

SET 1

Write an 8051 assembly level program to add 10 bytes of data.

Before execution:

The screenshot shows the Keil uVision IDE interface. At the top, the title bar reads "C:\Keil_v5\CS1\Examples\HELLO_ECE_Juxproj - uVision". The menu bar includes File, Edit, View, Project, Flash, Debug, peripherals, Tools, SVCS, Window, and Help. Below the menu is a toolbar with various icons for file operations, editing, and debugging. The main workspace is divided into three panes. The left pane, titled "Registers", displays a list of registers and their current values. The central pane shows assembly code for a program named "1P1.asm", which implements a loop to add 10 bytes of data. The right pane, titled "Memory 1", shows a memory dump starting at address 0x30h, displaying hexadecimal and ASCII representations of memory contents. The status bar at the bottom indicates the simulation time as 0.000000000 sec and the current instruction as L1 C27.

After execution:

C:\Keil_u5\CS1\Examples\HELLO\ECE_Uproj - uVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
r0	0x00
r1	0x3a
r2	0x3d
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x3d
b	0x00
np	0x07
np_max	0x07
PC	0x0010
auxr1	0x00
dptr	0x0000
states	65
sec	0.000023
psw	0x01

```

1 // to add 10 bytes of data
2
3     org 00h
4     mov r0, #0ah ; count register
5     mov r1, #30h ; pointer to data
6     mov r2, #00h ; to store answer
7     mov r3, #00h ; to store carry
8 repeat: add a, #r1
9         jnc next
10        inc r3
11 next:  inc r1
12        djnz r0, repeat
13        mov r2, a
14        end

```

Project Fund... Regis...

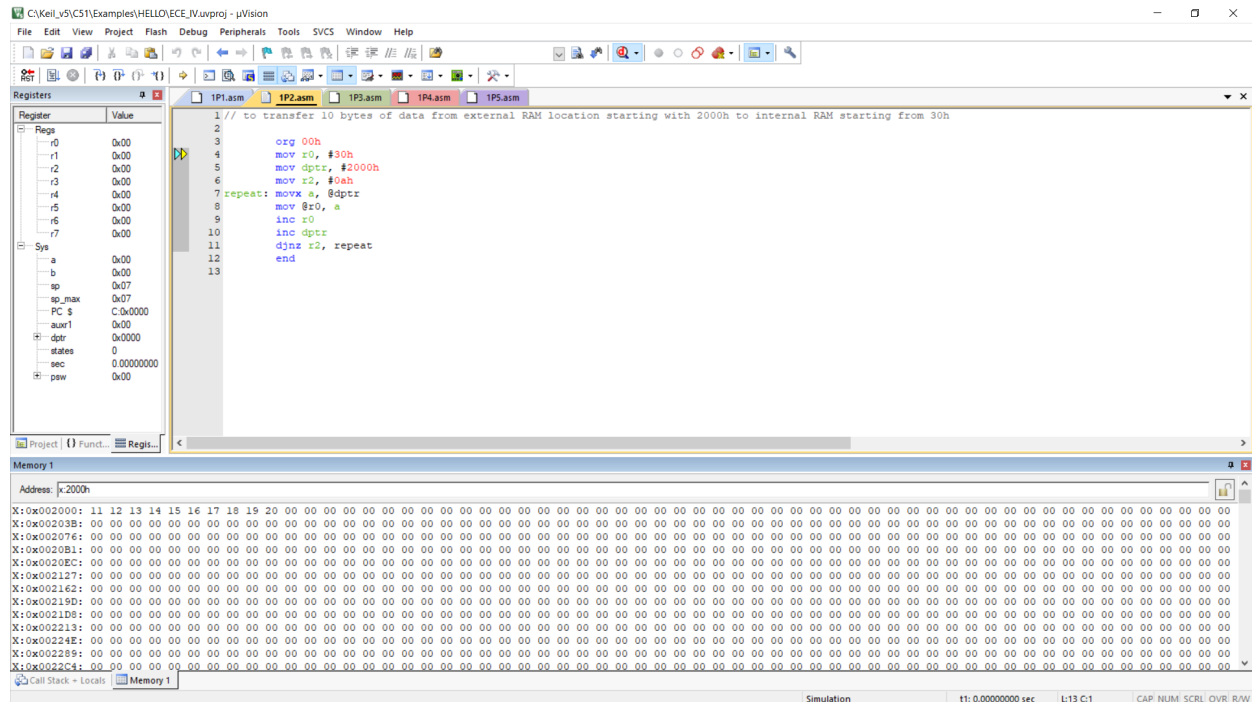
Memory 1

Address: 0x30h

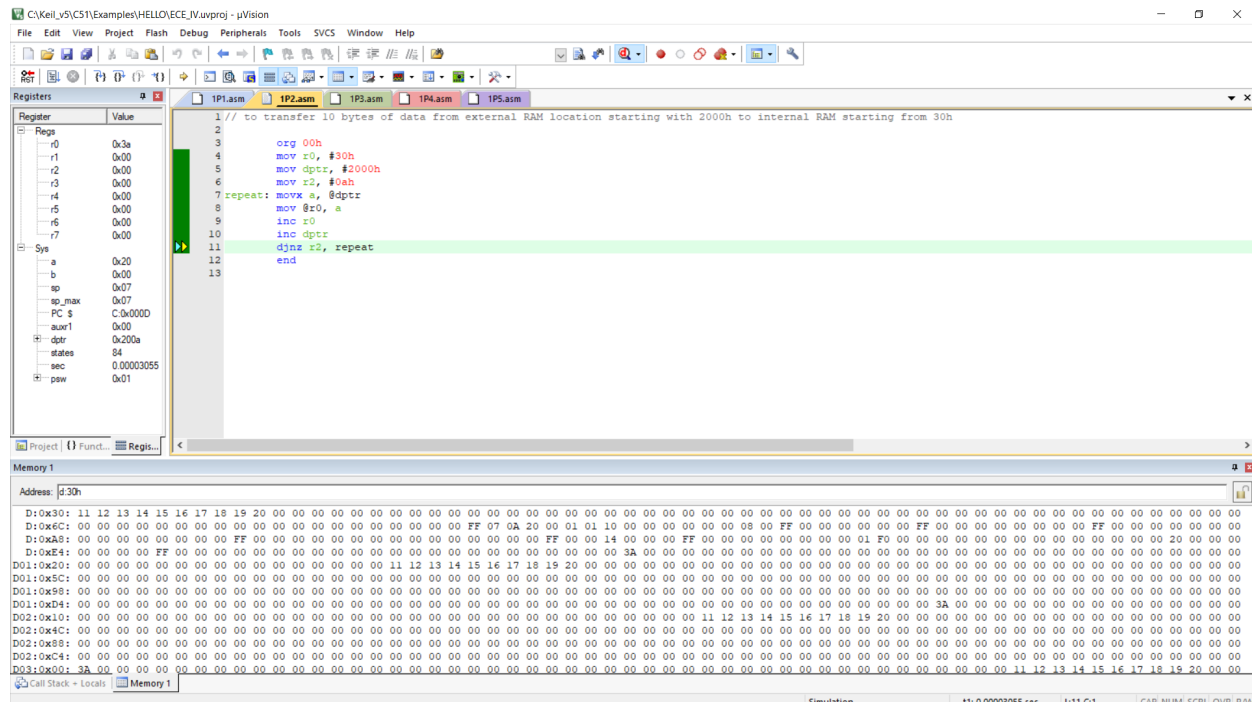
D1:0x30:	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	4
----------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---

Write an 8051 assembly level program to transfer 10 bytes of data from external RAM location starting with 2000h to internal RAM starting from 30h.

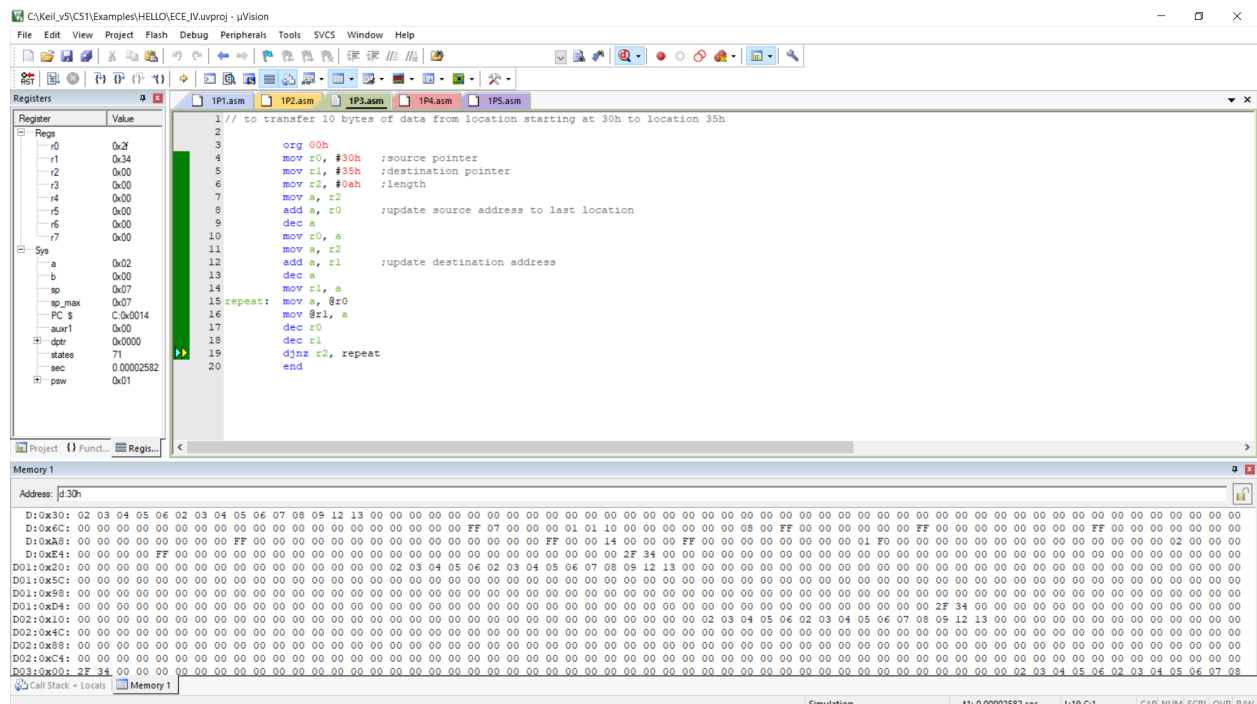
Before execution:



After execution:

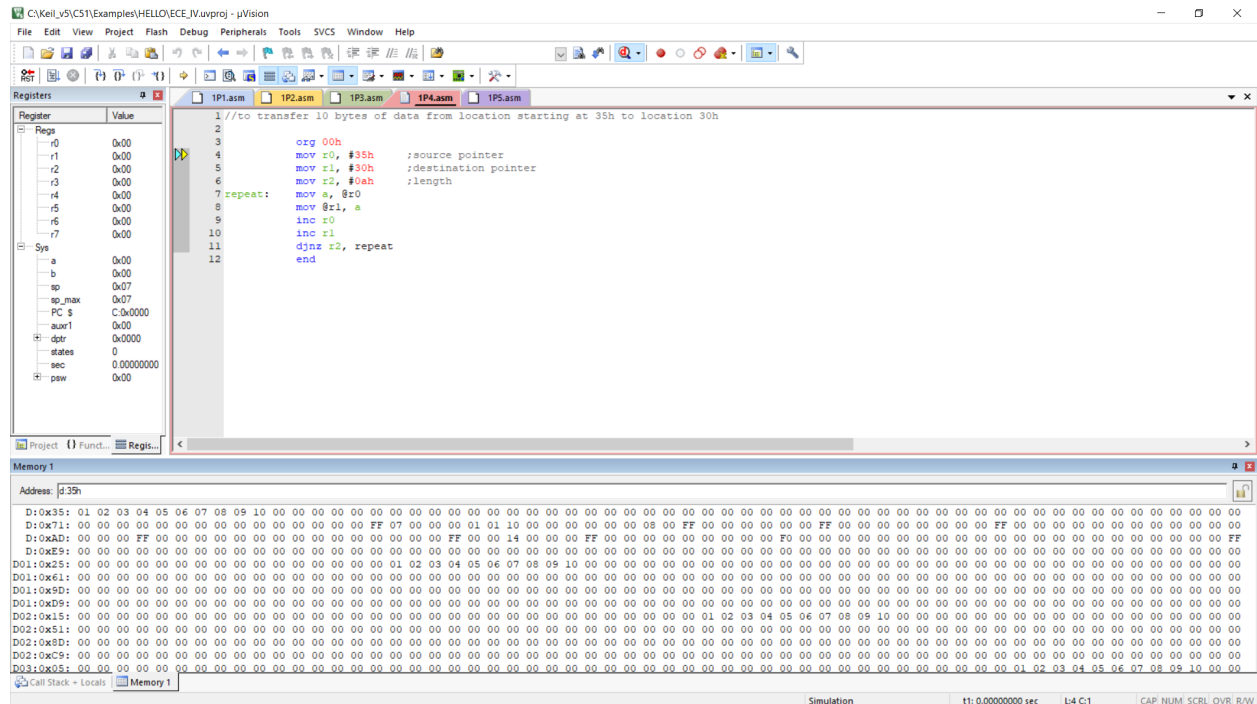


Before execution:

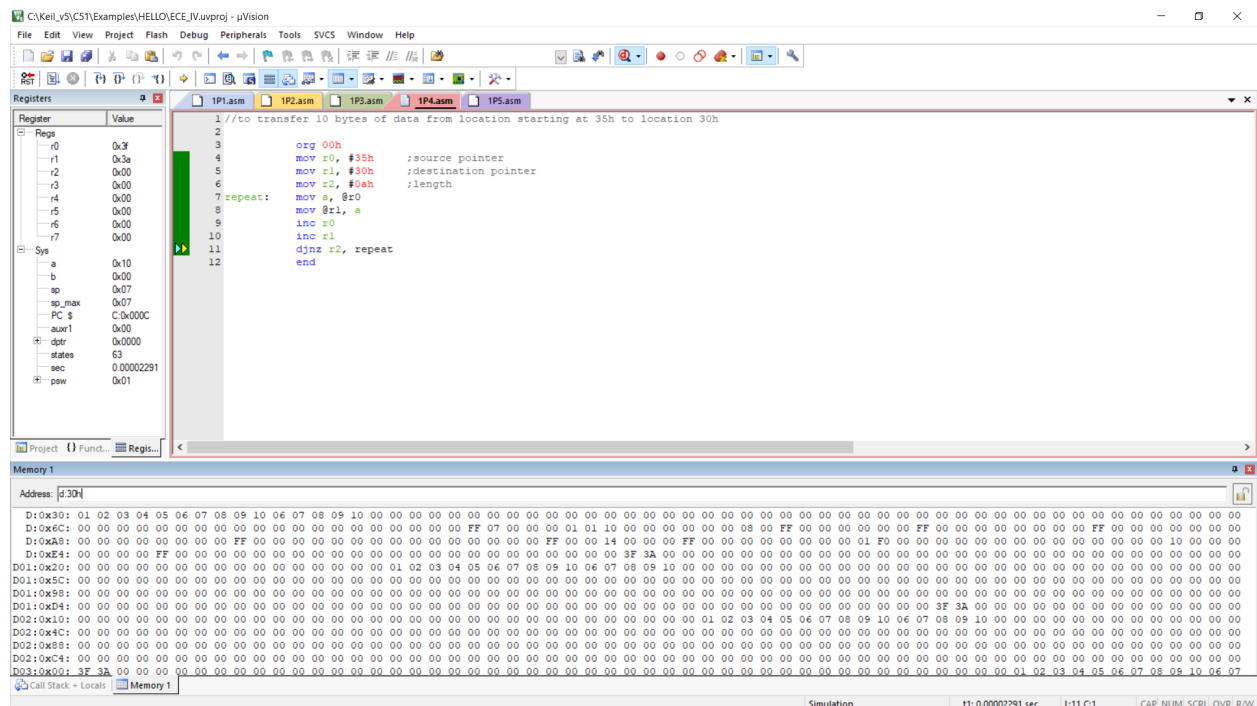


Write an 8051 assembly level program to transfer 10 bytes of data from location starting at 35h to location 30h.

Before execution:

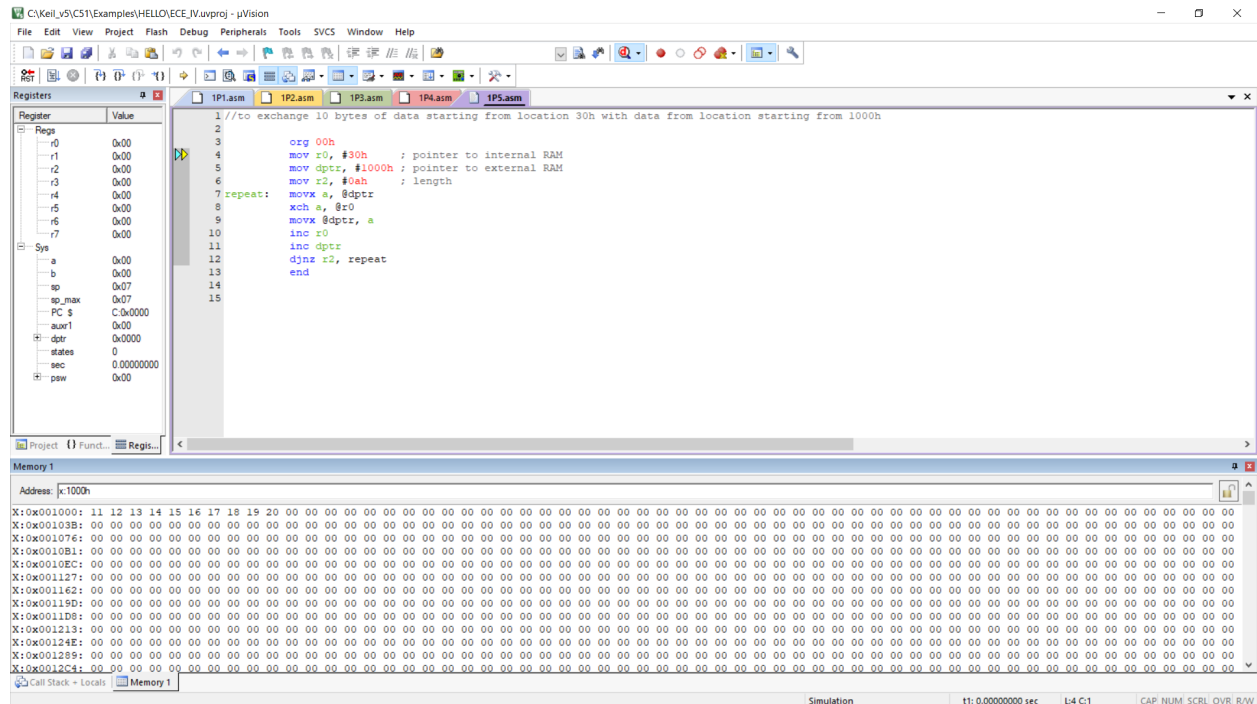


After execution:

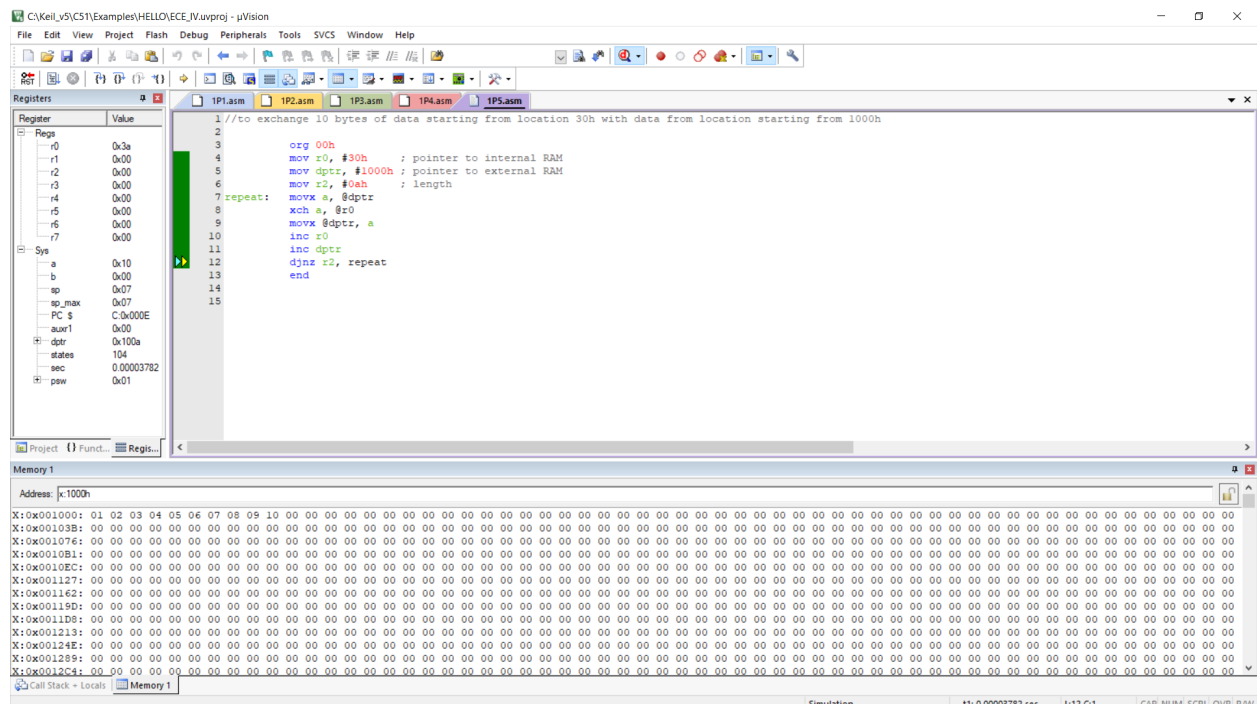


Write an 8051 assembly level program to exchange 10 bytes of data from location starting at 30h with data from location starting from 1000h

Before execution:



After execution:



Write an 8051 assembly level program to transfer 10 bytes of data starting from location 8000h to location 9000h within the external memory

Before execution:

The screenshot shows the Keil uVision IDE with the assembly code for 1P1.asm. The code is as follows:

```
1 // 10 bytes of data from location 8000h to 9000h within the external memory
2
3 org 0000h
4 mov r0, #0ah
5 mov dpl, #00h
6 repeat: mov dph, #80h
7 movx a, @dptr
8 mov dph, #90h
9 movx @dptr, a
10 inc dpl
11 djnz r0, repeat
12 end
```

The registers window shows the following values:

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x00
b	0x00
sp	0x07
sp_max	0x07
PC	0x0000
aux1	0x00
dptr	0x0000
status	0
sec	0.00000000
psw	0x00

The memory window shows the memory dump starting at address 8000h, with all bytes being 00.

After execution:

The screenshot shows the Keil uVision IDE with the assembly code for 1P1.asm. The code is as follows:

```
1 // to add N bytes of data taking into account the possible carry output
2
3 org 00h
4 mov r0, #20h ; count register
5 mov r1, #30h ; pointer to data
6 mov r2, #00h ; to store answer
7 mov r3, #00h ; to store carry
8 repeat: add a, @r1
9 jnc next
10 inc r3
11 next: inc r1
12 djnz r0, repeat
13 mov r2, a
14 end
```

The registers window shows the following values:

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x10
b	0x00
sp	0x07
sp_max	0x07
PC	0x0011
aux1	0x00
dptr	0x900a
status	113
sec	0.00004109
psw	0x01

The memory window shows the memory dump starting at address 8000h, with all bytes being 00.