1	addi a2,x0,1
0x00100493	addi x9 x0 1	addi s1,x0,1
0x00b65963	bge x12 x11 16	loop:bge a2,s1,done
0x00160613	addi x12 x12 1	addi a2,a2,1
0x02c464b3	mul x9 x9 x12	mul s1,s1,a2
0xffff006f	jal x0 -12	jal x0,loop

console output

Registers Memory

zero 0

ra (x1) 0

sp (x2) 2147483632

gp (x3) 268435456

tp (x4) 0

t0 (x5) 0

t1 (x6) 0

t2 (x7) 0

s0 (x8) 0

s1 (x9) 3628800

a0 (x10) 0

a1 (x11) 10

a2 (x12) 10

a3 (x13) 0

a4 (x14) 0

a5 (x15) 0

a6 (x16) 0

Display Settings Decimal

3.

addi a0,x0,11 //a0 is the upper border

addi a1,x0,2 //a1 is the first Prime Number 2

addi s1,s1,2 //s1 is the result

outer: addi a1,a1,1 //a1=a1+1

bgt a1,a0,done //a1>a0 the program is done

addi a2,x0,2 //a2=2

inner:beq a2,a1, cal//a2==a1 jump to cal

rem a3,a1,a2 //a3=a1%a2

beq a3,x0,outer //a3==0 jump to outer

addi a2,a2,1 // a2=a2+1

jal x0,inner // jump to inner

cal: add s1,s1,a1 //s1=s1+a1

jal x0,outer

done:

The upper border is 11. 'outer' is to traverse form 2 to 11. 'inner' is to make sure whether the number is prime or not.The result is 28.

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00b00513	addi x10 x0 11	addi a0,x0,11
0x00200593	addi x11 x0 2	addi a1,x0,2
0x00246493	addi x9 x9 2	addi s1,s1,2
0x00158593	addi x11 x11 1	outer: addi a1,a1,1
0x02b54263	blt x10 x11 36	bgt a1,a0,done
0x00200613	addi x12 x0 2	addi a2,x0,2
0x00b60a63	beq x12 x11 20	inner:beq a2,s1, cal
0x02c5e6b3	rem x13 x11 x12	rem a3,a1,a2
0xfe0686a3	beq x13 x0 -20	beq a3,x0,outer
0x00160613	addi x12 x12 1	addi a2,a2,1
0xffff006f	jal x0 -16	jal x0,inner
0x00b464b3	add x9 x9 x11	cal: add s1,s1,a1
0xfddff06f	jal x0 -36	jal x0,outer

console output

Registers Memory

zero 0

ra (x1) 0

sp (x2) 2147483632

gp (x3) 268435456

tp (x4) 0

t0 (x5) 0

t1 (x6) 0

t2 (x7) 0

s0 (x8) 0

s1 (x9) 28

a0 (x10) 11

a1 (x11) 12

a2 (x12) 11

a3 (x13) 1

a4 (x14) 0

a5 (x15) 0

a6 (x16) 0

Display Settings Decimal

4.

addi a0,x0,4 //a0 is the upper border.N terms.

addi a3,x0,1 //a3=1

addi s1,x0,1 //s1 is the result. s1=1

addi a4,x0,-3//r=-3

addi a1,x0,2 //a1=2

loop:bgt a1,a0,done // a1>a0 the program is done

mul a3,a3,a4 //a3=a3*(-3)

add s1,s1,a3 //s1=s1+a3

addi a1,a1,1 //a1=a1+1

jal x0,loop //loop

done:

When N=4, the result is -20.

Editor
Simulator

Run
Step
Prev
Reset
Dump

Registers Memory

Machine Code	Basic Code	Original Code
0x00400513	addi x10 x0 4	addi a0,x0,4
0x00100693	addi x13 x0 1	addi a3,x0,1
0x00100493	addi x9 x0 1	addi s1,x0,1
0xffd00713	addi x14 x0 -3	addi a4,x0,-3
0x00200593	addi x11 x0 2	addi a1,x0,2
0x00b54a63	blt x10 x11 20	loop:bgt a1,a0,done
0x02e686b3	mul x13 x13 x14	mul a3,a3,a4
0x00d484b3	add x9 x9 x13	add s1,s1,a3
0x00158593	addi x11 x11 1	addi a1,a1,1
0xff1ff06f	jal x0 -16	jal x0,loop

console output

zero	0
ra (x1)	0
sp (x2)	2147483632
gp (x3)	268435456
tp (x4)	0
t0 (x5)	0
t1 (x6)	0
t2 (x7)	0
a0 (x8)	0
s1 (x9)	-20
a0 (x10)	4
a1 (x11)	5
a2 (x12)	0
a3 (x13)	-27
a4 (x14)	-3
a5 (x15)	0
a6 (x16)	0