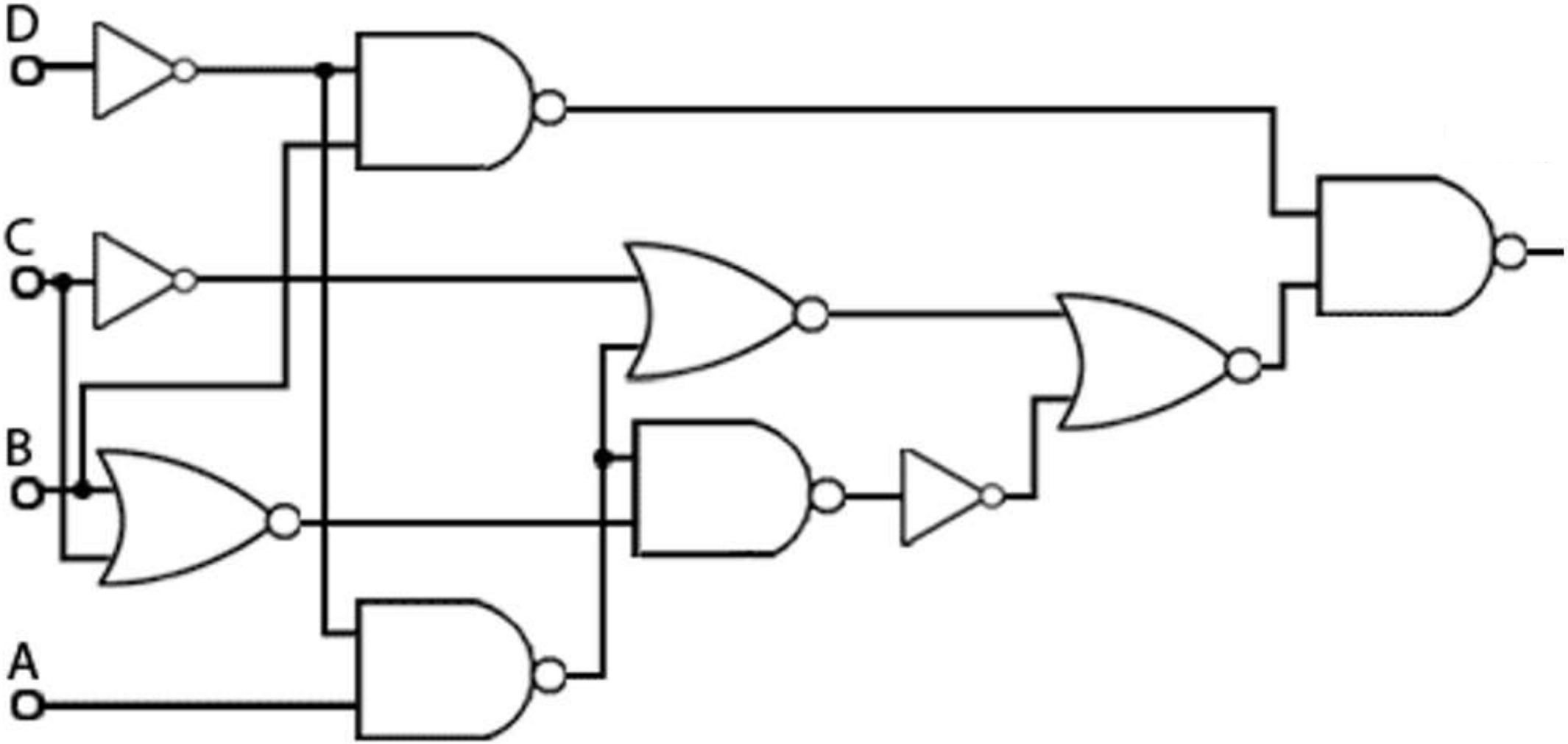
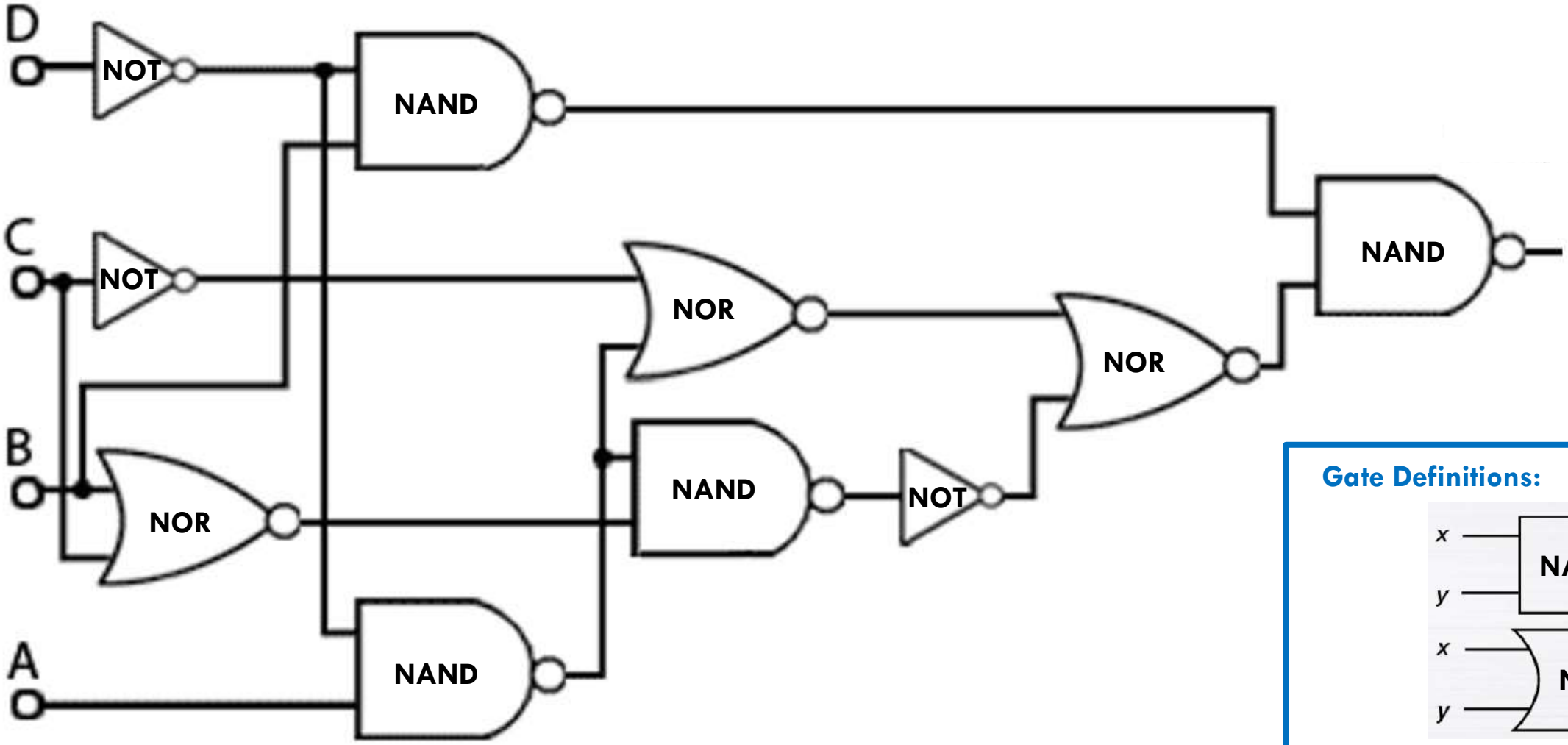
COS 284 TUTORIAL 5

SEMESTER TEST 2 RECAP

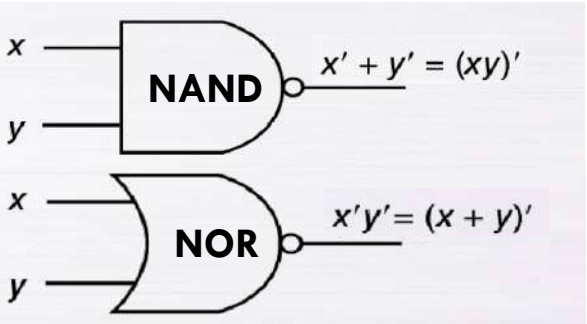
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



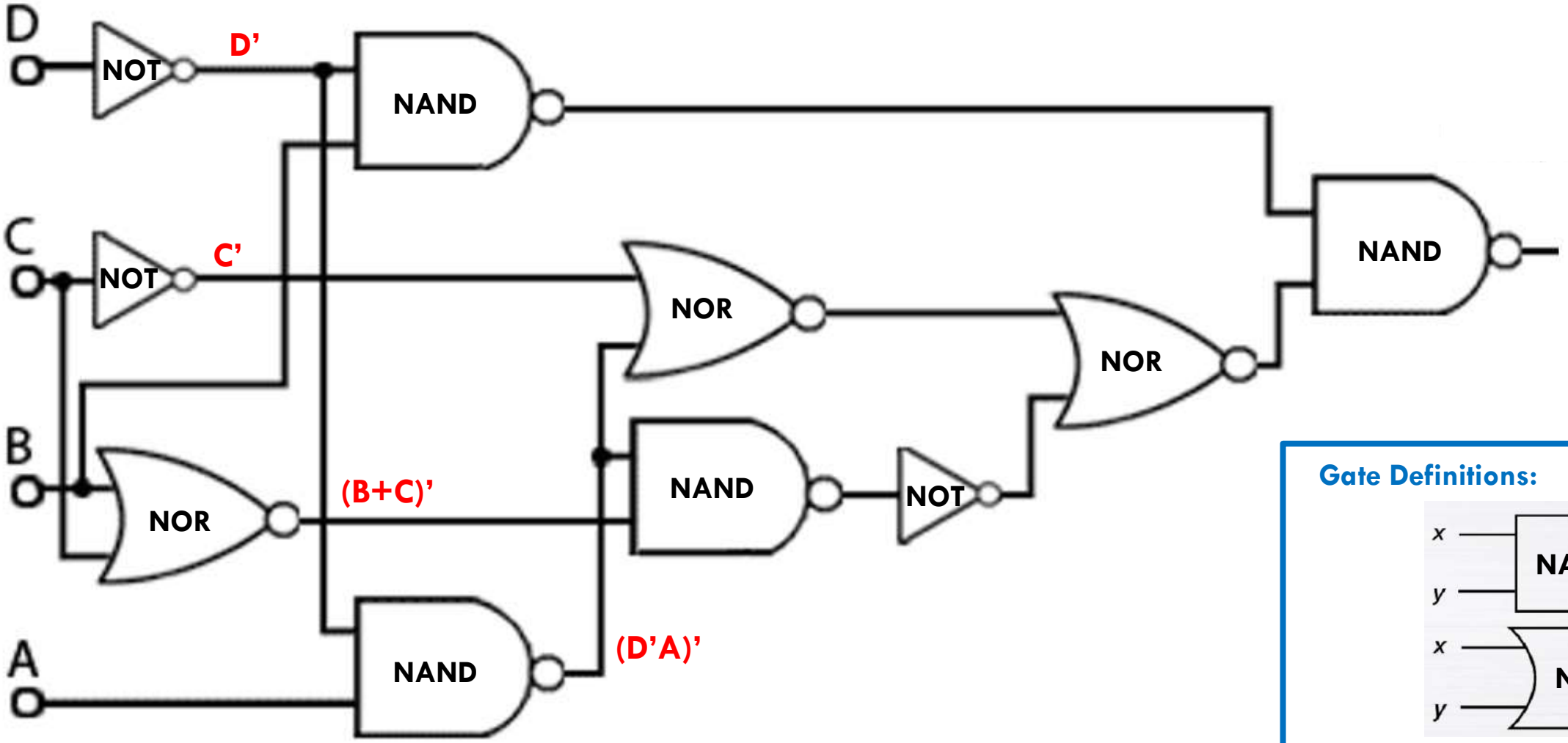
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



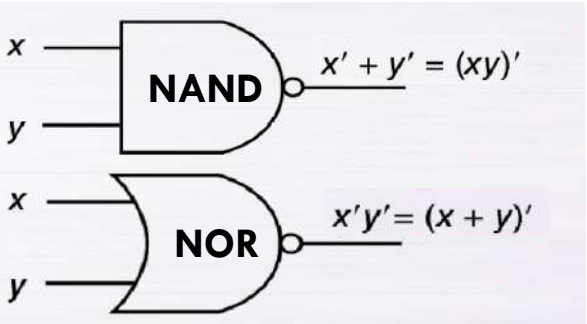
Gate Definitions:



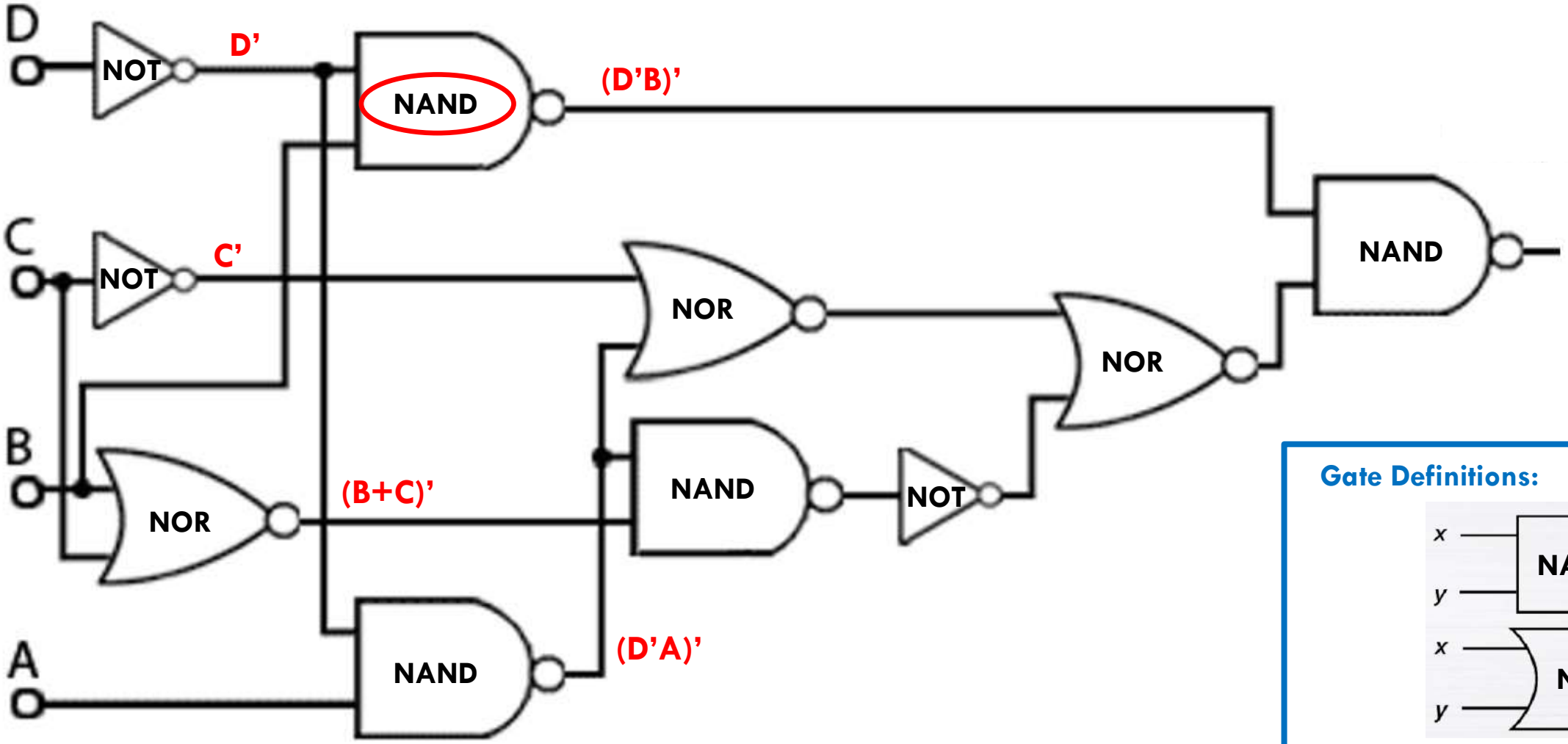
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



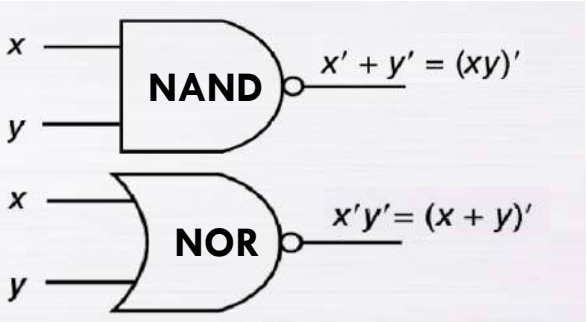
Gate Definitions:



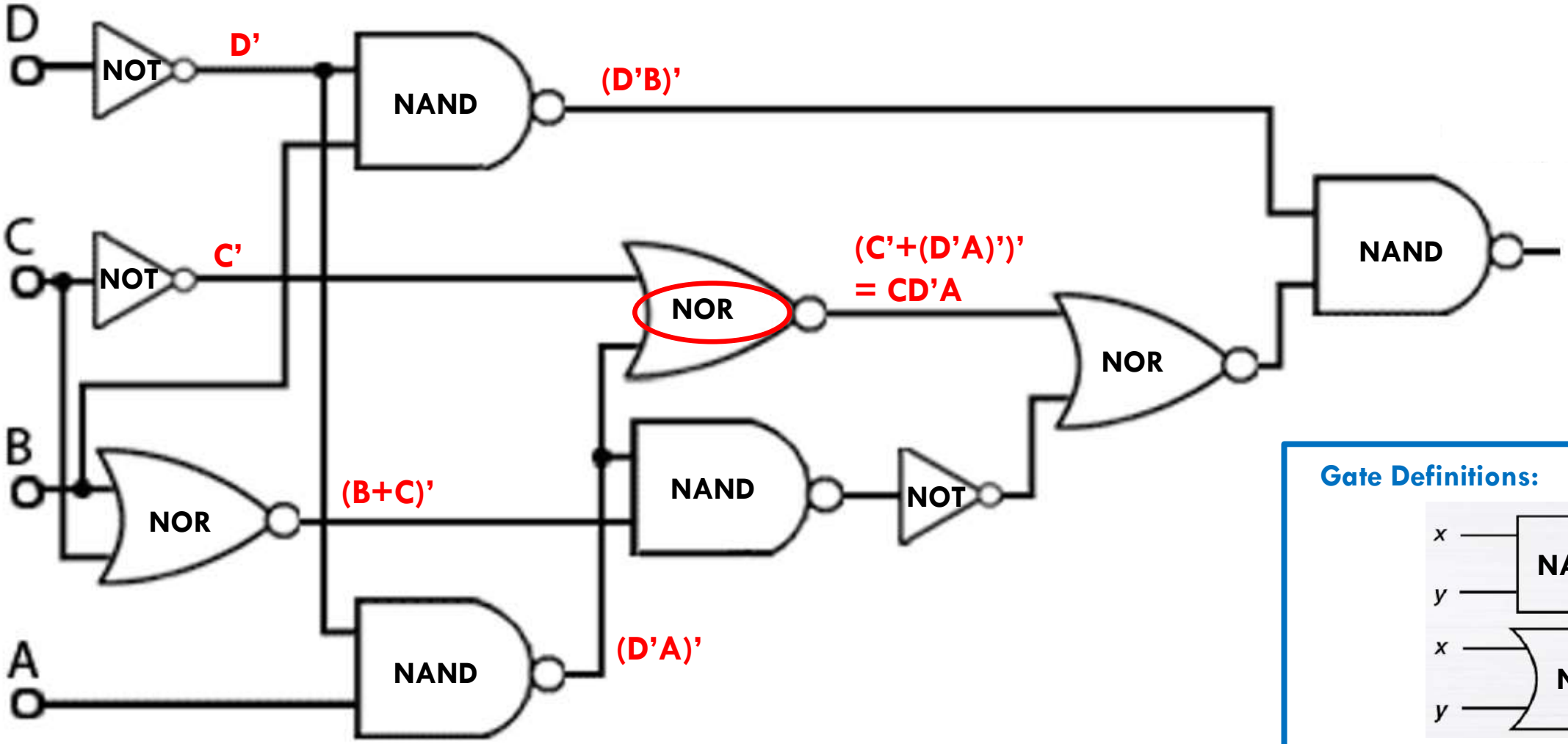
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



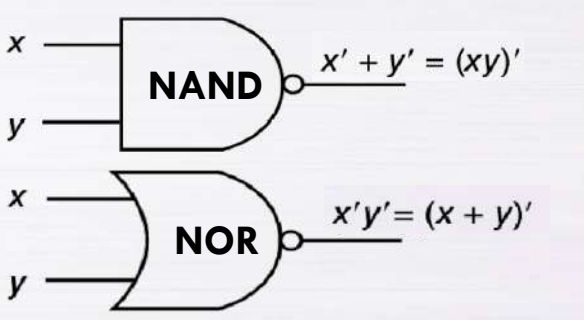
Gate Definitions:



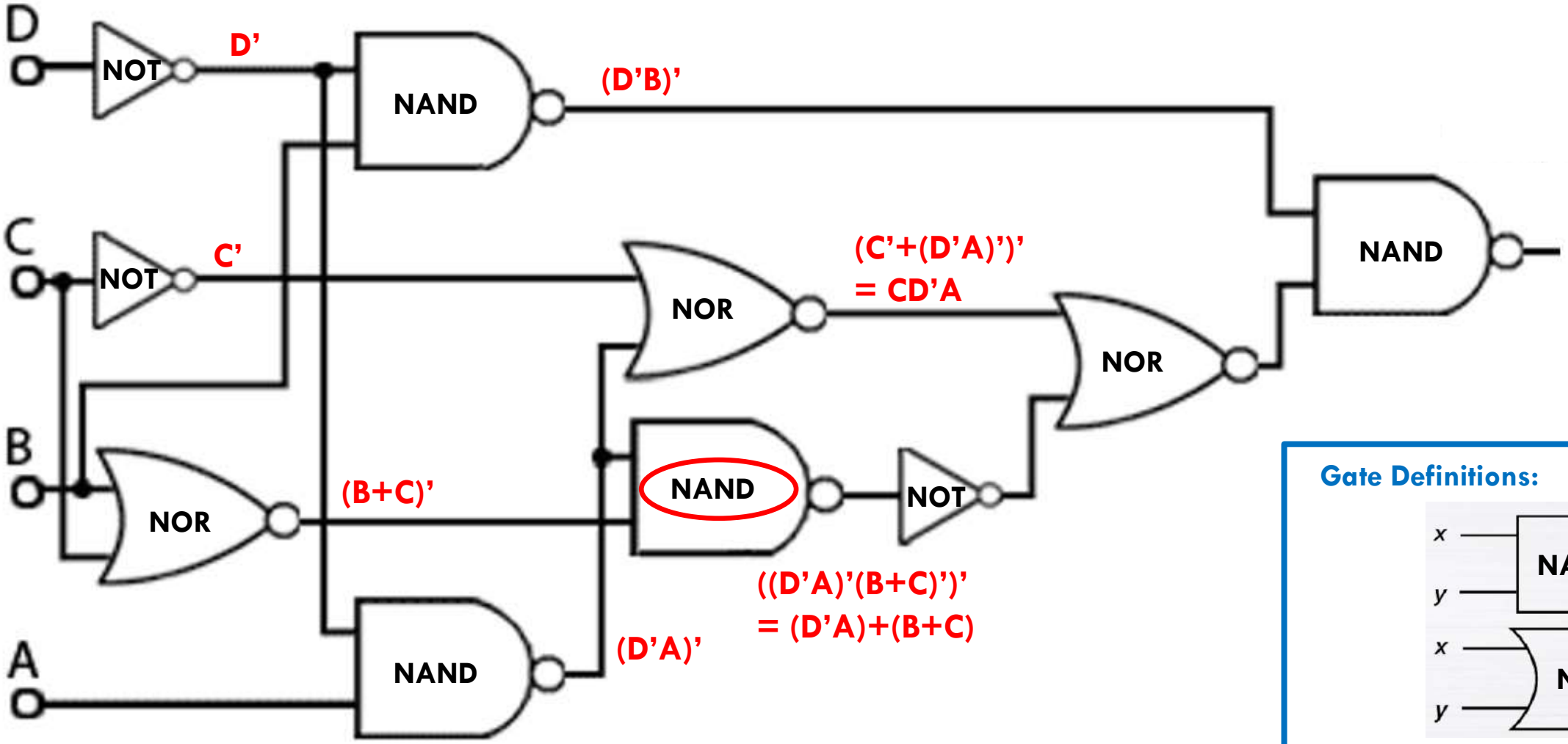
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



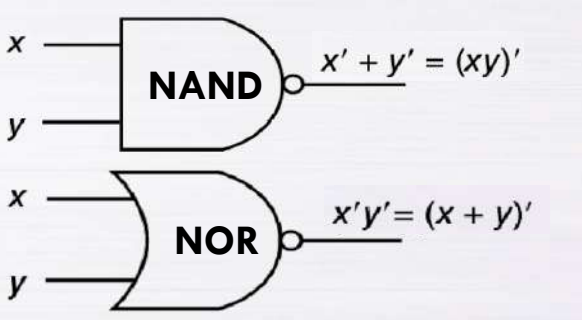
Gate Definitions:



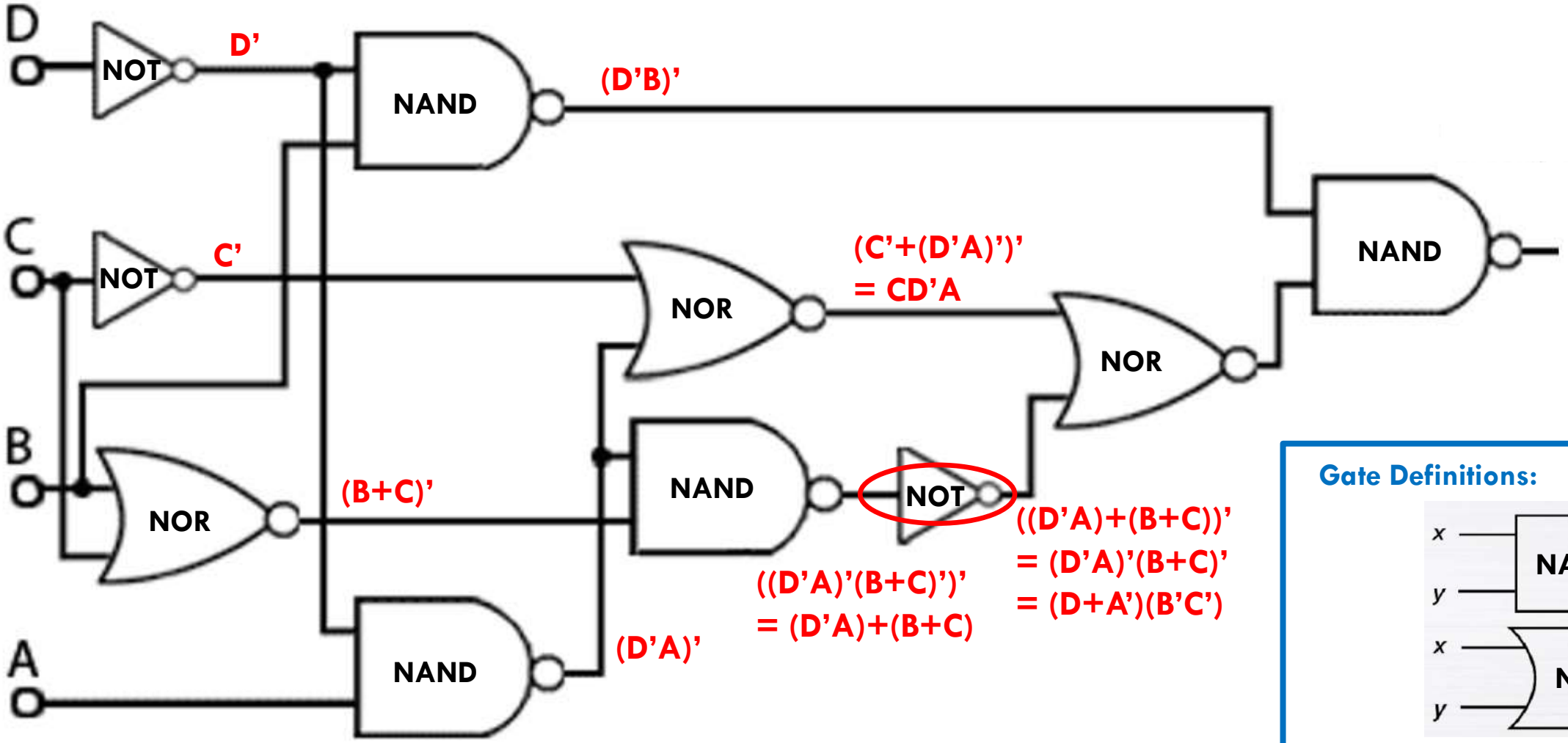
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



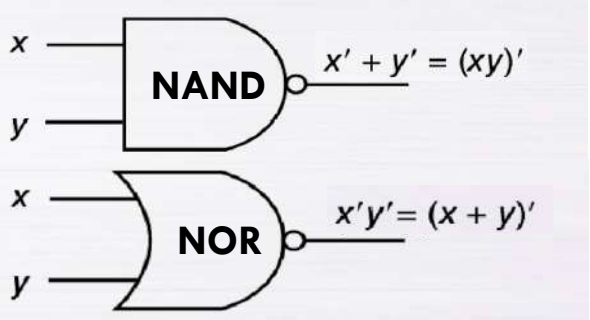
Gate Definitions:



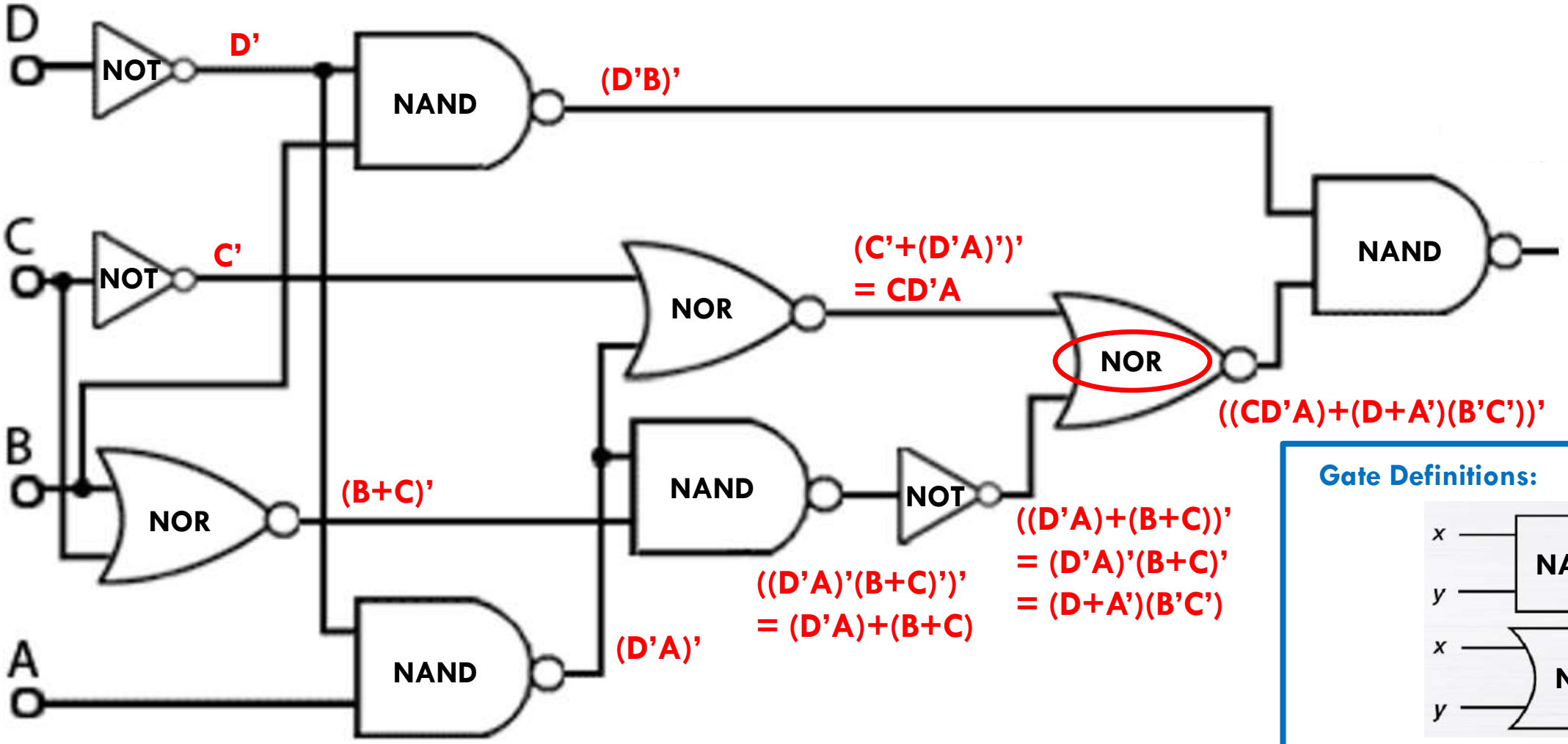
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



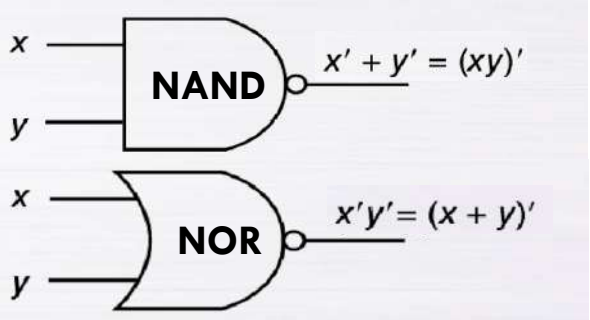
Gate Definitions:



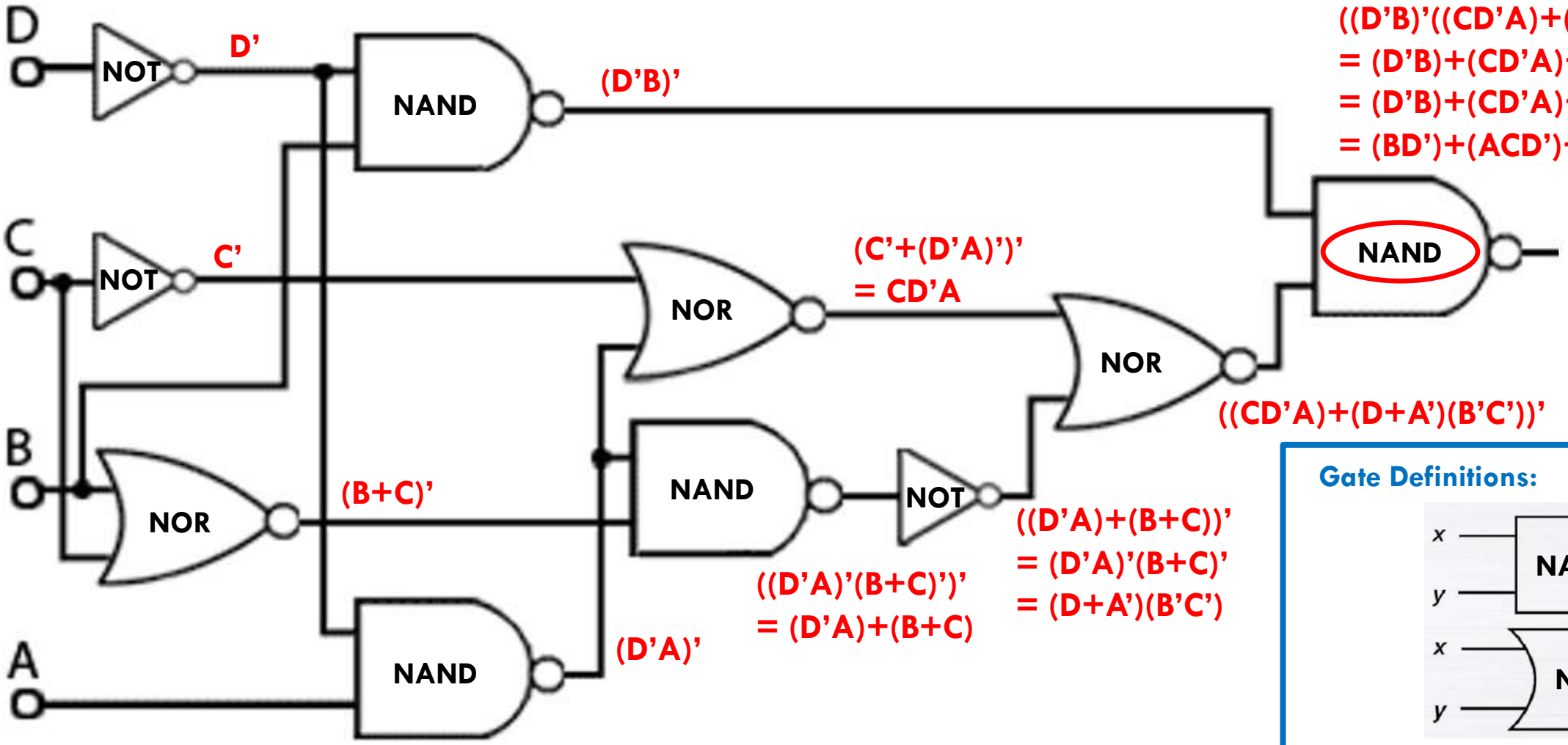
COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



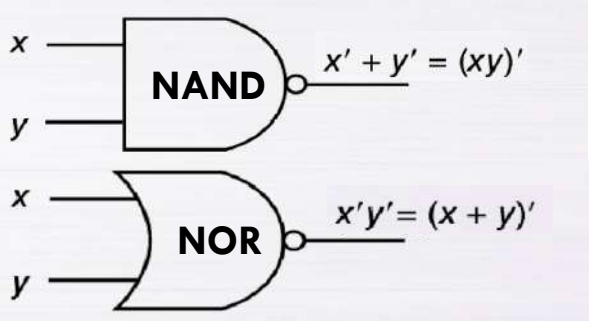
Gate Definitions:



COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT



Gate Definitions:



COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT

(BD') + (ACD') + (B'C'D) + (A'B'C')

AB \ CD	00	01	11	10
00				
01				
11				
10				

COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT

(BD') + (ACD') + (B'C'D) + (A'B'C')

AB \ CD	00	01	11	10
00				
01	1			1
11	1			1
10				

COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT

(BD') + (ACD') + (B'C'D) + (A'B'C')

AB \ CD	00	01	11	10
00				
01	1			1
11	1			1
10				1

COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT

(BD') + (ACD') + (B'C'D) + (A'B'C')

AB \ CD	00	01	11	10
00		1		
01	1			1
11	1			1
10		1		1

COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT

(BD') + (ACD') + (B'C'D) + (A'B'C')

AB \ CD	00	01	11	10
00	1	1		
01	1			1
11	1			1
10		1		1

COMPLETE THE KARNAUGH MAP SUCH THAT IT REPRESENTS THE FUNCTION OF THE GIVEN CIRCUIT

(BD') + (ACD') + (B'C'D) + (A'B'C')

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	0	0	1
11	1	0	0	1
10	0	1	0	1

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **IMMEDIATE ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Immediate addressing is where the data is part of the instruction.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **IMMEDIATE ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Immediate addressing is where the data is part of the instruction.

Answer: 66

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **DIRECT ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 12
Index register : 34
```

Definition:

Direct addressing is where the address of the data is given in the instruction.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **DIRECT ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

Base register : 12
Index register : 34

Definition:

Direct addressing is where the address of the data is given in the instruction.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **DIRECT ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 12
Index register : 34
```

Definition:

Direct addressing is where the address of the data is given in the instruction.

Answer: b6

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **INDIRECT ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Indirect addressing gives the address of the address of the data in the instruction.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **INDIRECT ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Indirect addressing gives the address of the address of the data in the instruction.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **INDIRECT ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Indirect addressing gives the address of the address of the data in the instruction.

Answer: ff

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **INDEXED ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Indexed addressing uses a register (implicitly or explicitly) as an offset, which is added to the address in the operand to determine the effective address of the data.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **INDEXED ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Indexed addressing uses a register (implicitly or explicitly) as an offset, which is added to the address in the operand to determine the effective address of the data.

Load Address: 66+12 = 78

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **INDEXED ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Indexed addressing uses a register (implicitly or explicitly) as an offset, which is added to the address in the operand to determine the effective address of the data.

Load Address: 66+12 = 78

Answer: 27

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **BASED ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Based addressing is similar except that a base register is used instead of an index register.

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **BASED ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Based addressing is similar except that a base register is used instead of an index register.

Load Address: 66+34 = 9A

A COMPUTER EXECUTES THE INSTRUCTION **LOAD 66**. WHAT DATA WILL BE LOADED INTO THE ACCUMULATOR IF **BASED ADDRESSING** IS USED?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
```

Definition:

Based addressing is similar except that a base register is used instead of an index register.

Load Address: 66+34 = 9A

Answer: 6b

SUPPOSE THE COMPUTER EXECUTES **ADD R1** USING **REGISTER ADDRESSING**. WHAT DATA VALUE WILL BE ADDED TO THE ACCUMULATOR?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
R0             : 10
R1             : 01
R2             : AB
R3             : 22
```

Definition:

Register addressing is where the data is located in a register.

SUPPOSE THE COMPUTER EXECUTES **ADD R1** USING **REGISTER ADDRESSING**. WHAT DATA VALUE WILL BE ADDED TO THE ACCUMULATOR?

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
R0             : 10
R1             : 01
R2             : AB
R3             : 22
```

Definition:

Register addressing is where the data is located in a register.

Answer: 01

SUPPOSE THE COMPUTER EXECUTES **ADD R1** USING **REGISTER-INDIRECT ADDRESSING**. WHAT IS THE **EFFECTIVE ADDRESS OF THE OPERAND THAT WILL BE ADDED TO THE ACCUMULATOR?**

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
R0             : 10
R1             : 01
R2             : AB
R3             : 22
```

Definition:

Register indirect addressing uses a register to store the address of the data.

SUPPOSE THE COMPUTER EXECUTES **ADD R1** USING **REGISTER-INDIRECT ADDRESSING**. WHAT IS THE **EFFECTIVE ADDRESS OF THE OPERAND THAT WILL BE ADDED TO THE ACCUMULATOR?**

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
R0             : 10
R1             : 01
R2             : AB
R3             : 22
```

Definition:

Register indirect addressing uses a register to store the address of the data.

SUPPOSE THE COMPUTER EXECUTES **ADD R1** USING **REGISTER-INDIRECT ADDRESSING**. WHAT IS THE **EFFECTIVE ADDRESS OF THE OPERAND THAT WILL BE ADDED TO THE ACCUMULATOR?**

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
R0             : 10
R1             : 01
R2             : AB
R3             : 22
```

Definition:

Register indirect addressing uses a register to store the address of the data.

Answer: 01

REMARK ON REGISTER-INDIRECT ADDRESSING

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	03	0f	a8	4f	7d	f6	40	93	38	8c	c7	b4	70	82	3e	a9
1_	e4	e4	f3	32	b9	34	69	46	a7	0f	49	4d	c1	c9	77	aa
2_	e4	c3	41	60	12	51	77	09	0d	7b	ad	8f	ac	5c	d1	7b
3_	b0	de	87	53	7a	6a	be	82	72	b8	fd	73	7c	f0	a5	9d
4_	e1	ed	0e	0e	66	d8	ad	eb	c2	ef	0b	1b	8b	23	23	8b
5_	95	aa	9f	11	d6	66	57	f1	f6	dc	6e	bc	e7	b5	2b	2c
6_	65	8c	34	a1	eb	26	b6	d6	10	74	8e	1b	08	e6	66	49
7_	19	63	0d	41	bd	e5	79	6d	27	d3	c8	f8	d9	84	4c	fd
8_	d0	fb	ea	b3	e3	75	5a	7d	f7	8a	3e	52	da	b6	0c	79
9_	d9	87	72	c5	93	ec	03	4a	65	04	6b	5d	94	32	f4	52
A_	ba	e6	96	ef	63	dd	f2	c0	3c	53	02	00	5a	9d	9f	0d
B_	fa	0e	f1	77	1f	f6	ff	50	6c	1b	b0	8b	48	f9	d2	bc
C_	ec	64	7f	fc	8e	7a	e0	9b	83	6e	2f	2c	60	23	30	8e
D_	1a	fa	e1	6b	eb	10	b6	53	f2	85	db	e4	48	e9	aa	67
E_	ef	36	db	62	51	d8	7a	d4	a0	b4	04	31	e0	1b	83	7e
F_	8c	e0	76	93	8d	ab	b0	6a	96	fe	a3	ba	eb	e4	a7	5b

```
Base  register : 34
Index register : 12
R0             : 10
R1             : 01
R2             : AB
R3             : 22
```

Correct Definition:

Register indirect addressing uses a register to store the address of the data.

Faulty Definition on Lecture Slides:

~~*Register indirect addressing* uses a register to store the address of the address of the data.~~

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 0
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC :

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 0
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 2

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 2

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 4

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 6

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 8

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 0

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 0

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 1

CONSIDER THE FOLLOWING MARIE PROGRAM. WHAT WOULD THE OUTPUT BE IF THE **USER ENTERED 2**?

Instruction	Meaning
Load X	Load contents of address X into AC.
Store X	Store the contents of AC at address X.
Add X	Add the contents of address X to AC.
Subt X	Subtract the contents of address X from AC.
Input	Input a value from the keyboard into AC.
Output	Output the value in AC to the display.
Halt	Terminate program.
Skipcond	Skip next instruction on condition.
Jump X	Load the value of X into PC.

Skipcond 000 : skip next instruction if AC is negative

Skipcond 400 : skip next instruction if AC = 0

Skipcond 800 : skip next instruction if AC is positive

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 2
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

AC : 1

WHAT IS THE HEXADECIMAL VALUE STORED AT ADDRESS 103?

Hex	Instruction	Meaning
1	Load X	Load contents of address X into AC.
2	Store X	Store the contents of AC at address X.
3	Add X	Add the contents of address X to AC.
4	Subt X	Subtract the contents of address X from AC.
5	Input	Input a value from the keyboard into AC.
6	Output	Output the value in AC to the display.
7	Halt	Terminate program.
8	Skipcond	Skip next instruction on condition.
9	Jump X	Load the value of X into PC.

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 0
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

WHAT IS THE HEXADECIMAL VALUE STORED AT ADDRESS 103?

Hex	Instruction	Meaning
1	Load X	Load contents of address X into AC.
2	Store X	Store the contents of AC at address X.
3	Add X	Add the contents of address X to AC.
4	Subt X	Subtract the contents of address X from AC.
5	Input	Input a value from the keyboard into AC.
6	Output	Output the value in AC to the display.
7	Halt	Terminate program.
8	Skipcond	Skip next instruction on condition.
9	Jump X	Load the value of X into PC.

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 0
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

WHAT IS THE HEXADECIMAL VALUE STORED AT ADDRESS 103?

Hex	Instruction	Meaning
1	Load X	Load contents of address X into AC.
2	Store X	Store the contents of AC at address X.
3	Add X	Add the contents of address X to AC.
4	Subt X	Subtract the contents of address X from AC.
5	Input	Input a value from the keyboard into AC.
6	Output	Output the value in AC to the display.
7	Halt	Terminate program.
8	Skipcond	Skip next instruction on condition.
9	Jump X	Load the value of X into PC.

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 0
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

WHAT IS THE HEXADECIMAL VALUE STORED AT ADDRESS 103?

Hex	Instruction	Meaning
1	Load X	Load contents of address X into AC.
2	Store X	Store the contents of AC at address X.
3	Add X	Add the contents of address X to AC.
4	Subt X	Subtract the contents of address X from AC.
5	Input	Input a value from the keyboard into AC.
6	Output	Output the value in AC to the display.
7	Halt	Terminate program.
8	Skipcond	Skip next instruction on condition.
9	Jump X	Load the value of X into PC.

Hex address	Label	Instruction
100	Read	Input
101		Store W
102	Quadruple	Add W
103		Add W
104		Add W
105	Compare	Subt X
106		Skipcond 400
107		Jump Else
108	Match	Load Y
109		Output
10A		Halt
10B	Else	Load Z
10C		Output
10D		Halt
10E	W	Dec 0
10F	X	Dec 8
110	Y	Dec 1
111	Z	Dec 0

Answer : opcode address : 310E