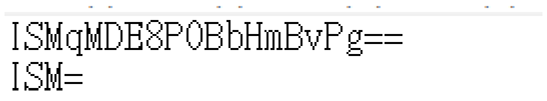
1.Read in the file. Use strlen() so that we can read in data line by line. We use a 2D char array to store each line.

2.Include <stdarg.h> to use getopt() for easily parsing command line arguments in shell scripts. If getopt() finds an option character in argv that was not included in optstring, or if it detects a missing option argument, it returns '?' and sets the external variable optopt to the actual option character. If the first character of optstring is a colon , then getopt() returns ':' instead of '?' to indicate a missing option argument.

3.Because our input is string type, we have to convert it to int type. Use the ASCII code to know the relationships between characters and integers.

4.Because every 3 hexadicimal characters (4 bits each) can convert into 2 base64 characters (6 bits each), we use group = strlen(strLine[n])/3 and remain = strlen(strLine[n])%3 to represent how many groups of base64 characters and how many numbers are left.

5.The convertion is kind of trivial. Say there are three binary numbers 1010 0101 0110. We know we should let 101001 and 010110 be group respectively. So we left shift 1010 for two digits (101000) and right shift 0101 for two digits (0001), then add both(101001). Next, left shift 0101 for four bits(01010000) and add 0110, we will get 100110. To avoid overflow problem, we should let the number module 4 before left shifting. Voilà! We have reached our goal!



```
ISMqMDE8POBbHmBvPg==
ISM=
```

6.Now we just turn the hexadicimal to base64 index. We then have to check the base64 table to see what character should it be for each index. That's why we call the function int2base64.

7.Finally use fprintf() to write the results into the output file. Remember to close the fin and fout stream to release the resource.