CO PROJECT – RISC-V SIMULATOR

TEAM NAME: TECH PHANTOMS

TEAM MEMBERS: LELLA HEMASRI SAI - CS20B020

KANCHARLA NAGA VANI SARANYA - CS20B016

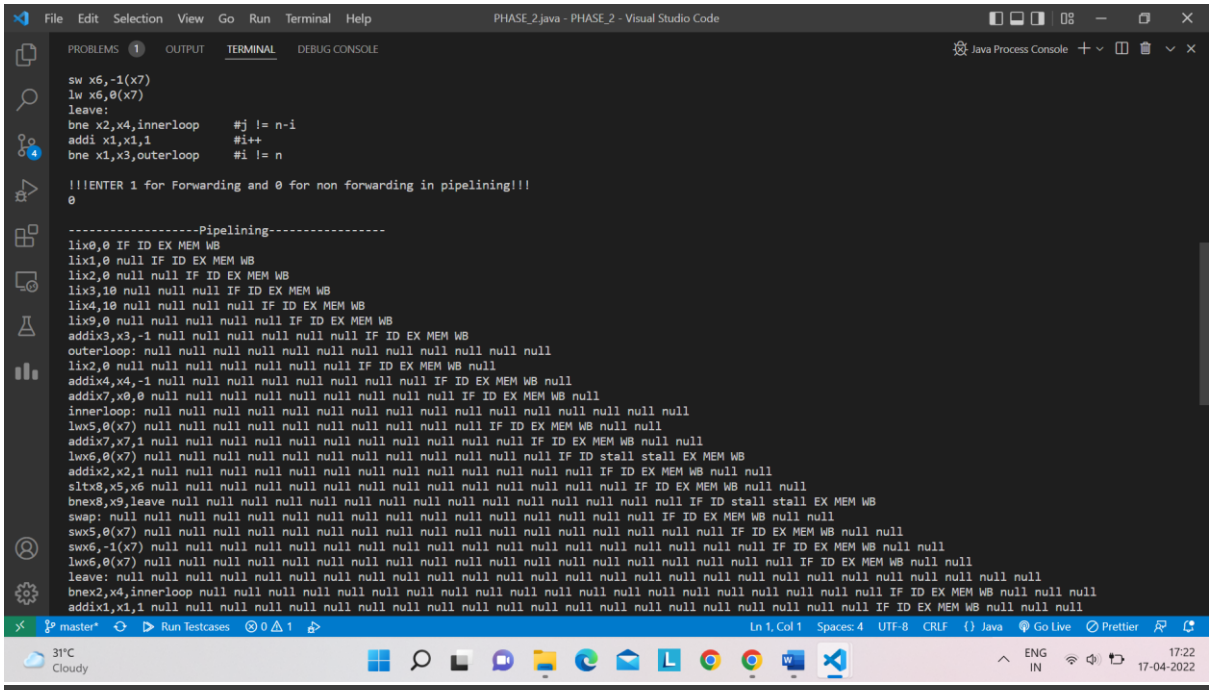
WHAT WE TRIED:

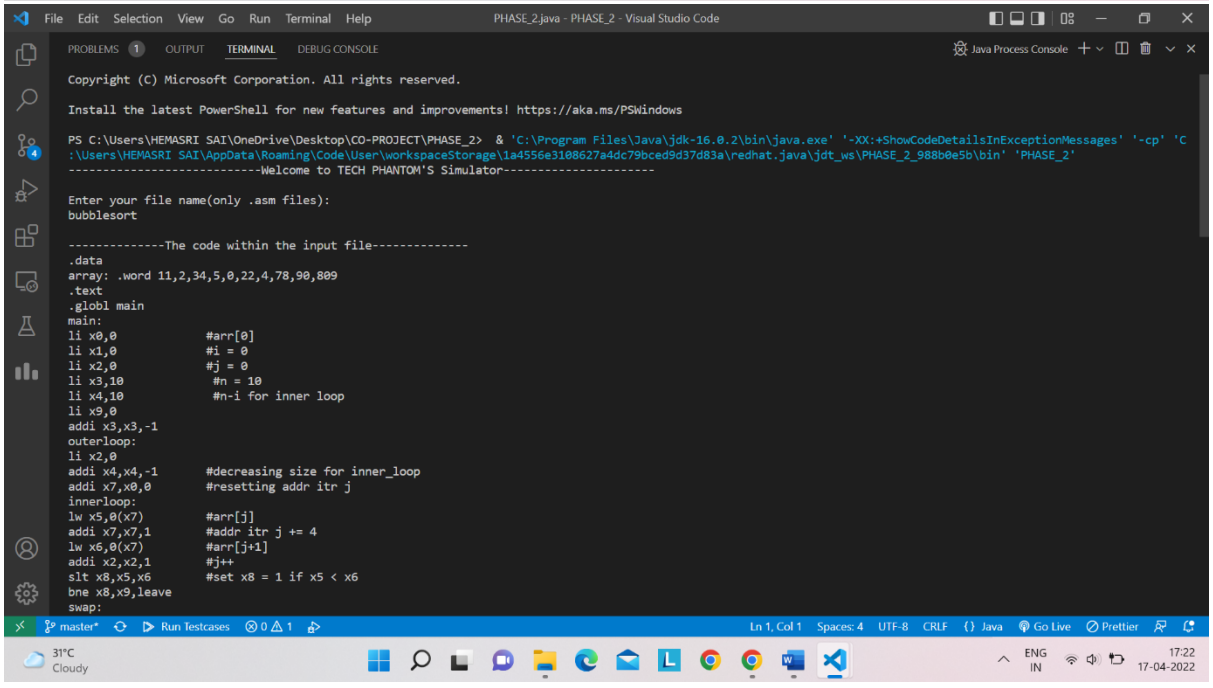
- After taking the input of the file, the code within the input file is being stored in an array.
- Then, we do check for data and text sections and if they are correct proceed to the next part.
- We check for the main index to start and then store the instructions into another array.
- While the program counter is less than or equal to the total number of lines in the code, we implement the code within the input file by updating the pipeline 2-D array.
- Now, we fill the pipeline 2-D array according to the instructions.
- We then check for the instruction in each line and then check if the source register is same as the destination register of the previous instruction.
- If in case they are same, then we check whether the instruction is branch instruction in both forwarding and non-forwarding cases.
- Now we check where the previous instruction's IF, ID, EX, MEM, WB are located and if the previous instruction's contain stalls, then continue them further.
- Now, fill the pipeline 2-D array accordingly.
- While filling the 2-D array, we also check for stalls and continue them in further rows.
- The same method is followed every instruction line.
- Finally, we checked where the last WB is located and calculated the number of clock cycles. And then we calculated total number of stalls, number of instructions per cycle and found the list of instructions where stalls occurred.

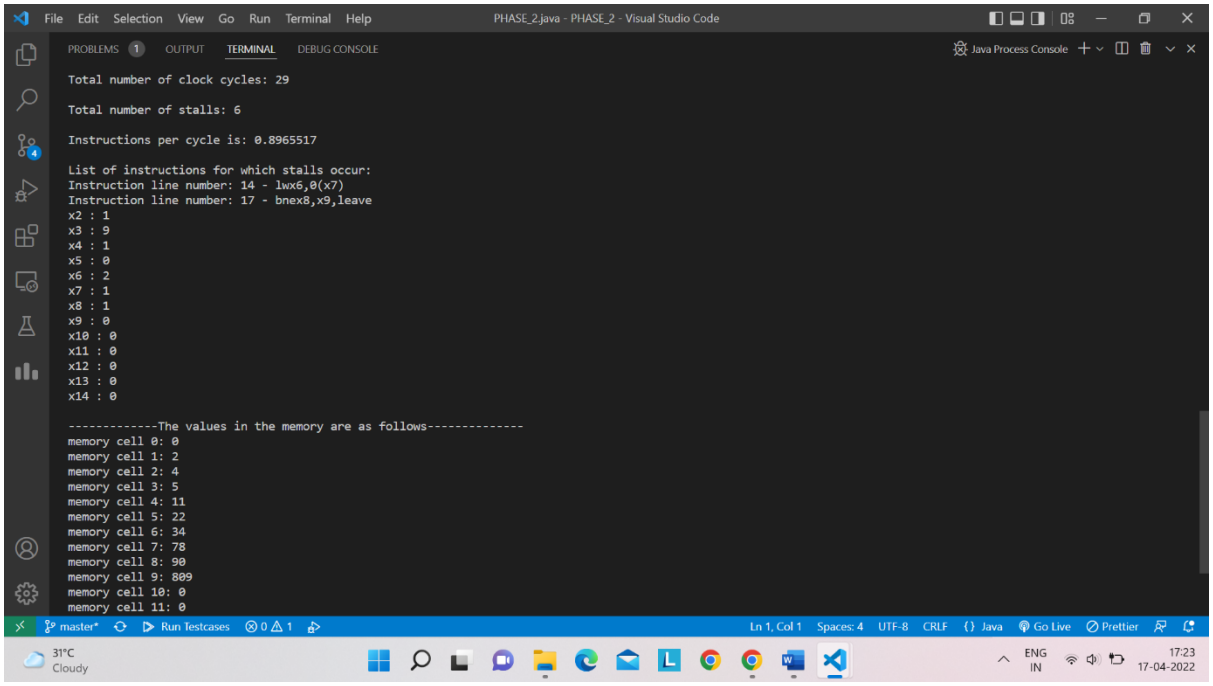
WHAT DID NOT WORK:

- The stalls for every instruction are being printed but they aren't being continued to further rows that is if a instruction results in 2 stalls in 11^{th} and 12^{th} location in 3^{rd} row, then all the instructions in the succeeding rows should contain stalls in 11^{th} and 12^{th} location, but here in this code those stalls are being re-written by null or any one of the IF/ID/EX/MEM/WB.
- In this implementation, we are able to cross-check between source and destination registers only for current instruction line and previous instruction line but not the line before the previous instruction line. If we try to do that, the stalls are not being seen in the output.

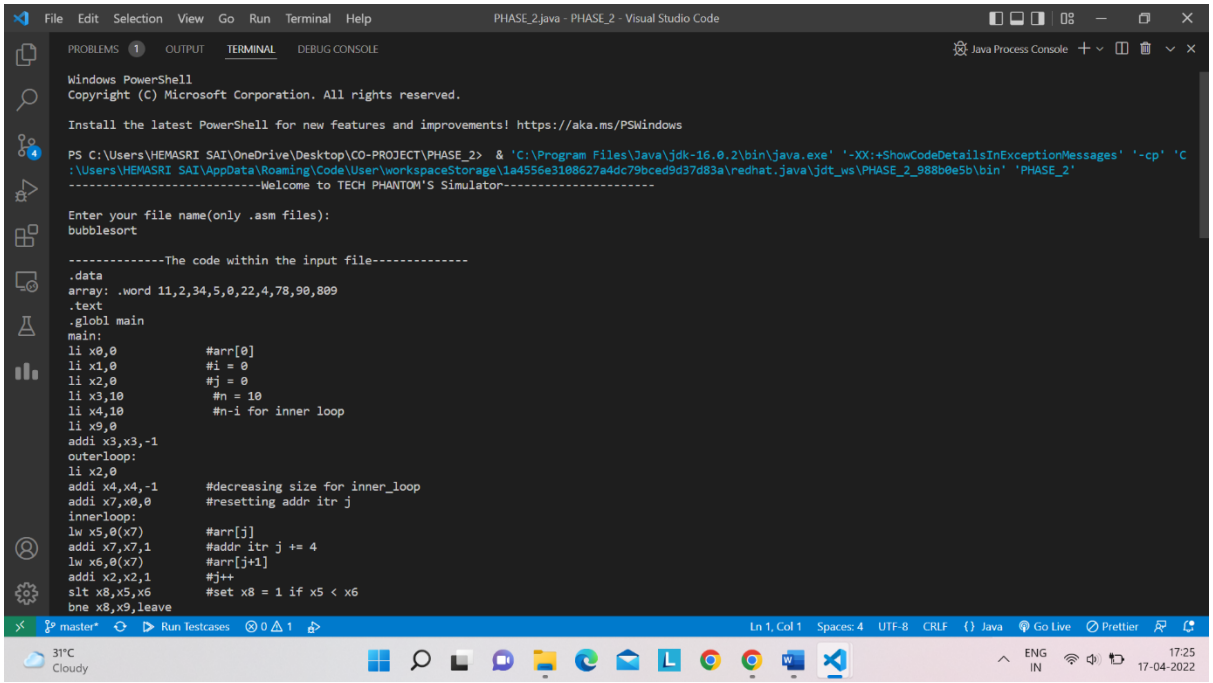
OUTPUT WITHOUT PIPELINING FORWARDING:

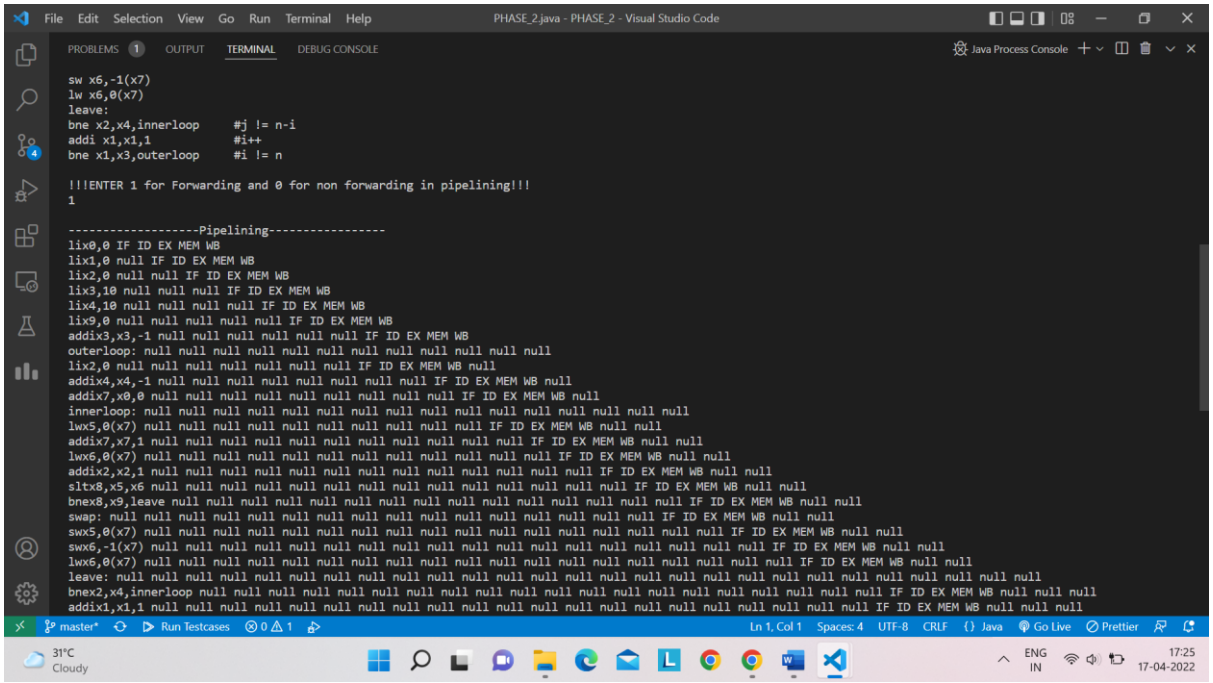


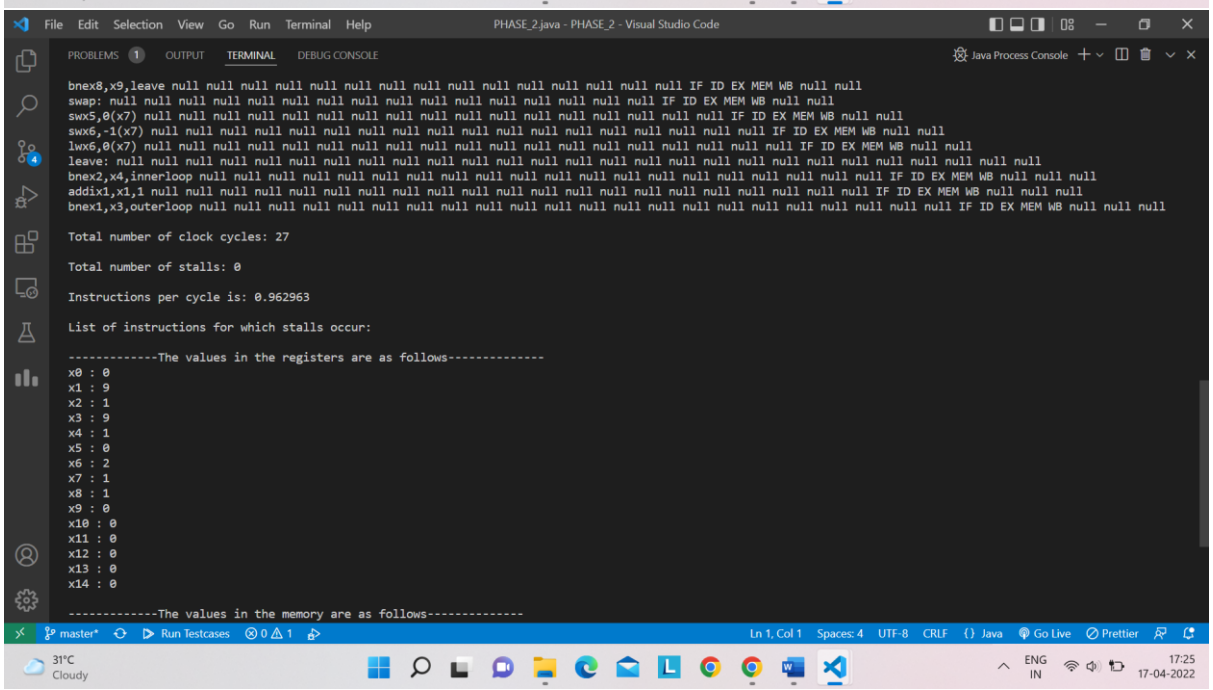


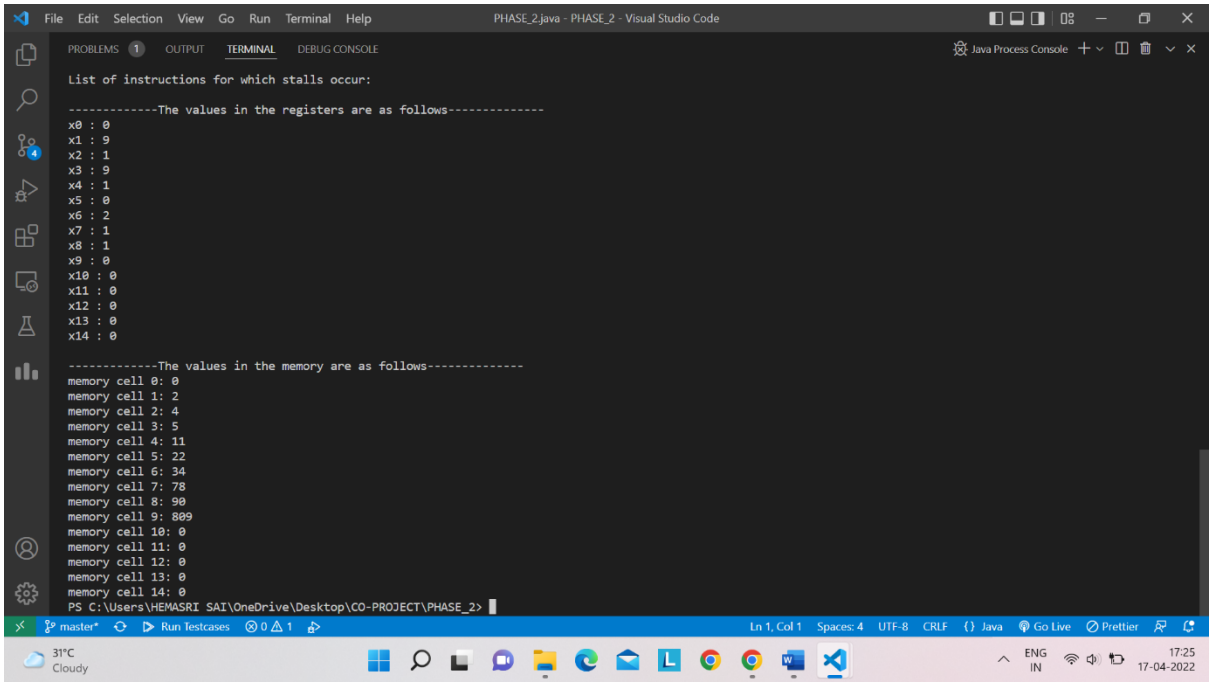


OUTPUT WITH PIPELINE FORWARDING:




File Edit Selection View Go Run Terminal Help
PHASE_2.java - PHASE_2 - Visual Studio Code
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
Java Process Console
sw x6,-1(x7)
lw x6,0(x7)
leave:
bne x2,x4,innerloop #j != n-i
addi x1,x1,1 #i++
bne x1,x3,outerloop #i != n
!!!ENTER 1 for Forwarding and 0 for non forwarding in pipelining!!!
1
------------------Pipelining----------------
lix0,0 IF ID EX MEM WB
lix1,0 null IF ID EX MEM WB
lix2,0 null null IF ID EX MEM WB
lix3,10 null null null IF ID EX MEM WB
lix4,10 null null null null IF ID EX MEM WB
lix9,0 null null null null null IF ID EX MEM WB
addix3,x3,-1 null null null null null null IF ID EX MEM WB
outerloop: null null null null null null null null null null null null null
lix2,0 null null null null null null null IF ID EX MEM WB null
addix4,x4,-1 null null null null null null null null IF ID EX MEM WB null
addix7,x0,0 null null null null null null null null null IF ID EX MEM WB null
innerloop: null null null null null null null null null null null null null null null
lwx5,0(x7) null null null null null null null null null null IF ID EX MEM WB null null
addix7,x7,1 null null null null null null null null null null null IF ID EX MEM WB null null
lwx6,0(x7) null null null null null null null null null null null null IF ID EX MEM WB null null
addix2,x2,1 null null null null null null null null null null null null null IF ID EX MEM WB null null
sltx8,x5,x6 null null null null null null null null null null null null null null IF ID EX MEM WB null null
bnex8,x9,leave null null null null null null null null null null null null null null null IF ID EX MEM WB null null
swap: null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
swx5,0(x7) null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
swx6,-1(x7) null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
lwx6,0(x7) null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
leave: null null null null null null null null null null null null null null null null null null null null null null null null null
bnex2,x4,innerloop null null null null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null null
addix1,x1,1 null null null null null null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null null
master* Run Testcases 0 1
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier
31°C Cloudy
ENG IN
17:25 17-04-2022


File Edit Selection View Go Run Terminal Help
PHASE_2.java - PHASE_2 - Visual Studio Code
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
Java Process Console
bnex8,x9,leave null null null null null null null null null null null null null null null IF ID EX MEM WB null null
swap: null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
swx5,0(x7) null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
swx6,-1(x7) null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
lwx6,0(x7) null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null
leave: null null null null null null null null null null null null null null null null null null null null null null null null null
bnex2,x4,innerloop null null null null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null null
addix1,x1,1 null null null null null null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null null
bnex1,x3,outerloop null null null null null null null null null null null null null null null null null null null null null null null null null IF ID EX MEM WB null null null
Total number of clock cycles: 27
Total number of stalls: 0
Instructions per cycle is: 0.962963
List of instructions for which stalls occur:
------------The values in the registers are as follows--------------
x0 : 0
x1 : 9
x2 : 1
x3 : 9
x4 : 1
x5 : 0
x6 : 2
x7 : 1
x8 : 1
x9 : 0
x10 : 0
x11 : 0
x12 : 0
x13 : 0
x14 : 0
------------The values in the memory are as follows--------------
master* Run Testcases 0 1
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier
31°C Cloudy
ENG IN
17:25 17-04-2022


File Edit Selection View Go Run Terminal Help
PHASE_2.java - PHASE_2 - Visual Studio Code
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
Java Process Console
List of instructions for which stalls occur:
------------The values in the registers are as follows--------------
x0 : 0
x1 : 9
x2 : 1
x3 : 9
x4 : 1
x5 : 0
x6 : 2
x7 : 1
x8 : 1
x9 : 0
x10 : 0
x11 : 0
x12 : 0
x13 : 0
x14 : 0
------------The values in the memory are as follows--------------
memory cell 0: 0
memory cell 1: 2
memory cell 2: 4
memory cell 3: 5
memory cell 4: 11
memory cell 5: 22
memory cell 6: 34
memory cell 7: 78
memory cell 8: 90
memory cell 9: 809
memory cell 10: 0
memory cell 11: 0
memory cell 12: 0
memory cell 13: 0
memory cell 14: 0
PS C:\Users\HEMASRI SAI\OneDrive\Desktop\CO-PROJECT\PHASE_2>
master* Run Testcases 0 1
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Java Go Live Prettier
31°C Cloudy
ENG IN
17:25 17-04-2022