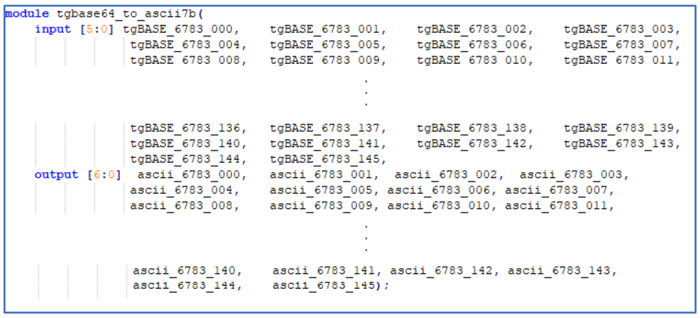
COMP311-1: Logic Circuit Design
Fall 2019, Prof. Taigon Song
Project 3. Due: Dec. 19, 9:59am [Total: 210 + 40 points]
(example)

[2016116783]
KIM SEONGROK

No.	Checksheet item	Answer	Points
1.	ascii (part 1)	0 0 33 100 108 114 111 87 32 111 108 108 101 72	10
2.	tgBASE (part 1)	0 0 1 41 49 55 52 34 0 52 49 49 42 19	10
3.	oneBig (part 1)	0 141 4955 5234 52 49 49 4219	10
4.	encrypt (part 1)	0 9367 8490 704 5130 909 10137	10
5.	DoneBig (part 2)	1426 2427 503 300 4656 43 5851 101	10
6.	DtgBase (part 2)	14 26 24 27 5 3 3 0 46 56 0 43 58 51 1 1	10
7.	Dascii (part 2)	67 79 77 80 51 49 49 32 105 115 32 102 117 110 33 33	10
8.	decrypted message (part 2)	COMP311 is fun!!	10
9.	[Modelsim] ascii (part 1)	Y	5
10.	[Modelsim] tgBase (part 1)	Y	5
11.	[Modelsim] oneBig (part 1)	Y	5
12.	[Modelsim] encrypt (part 1)	Y	5
13.	[Modelsim] DoneBig (part 2)	Y	5
14.	[Modelsim] DtgBase (part 2)	Y	5
15.	[Modelsim] Dascii (part 2)	Y	5
16.	[Modelsim] output file (part 2)	Y	5
17.	p, q (part 3)	97,109	10
18.	d (part 3)	233	10
19.	decrypted message (part 3)		10
	Well done on your final Project! Have a safe2020winter break!		
21.	[Modelsim] p, q (part 3)	Y	20
22.	[Modelsim] d (part 3)	Y	20
23.	[Modelsim] message print	Y	20
24	[Bonus]	40	40

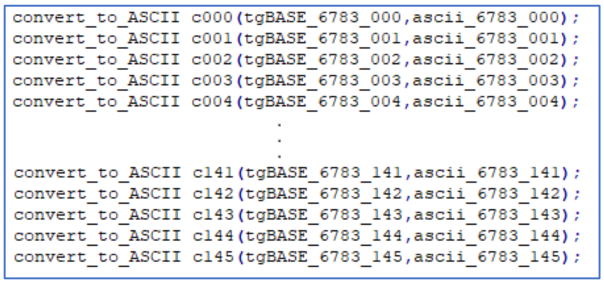
Part1. ASCII 7bit to tgBASE 6bit

Input and output part



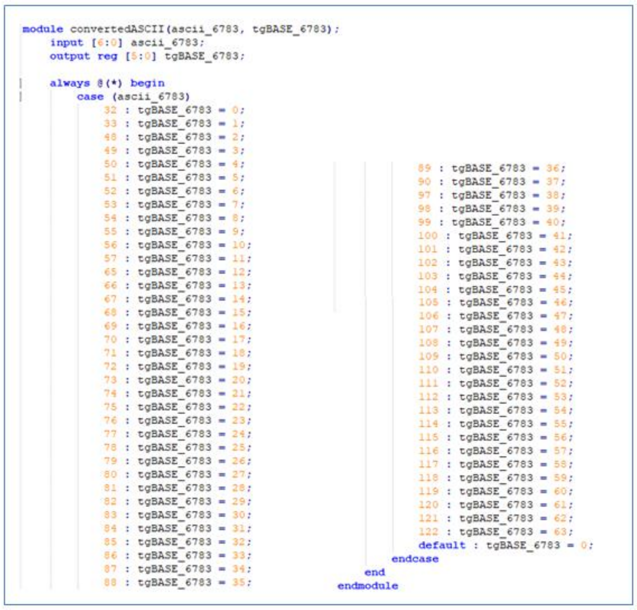
```
module tgbase64_to_ascii7b(
    input [5:0] tgBASE_6783_000,    tgBASE_6783_001,    tgBASE_6783_002,    tgBASE_6783_003,
                 tgBASE_6783_004,    tgBASE_6783_005,    tgBASE_6783_006,    tgBASE_6783_007,
                 tgBASE_6783_008,    tgBASE_6783_009,    tgBASE_6783_010,    tgBASE_6783_011,
                                               .
                                               .
                                               .

                 tgBASE_6783_136,    tgBASE_6783_137,    tgBASE_6783_138,    tgBASE_6783_139,
                 tgBASE_6783_140,    tgBASE_6783_141,    tgBASE_6783_142,    tgBASE_6783_143,
                 tgBASE_6783_144,    tgBASE_6783_145,
    output [6:0]  ascii_6783_000,    ascii_6783_001,  ascii_6783_002,  ascii_6783_003,
                 ascii_6783_004,     ascii_6783_005, ascii_6783_006, ascii_6783_007,
                 ascii_6783_008,     ascii_6783_009, ascii_6783_010, ascii_6783_011,
                                               .
                                               .
                                               .

                 ascii_6783_140,     ascii_6783_141, ascii_6783_142, ascii_6783_143,
                 ascii_6783_144,     ascii_6783_145);
```

Perform converting using defined module



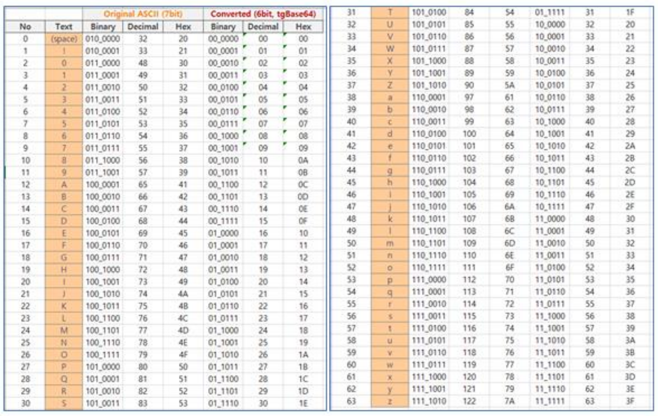
```
convert_to_ASCII c000(tgBASE_6783_000,ascii_6783_000);
convert_to_ASCII c001(tgBASE_6783_001,ascii_6783_001);
convert_to_ASCII c002(tgBASE_6783_002,ascii_6783_002);
convert_to_ASCII c003(tgBASE_6783_003,ascii_6783_003);
convert_to_ASCII c004(tgBASE_6783_004,ascii_6783_004);
                              .
                              .

convert_to_ASCII c141(tgBASE_6783_141,ascii_6783_141);
convert_to_ASCII c142(tgBASE_6783_142,ascii_6783_142);
convert_to_ASCII c143(tgBASE_6783_143,ascii_6783_143);
convert_to_ASCII c144(tgBASE_6783_144,ascii_6783_144);
convert_to_ASCII c145(tgBASE_6783_145,ascii_6783_145);
```

Define converting table from ascii to tgBASE



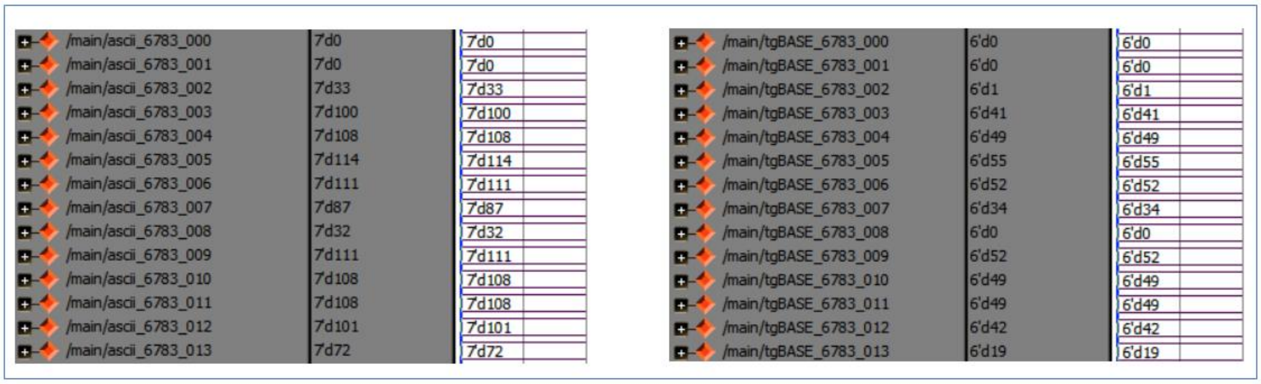
```
module convertedASCII(ascii_6783, tgBASE_6783);
    input [6:0] ascii_6783;
    output reg [5:0] tgBASE_6783;

    always @(*) begin
        case (ascii_6783)
            32 : tgBASE_6783 = 0;
            33 : tgBASE_6783 = 1;
            48 : tgBASE_6783 = 2;
            49 : tgBASE_6783 = 3;
            50 : tgBASE_6783 = 4;
            51 : tgBASE_6783 = 5;
            52 : tgBASE_6783 = 6;
            53 : tgBASE_6783 = 7;
            54 : tgBASE_6783 = 8;
            55 : tgBASE_6783 = 9;
            56 : tgBASE_6783 = 10;
            57 : tgBASE_6783 = 11;
            65 : tgBASE_6783 = 12;
            66 : tgBASE_6783 = 13;
            67 : tgBASE_6783 = 14;
            68 : tgBASE_6783 = 15;
            69 : tgBASE_6783 = 16;
            70 : tgBASE_6783 = 17;
            71 : tgBASE_6783 = 18;
            72 : tgBASE_6783 = 19;
            73 : tgBASE_6783 = 20;
            74 : tgBASE_6783 = 21;
            75 : tgBASE_6783 = 22;
            76 : tgBASE_6783 = 23;
            77 : tgBASE_6783 = 24;
            78 : tgBASE_6783 = 25;
            79 : tgBASE_6783 = 26;
            80 : tgBASE_6783 = 27;
            81 : tgBASE_6783 = 28;
            82 : tgBASE_6783 = 29;
            83 : tgBASE_6783 = 30;
            84 : tgBASE_6783 = 31;
            85 : tgBASE_6783 = 32;
            86 : tgBASE_6783 = 33;
            87 : tgBASE_6783 = 34;
            88 : tgBASE_6783 = 35;
            89 : tgBASE_6783 = 36;
            90 : tgBASE_6783 = 37;
            97 : tgBASE_6783 = 38;
            98 : tgBASE_6783 = 39;
            99 : tgBASE_6783 = 40;
            100 : tgBASE_6783 = 41;
            101 : tgBASE_6783 = 42;
            102 : tgBASE_6783 = 43;
            103 : tgBASE_6783 = 44;
            104 : tgBASE_6783 = 45;
            105 : tgBASE_6783 = 46;
            106 : tgBASE_6783 = 47;
            107 : tgBASE_6783 = 48;
            108 : tgBASE_6783 = 49;
            109 : tgBASE_6783 = 50;
            110 : tgBASE_6783 = 51;
            111 : tgBASE_6783 = 52;
            112 : tgBASE_6783 = 53;
            113 : tgBASE_6783 = 54;
            114 : tgBASE_6783 = 55;
            115 : tgBASE_6783 = 56;
            116 : tgBASE_6783 = 57;
            117 : tgBASE_6783 = 58;
            118 : tgBASE_6783 = 59;
            119 : tgBASE_6783 = 60;
            120 : tgBASE_6783 = 61;
            121 : tgBASE_6783 = 62;
            122 : tgBASE_6783 = 63;
            default : tgBASE_6783 = 0;
        endcase
    end
endmodule
```

Followed by given table set



		Original ASCII (7bit)			Converted (6bit, tgBase64)		
No	Text	Binary	Decimal	Hex	Binary	Decimal	Hex
0	(space)	010_0000	32	20	00_0000	00	00
1	!	010_0001	33	21	00_0001	01	01
2	0	011_0000	48	30	00_0010	02	02
3	1	011_0001	49	31	00_0011	03	03
4	2	011_0010	50	32	00_0100	04	04
5	3	011_0011	51	33	00_0101	05	05
6	4	011_0100	52	34	00_0110	06	06
7	5	011_0101	53	35	00_0111	07	07
8	6	011_0110	54	36	00_1000	08	08
9	7	011_0111	55	37	00_1001	09	09
10	8	011_1000	56	38	00_1010	10	0A
11	9	011_1001	57	39	00_1011	11	0B
12	A	100_0001	65	41	00_1100	12	0C
13	B	100_0010	66	42	00_1101	13	0D
14	C	100_0011	67	43	00_1110	14	0E
15	D	100_0100	68	44	00_1111	15	0F
16	E	100_0101	69	45	01_0000	16	10
17	F	100_0110	70	46	01_0001	17	11
18	G	100_0111	71	47	01_0010	18	12
19	H	100_1000	72	48	01_0011	19	13
20	I	100_1001	73	49	01_0100	20	14
21	J	100_1010	74	4A	01_0101	21	15
22	K	100_1011	75	4B	01_0110	22	16
23	L	100_1100	76	4C	01_0111	23	17
24	M	100_1101	77	4D	01_1000	24	18
25	N	100_1110	78	4E	01_1001	25	19
26	O	100_1111	79	4F	01_1010	26	1A
27	P	101_0000	80	50	01_1011	27	1B
28	Q	101_0001	81	51	01_1100	28	1C
29	R	101_0010	82	52	01_1101	29	1D
30	S	101_0011	83	53	01_1110	30	1E
31	T	101_0100	84	54	01_1111	31	1F
32	U	101_0101	85	55	10_0000	32	20
33	V	101_0110	86	56	10_0001	33	21
34	W	101_0111	87	57	10_0010	34	22
35	X	101_1000	88	58	10_0011	35	23
36	Y	101_1001	89	59	10_0100	36	24
37	Z	101_1010	90	5A	10_0101	37	25
38	a	110_0001	97	61	10_0110	38	26
39	b	110_0010	98	62	10_0111	39	27
40	c	110_0011	99	63	10_1000	40	28
41	d	110_0100	100	64	10_1001	41	29
42	e	110_0101	101	65	10_1010	42	2A
43	f	110_0110	102	66	10_1011	43	2B
44	g	110_0111	103	67	10_1100	44	2C
45	h	110_1000	104	68	10_1101	45	2D
46	i	110_1001	105	69	10_1110	46	2E
47	j	110_1010	106	6A	10_1111	47	2F
48	k	110_1011	107	6B	11_0000	48	30
49	l	110_1100	108	6C	11_0001	49	31
50	m	110_1101	109	6D	11_0010	50	32
51	n	110_1110	110	6E	11_0011	51	33
52	o	110_1111	111	6F	11_0100	52	34
53	p	111_0000	112	70	11_0101	53	35
54	q	111_0001	113	71	11_0110	54	36
55	r	111_0010	114	72	11_0111	55	37
56	s	111_0011	115	73	11_1000	56	38
57	t	111_0100	116	74	11_1001	57	39
58	u	111_0101	117	75	11_1010	58	3A
59	v	111_0110	118	76	11_1011	59	3B
60	w	111_0111	119	77	11_1100	60	3C
61	x	111_1000	120	78	11_1101	61	3D
62	y	111_1001	121	79	11_1110	62	3E
63	z	111_1010	122	7A	11_1111	63	3F

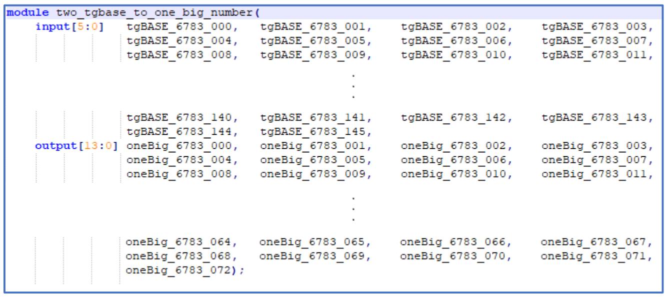
Output view



```
/main/ascii_6783_000   7d0     7d0          /main/tgBASE_6783_000   6'd0    6'd0
/main/ascii_6783_001   7d0     7d0          /main/tgBASE_6783_001   6'd0    6'd0
/main/ascii_6783_002   7d33    7d33         /main/tgBASE_6783_002   6'd1    6'd1
/main/ascii_6783_003   7d100   7d100        /main/tgBASE_6783_003   6'd41   6'd41
/main/ascii_6783_004   7d108   7d108        /main/tgBASE_6783_004   6'd49   6'd49
/main/ascii_6783_005   7d114   7d114        /main/tgBASE_6783_005   6'd55   6'd55
/main/ascii_6783_006   7d111   7d111        /main/tgBASE_6783_006   6'd52   6'd52
/main/ascii_6783_007   7d87    7d87         /main/tgBASE_6783_007   6'd34   6'd34
/main/ascii_6783_008   7d32    7d32         /main/tgBASE_6783_008   6'd0    6'd0
/main/ascii_6783_009   7d111   7d111        /main/tgBASE_6783_009   6'd52   6'd52
/main/ascii_6783_010   7d108   7d108        /main/tgBASE_6783_010   6'd49   6'd49
/main/ascii_6783_011   7d108   7d108        /main/tgBASE_6783_011   6'd49   6'd49
/main/ascii_6783_012   7d101   7d101        /main/tgBASE_6783_012   6'd42   6'd42
/main/ascii_6783_013   7d72    7d72         /main/tgBASE_6783_013   6'd19   6'd19
```

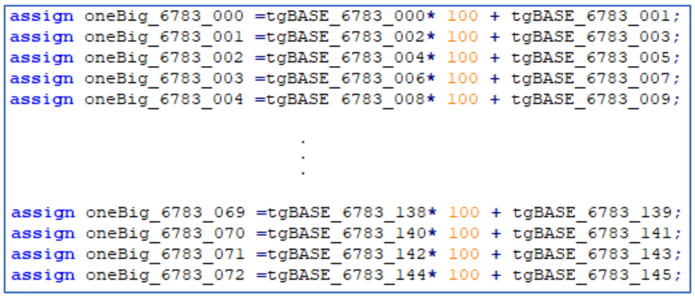
tgBASE 6bit to OneBigNumber 14bit

Input and output part



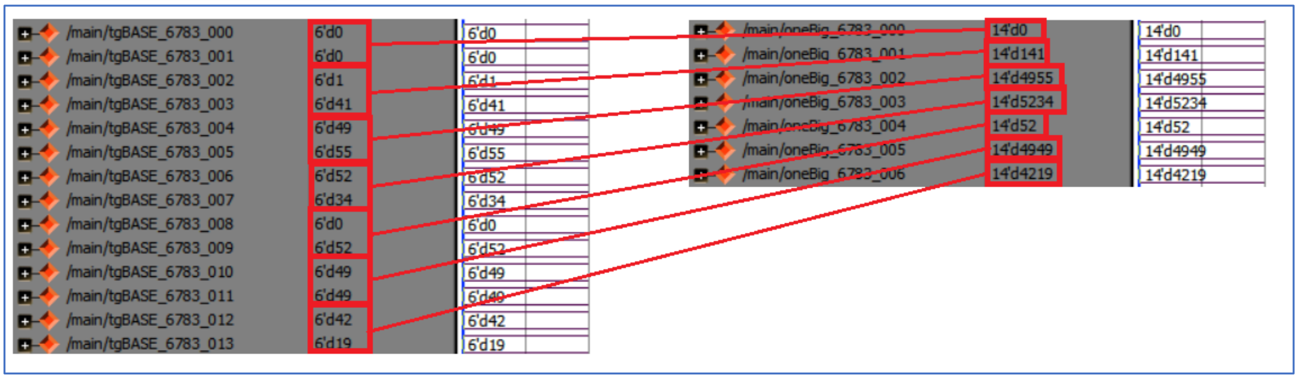
```
module two_tgbase_to_one_big_number(
    input[5:0]    tgBASE_6783_000,   tgBASE_6783_001,    tgBASE_6783_002,    tgBASE_6783_003,
                  tgBASE_6783_004,   tgBASE_6783_005,    tgBASE_6783_006,    tgBASE_6783_007,
                  tgBASE_6783_008,   tgBASE_6783_009,    tgBASE_6783_010,    tgBASE_6783_011,
                                            .
                                            .
                                            .
                  tgBASE_6783_140,   tgBASE_6783_141,    tgBASE_6783_142,    tgBASE_6783_143,
                  tgBASE_6783_144,   tgBASE_6783_145,
    output[13:0]  oneBig_6783_000,   oneBig_6783_001,    oneBig_6783_002,    oneBig_6783_003,
                  oneBig_6783_004,   oneBig_6783_005,    oneBig_6783_006,    oneBig_6783_007,
                  oneBig_6783_008,   oneBig_6783_009,    oneBig_6783_010,    oneBig_6783_011,
                                            .
                                            .
                                            .
                  oneBig_6783_064,   oneBig_6783_065,    oneBig_6783_066,    oneBig_6783_067,
                  oneBig_6783_068,   oneBig_6783_069,    oneBig_6783_070,    oneBig_6783_071,
                  oneBig_6783_072);
```

Link two tgBASEs into oneBigNumber



```
assign oneBig_6783_000 =tgBASE_6783_000* 100 + tgBASE_6783_001;
assign oneBig_6783_001 =tgBASE_6783_002* 100 + tgBASE_6783_003;
assign oneBig_6783_002 =tgBASE_6783_004* 100 + tgBASE_6783_005;
assign oneBig_6783_003 =tgBASE_6783_006* 100 + tgBASE_6783_007;
assign oneBig_6783_004 =tgBASE_6783_008* 100 + tgBASE_6783_009;
                              .
                              .
                              .
assign oneBig_6783_069 =tgBASE_6783_138* 100 + tgBASE_6783_139;
assign oneBig_6783_070 =tgBASE_6783_140* 100 + tgBASE_6783_141;
assign oneBig_6783_071 =tgBASE_6783_142* 100 + tgBASE_6783_143;
assign oneBig_6783_072 =tgBASE_6783_144* 100 + tgBASE_6783_145;
```

Output view

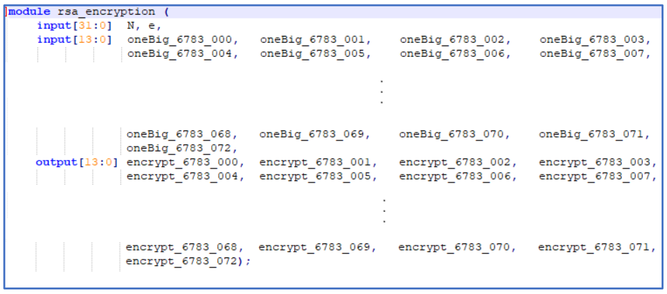


```
/main/tgBASE_6783_000   6'd0     6'd0
/main/tgBASE_6783_001   6'd0     6'd0
/main/tgBASE_6783_002   6'd1     6'd1
/main/tgBASE_6783_003   6'd41    6'd41
/main/tgBASE_6783_004   6'd49    6'd49
/main/tgBASE_6783_005   6'd55    6'd55
/main/tgBASE_6783_006   6'd52    6'd52
/main/tgBASE_6783_007   6'd34    6'd34
/main/tgBASE_6783_008   6'd0     6'd0
/main/tgBASE_6783_009   6'd52    6'd52
/main/tgBASE_6783_010   6'd49    6'd49
/main/tgBASE_6783_011   6'd49    6'd49
/main/tgBASE_6783_012   6'd42    6'd42
/main/tgBASE_6783_013   6'd19    6'd19
```

```
/main/oneBig_6783_000   14'd0      14'd0
/main/oneBig_6783_001   14'd141    14'd141
/main/oneBig_6783_002   14'd4955   14'd4955
/main/oneBig_6783_003   14'd5234   14'd5234
/main/oneBig_6783_004   14'd52     14'd52
/main/oneBig_6783_005   14'd4949   14'd4949
/main/oneBig_6783_006   14'd4219   14'd4219
```

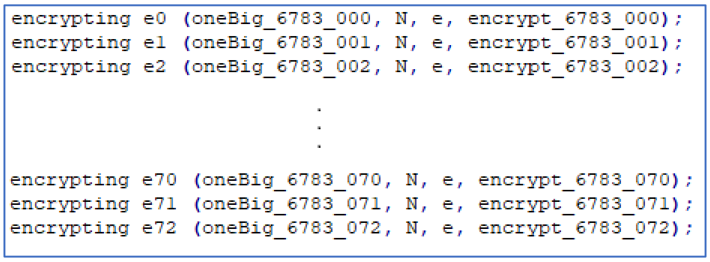
Encrypting OneBigNumber 14bit

Input and output part



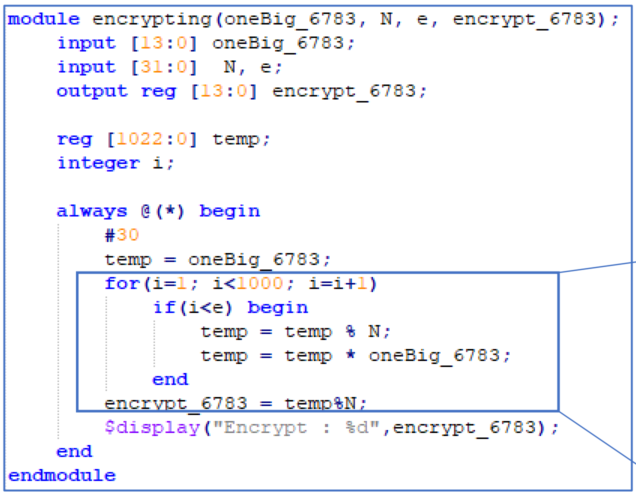
```
module rsa_encryption (
    input[31:0]  N, e,
    input[13:0]  oneBig_6783_000,   oneBig_6783_001,    oneBig_6783_002,    oneBig_6783_003,
                 oneBig_6783_004,   oneBig_6783_005,    oneBig_6783_006,    oneBig_6783_007,
                                           .
                                           .
                                           .
                 oneBig_6783_068,   oneBig_6783_069,    oneBig_6783_070,    oneBig_6783_071,
                 oneBig_6783_072,
    output[13:0] encrypt_6783_000,  encrypt_6783_001,   encrypt_6783_002,   encrypt_6783_003,
                 encrypt_6783_004,  encrypt_6783_005,   encrypt_6783_006,   encrypt_6783_007,
                                           .
                                           .
                                           .
                 encrypt_6783_068,  encrypt_6783_069,   encrypt_6783_070,   encrypt_6783_071,
                 encrypt_6783_072);
```

Encrypting OneBigNumber



```
encrypting e0  (oneBig_6783_000, N, e, encrypt_6783_000);
encrypting e1  (oneBig_6783_001, N, e, encrypt_6783_001);
encrypting e2  (oneBig_6783_002, N, e, encrypt_6783_002);
                              .
                              .
                              .
encrypting e70 (oneBig_6783_070, N, e, encrypt_6783_070);
encrypting e71 (oneBig_6783_071, N, e, encrypt_6783_071);
encrypting e72 (oneBig_6783_072, N, e, encrypt_6783_072);
```

Encrypting Algorithm



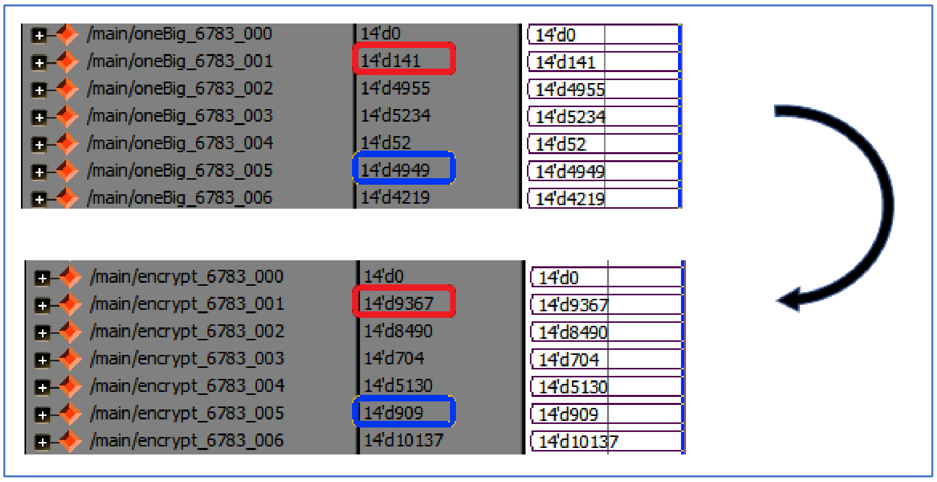
```
module encrypting(oneBig_6783, N, e, encrypt_6783);
    input [13:0] oneBig_6783;
    input [31:0]  N, e;
    output reg [13:0] encrypt_6783;

    reg [1022:0] temp;
    integer i;

    always @(*) begin
        #30
        temp = oneBig_6783;
        for(i=1; i<1000; i=i+1)
            if(i<e) begin
                temp = temp % N;
                temp = temp * oneBig_6783;
            end
        encrypt_6783 = temp%N;
        $display("Encrypt : %d",encrypt_6783);
    end
endmodule
```

Encryption
(N=10403, e = 71)

$$c = m^e \mod N$$

i=1 : Temp = oneBig
i=2 : Temp = (oneBig mod N) * oneBig
= oneBig^2 mod N
...
i=e : Temp = oneBig^e mod N

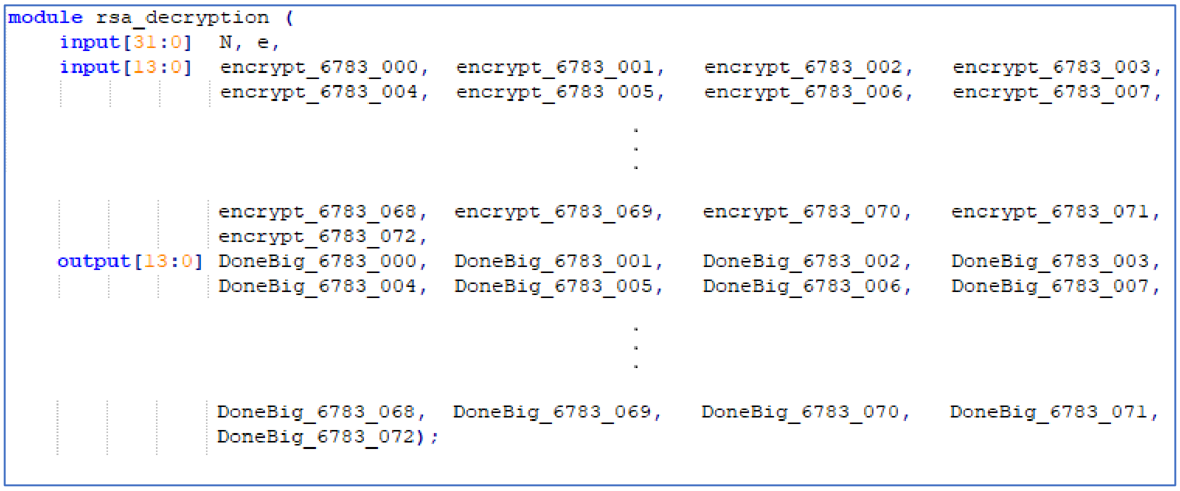
Output view



```
/main/oneBig_6783_000    14'd0        14'd0
/main/oneBig_6783_001    14'd141      14'd141
/main/oneBig_6783_002    14'd4955     14'd4955
/main/oneBig_6783_003    14'd5234     14'd5234
/main/oneBig_6783_004    14'd52       14'd52
/main/oneBig_6783_005    14'd4949     14'd4949
/main/oneBig_6783_006    14'd4219     14'd4219

/main/encrypt_6783_000   14'd0        14'd0
/main/encrypt_6783_001   14'd9367     14'd9367
/main/encrypt_6783_002   14'd8490     14'd8490
/main/encrypt_6783_003   14'd704      14'd704
/main/encrypt_6783_004   14'd5130     14'd5130
/main/encrypt_6783_005   14'd909      14'd909
/main/encrypt_6783_006   14'd10137    14'd10137
```

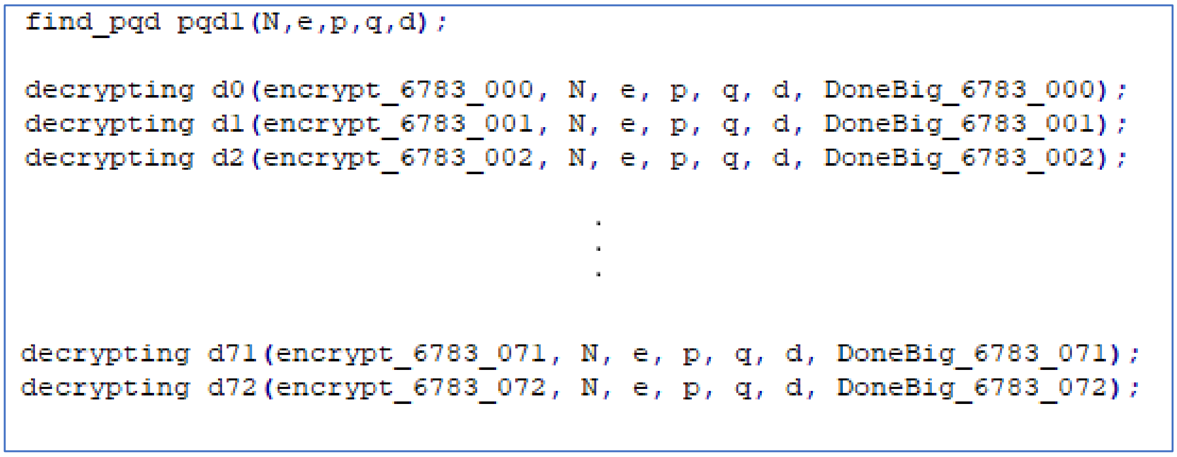
Part2. Decrypting Encrypted 14bit

Input and Output part



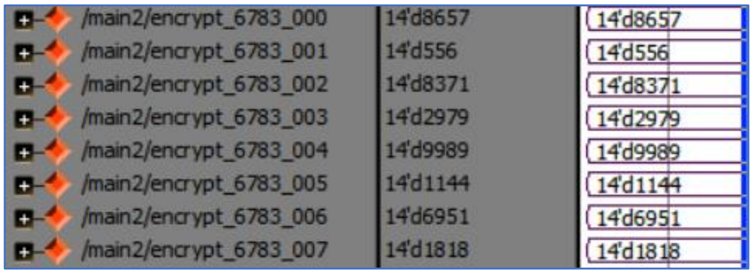
```
module rsa_decryption (
    input[31:0]  N, e,
    input[13:0]  encrypt_6783_000,  encrypt_6783_001,   encrypt_6783_002,   encrypt_6783_003,
                 encrypt_6783_004,  encrypt_6783_005,   encrypt_6783_006,   encrypt_6783_007,
                                               .
                                               .
                                               .
                 encrypt_6783_068,  encrypt_6783_069,   encrypt_6783_070,   encrypt_6783_071,
                 encrypt_6783_072,
    output[13:0] DoneBig_6783_000,  DoneBig_6783_001,   DoneBig_6783_002,   DoneBig_6783_003,
                 DoneBig_6783_004,  DoneBig_6783_005,   DoneBig_6783_006,   DoneBig_6783_007,
                                               .
                                               .
                                               .
                 DoneBig_6783_068,  DoneBig_6783_069,   DoneBig_6783_070,   DoneBig_6783_071,
                 DoneBig_6783_072);
```

Find p, q, d from N, e and Decrypting with them

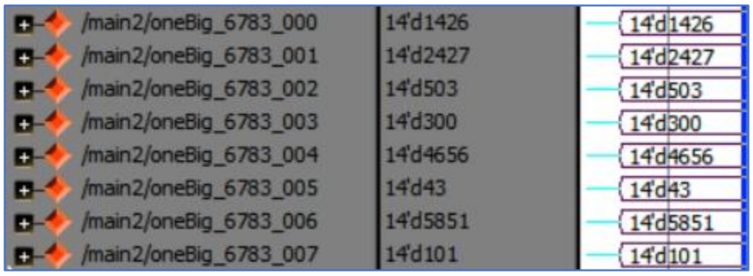


```
find_pqd pqd1(N,e,p,q,d);

decrypting d0(encrypt_6783_000, N, e, p, q, d, DoneBig_6783_000);
decrypting d1(encrypt_6783_001, N, e, p, q, d, DoneBig_6783_001);
decrypting d2(encrypt_6783_002, N, e, p, q, d, DoneBig_6783_002);

                              .
                              .
                              .

decrypting d71(encrypt_6783_071, N, e, p, q, d, DoneBig_6783_071);
decrypting d72(encrypt_6783_072, N, e, p, q, d, DoneBig_6783_072);
```

Output view



```
/main2/encrypt_6783_000   14'd8657   14'd8657
/main2/encrypt_6783_001   14'd556    14'd556
/main2/encrypt_6783_002   14'd8371   14'd8371
/main2/encrypt_6783_003   14'd2979   14'd2979
/main2/encrypt_6783_004   14'd9989   14'd9989
/main2/encrypt_6783_005   14'd1144   14'd1144
/main2/encrypt_6783_006   14'd6951   14'd6951
/main2/encrypt_6783_007   14'd1818   14'd1818
```



```
/main2/oneBig_6783_000    14'd1426   14'd1426
/main2/oneBig_6783_001    14'd2427   14'd2427
/main2/oneBig_6783_002    14'd503    14'd503
/main2/oneBig_6783_003    14'd300    14'd300
/main2/oneBig_6783_004    14'd4656   14'd4656
/main2/oneBig_6783_005    14'd43     14'd43
/main2/oneBig_6783_006    14'd5851   14'd5851
/main2/oneBig_6783_007    14'd101    14'd101
```

Decryted Onebig 14 bit to tgBASE 6bit

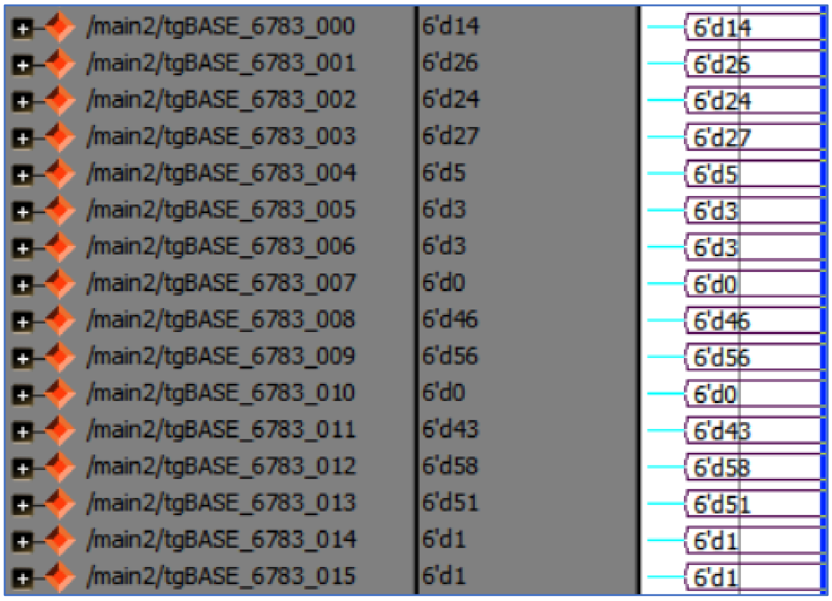
Input and Output part

```
module one_big_number_to_two_tgbase(
    input[13:0]  oneBig_6783_000,   oneBig_6783_001,    oneBig_6783_002,    oneBig_6783_003,
                 oneBig_6783_004,   oneBig_6783_005,    oneBig_6783_006,    oneBig_6783_007,
                                              .
                                              .
                                              .
                 oneBig_6783_068,   oneBig_6783_069,    oneBig_6783_070,    oneBig_6783_071,
                 oneBig_6783_072,
    output[5:0]  tgBASE_6783_000,   tgBASE_6783_001,    tgBASE_6783_002,    tgBASE_6783_003,
                 tgBASE_6783_004,   tgBASE_6783_005,    tgBASE_6783_006,    tgBASE_6783_007,
                                              .
                                              .
                                              .
                 tgBASE_6783_140,   tgBASE_6783_141,    tgBASE_6783_142,    tgBASE_6783_143,
                 tgBASE_6783_144,   tgBASE_6783_145);
```

Divide OneBig 14bit to two tgbBASEs 6bit

```
assign tgBASE_6783_000 = oneBig_6783_000 / 100; assign tgBASE_6783_001 = oneBig_6783_000 % 100;
assign tgBASE_6783_002 = oneBig_6783_001 / 100; assign tgBASE_6783_003 = oneBig_6783_001 % 100;
assign tgBASE_6783_004 = oneBig_6783_002 / 100; assign tgBASE_6783_005 = oneBig_6783_002 % 100;
                                         .
                                         .
                                         .
assign tgBASE_6783_140 = oneBig_6783_070 / 100; assign tgBASE_6783_141 = oneBig_6783_070 % 100;
assign tgBASE_6783_142 = oneBig_6783_071 / 100; assign tgBASE_6783_143 = oneBig_6783_071 % 100;
assign tgBASE_6783_144 = oneBig_6783_072 / 100; assign tgBASE_6783_145 = oneBig_6783_072 % 100;
```

Output view



```
/main2/tgBASE_6783_000   6'd14   6'd14
/main2/tgBASE_6783_001   6'd26   6'd26
/main2/tgBASE_6783_002   6'd24   6'd24
/main2/tgBASE_6783_003   6'd27   6'd27
/main2/tgBASE_6783_004   6'd5    6'd5
/main2/tgBASE_6783_005   6'd3    6'd3
/main2/tgBASE_6783_006   6'd3    6'd3
/main2/tgBASE_6783_007   6'd0    6'd0
/main2/tgBASE_6783_008   6'd46   6'd46
/main2/tgBASE_6783_009   6'd56   6'd56
/main2/tgBASE_6783_010   6'd0    6'd0
/main2/tgBASE_6783_011   6'd43   6'd43
/main2/tgBASE_6783_012   6'd58   6'd58
/main2/tgBASE_6783_013   6'd51   6'd51
/main2/tgBASE_6783_014   6'd1    6'd1
/main2/tgBASE_6783_015   6'd1    6'd1
```

tgBASE 6bit to ASCII 7bit

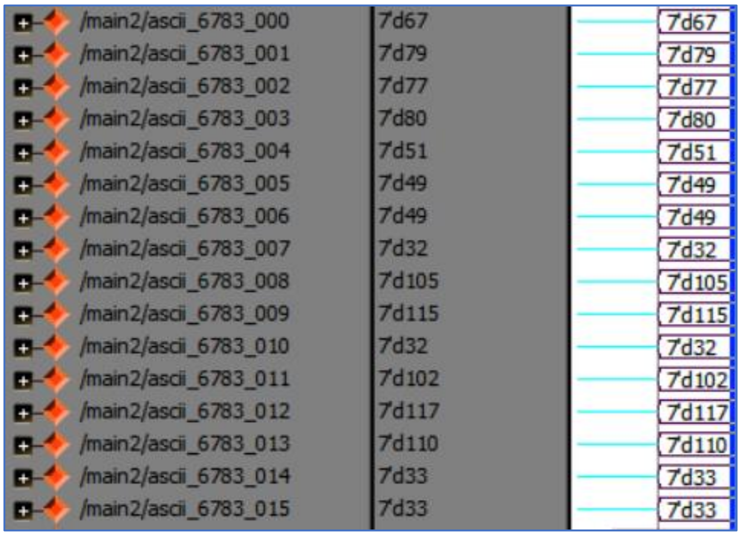
Input and Output part

```
module tgbase64_to_ascii7b(
    input [5:0] tgBASE_6783_000,    tgBASE_6783_001,    tgBASE_6783_002,    tgBASE_6783_003,
                tgBASE_6783_004,    tgBASE_6783_005,    tgBASE_6783_006,    tgBASE_6783_007,

                                              .
                                              .
                                              .

                tgBASE_6783_140,    tgBASE_6783_141,    tgBASE_6783_142,    tgBASE_6783_143,
                tgBASE_6783_144,    tgBASE_6783_145,
    output [6:0] ascii_6783_000,    ascii_6783_001,  ascii_6783_002,  ascii_6783_003,
                ascii_6783_004,     ascii_6783_005,  ascii_6783_006,  ascii_6783_007,
                ascii_6783_008,     ascii_6783_009,  ascii_6783_010,  ascii_6783_011,

                                              .
                                              .
                                              .

                ascii_6783_140,     ascii_6783_141,  ascii_6783_142,  ascii_6783_143,
                ascii_6783_144,     ascii_6783_145);
```

Perform converting by using pre-defined module

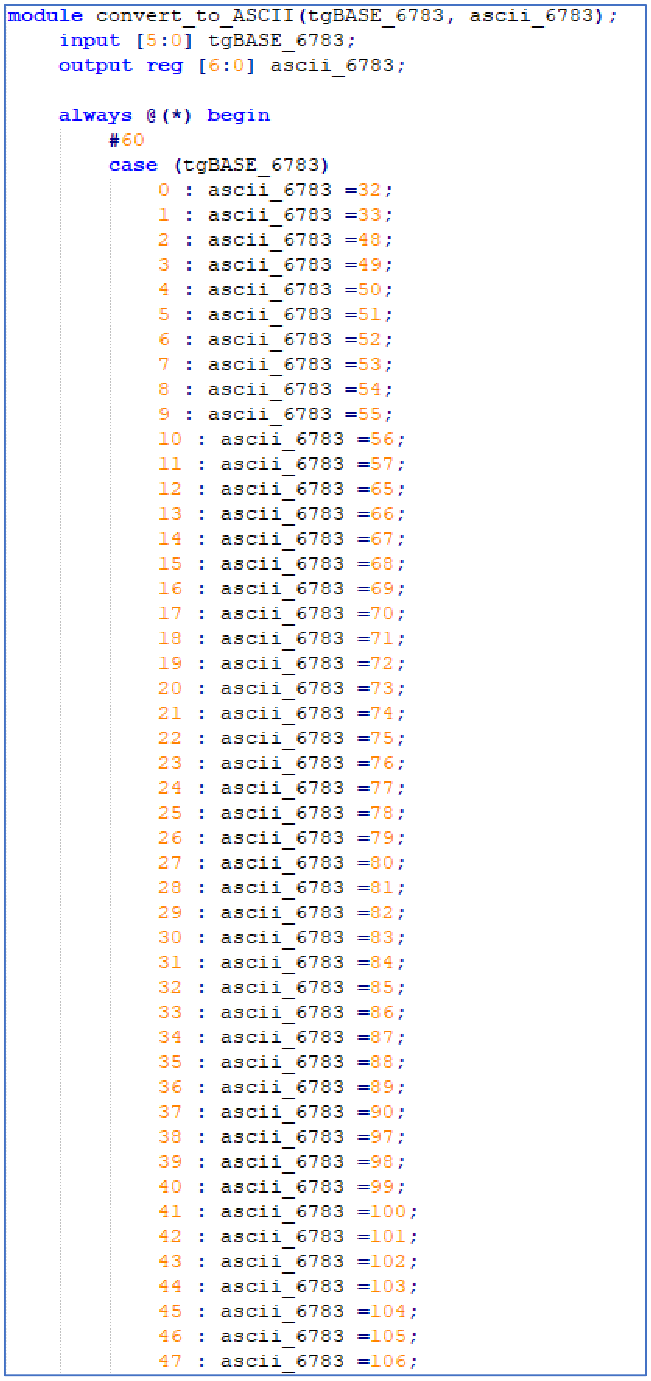
```
convert_to_ASCII c000(tgBASE_6783_000,ascii_6783_000);
convert_to_ASCII c001(tgBASE_6783_001,ascii_6783_001);
convert_to_ASCII c002(tgBASE_6783_002,ascii_6783_002);
convert_to_ASCII c003(tgBASE_6783_003,ascii_6783_003);
convert_to_ASCII c004(tgBASE_6783_004,ascii_6783_004);

                                 .
                                 .
                                 .

convert_to_ASCII c141(tgBASE_6783_141,ascii_6783_141);
convert_to_ASCII c142(tgBASE_6783_142,ascii_6783_142);
convert_to_ASCII c143(tgBASE_6783_143,ascii_6783_143);
convert_to_ASCII c144(tgBASE_6783_144,ascii_6783_144);
convert_to_ASCII c145(tgBASE_6783_145,ascii_6783_145);
```

Output View

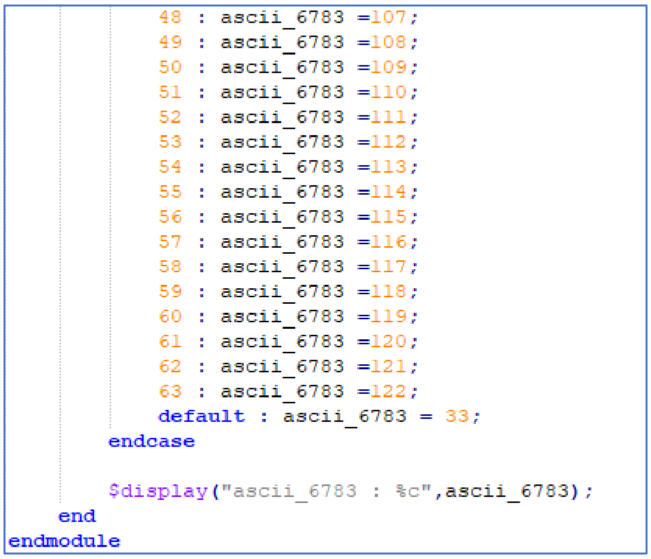


```
/main2/ascii_6783_000   7d67    7d67
/main2/ascii_6783_001   7d79    7d79
/main2/ascii_6783_002   7d77    7d77
/main2/ascii_6783_003   7d80    7d80
/main2/ascii_6783_004   7d51    7d51
/main2/ascii_6783_005   7d49    7d49
/main2/ascii_6783_006   7d49    7d49
/main2/ascii_6783_007   7d32    7d32
/main2/ascii_6783_008   7d105   7d105
/main2/ascii_6783_009   7d115   7d115
/main2/ascii_6783_010   7d32    7d32
/main2/ascii_6783_011   7d102   7d102
/main2/ascii_6783_012   7d117   7d117
/main2/ascii_6783_013   7d110   7d110
/main2/ascii_6783_014   7d33    7d33
/main2/ascii_6783_015   7d33    7d33
```

Perform converting by using pre-defined module



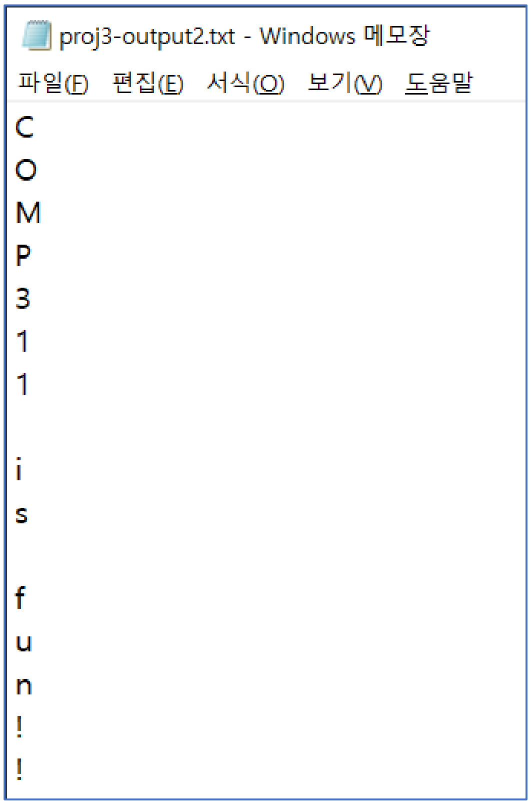
```
module convert_to_ASCII(tgBASE_6783, ascii_6783);
    input [5:0] tgBASE_6783;
    output reg [6:0] ascii_6783;

    always @(*) begin
        #60
        case (tgBASE_6783)
            0 : ascii_6783 =32;
            1 : ascii_6783 =33;
            2 : ascii_6783 =48;
            3 : ascii_6783 =49;
            4 : ascii_6783 =50;
            5 : ascii_6783 =51;
            6 : ascii_6783 =52;
            7 : ascii_6783 =53;
            8 : ascii_6783 =54;
            9 : ascii_6783 =55;
            10 : ascii_6783 =56;
            11 : ascii_6783 =57;
            12 : ascii_6783 =65;
            13 : ascii_6783 =66;
            14 : ascii_6783 =67;
            15 : ascii_6783 =68;
            16 : ascii_6783 =69;
            17 : ascii_6783 =70;
            18 : ascii_6783 =71;
            19 : ascii_6783 =72;
            20 : ascii_6783 =73;
            21 : ascii_6783 =74;
            22 : ascii_6783 =75;
            23 : ascii_6783 =76;
            24 : ascii_6783 =77;
            25 : ascii_6783 =78;
            26 : ascii_6783 =79;
            27 : ascii_6783 =80;
            28 : ascii_6783 =81;
            29 : ascii_6783 =82;
            30 : ascii_6783 =83;
            31 : ascii_6783 =84;
            32 : ascii_6783 =85;
            33 : ascii_6783 =86;
            34 : ascii_6783 =87;
            35 : ascii_6783 =88;
            36 : ascii_6783 =89;
            37 : ascii_6783 =90;
            38 : ascii_6783 =97;
            39 : ascii_6783 =98;
            40 : ascii_6783 =99;
            41 : ascii_6783 =100;
            42 : ascii_6783 =101;
            43 : ascii_6783 =102;
            44 : ascii_6783 =103;
            45 : ascii_6783 =104;
            46 : ascii_6783 =105;
            47 : ascii_6783 =106;
```



```
            48 : ascii_6783 =107;
            49 : ascii_6783 =108;
            50 : ascii_6783 =109;
            51 : ascii_6783 =110;
            52 : ascii_6783 =111;
            53 : ascii_6783 =112;
            54 : ascii_6783 =113;
            55 : ascii_6783 =114;
            56 : ascii_6783 =115;
            57 : ascii_6783 =116;
            58 : ascii_6783 =117;
            59 : ascii_6783 =118;
            60 : ascii_6783 =119;
            61 : ascii_6783 =120;
            62 : ascii_6783 =121;
            63 : ascii_6783 =122;
            default : ascii_6783 = 33;
        endcase

        $display("ascii_6783 : %c",ascii_6783);
    end
endmodule
```

Converting reversely from tgBASE to ASCII using given converting table

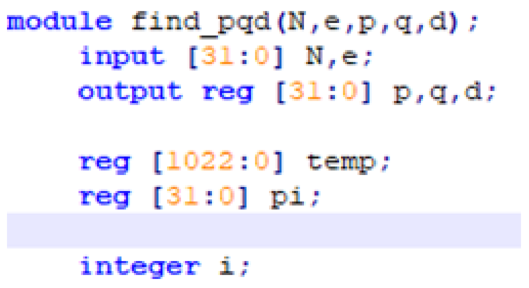
Output file


proj3-output2.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말

```
C
O
M
P
3
1
1

i
s

f
u
n
!
!
```

Part 3. Decrypting Practice

1. Declare module that find p, q, and d

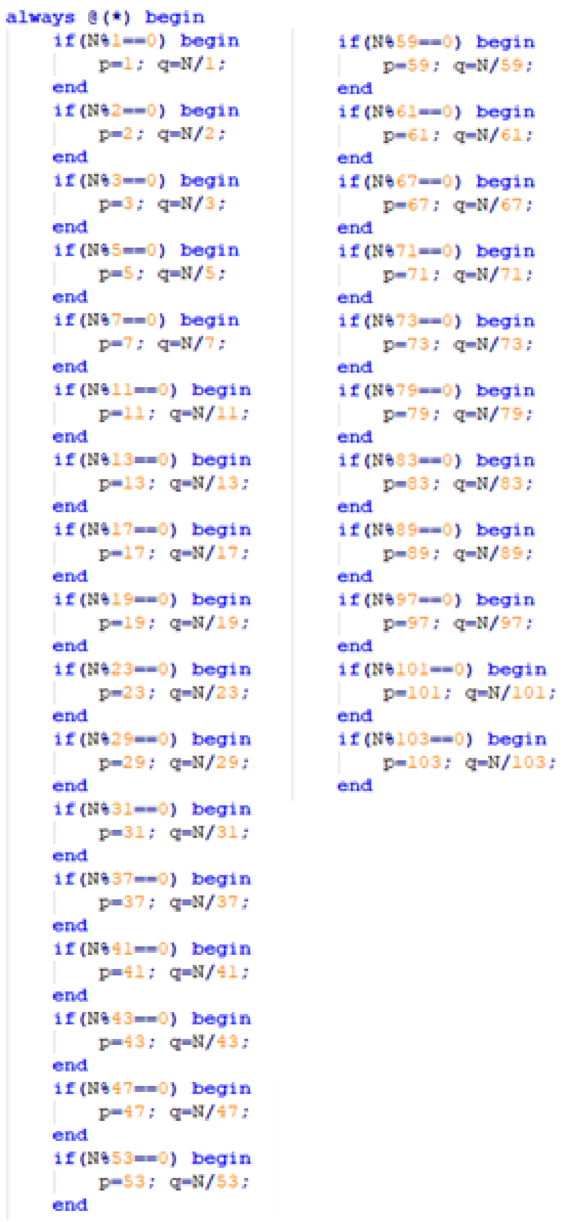


```
module find_pqd(N,e,p,q,d);
    input [31:0] N,e;
    output reg [31:0] p,q,d;

    reg [1022:0] temp;
    reg [31:0] pi;

    integer i;
```

Get 32bit decimal given input of N and e.
P,q,and d are also be 32bit decimal value.

Set temp, pi and I value
that will be used for calculating

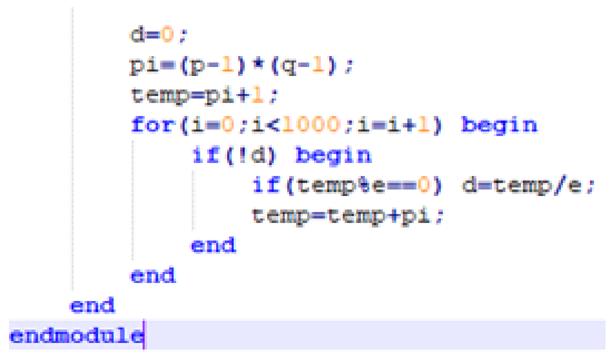
2. find p and q without loop statement



```
always @(*) begin
    if(N%1==0) begin
        p=1; q=N/1;
    end
    if(N%2==0) begin
        p=2; q=N/2;
    end
    if(N%3==0) begin
        p=3; q=N/3;
    end
    if(N%5==0) begin
        p=5; q=N/5;
    end
    if(N%7==0) begin
        p=7; q=N/7;
    end
    if(N%11==0) begin
        p=11; q=N/11;
    end
    if(N%13==0) begin
        p=13; q=N/13;
    end
    if(N%17==0) begin
        p=17; q=N/17;
    end
    if(N%19==0) begin
        p=19; q=N/19;
    end
    if(N%23==0) begin
        p=23; q=N/23;
    end
    if(N%29==0) begin
        p=29; q=N/29;
    end
    if(N%31==0) begin
        p=31; q=N/31;
    end
    if(N%37==0) begin
        p=37; q=N/37;
    end
    if(N%41==0) begin
        p=41; q=N/41;
    end
    if(N%43==0) begin
        p=43; q=N/43;
    end
    if(N%47==0) begin
        p=47; q=N/47;
    end
    if(N%53==0) begin
        p=53; q=N/53;
    end
    if(N%59==0) begin
        p=59; q=N/59;
    end
    if(N%61==0) begin
        p=61; q=N/61;
    end
    if(N%67==0) begin
        p=67; q=N/67;
    end
    if(N%71==0) begin
        p=71; q=N/71;
    end
    if(N%73==0) begin
        p=73; q=N/73;
    end
    if(N%79==0) begin
        p=79; q=N/79;
    end
    if(N%83==0) begin
        p=83; q=N/83;
    end
    if(N%89==0) begin
        p=89; q=N/89;
    end
    if(N%97==0) begin
        p=97; q=N/97;
    end
    if(N%101==0) begin
        p=101; q=N/101;
    end
    if(N%103==0) begin
        p=103; q=N/103;
    end
```

Because N is smaller than 10000,
maximum value of p and q is 100,
and the number of prime number is less.

3. find d with euclidean algorithm



```
        d=0;
        pi=(p-1)*(q-1);
        temp=pi+1;
        for(i=0;i<1000;i=i+1) begin
            if(!d) begin
                if(temp%e==0) d=temp/e;
                temp=temp+pi;
            end
        end
    end
endmodule
```

Initialize d, pi and temp value

Overall algorithm :

$$ed \bmod pi = 1$$

$$1 \bmod e \neq 0$$

$$(1 + pi) \bmod e \neq 0$$

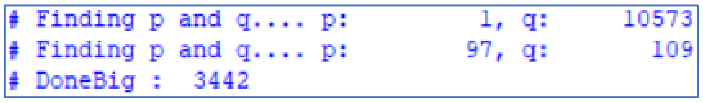
$$(1 + pi + pi) \bmod e \neq 0$$

$$\vdots$$

$$temp \bmod e = 0 \qquad d = temp \div e$$

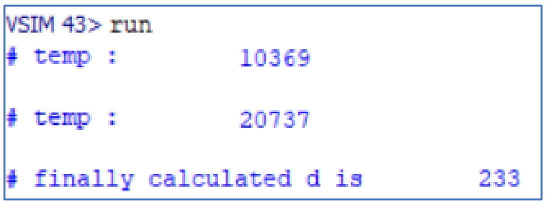
Finding p, q and d from N, e

Finding p and q (in this case, (97, 109) is only case of prime set at 3442)



```
# Finding p and q.... p:          1, q:      10573
# Finding p and q.... p:         97, q:        109
# DoneBig :   3442
```

Finding d (in this case, temp is pi * 2 +1 = 20737, which is perfectly divided by e=89)



```
VSIM 43> run
# temp :          10369

# temp :          20737

# finally calculated d is          233
```

Message Print

Part 1

Input

```
# input : 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001001000110010111011001101100110111101
0000101011111101111111100101101100110010001000010000000000000000
```

Input to ascii

```
# ascii :   0
# ascii :   0
# ascii :  33
# ascii : 100
# ascii : 108
# ascii : 114
# ascii : 111
# ascii :  87
# ascii :  32
# ascii : 111
# ascii : 108
# ascii : 108
# ascii : 101
# ascii :  72
```

Ascii to tgBASE

```
# tgBASE_6783 :  0
# tgBASE_6783 :  0
# tgBASE_6783 :  1
# tgBASE_6783 : 41
# tgBASE_6783 : 49
# tgBASE_6783 : 55
# tgBASE_6783 : 52
# tgBASE_6783 : 34
# tgBASE_6783 :  0
# tgBASE_6783 : 52
# tgBASE_6783 : 49
# tgBASE_6783 : 49
# tgBASE_6783 : 42
# tgBASE_6783 : 19
```

tgBASE to OneBig

```
# OneBig :     0
# OneBig :   141
# OneBig :  4955
# OneBig :  5234
# OneBig :    52
# OneBig :  4949
# OneBig :  4219
```

OneBig to Encrypt

```
# Encrypt :     0
# Encrypt :  9367
# Encrypt :  8490
# Encrypt :   704
# Encrypt :  5130
# Encrypt :   909
# Encrypt : 10137
```

Encrypt to Decrypt

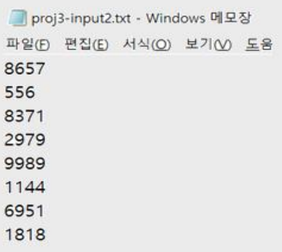
```
# DoneBig :     0
# DoneBig :   141
# DoneBig :  4955
# DoneBig :  5234
# DoneBig :    52
# DoneBig :  4949
# DoneBig :  4219
```

output

```
# output : 00000000000000
# output : 00000010001101
# output : 01001101011011
# output : 01010001110010
# output : 00000000110100
# output : 01001101010101
# output : 01000001111011
```

Part 2

Input file



proj3-input2.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움

```
8657
556
8371
2979
9989
1144
6951
1818
```

Decrypting

```
# DoneBig :  1426
# DoneBig :  2427
# DoneBig :   503
# DoneBig :   300
# DoneBig :  4656
# DoneBig :    43
# DoneBig :  5851
# DoneBig :   101
```

To tgBASE

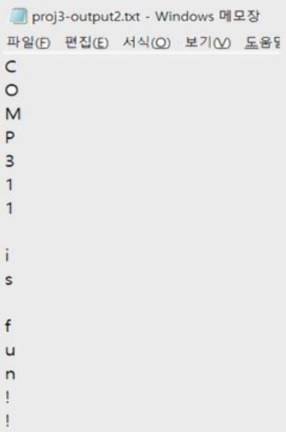
```
# tgBASE : 14
# tgBASE : 26
# tgBASE : 24
# tgBASE : 27
# tgBASE :  5
# tgBASE :  3
# tgBASE :  3
# tgBASE :  0
# tgBASE : 46
# tgBASE : 56
# tgBASE :  0
# tgBASE : 43
# tgBASE : 58
# tgBASE : 51
# tgBASE :  1
# tgBASE :  1
```

To ascii code

```
# ascii_6783 : !
# ascii_6783 : !
# ascii_6783 : n
# ascii_6783 : u
# ascii_6783 : f
# ascii_6783 :  
# ascii_6783 : s
# ascii_6783 : i
# ascii_6783 :  
# ascii_6783 : 1
# ascii_6783 : 1
# ascii_6783 : 3
# ascii_6783 : P
# ascii_6783 : M
# ascii_6783 : O
# ascii_6783 : C
```

reversed because of the delay

output file



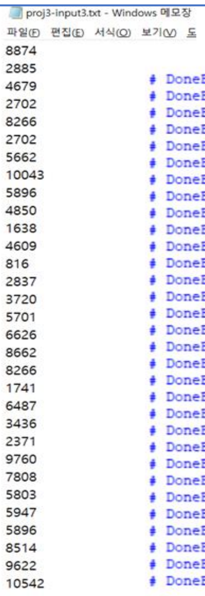
proj3-output2.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

```
C
O
M
P
3
1
1

i
s

f
u
n
!
!
```

Part 3

Input



proj3-input3.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도

```
8874
2885
4679
2702
8266
2702
5662
10043
5896
4850
1638
4609
816
2837
3720
5701
6626
8662
8266
1741
6487
3436
2371
9760
7808
5803
5947
5896
8514
9622
10542
```

Decrypting

```
# DoneBig :  3442
# DoneBig :  4949
# DoneBig :    41
# DoneBig :  5251
# DoneBig :  4200
# DoneBig :  5251
# DoneBig :    62
# DoneBig :  5258
# DoneBig :  5500
# DoneBig :  1746
# DoneBig :  5138
# DoneBig :  4900
# DoneBig :  2755
# DoneBig :  5247
# DoneBig :  4240
# DoneBig :  5701
# DoneBig :    19
# DoneBig :  3859
# DoneBig :  4200
# DoneBig :  3800
# DoneBig :  5638
# DoneBig :  4342
# DoneBig :     4
# DoneBig :   204
# DoneBig :   260
# DoneBig :  4651
# DoneBig :  5742
# DoneBig :  5500
# DoneBig :  3955
# DoneBig :  4238
# DoneBig :  4801
```

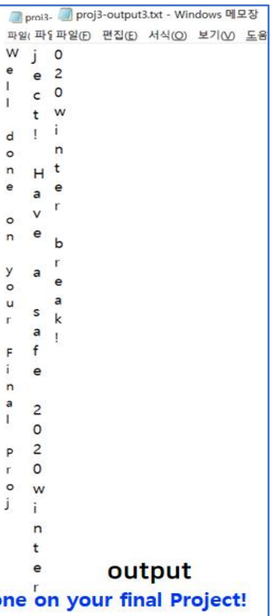
To tgBASE

```
# tgBASE :  1    # tgBASE : 34
# tgBASE :  0    # tgBASE : 42
# tgBASE : 19    # tgBASE : 49
# tgBASE : 38    # tgBASE : 49
# tgBASE : 59    # tgBASE :  0
# tgBASE : 42    # tgBASE : 41
# tgBASE :  0    # tgBASE : 52
# tgBASE : 38    # tgBASE : 51
# tgBASE :  0    # tgBASE : 42
# tgBASE : 56    # tgBASE :  0
# tgBASE : 38    # tgBASE : 52
# tgBASE : 43    # tgBASE : 51
# tgBASE : 42    # tgBASE :  0
# tgBASE :  0    # tgBASE : 62
# tgBASE :  4    # tgBASE : 52
# tgBASE :  2    # tgBASE : 58
# tgBASE :  4    # tgBASE : 55
# tgBASE :  2    # tgBASE :  0
# tgBASE : 60    # tgBASE : 17
# tgBASE : 46    # tgBASE : 46
# tgBASE : 51    # tgBASE : 51
# tgBASE : 57    # tgBASE : 38
# tgBASE : 42    # tgBASE : 49
# tgBASE : 55    # tgBASE :  0
# tgBASE :  0    # tgBASE : 27
# tgBASE : 39    # tgBASE : 55
# tgBASE : 55    # tgBASE : 52
# tgBASE : 42    # tgBASE : 47
# tgBASE : 38    # tgBASE : 42
# tgBASE : 48    # tgBASE : 40
# tgBASE :  1    # tgBASE : 57
```

To ASCII

```
# ascii_6783 : !      # ascii_6783 :  
# ascii_6783 : k      # ascii_6783 : !
# ascii_6783 : a      # ascii_6783 : t
# ascii_6783 : e      # ascii_6783 : c
# ascii_6783 : r      # ascii_6783 : e
# ascii_6783 : b      # ascii_6783 : j
# ascii_6783 :        # ascii_6783 : o
# ascii_6783 : r      # ascii_6783 : r
# ascii_6783 : e      # ascii_6783 : P
# ascii_6783 : t      # ascii_6783 :  
# ascii_6783 : n      # ascii_6783 : l
# ascii_6783 : i      # ascii_6783 : a
# ascii_6783 : w      # ascii_6783 : n
# ascii_6783 : 0      # ascii_6783 : i
# ascii_6783 : 2      # ascii_6783 : F
# ascii_6783 : 0      # ascii_6783 :  
# ascii_6783 : 2      # ascii_6783 : r
# ascii_6783 :        # ascii_6783 : u
# ascii_6783 : e      # ascii_6783 : o
# ascii_6783 : f      # ascii_6783 : y
# ascii_6783 : a      # ascii_6783 :  
# ascii_6783 : s      # ascii_6783 : n
# ascii_6783 :        # ascii_6783 : o
# ascii_6783 : a      # ascii_6783 :  
# ascii_6783 :        # ascii_6783 : e
# ascii_6783 : e      # ascii_6783 : n
# ascii_6783 : v      # ascii_6783 : o
# ascii_6783 : a      # ascii_6783 : d
# ascii_6783 : H      # ascii_6783 :  
                      # ascii_6783 : l
                      # ascii_6783 : l
                      # ascii_6783 : e
                      # ascii_6783 : W
```

output



proj3- proj3-output3.txt - Windows 메모장

파일(파일 파일(F) 편집(E) 서식(O) 보기(V) 도움

```
W j 0
e e 2
l c 0
l t w
  ! i
d   n
o H t
n a e
e v r
  e  
o   b
n a r
    e
y s a
o a k
u f !
r e
   
F 2
i 0
n 2
a 0
l w
  i
P n
r t
o e
j r
```

Well done on your final Project!

Have a safe 2020winter break!