lab13

```
1 // EE231002 Lab13. Text Decoding
2 // 108061213, 劉奕緯
3 // Dec. 19, 2019
4
5 #include <stdio.h>              // getchar();  putchar();
6
7 // output original text from four number
8 void Print(unsigned char code[4], int i);
9 // transfer code to number
10 unsigned char decoder(unsigned char c);
11
12 int main (void)
   int main(void)
13 {
14     unsigned char code[4];      // store encoded texts
15     int i = 0;                  // #valid code in code[4] array
16     char ch;                    // buffer for input
17
18     // discard first line
19     while ((ch = getchar()) != '\n');
20     // input the text till find '-'
21     while ((ch = getchar()) != '-') {
22         if (ch == '=') {        // no more text
23             Print(code, i);     // output
24             i = 0;              // clear code array
25             while (getchar() != '\n');
26                                 // since no more code afterward
27         }
28         else if ((ch = decoder(ch)) >= 0) {
29                                 // if we obtain a valid code
30             code[i++] = ch;     // store it in code array
31             if (i == 4) {       // while receiving four codes
32                 Print(code, i); // print out the original text
33                 i = 0;          // clear code array
34             }
35         }
36     }
37     // discard the last line and end file
38     while ((ch = getchar()) != '\n');
39     return 0;
40 }
   Need a blank line here.
41 // transfer code to number
42 unsigned char decoder(unsigned char c)
43 {
44     if ('A' <= c && c <= 'Z')
45         return c - 'A';         // 'A' is 0, B is 1, and so on.
46     if ('a' <= c && c <= 'z')
```

```
47          return c - 'a' + 0x1A;  // 'a' is 0x1A, b is 0x1B, and so on.
48      if (48 <= c && c <= 57)
49          return c - '0' + 0x34;  // '0' is 0x34, '1' is 0x35, and so on.
50      if (c == '+')
51          return 0x3E;            // given
52      if (c == '/')
53          return 0x3F;            // given
54      return -1;                  // invalid code, i assigned it to -1.
55  }
    Need a blank line here.
56  // output three letters from four codes
57  void Print(unsigned char code[4], int i)
58  {
59      unsigned char ascii[3];     // original texts
60
61      if (i == 3) code[3] = 0;    // we have one '='
62      if (i == 2) {               // we have two '='
63          code[2] = 0;
64          code[3] = 0;
65      }
66      // below three line is recovering split codes back to original texts.
67      ascii[0] = (code[0] << 2) | (code[1] >> 4);
68      ascii[1] = (code[1] << 4) | (code[2] >> 2);
69      ascii[2] = (code[2] << 6) | code[3];
70      // output the three original texts
71      putchar(ascii[0]);
72      if (i > 2)
73          putchar(ascii[1]);
74      if (i > 3)
75          putchar(ascii[2]);
76  }
77
```

[Format] can be improved.

Score: 97