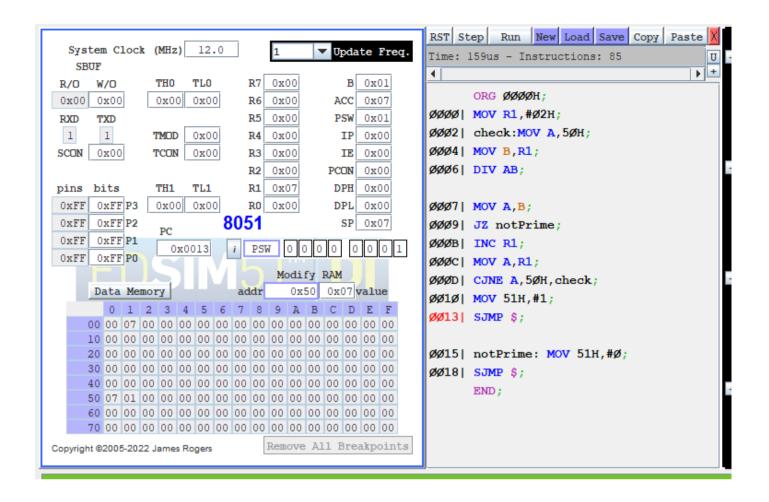
Assignment-3

Q1 Implement an assembly language program to checkwhether the given number is prime or not.

```
ORG 0000H;
MOV R1,#02H;
check: MOV A,50H;
MOV B,R1;
DIV AB;
MOV A,B;JZ
notPrime;
INC R1;
MOV A,R1;
CJNE
A,50H,check;
MOV 51H,#1;
SJMP $;
notPrime: MOV 51H,#0;
SJMP $;
END;
```



Q2 Write an assembly language program for "Addition of Array of numbers" in Edsim51.

```
ORG 0000H;

MOV R1,#50H;

MOV R0,#60H;

MOV @R0,#0H;

MOV A,@R1;

MOV R4,A;

repeat: INC R1;

MOV A,@R0;

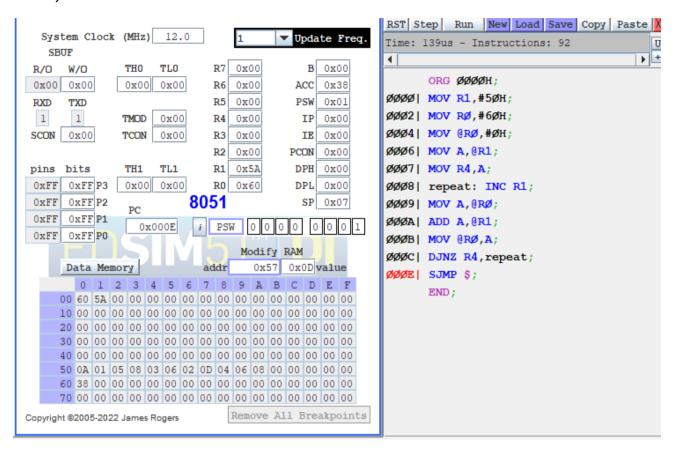
ADD A,@R1;

MOV @R0,A;
```

DJNZ R4, repeat;

SJMP\$;

END;



Q3 Move a block of data from one memory location to other where memory is addressed through indirect mode in both overlapping and non-overlapping case.

Non-Overlapping-

MOV R0,#50H;

MOV A,@R0; MOV

R4,A;

INC R0;

MOV R1,#61H;

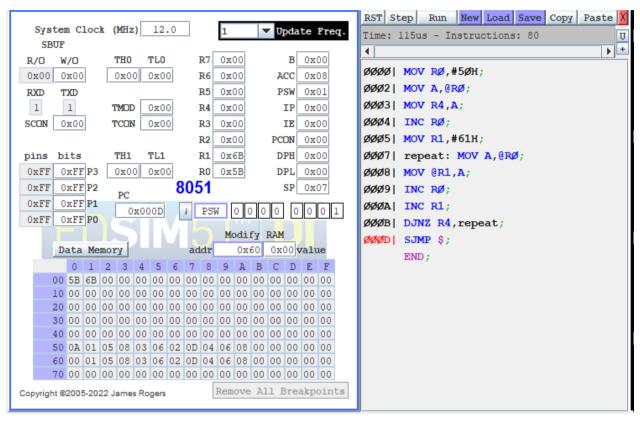
repeat: MOV A,@R0;

MOV @R1,A;

```
INC R0;
INC R1;
DJNZ R4,repeat;
```

SJMP \$;

END;



Overlapping-

MOV R0,#50H;

MOV A,@R0;

MOV R4,A;

MOV R0,#6AH;

MOV R1,#6DH;

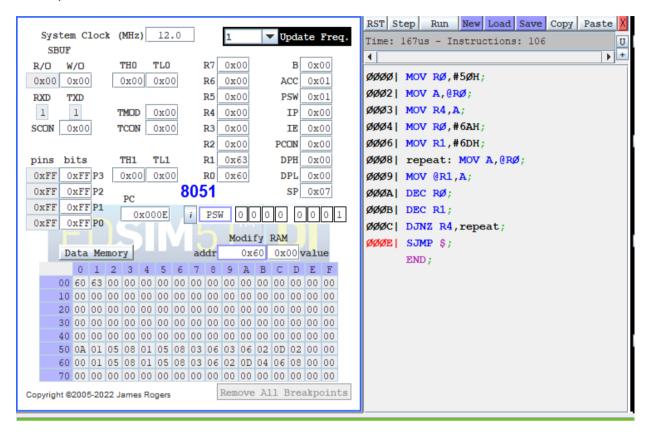
repeat: MOV A,@R0;

MOV @R1,A;

DEC R0;

DEC R1;
DJNZ R4,repeat;
SJMP \$;

END;



Q4 Implement an assembly language program to "exchange data blocks of 10.

MOV R0,#50H;

MOV A,@R0;

MOV R4,A;

INC R0;

MOV R1,#61H;

repeat: MOV A,@R0;

MOV B,@R1;

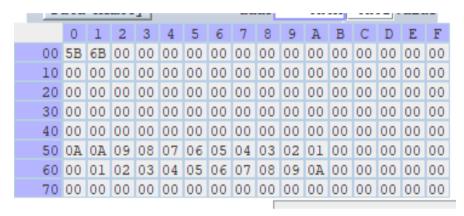
MOV @R1,A; MOV @R0,B; INC R0; INC R1;

DJNZ R4, repeat;

SJMP \$;

END;

Before:



After:

