

Assignment-3

Q1 Implement an assembly language program to check whether 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;
```

The screenshot shows the Edsim51 microsimulator interface. The left pane displays the 8051 register set and memory. The right pane shows the assembly code being executed.

Registers and Memory:

Register	Value
R7	0x00
R6	0x00
R5	0x00
R4	0x00
R3	0x00
R2	0x00
R1	0x07
R0	0x00
B	0x01
ACC	0x07
PSW	0x01
IP	0x00
IE	0x00
PCON	0x00
DPH	0x00
DPL	0x00
SP	0x07

Memory (Data Memory):

Address	Value
00	00
01	00
02	07
03	00
04	00
05	00
06	00
07	00
08	00
09	00
0A	00
0B	00
0C	00
0D	00
0E	00
0F	00
10	00
11	00
12	00
13	00
14	00
15	00
16	00
17	00
18	00
19	00
1A	00
1B	00
1C	00
1D	00
1E	00
1F	00
20	00
21	00
22	00
23	00
24	00
25	00
26	00
27	00
28	00
29	00
2A	00
2B	00
2C	00
2D	00
2E	00
2F	00
30	00
31	00
32	00
33	00
34	00
35	00
36	00
37	00
38	00
39	00
3A	00
3B	00
3C	00
3D	00
3E	00
3F	00
40	00
41	00
42	00
43	00
44	00
45	00
46	00
47	00
48	00
49	00
4A	00
4B	00
4C	00
4D	00
4E	00
4F	00
50	07
51	01
52	00
53	00
54	00
55	00
56	00
57	00
58	00
59	00
5A	00
5B	00
5C	00
5D	00
5E	00
5F	00
60	00
61	00
62	00
63	00
64	00
65	00
66	00
67	00
68	00
69	00
6A	00
6B	00
6C	00
6D	00
6E	00
6F	00
70	00
71	00
72	00
73	00
74	00
75	00
76	00
77	00
78	00
79	00
7A	00
7B	00
7C	00
7D	00
7E	00
7F	00

Assembly Code:

```

ORG 0000H;
0000| MOV R1,#02H;
0002| check:MOV A,50H;
0004| MOV B,R1;
0006| DIV AB;
0007| MOV A,B;
0009| JZ notPrime;
000B| INC R1;
000C| MOV A,R1;
000D| CJNE A,50H,check;
0010| MOV 51H,#1;
0013| SJMP $;
0015| notPrime: MOV 51H,#0;
0018| 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;

The screenshot displays the 8051 simulator interface. On the left, the register file shows R0-R7, ACC, PSW, IE, PCON, DPH, DPL, and SP. The PC is 0x000E. The memory window shows a table of memory addresses and values. On the right, the assembly code is displayed, showing the execution of the program.

System Clock (MHz): 12.0
Update Freq. 1

8051

Modify RAM
Data Memory
addr 0x57 0x0D value

Remove All Breakpoints

Assembly Code:

```
ORG 0000H;  
0000| MOV R1,#50H;  
0002| MOV R0,#60H;  
0004| MOV @R0,#0H;  
0006| MOV A,@R1;  
0007| MOV R4,A;  
0008| repeat: INC R1;  
0009| MOV A,@R0;  
000A| ADD A,@R1;  
000B| MOV @R0,A;  
000C| DJNZ R4,repeat;  
000E| 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;

```

The screenshot displays the 8051 SIM501 simulator interface. The top section shows the System Clock (MHz) set to 12.0 and the Update Freq. dropdown set to 1. Below this, the SBUF register is shown with R/O and W/O bits set to 0x00. The I/O pins (RXD, TXD, SC0N) are also shown with their respective values. The main register window displays the 8051 registers: R0-R7, ACC, PSW, IP, IE, PCON, DPH, DPL, SP, TH0, TL0, TMOD, TCON, TH1, TL1, PC, and P0-P3. The PC register is highlighted with the value 0x000D. The Data Memory window shows a table of memory addresses (00-70) and their corresponding values. The assembly code window on the right shows the following code:

```

0000 MOV R0,#50H;
0002 MOV A,@R0;
0003 MOV R4,A;
0004 INC R0;
0005 MOV R1,#61H;
0007 repeat: MOV A,@R0;
0008 MOV @R1,A;
0009 INC R0;
000A INC R1;
000B DJNZ R4,repeat;
000D 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;

```

The screenshot displays the 8051 simulator interface. On the left, the register file is shown with values for R0 through R7, ACC, PSW, IP, IE, PCON, DPH, DPL, and SP. The PC (Program Counter) is highlighted with the value 8051. Below the registers is a memory window showing a table of addresses and values. On the right, the assembly code is displayed, showing the execution of the program.

System Clock (MHz): 12.0
 SBUF: 0x00
 R/O: 0x00, W/O: 0x00
 RXD: 1, TXD: 1
 SC0N: 0x00
 TH0: 0x00, TL0: 0x00
 TMOD: 0x00, TCON: 0x00
 pins: 0xFF, bits: 0xFF
 P3: 0xFF, P2: 0xFF, P1: 0xFF, P0: 0xFF
 TH1: 0x00, TL1: 0x00
 PC: 0x000E
 PSW: 000001
 R7: 0x00, R6: 0x00, R5: 0x00, R4: 0x00, R3: 0x00, R2: 0x00, R1: 0x63, R0: 0x60
 B: 0x00, ACC: 0x01, PSW: 0x01, IP: 0x00, IE: 0x00, PCON: 0x00, DPH: 0x00, DPL: 0x00, SP: 0x07

Modify RAM
 Data Memory
 addr: 0x60, value: 0x00

Copyright ©2005-2022 James Rogers
 Remove All Breakpoints

Assembly Code:

```

0000 MOV R0,#50H;
0002 MOV A,@R0;
0003 MOV R4,A;
0004 MOV R0,#6AH;
0006 MOV R1,#6DH;
0008 repeat: MOV A,@R0;
0009 MOV @R1,A;
000A DEC R0;
000B DEC R1;
000C DJNZ R4,repeat;
000E 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:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	5B	6B	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	0A	0A	09	08	07	06	05	04	03	02	01	00	00	00	00	00
60	00	01	02	03	04	05	06	07	08	09	0A	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

After:

System Clock (MHz) 12.0
SBUF
R/O W/O TH0 TL0 R7 0x00 B 0x0A
0x00 0x00 0x00 0x00 R6 0x00 ACC 0x01
RXD TXD 1 1 TMOD 0x00 R5 0x00 PSW 0x01
SCON 0x00 TCON 0x00 R4 0x00 IP 0x00
pins bits TH1 TL1 R3 0x00 IE 0x00
0xFF 0xFF P3 0x00 0x00 R2 0x00 PCON 0x00
0xFF 0xFF P2 8051 DPH 0x00
0xFF 0xFF P1 PC 0x0011 PSW 0 0 0 0 0 0 0 1
0xFF 0xFF P0 Modify RAM
Data Memory addr 0x6A 0x01 value

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	5B	6B	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	0A	01	02	03	04	05	06	07	08	09	0A	00	00	00	00	00
60	00	0A	09	08	07	06	05	04	03	02	01	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

RST Step Run New Load Save Copy Paste X
Time: 161us - Instructions: 103
0000 MOV R0,#50H;
0002 MOV A,@R0;
0003 MOV R4,A;
0004 INC R0;
0005 MOV R1,#61H;
0007 repeat: MOV A,@R0;
0008 MOV B,@R1;
000A MOV @R1,A;
000B MOV @R0,B;
000D INC R0;
000E INC R1;
000F DJNZ R4,repeat;
0011 SJMP \$;
END;

Copyright ©2005-2022 James Rogers
Remove All Breakpoints