

# Variables, What are They?

CS143: lecture 3

Byron Zhong, June 15

# Variables

## Review



Memory Unit



# Variables

## Review

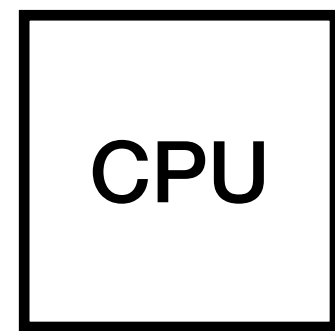
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
320	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
350	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
380	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
410	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
440	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
470	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
500	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
530	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
560	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
590	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
620	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

# Variables

## Review

```
int number;  
number = 10;  
number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

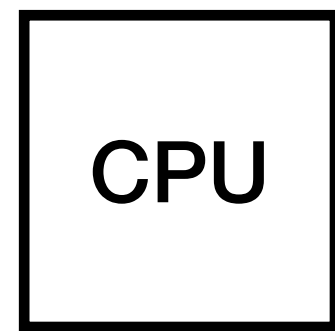


# Variables

## Review

```
→ int number;  
   number = 10;  
   number = number + 5;  
  
   int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

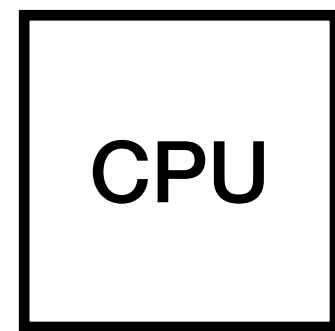


# Variables

## Review

```
int number;  
→ number = 10;  
number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	10				210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



write 10 to 246



# Variables

## Review

```
int number;  
number = 10;  
→ number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	10				210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



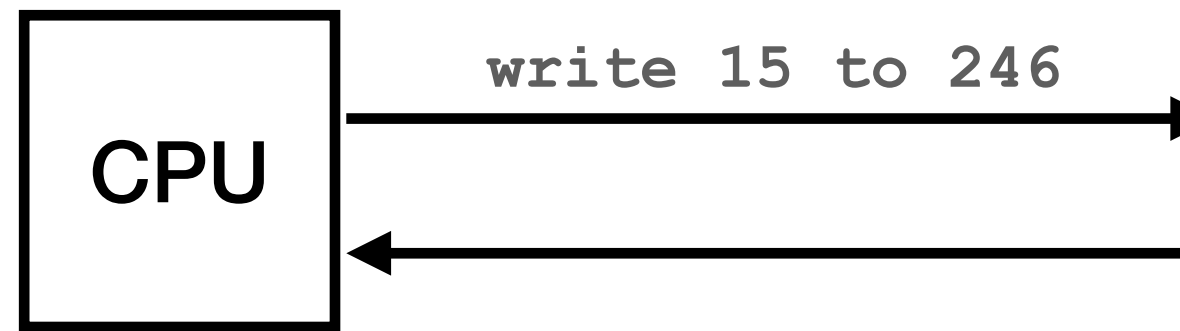


# Variables

## Review

```
int number;  
number = 10;  
→ number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	15				210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



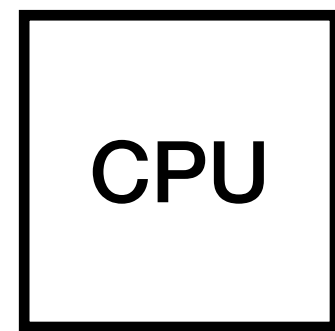
# Variables

## Review

```
int number;  
number = 10;  
number = number + 5;
```

→ `int another = number * 2;`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	15				210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
320	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
350	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
380	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
410	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
440	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
470	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
500	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
530	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
560	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
590	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
620	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



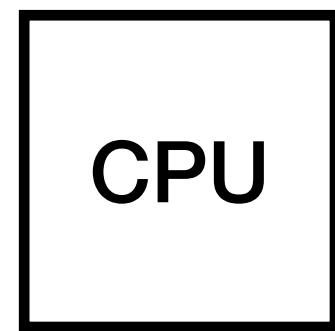
# Variables

## Review

```
int number;  
number = 10;  
number = number + 5;
```

→ `int another = number * 2;`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	15				210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	30				125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



write 30 to 399



# Variables

## In Memory

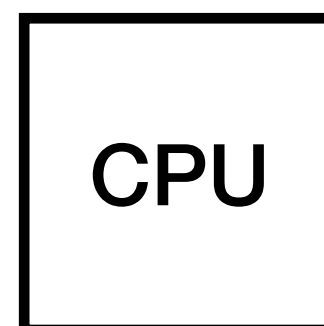
- Variable is a named location in memory.
- Compiler keeps a table of variable names and their locations.
- A declaration (`int number;`) tells the compiler to find a space in memory.
- But how much space?

# Variables

## In Memory

```
int number;  
number = 10;  
→ number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	10				210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



read from 246

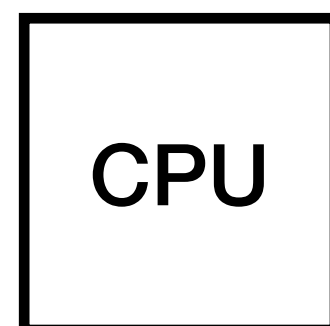
10

# Variables

## In Memory

```
int number;  
number = 10;  
→ number = number + 5;  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	0	0	0	10	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



read ? from 246

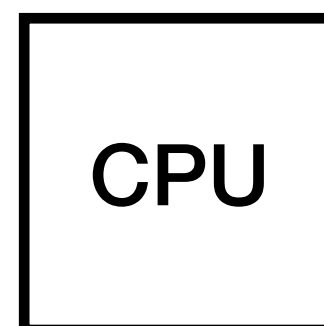
10

# Variables

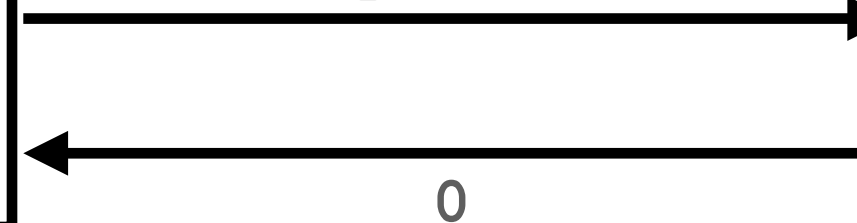
## In Memory

```
int number;  
number = 10;  
→ number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	0	0	0	10	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



read 2 bytes from 246



# Variables

## In Memory

```
int number;  
number = 10;  
→ number = number + 5;  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	0	0	0	10	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



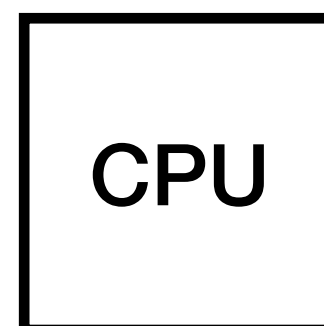


# Variables

## In Memory

```
int number;  
number = 10;  
→ number = number + 5;  
  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	0	0	0	10	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



read 8 bytes from 246

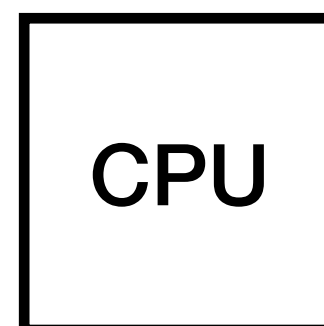
1021012111998

# Variables

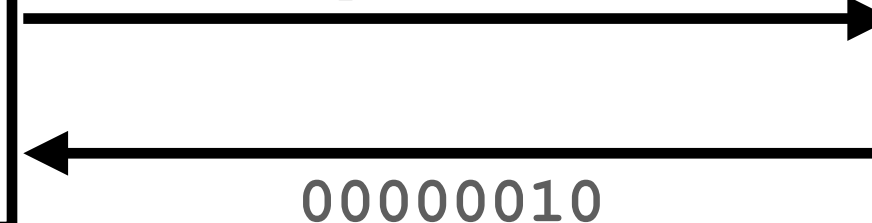
## In Memory

```
int number;  
number = 10;  
→ number = number + 5;  
int another = number * 2;
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	0	0	0	10	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



read 4 spaces from 246



# Variables

## In Memory

- A “location” is a point
- The type `int` specifies the size of the variable in memory, and compiler uses this size to
  - find space
  - read (n bytes after the location) from memory
  - write (n bytes after the location) to memory
- One variable, one type

# Variables

## C Types and Their Sizes

type name	size (bytes)
char	1
short	2
int	4
long	8
float	4
double	8

} These are numbers with  
decimal points

# Variables

## In Memory

- Types also tell the compiler how to interpret the number
  - Is it a floating point number?
  - Is it an integer?
  - Is it negative?
- ...more on this later

# Variables

## In Memory

- Variable is a named location in memory.
- Compiler keeps a table of names and their locations **and types**.
- A declaration (`int number;`) tells the compiler to find a space **enough for 4 bytes** in memory.
- But how much space? — **Depends on the type!**
- For this reason, we want one variable, one type.

# Variables

## C Types and Their Sizes

- Why do we need numbers with different sizes?
- They have different ranges. We want to be space-efficient.
- `char` is a number?
- Yes, `char` is a number.

# Variables

## American Standard Code for Information Interchange

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL	32	SPACE	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

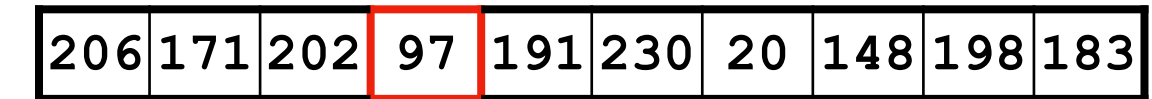


# Variables

## American Standard Code for Information Interchange

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL	32	SPACE	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

char c = 'a';



# Variables

## American Standard Code for Information Interchange

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL	32	SPACE	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

'b' - 'a' == 1      Equivalent      98 - 97 == 1  
int x = 'a';      ←————→      int x = 97;

- char is a number.
- Fundamental difference between 'a' and "a"
- 'a' is just a number translated according to the ASCII table
  - In Python, `ord('a')` and `chr(97)`
- "a" is an array of chars.
- Don't write 97 when you could write 'a', much clearer

# Variables

## How about a lot of variables?

- Say you want to find space for 26 integers

# Variables

## How about a lot of variables? Naive Solution

```
int a0 = 0;
int a1 = 0;
int a2 = 0;
int a3 = 0;
int a4 = 0;
int a5 = 0;
int a6 = 0;
int a7 = 0;
int a8 = 0;
int a9 = 0;
int a10 = 0;
int a11 = 0;
int a12 = 0;
int a13 = 0;
int a14 = 0;
int a15 = 0;
int a16 = 0;
int a17 = 0;
int a18 = 0;
int a19 = 0;
int a20 = 0;
int a21 = 0;
int a22 = 0;
int a23 = 0;
int a24 = 0;
int a25 = 0;
```

```
int a0 = 0, a1 = 0, a2 = 0, a3 = 0, a4 = 0, a5 = 0, a6 = 0, a7 = 0,
    a8 = 0, a9 = 0, a10 = 0, a11 = 0, a12 = 0, a13 = 0, a14 = 0, a15 = 0,
    a16 = 0, a17 = 0, a18 = 0, a19 = 0, a20 = 0, a21 = 0, a22 = 0, a23 = 0,
    a24 = 0; a25 = 0;
```

- What's wrong with this?

```
if (c == 'a') {
    a0 += 1;
} else if (c == 'b') {
    a1 += 1;
} else if (c == 'c') {
    a2 += 1;
} else if (c == 'd') {
    a3 += 1;
} else if (c == 'e') {
    ...
}
```

# Variables

## How about a lot of variables?

- Say you want to find space for 26 integers
- `int number[26];` declares 26 integers!
- Compiler will find a block of space large enough for 26 integers
  - $26 * \text{sizeof}(\text{int}) = 26 * 4 = 104$  bytes!
- `number[3]` accesses the fourth integer in the block

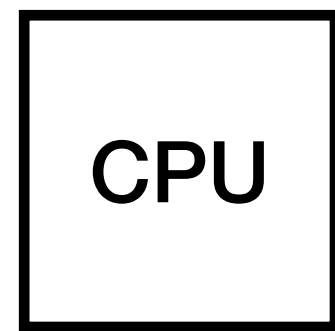
# Variables

## In Memory

→ `int number[4];`

`number[2] = 2;`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



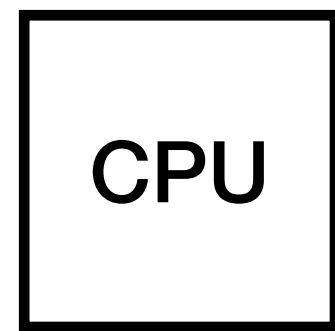
# Variables

## In Memory

→ `int number[4];`

`number[2] = 2;`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



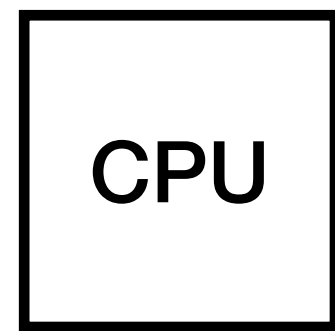
# Variables

## In Memory

```
int number[4];
```

→ number[2] = 2;

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	0	0	0	2	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36





# Variables

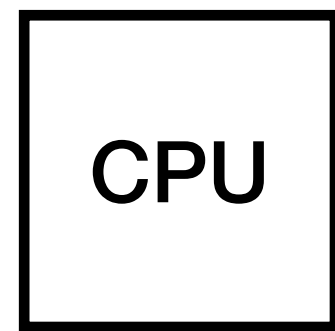
## In Memory

```
int number[4];
```

→ `number[2] = 2;`

```
printf("%d", number[2]);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	0	0	0	2	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



# Variables

## In Memory

```
int number[4];
```

```
number[2] = 2;
```

```
→ printf("%d", number[2]);
```

number starts at 212  
You want number[2]  
Each int is 4 bytes  
So, number[2] starts at  
 $212 + 2 * 4 = 220$

CPU

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	0	0	0	2	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

# Variables

## In Memory

```
int number[4];
```

```
number[2] = 2;
```

```
→ printf("%d", number[2]);
```

number starts at 212  
You want number[2]  
Each int is 4 bytes  
So, number[2] starts at  
 $212 + 2 * 4 = 220$

CPU

read 4 bytes from 220

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	0	0	0	2	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

# Variables

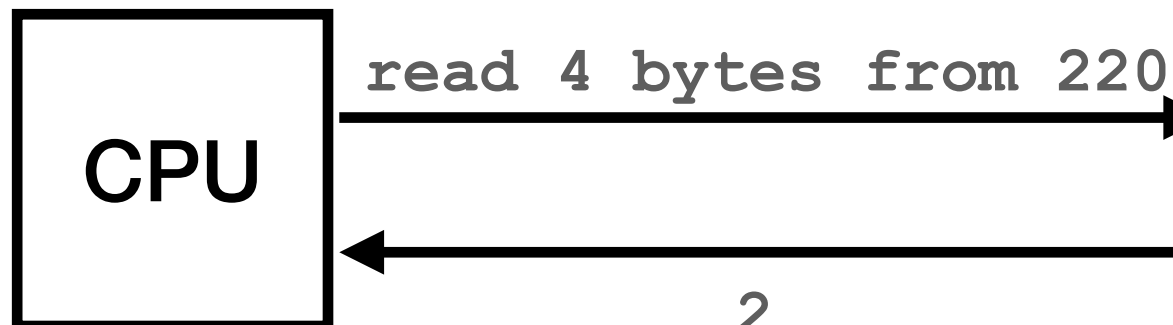
## In Memory

```
int number[4];
```

```
number[2] = 2;
```

```
→ printf("%d", number[2]);
```

number starts at 212  
You want number[2]  
Each int is 4 bytes  
So, number[2] starts at  
 $212 + 2 * 4 = 220$



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	0	0	0	2	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

# Variables

## How about a lot of variables?

- `int number[26];` declares an *array* of 26 integers.
- An array is a *contiguous* block of elements of the *same type* (element type).
- To get a specific element  $i$ , (`number[i]`), read from (starting location +  $i$  \* size of element type)
- Sanity check: why fixed length?
- Sanity check: why same type?
- Sanity check: why does index start with 0?

# Interlude: I/O

## getchar/putchar

- In library: `#include <stdio.h>`
- `c = getchar()`; reads 1 character from the standard input
- `putchar(c)`; prints `c` to the standard output.
- **Caveat:** `int getchar(void)`;
  - Why does `getchar` return an `int`??
  - Needs to signal EOF
  - `getchar` needs to return all possible character + 1 special character EOF
  - `char` is not enough, so we use `int`

# Interlude: I/O

## `printf`

- `printf` is an output formatting function.
  - `printf("format", v1, v2, v3, ...);`
  - Format string contains characters and placeholders starting with %
- `printf("hello: %s %c %d %f\n", "world", 'a', 2, 2.5);`
  - `hello: world a 2 2.5`
  - note the `\n` character

# Interlude: I/O

## printf

- Some common specifiers

Specifier	Value Type
%d	int
%c	char
%hd	short
%s	string***
%f	float
%f	double



# Interlude: I/O

## `printf`

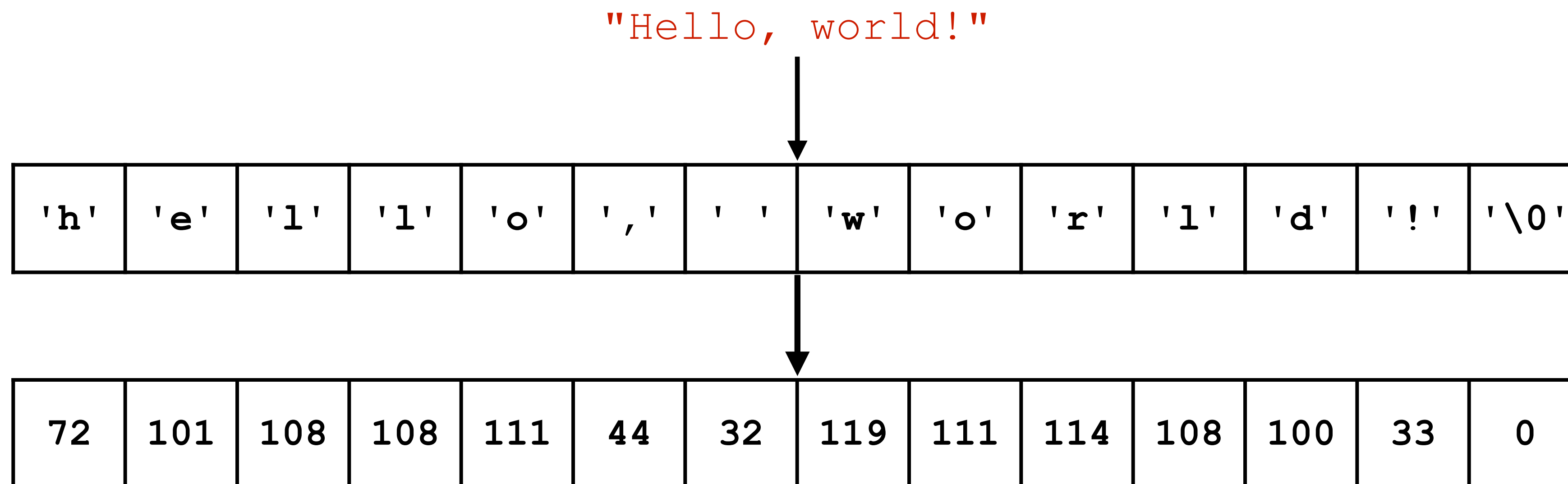
- Fancier formats also available

Specifier	Format
<code>%d</code>	as is
<code>%6d</code>	at least 6 characters wide
<code>%-6d</code>	at least 6 characters wide, left adjusted
<code>%06d</code>	at least 6 characters wide, leading 0
<code>%f</code>	double

# Variables

## Strings

- String is an array of characters with a special character (' \0 ') at the end

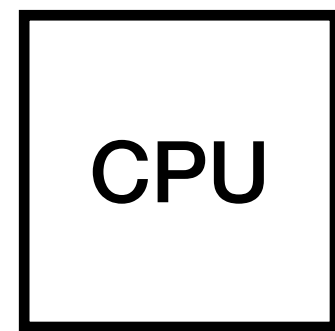


# Variables

## Strings

```
→ char s[14] =  
    "Hello, world!";  
  
printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

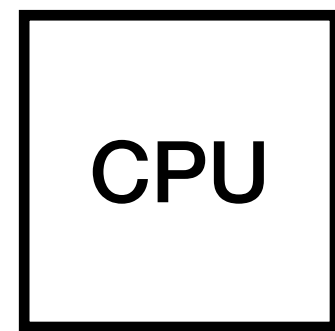


# Variables

## Strings

```
→ char s[14] =  
    "Hello, world!";  
  
printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	82	110	244	93	183	172	154	137	57	50	158	107	81	67	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

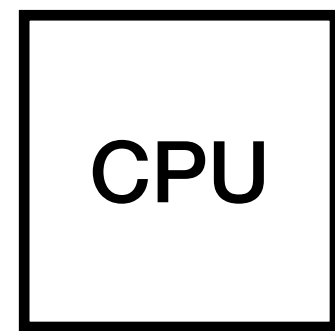


# Variables

## Strings

```
→ char s[14] =  
    "Hello, world!";  
  
printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



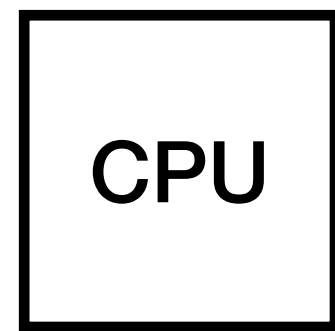
# Variables

## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



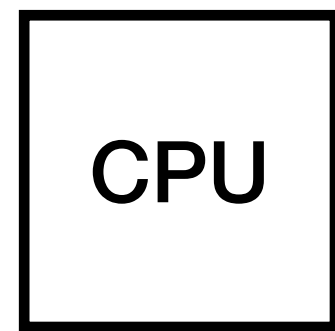
# Variables

## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36



# Variables

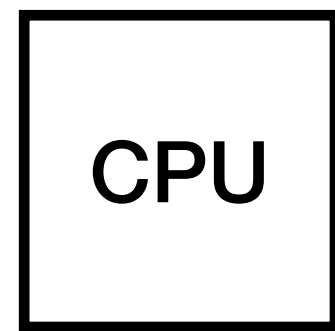
## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

He





# Variables

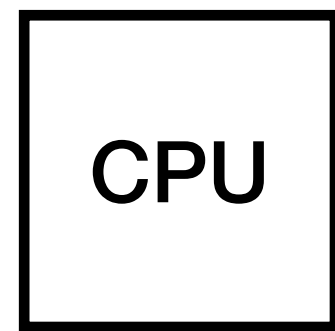
## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hel



# Variables

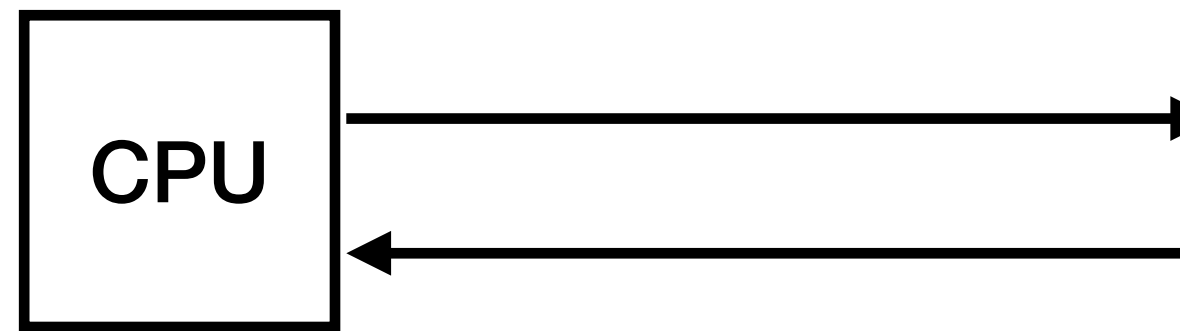
## Strings

```
char s[14] =  
    "Hello, world!";
```

→ `printf("%s\n", s);`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hell



# Variables

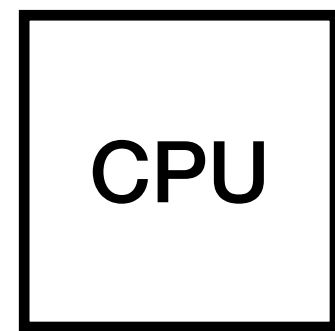
## Strings

```
char s[14] =  
    "Hello, world!";
```

→ `printf("%s\n", s);`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hello



# Variables

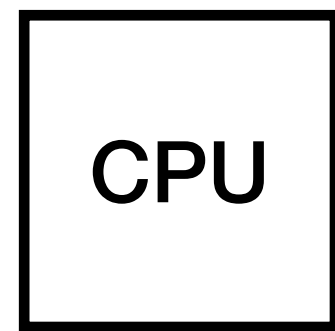
## Strings

```
char s[14] =  
    "Hello, world!";
```

→ `printf("%s\n", s);`

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hello,



# Variables

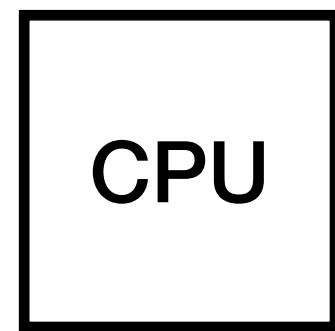
## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hello,



# Variables

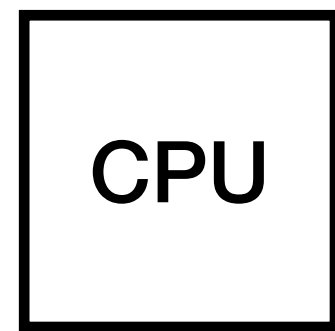
## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hello, world!



# Variables

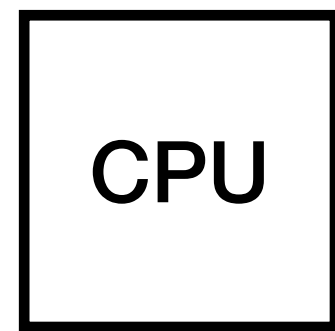
## Strings

```
char s[14] =  
    "Hello, world!";
```

```
→ printf("%s\n", s);
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	11	119	115	102	115	31	58	127	49	108	161	128	120	139	49	138	165	8	173	1	167	160	18	148	131	240	49	227	116	84
30	28	77	126	184	77	46	7	95	127	117	145	164	77	6	98	12	232	252	238	143	39	49	225	210	142	227	28	182	82	81
60	119	209	67	25	13	220	39	84	249	190	14	10	23	72	200	64	173	216	129	195	171	37	251	217	205	210	229	41	198	78
90	99	36	153	109	28	194	94	28	170	94	224	34	24	87	70	112	107	213	48	195	164	117	202	169	104	238	240	56	30	142
120	220	135	65	195	236	211	117	131	206	198	96	83	142	32	50	165	64	177	18	205	210	143	210	165	221	227	238	132	14	131
150	4	103	109	183	254	173	250	88	34	47	176	173	193	72	233	103	233	174	234	98	25	130	10	149	230	103	54	59	107	99
180	142	20	16	142	38	19	198	85	164	89	134	171	213	75	8	112	20	197	29	204	64	210	47	244	188	237	99	150	207	109
210	168	161	208	83	103	221	140	239	166	233	93	153	173	180	69	27	69	136	210	93	80	58	47	102	1	41	67	21	74	79
240	32	63	212	256	52	168	244	19	75	228	210	121	119	98	133	226	77	43	105	72	55	227	140	169	6	98	188	222	108	98
270	97	140	8	67	239	106	155	72	10	10	10	11	44	32	11	11	11	10	10	33	0	123	227	220	240	30	119	11	119	243
300	206	46	218	50	170	218	247	34	188	65	34	247	192	117	35	67	146	26	120	150	48	166	174	33	138	50	197	120	155	163
330	114	152	68	122	254	126	76	131	131	73	183	86	144	107	232	247	41	31	32	169	97	32	146	216	197	167	90	1	20	231
360	152	107	192	60	137	191	4	240	122	76	101	251	118	197	71	183	12	89	90	10	19	190	204	17	170	178	236	108	16	26
390	206	171	202	187	191	230	20	148	198	183	243	71	136	125	204	230	72	239	151	128	63	126	205	111	74	248	95	21	170	152
420	224	250	208	211	99	72	232	60	8	38	108	5	41	15	71	195	26	67	84	139	100	173	199	206	219	60	235	141	39	208
450	183	72	75	12	99	165	165	246	0	117	204	38	181	2	129	111	121	88	28	127	191	64	244	205	76	107	103	146	147	181
480	163	161	77	44	145	98	89	166	180	151	152	99	225	169	130	73	41	238	20	17	253	139	6	166	157	45	89	138	159	34
510	203	9	70	34	43	19	169	72	160	21	250	34	197	157	113	192	27	52	26	35	84	95	39	171	99	246	234	45	214	47
540	39	195	93	109	19	185	68	206	192	224	235	187	140	68	205	129	55	41	30	143	198	222	78	25	2	199	94	102	149	0
570	150	28	18	225	182	94	46	81	198	66	252	90	110	65	161	241	36	17	151	225	130	89	67	151	160	15	195	160	212	117
600	133	40	139	147	250	50	122	4	99	11	201	8	6	225	104	153	146	103	67	1	155	165	103	158	55	89	23	110	133	58
630	118	33	145	42	247	93	21	202	196	21	203	63	248	155	96	185	138	179	150	134	139	37	36	132	140	7	211	167	11	99
660	187	210	219	52	121	39	24	125	27	61	171	110	26	169	192	6	148	249	190	24	8	9	229	107	10	244	191	166	137	27
690	167	65	33	66	149	133	173	128	151	196	207	41	108	219	46	208	13	209	105	154	19	175	135	42	63	36	143	254	32	215
720	215	40	157	139	175	235	235	92	197	203	68	109	241	194	92	231	168	173	30	206	113	97	81	150	145	163	237	208	133	36

Hello, world!



# Variables

## Strings

- String is an array of characters with a special character ( ' \ 0 ' ) at the end
- Arrays in C run on an "honor system"
  - An array is represented by its starting location
  - C doesn't keep track of the length of the array
  - Either there is something at the end (terminator) (NUL-terminator)
  - or you provide the length explicitly



# Variables

## Demo

- Demo (if we have time)
  - printf
  - wacky array access

# Variables

## In one slide

- A variable is a named location in memory.
  - A variable has a type (thereby size) and a location in memory.
- A variable needs to be *declared* before use. Syntax: `type name;`
  - The first assignment to a variable is called *initialization*.
  - A variable contains junk between declaration and initialization.
- An *array* is a *contiguous* block of elements of the *same type*.  
Syntax: `type name [number];`
  - Fixed size
  - Access/modify by index, syntax: `name [index]`. This index is not checked.
- A *string* is a NUL-terminated array of characters.

# Week 1

## Summary

- C
  - Basic syntax, control-flow, declarations, statements, ...
  - Compiling C and running executable
- Shell
  - Navigation, I/O redirection, (scripting)
  - Git
- Variables