

**소스파일**

```
#include <iostream> // cout 사용
#include <fstream> // 파일 입력을 위해 사용
#include <string> // string 클래스 사용
#include <conio.h> // getch() 사용
#include <iomanip>
using namespace std;

struct WordPair {
    string eng;
    string kor;
};

#define MAXWORDS 200
class MyDic {
    WordPair words[MAXWORDS];
    int nWords; // 등록 단어 수

public:
    MyDic() { nWords = 0; }
    void add(string e, string k) { // 단어 등록
        words[nWords].eng = e;
        words[nWords].kor = k;
        nWords++;
    }
    void load(const char* filename) {
        ifstream f(filename);
        f >> nWords;
        for (int i = 0; i < nWords; i++)
            f >> words[i].eng >> words[i].kor;
        f.close();
    }
    void store(char* filename) {
        ofstream f(filename);
        f << nWords << endl;
        for (int i = 0; i < nWords; i++)
            f << setw(20) << words[i].eng << " " << words[i].kor
              << endl;
        f.close();
    }
    int count() {
        return nWords;
    }
    string getEng(int a) {
```

```

        return words[a].eng;
    }
    string getKor(int a) {
        return words[a].kor;
    }
};

class Hangman {
    string progress[64]; // 행맨 진행 상태 출력 그림
    string problem; // 문제 단어
    string answer; // 현재 진행 중인 단어
    string guessed; // 전체 알파벳의 현재까지 예측 상황
    int nTries; // 현재까지 틀린 횟수
    const int maxTries = 7; // 최대 허용 시도 횟수 제한
    void load(const char* progName = "HangmanProgress.txt") {
        ifstream fs(progName);
        if (fs) {
            getline(fs, progress[0]); // 첫 번째 행은 주석
            for (int i = 0; i < 64; i++)
                getline(fs, progress[i]);
        }
    }
    void print() {
        system("cls");
        cout << " <Hangman Game>\n";
        for (int i = 0; i < 8; i++)
            cout << 'Wt' << progress[nTries * 8 + i] << endl;
        cout << "WnWt" << answer;
        cout << "WnWn " << guessed;
    }
    int countMatched(char ch) {
        int nMatched = 0;
        for (int pos = -1; ; ) {
            pos = problem.find(ch, pos + 1);
            if (pos < 0) break;
            answer[pos] = ch; // 현재 단어를 갱신(맞힌 문자 열기)
            nMatched++;
        }
        return nMatched;
    }
    void guess() {
        char ch = _getch();
        if (ch >= 'a' && ch <= 'z') {
            int pos = guessed.find(ch);
            if (pos < 0) { // 아직 추측하지 않은 문자
                guessed[ch - 'a'] = ch; // 예측한 문자 해당 위치 열기
            }
        }
    }
};

```

```

        if (countMatched(ch) == 0) nTries++; // 실패 횟수 증가
    }
}

public:
    void play(string prob, string solve) {
        load();
        problem = prob;
        answer = string(problem.length(), '-');// 생성자 사용
        guessed = string(26, '.');// 생성자 사용
        nTries = 0;
        while (nTries < maxTries && answer != problem) {
            print();
            guess();
        }
        print();
        cout << "WnWt" << ((nTries == maxTries) ? "실패" : "정답") << endl;
        cout << "정답 : " << prob << endl;
        cout << "뜻 : " << solve << endl;
    }
};

int main() {
    Hangman game;
    MyDic dic;
    srand((unsigned int)time(NULL));
    dic.load("tmp.dic");
    int i = rand() % dic.count();
    game.play(dic.getEng(i), dic.getKor(i));
    return 0;
}

```

## tmp.dic

```

5
hello 안녕
computer 컴퓨터
school 학교
student 학생
coffee 커피

```

```
<Hangman Game>
+---+
|   |
O
|
+-----+
|
B+-----+
|
school

...c....h....l...o...st.....
정답
맞 : school
틀 : 학교
```

```
<Hangman Game>
+---+
|
|
0
|
/  \
|  \
/  \
|
0+-----
-offee

a..def.....mno.qr.t.....
실패
정답 : coffee
옳 : 커피
```