Name: JAHANVI AGRAWAL
Roll No.: IMT2019506

THE PROGRAM IMPLEMENTED using the IAS architecture:

```
a=15;
b=5;
if( a>=b ) c = a - b;
else c = a + b;
return c;
```

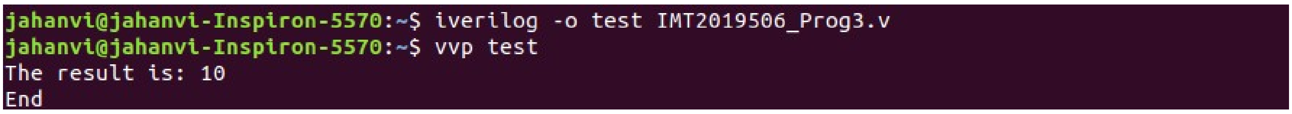
Corresponding implementation using ISA intructions in the memory:

```
LOAD M(8)
SUB M(9)
JUMP + M(2,20:39)
ADD M(9)
ADD M(9)
STOR M(10)
HALT()
```

Memory Location	**LHS Instruction**	**RHS Instruction**
0	LOAD M(8)	SUB M(9)
1	JUMP + M(2,20:39)	ADD M(9)
2	ADD M(9)	STOR M(10)
3	XXXXXXXXXX	HALT()
8	15	
9	5	

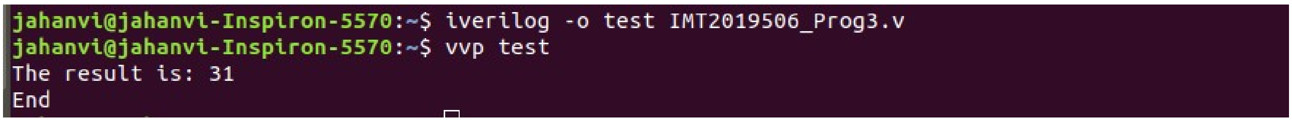
OUTPUT:

This is the snapshot of the terminal.



```
jahanvi@jahanvi-Inspiron-5570:~$ iverilog -o test IMT2019506_Prog3.v
jahanvi@jahanvi-Inspiron-5570:~$ vvp test
The result is: 10
End
```

Now if we change the inputs to a=15 and b=16 in the .v file, we get:



```
jahanvi@jahanvi-Inspiron-5570:~$ iverilog -o test IMT2019506_Prog3.v
jahanvi@jahanvi-Inspiron-5570:~$ vvp test
The result is: 31
End
```

Memory Allocations and Assumptions:

- Initially PC is set to 0.
- Memory locations from 0 to 3 are used for storing the instructions and the next locations are for data storage (this can be changed).
- At location 8, “a” is stored as a 40 bit binary number (here a=15)
- At location 9, “b” is stored as a 40 bit binary number (here b=5)
- At location 10, the value of “c” is stored at the end.

- If the LHS or RHS instruction is 20’bX, it means that there is no instruction there.

Explaination:
Initially AC is loaded with “a” from the location 8 and then “b” stored at location 9 is subtracted from the AC getting a-b in the AC.
For the if-else loop, using the JUMP + M(X,20:39), if the content of AC is non-negative (i.e. a-b>=0 i.e. a>=b), we JUMP to right half of location 2 and we store a-b at location 10. Or if the content of AC is negative, we add the value of b from location 9 two times so as to get a+b in the AC. After this, it is stored at location 10 and the program halts.