

C프로그래밍(2)

기말 프로젝트
TYPING GAME

학 번 : 1805040

학 반 : 2반

이 름 : 김 준 영

```

1  #pragma comment(lib, "libmysql.lib") // 데이터베이스를 사용하기 위해 mysql 라이브러리 파일 호출
2  #include <stdio.h> // 표준 입출력함수 printf(), scanf()를 사용하기 위해 선언
3  #include <stdlib.h> // system()함수를 사용하기 위해 선언
4  #include <mysql.h> // 데이터베이스를 사용하기 위해 선언
5  #include <time.h> // srand(time)함수를 사용하기 위해 선언
6  // 연산속도 측정을 하는 clock()를 사용하기 위해 선언
7  #include <Windows.h> // gotoxy(), Sleep(), removeCursor() 함수를 사용하기 위해 선언
8  #include <conio.h> // 비표준 입출력함수 _kbhit() 함수를 사용하기 위해 선언
9  #include <string.h> // strcmp() 함수를 사용하기 위해 선언
10
11 // 데이터베이스 필요
12 #define DB_HOST "127.0.0.1" // 루프백 주소 / 자기 로컬 주소
13 #define DB_USER "root" // 아이디
14 #define DB_PASS "1234" // 비밀번호 desctop 1234
15 #define DB_NAME "login" // 데이터베이스 이름
16 #define CHOP(x) x[strlen(x) - 1] = ' ' // 엔터키를 눌렀을 때 배열에 공백이 들어가기 위한 함수
17
18 // 게임필요
19 #define wordmax 103 // 단어의 개수
20 #define wordtime 40000 // 단어가 뜨는 속도
21
22 void gotoxy(int x, int y) { // 커서 위치 조정 함수
23     COORD Pos = { x - 1, y - 1 };
24     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
25 }
26
27 void removeCursor() { // 콘솔에서 커서 삭제하는 함수
28     CONSOLE_CURSOR_INFO curInfo;
29     GetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &curInfo);
30     curInfo.bVisible = 0;
31     SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &curInfo);
32 }
33
34 void help();// 도움말
35 void gamemain();// 게임메인
36 void word_io();// 단어 입출력 해주는 함수
37 void design();// 인터페이스 화면
38 void rank(); // 랭킹 화면
39 void menu();// 게임 시작 초기 화면
40 void setting(); // 시간 설정 메뉴
41 void removeCursor();// printf에서 단어를 뜨게할때 커서 제거
42 int database(); // 데이터베이스 함수
43
44 // DB에 쓰일 변수
45 int query_stat; // 쿼리가 제대로 넘어갔는지 오류체크
46 char ans; // 로그인 Y/N 유무 확인
47 char id[20]; // 아이디 입력
48 char pass[20]; // 비밀번호 입력
49 char check[255]; // 아이디, 비밀번호 체크를 쿼리로 날리기 위해 임시버퍼 선언
50 char query[255]; // 아이디, 비밀번호를 쿼리로 날리기 위해 임시버퍼 선언
51 char query1[255]; // 회원가입시 아이디와 점수를 쿼리로 날리기 위해 임시버퍼 선언
52 char query2[255]; // 최고 점수를 기록하기 위해 아이디와 점수를 쿼리로 날리기 위해 임시버퍼 선언
53 char query3[255]; // 로그인을 했을 때 점수를 불러오기 위해 임시버퍼 선언
54 int dbscore; // DB에 저장될 점수
55 int dbtimer; // DB에 저장될 시간
56 MYSQL* connection = NULL, conn; // 연결 정보 관리
57 MYSQL_RES* sql_result; // 결과 값을 메모리에 저장
58 MYSQL_ROW sql_row; // 행 단위 불러오기
59 int count = 1; //로그인 화면을 한 번만 뜨게 하기 위해 선언
60
61 // 게임 필요 변수
62 int gap; // 게임 시간 카운터
63 int wordi[100]; //한 번 출력된 단어들은 다시 출력이 안되게 하기 위해 선언
64 int x, y; // gotoxy 의 x값과 y값
65 int i; // 단어 수
66 int x2, y2; // x,y 값을 다시 불러올 변수
67 int score; // 게임점수를 0으로 초기화
68 int h; // 초기화면에서 메뉴선택을 입력 받을 변수
69 int f; // 색깔
70 int f2; // 색깔 값을 다시 불러올 변수
71 int sword; // scan 변수의 인덱스 값을 증가시키기 위한 변수
72 char scan[50]; // 키보드로 부터 단어를 입력받을 변수
73 char remem[wordmax][10]; // 게임을 시작할 때 단어를 임시 저장하고 출력하기 위한 2차원 배열
74 int timer = 40; // 게임의 남은 시간을 설정하는 변수 (초기값 : 40초)
75 time_t startTime, endTime; // 게임 시간 제한
76 // 단어 수 총 103개
77 // 2차원 배열 word row = Wordmax(103), col = 11
78 // row의 값은 총 103개의 단어가 들어가기 때문에 103으로 설정
79 // 단어의 최대 글자 수는 10개이지만 문자열은 끝에 \0이 들어가기 때문에 col값은 11으로 설정
80 char word[wordmax][11] = { "dog", "cat", "bottle", "phone", "robot", "green", "elephant", "include", "sky", "game",
    "greenday", "kimchanho", "max",
81
82 "knife", "glass", "class", "art", "smart", "bell", "carry", "climb", "between", "blow", "album", "ralo", "among", "animal",
    "any", "box",
83

```

```

84  "and", "board", "body", "monsrat", "classmate", "city", "boy", "bridge", "clean",
85
86  "club", "coat", "bright", "coin", "chopstick", "coffee", "cold", "chance", "chalk", "chair", "cheap", "blue",
87
88  "before", "bowl", "aunt", "as", "away", "bicycle", "paka", "card", "hold",
89
90  "chose", "come", "drink", "give", "get", "hurt", "lay", "had", "feed", "lend", "rand", "wsing", "throw", "wet", "tell",
91
92  "set", "wind", "wear", "write", "spend", "stand", "worn", "win", "sweep", "account", "achieve", "across", "accept", "above",
    "ability", "abuse",
93
94  "abnormal", "absurd", "acceptance", "according", "absent", "nation", "past", "value", "though", "person", "machine", "stand",
    "null" };
95
96  int main() {
97      //로그인 화면을 한 번만 뜨게 하기 위해
98      if (count == 1) { // 만약 count가 1이면
99          database(); // 데이터베이스 함수 호출
100         sprintf(query3, "select score, time from rank where id = '%s'", id); // query3에 rank 테이블에 있는 로그인한 id의 점수값
            을 가져옴
101         query_stat = mysql_query(connection, query3); // query3이 정상적으로 값이 넘어갔는지 확인
102         sql_result = mysql_store_result(connection); // select score from rank where id = '%s' 메모리에 저장
103
104         while ((sql_row = mysql_fetch_row(sql_result)) != NULL) { // sql_row가 NULL값이 아닐 때 까지 출력
105             dbscore = atoi(sql_row[0]); // sql_row[0] 값을 dbscore에 저장
106             dbtimer = atoi(sql_row[1]); // sql_row[1] 값을 dbtimer에 저장
107             //atoi : 문자스tring 함수를 정수로 변환
108         }
109         mysql_free_result(sql_result); // 메모리 해제
110     }
111
112     srand((unsigned)time(NULL)); // 랜덤 함수
113     system("mode con cols=120 lines=30"); // 화면 사이즈 조절
114     design(); // 인터페이스 화면 함수 호출
115     Sleep(1500); // 인터페이스 화면 출력 후 1.5초 딜레이
116     gotoxy(60, 9);
117     removeCursor(); // 커서이동
118
119     while (1) { // 메뉴화면을 계속 출력하기 위한 반복문
120         menu(); // 메뉴화면 출력
121         gotoxy(70, 19);
122         scanf("%d", &h); // 메뉴 선택
123         fflush(stdin); // 표준 입력(키보드) 버퍼를 비워 버리는 함수
124         switch (h) { // Switch 선택문으로 설정화면 선택
125             case 1:
126                 score = 0; // 현재점수 0 으로 초기화
127                 gap = 0;
128                 gamemain(); // 게임시작 함수 호출
129                 break; // 점수 초기화, 게임시작 함수 호출 후 게임 시작
130             case 2:
131                 help(); // 도움말 함수 호출
132                 break; // 도움말 화면 출력
133             case 3:
134                 rank(); // 랭킹 화면 함수 호출
135                 break; // switch문 나가기
136             case 4:
137                 setting(); // 시간설정 함수 호출
138                 break; // switch문 나가기
139             case 5:
140                 mysql_close(connection); // mysql 연결 종료
141                 gotoxy(45, 25);
142                 printf("*****게임이 종료됩니다*****");
143                 _getch();
144                 return 0; // 게임종료
145             default: // 다른 키를 입력 했을 경우
146                 gotoxy(50, 21);
147                 printf("올바른 키가 아닙니다!"); // 경고창 출력
148                 _getch(); // 키보드로 부터 입력 대기
149                 system("cls"); // 화면 삭제
150                 break; // switch문 나가기
151         }
152     }
153     return 0;
154 }
155
156 void menu() { // 메뉴화면 함수
157     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); // 글자색을 바꾸는 함수, 흰색으로 출력
158     system("cls"); // 화면 지우기
159     design(); // 인터페이스 함수 호출
160     gotoxy(44, 3);
161     printf("%s님 안녕하세요!!", id);
162     gotoxy(44, 5);
163     printf("*****"); // 메뉴 화면 디자인
164     gotoxy(44, 6);
165     printf("*****TYPING GAME*****"); // 메뉴 화면 디자인

```

```
166     gotoxy(44, 7);
167     printf("*****"); // 메뉴 화면 디자인
168     gotoxy(50, 9);
169     printf(" 1. 게임시작");// 1. 게임시작
170     gotoxy(50, 11);
171     printf(" 2. 도움말"); // 2. 도움말
172     gotoxy(50, 13);
173     printf(" 3. 랭 킹"); // 3. 랭킹
174     gotoxy(50, 15);
175     printf(" 4. 시간설정"); // 4. 시간설정
176     gotoxy(50, 17);
177     printf(" 5. 게임종료"); // 5. 게임종료
178     gotoxy(45, 19);
179     printf(" Please enter the key [  ]");// 메뉴 선택
180     gotoxy(43, 21);
181     printf("[ 영남이공대학교 1805040 김준영 ]");
182 }
183
184 void design() { // 인터페이스 화면 함수
185
186     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); // 글자색을 바꾸는 함수, 흰색으로 출력
187     // 인터페이스 화면
188     printf
189         ( " |-----
190         -|Wn");
191     printf(" |
192         |Wn");
193     printf(" |
194         |Wn");
195     printf(" |
196         |Wn");
197     printf(" |
198         |Wn");
199     printf(" |
200         |Wn");
201     printf(" |
202         |Wn");
203     printf(" |
204         |Wn");
205     printf(" |
206         |Wn");
207     printf(" |
208         |Wn");
209     printf(" |
210         |Wn");
211     printf
212         ( " |-----
213         -|Wn");
214     printf(" |
215         |Wn");
216     printf(" |
217         |Wn");
218     printf(" |
219         |Wn");
220     printf(" |
221         |Wn");
222     printf(" |
223         |Wn");
224     printf
225         ( " |-----
226         -|");
```

```

218 }
219
220 void help() { // 도움말
221     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); // 글자색을 바꾸는 함수, 흰색으로 출력
222     system("cls"); // 화면 지우기
223     design(); // 인터페이스 화면 함수 호출
224     gotoxy(35, 7);
225     printf("- 무작위로 뜨는 단어들을 없애주세요!!"); // 화면 출력
226     gotoxy(35, 9);
227     printf("- 색마다 점수 배점이 다릅니다!!"); // 화면 출력
228     gotoxy(35, 11);
229
230     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1); // 글자색을 바꾸는 함수, 파랑색으로 출력
231     printf("파랑색 : 150점 "); // 화면 출력
232
233     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2); // 글자색을 바꾸는 함수, 초록색으로 출력
234     printf("초록색 : 200점 "); // 화면 출력
235
236     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3); // 글자색을 바꾸는 함수, 하늘색으로 출력
237     printf("하늘색 : 250점 "); // 화면 출력
238
239     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4); // 글자색을 바꾸는 함수, 빨강색으로 출력
240     printf("빨강색 : 300점 "); // 화면 출력
241
242     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); // 글자색을 바꾸는 함수, 흰색으로 출력
243     gotoxy(35, 15);
244     printf("- 최고점수가 저장됩니다."); // 화면 출력
245     gotoxy(35, 17);
246     printf("- 메인메뉴 랭킹에서 확인 가능합니다."); // 화면 출력
247     gotoxy(35, 19);
248     printf("- 아무키나 누르면 메인 메뉴로 돌아갑니다."); // 화면 출력
249     _getch(); // 키보드로 부터 입력 대기
250 }
251
252 void rank() { // 랭킹 화면 함수
253     int y = 7; // gotoxy의 y값 7로 초기화
254     int x = 1; // 순위를 나타내기 위한 변수 1로 초기화
255
256     system("cls"); // 화면 지우기
257     design(); // 인터페이스 화면 출력
258
259     query_stat = mysql_query(connection, "select * from rank order by score desc");
260     // 정상적으로 쿼리가 DB로 넘어가면 "select * from rank order by score dsec"문 출력
261     // 점수를 내림차순으로 출력
262     sql_result = mysql_store_result(connection); // select * from rank order by score desc 메모리에 저장
263
264     gotoxy(40, 3);
265     printf("-----랭      킹-----"); // 랭킹
266     gotoxy(48, 5);
267     printf("아이디"); // 아이디 출력
268     gotoxy(65, 5);
269     printf("점수"); // 점수 출력
270     gotoxy(72, 5);
271     printf("시간"); // 점수 출력
272     while ((sql_row = mysql_fetch_row(sql_result)) != NULL) { // sql_row가 NULL값이 아닐 때 까지 출력
273         gotoxy(45, y);
274         printf("%d", x); // 순위 표시 1부터 ~
275         gotoxy(48, y);
276         printf("%s", sql_row[0]); // ID 출력
277         gotoxy(65, y);
278         printf("%s", sql_row[1]); // 점수 출력
279         gotoxy(72, y);
280         printf("%s", sql_row[2]); // 점수 출력
281         x++; // 변수 o 증감
282         y++; // 변수 q 증감
283     }
284
285     gotoxy(40, 16);
286     printf("아무키나 누르면 메인 메뉴로 돌아갑니다."); // 화면 출력
287     _getch(); // 키보드로부터 입력대기
288     mysql_free_result(sql_result); // 메모리 해제
289 }
290
291 void setting() { // 시간 설정하는 함수
292     system("cls"); // 화면 지우기
293     design(); // 인터페이스 화면 호출
294     gotoxy(48, 10);
295     printf("현재 설정 시간 : %d", timer); // 현재 설정되어 있는 시간 출력
296     gotoxy(48, 12);
297     printf("시간을 설정하세요 : "); // 시간 설정 화면 출력
298     scanf("%d", &timer); // 시간 입력 ex) 10, 30, 60, 90초 단위로 설정 가능
299
300     system("cls"); // 설정이 완료되면 화면 지우기
301     design(); // 인터페이스 화면 출력
302     gotoxy(48, 10);

```

```

303     printf("시간이 %d 초로 설정되었습니다!", timer); // 시간이 입력한 값으로 설정 되었다는 화면 출력
304
305     gotoxy(48, 12);
306     printf("계속하시려면 아무 키나 누르세요."); // 아무 키나 누르세요
307     _getch(); // 키보드로부터 입력대기
308 }
309
310 void gamemain() { // 게임시작 함수
311     gotoxy(50, 9);
312     system("cls"); // 화면 지우기
313     design(); // 인터페이스 화면 함수 호출
314
315     gotoxy(50, 9);
316     printf(" 3초 후 시작 !!"); // 화면 출력
317     Sleep(1000); // 1초 기다립니다.
318
319     system("cls"); // 화면 지우기
320     design(); // 인터페이스 화면 함수 호출
321
322     gotoxy(50, 9);
323     printf(" 2초 후 시작 !!"); // 화면 출력
324     Sleep(1000); // 1초 기다립니다.
325
326     system("cls"); // 화면 지우기
327     design(); // 인터페이스 화면 함수 호출
328
329     gotoxy(50, 9);
330     printf(" 1초 후 시작 !!"); // 화면 출력
331     Sleep(1000); // 1초 기다립니다.
332
333     system("cls"); // 화면 지우기
334     startTime = clock(); // 시간 설정 초기화 clock 함수 호출
335     design(); // 인터페이스 화면 함수 호출
336
337     while (timer >= gap) { // 게임이 끝날 때 까지 반복
338         word_io(); // 단어를 입력하는 함수 호출
339     }
340     if (score > dbscore) { // 현재점수가 dbscore 보다 클 경우
341         dbscore = score; // 현재점수를 dbscore에 저장
342         if (timer != dbtimer) { // 설정시간이 dbscore와 같지 않을 경우
343             dbtimer = timer; // 설정시간을 dbtimer에 저장
344         }
345         sprintf(query2, "update rank set score = '%d', time = '%d' where id = '%s'", dbscore, dbtimer, id);
346         // query2에 rank 테이블에 아이디, 점수, 시간 값을 저장
347         query_stat = mysql_query(connection, query2); // query2이 정상적으로 값이 넘어갔는지 확인
348
349
350         system("cls"); // 화면 지우기
351         design(); // 인터페이스 화면 함수 호출
352         gotoxy(50, 11);
353         printf("----최 고 기 록!!!----"); // 최고기록 표시
354         Sleep(2000); // 2초간 기다립니다.
355     }
356     system("cls"); // 화면 지우기
357     design(); // 인터페이스 화면 함수 호출
358     gotoxy(50, 11);
359     printf("----게 임 종 료----"); // 게임이 종료되었다는 화면 출력
360     Sleep(2000); // 2초간 기다립니다.
361 }
362
363 void word_io() { // 단어를 입출력하는 함수
364
365     x = rand() % 105 + 5; // 단어를 x축 5에서 105까지 랜덤으로 출력
366     y = rand() % 19 + 3; // 단어를 y축 3에서 19까지 랜덤으로 출력
367     i = rand() % wordmax; // 0에서 103(wordmax = 103) 번째 단어를 랜덤으로 출력
368     f = rand() % 4 + 1; // 단어의 색을 파랑색, 초록색, 하늘색, 빨강색으로 랜덤 부여
369
370     if (wordi[i] != 1) { // 단어를 한 번만 뜨게 하는 배열 wordi 인덱스 값이 1이 아닌 경우
371         wordi[i] = 1; // 배열 wordi에 1을 저장
372
373         gotoxy(x, y); // 랜덤으로 설정한 x, y축으로 단어 출력
374         for (int j = 0; j < 3; j++) { // 반복문을 사용하기 위해 j를 선언하고 0으로 초기화, j가 3보다 클때까지 반복, 반복할
            때마다 j 1증감
375             if (j == 0) { // 만약 j가 0일때
376                 remem[i][j] = x; // 단어를 출력하기 위한 임시 2차원배열 remem에 단어의 x축 값 저장
377             }
378             else if (j == 1) { // 만약 j가 1일때
379                 remem[i][j] = y; // 단어를 출력하기 위한 임시 2차원배열 remem에 단어의 y축 값 저장
380             }
381             else if (j == 2) { // 만약 j가 3일때
382                 remem[i][j] = f; // 단어를 출력하기 위한 임시 2차원 배열 remem에 단어의 색깔 값을 저장
383             }
384         }
385         SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), f); // 글자색을 바꾸는 함수, 랜덤 색깔로 출력
386         printf("%s ", word[i]); //배열 word에 저장된 0 ~ 103열에 있는 단어를 출력

```



```

387     }
388
389     int p = 0; // 단어를 출력하기 위한 변수 p 선언 후 0으로 초기화
390     while (p < wordtime) { // p가 wordtime(40000)보다 클 때까지 반복
391         p++; // p 증감
392         i = 0; // 단어 수를 입력 받기 위해 0으로 초기화
393         int scanc = 0; // 키보드 입력 값을 받을 변수
394         endTime = clock(); // 코드가 끝난 시간 저장
395         gap = (double)(endTime - startTime) / (CLOCKS_PER_SEC); // 초 단위로 시간을 출력하기 위함
396
397         gotoxy(19, 2);
398         SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); // 글자색을 바꾸는 함수, 랜덤 색깔로 출력
399         printf("[ 남은 시간 : %d / %d ]", gap, timer); // [ 남은시간 : 0 / 40 ] 남은 시간을 알려줌 ggp은 1초마다 증가 / timer
        메뉴 4. 시간설정에서 설정 값 출력
400
401         if (_kbhit()) { // 만약 키보드가 입력 되었을 경우
402             scanc = _getch(); // 키보드 입력 값을 변수 scanc에 저장
403             if (scanc != 8 && scanc != 13) { // 만약 키보드로부터 scanc에 저장된 키 값이 아스키 코드 값 8(backspace(<-)) 그리
                고 13(carriage return(Wr))이 아닐 경우
404                 scan[sword++] = scanc; // 키보드 입력으로 부터 저장된 scanc 값을 문자를 입력받는 scan[sword == 0] 변수에 저장
                (반복을 진행할 때 마다 증감)
405                 // ex) Hello 입력 scan[0] = 'H', scan[1] = 'e', scan[2] = 'l', scan[3] = 'l', scan[4]
                = 'o'
406                 gotoxy(5, 25);
407                 SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); // 글자색을 바꾸는 함수, 흰색으로 출력
408                 printf("%s", scan); // scan변수에 저장된 키보드로 입력된 값을 출력해 줌
409             }
410             else if (scanc == 8) { // backspace 입력 됨 (만약 scanc에 저장된 키 값이 아스키 코드 8(backspace(<-))일 경우)
411                 gotoxy(1, 25);
412                 printf("|
                |Wn");
413                 // backspace가 입력 될 경우 단어가 정상적으로 지워지도록 하기 위해 출력
414                 gotoxy(5, 25);
415                 scan[--sword] = scanc; // 키보드 입력으로 부터 저장된 scanc 값을 문자를 입력받는 scan[sword - 1] 변수에 저장
                된 값을 삭제
416                 // ex) Heloo(오타)를 입력 scan[4] = 'o' 삭제 scan[3] = 'o' 삭제
417                 if (sword == -1) // 만약 sword가 -1일 경우
418                     sword = 0; // scan 변수의 인덱스 값을 나타내주는 변수이기 때문에 0으로 초기화 하면 다시 scan[0]부터 입력
                을 받음
419                 printf("%s", scan); // scan변수에 저장된 키보드로 입력된 값을 출력해 줌
420             }
421             else if (scanc == 13) { //만약 키보드로부터 scanc에 저장된 키 값이 아스키코드 값 13(Wr)일 경우
422                 sword = 0; // scan 변수의 인덱스 값을 나타내주는 변수이기 때문에 0으로 초기화 하면 다시 scan[0]부터 입력을 받
                음
423                 gotoxy(1, 25);
424                 printf("|
                |Wn");
425                 // Wr(줄바꿈)이 입력 될 경우 단어를 다시 정상적으로 입력 받기 위해서 출력
426                 for (i = 0; i <= wordmax; i++) { // 단어 수(i)를 0으로 초기화, 단어 수(i)가 최대 단어수 wordmax(103)보다 크거
                나 같을 때까지 단어 수(i++)를 증감하면서 반복
427                     if (!strcmp(scan, word[i])) { // word[i]에 있는 단어와 키보드로 부터 입력 된 scan에 저장된 값이 같을 경우
428                         x2 = remem[i][0]; // x 좌표를 다시 불러올 변수 x2에 임시 배열 remem[i][0] 값을 저장
429                         y2 = remem[i][1]; // y 좌표를 다시 불러올 변수 x2에 임시 배열 remem[i][1] 값을 저장
430                         f2 = remem[i][2]; // 색을 다시 불러올 변수 f2에 임시 배열 remem[i][2] 값을 저장
431
432                         gotoxy(x2, y2); // 임시 배열 remem에 저장 받은 복사된 랜덤 x2, y2축 값을 출력
433                         printf("
                        "); // 복사된 x, y 축 값에 가서 출력된 단어를 공백 처리로 출력하여 지움
434                         gotoxy(1, 25);
435                         printf("|
                        |Wn");
436                         // Wr(줄바꿈)이 입력 될 단어를 정상적으로 입력받기 위해서 출력
437                         switch (f2) { // 임시 배열 remem에 저장 되고 복사된 랜덤 색깔 지정 변수 f2에 Switch문 생성
438                             // 기존의 f (색을 1 ~ 4로 랜덤 부여 됨)
439                             case 1: score += 150; break; // 파랑색 글자에 현재점수 150점 부여
440                             case 2: score += 200; break; // 초록색 글자에 현재점수 200점 부여
441                             case 3: score += 250; break; // 하늘색 글자에 현재점수 250점 부여
442                             case 4: score += 300; break; // 빨강색 글자에 현재점수 300점 부여
443                             default: break; //Swith문 빠져나옴
444                         }
445                         gotoxy(3, 2);
446                         SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15); //글자 색을 바꿔주는 함수, 흰색으로 설
                        정
447                         printf("현재점수 : %d", score); // 현재점수를 나타내 줌
448                     }
449                 }
450                 for (i = 0; i < 50; i++) { // i를 0으로 초기화, i가 50보다 클때까지 i를 증감하면서 반복
451                     scan[i] = NULL; // scan[i]에 저장된 값을 키보드에 입력된 값을 NULL처리
452                 }
453             }
454         }
455     }
456 }
457
458 int database() { // 데이터베이스 함수
459

```

```

460     mysql_init(&conn);
461
462     //한글문자셋 설정 (오류방지)
463     mysql_options(&conn, MYSQL_SET_CHARSET_NAME, "euckr");
464     mysql_options(&conn, MYSQL_INIT_COMMAND, "SET NAMES euckr");
465
466     //연결 mysql_real_connect
467     connection = mysql_real_connect(&conn, DB_HOST, DB_USER, DB_PASS, DB_NAME, 3306, NULL, 0);
468
469     system("mode con: cols=34 lines=20"); // 화면 사이즈 조절
470
471     if (connection == NULL) { // 연결 정보가 없을 때
472         fprintf(stderr, "실패원인 : %s\n", mysql_error(&conn)); // 실패 원인 출력
473         return 1; // 비정상 종료
474     }
475
476     query_stat = mysql_query(connection, "select * from login"); // 정상적으로 쿼리가 DB로 넘어가면 "select * from login" 문
출력
477     if (query_stat != 0) { // 만약 query_stat에 0값이 넘어가지 않으면 ( 0값이 넘어가면 정상 )
478         fprintf(stderr, "실패원인 : %s\n", mysql_error(&conn)); // 실패 원인 출력
479         return 1; // 비정상 종료
480     }
481
482     while (1) {
483
484         query_stat = mysql_query(connection, "select * from login"); // 정상적으로 쿼리가 DB로 넘어가면 "select * from login"
문 출력
485         printf("Wt   회원 목록\n"); // 회원 목록 출력
486         printf("Wt [ 아이디 ]\n"); // 아이디 출력
487         sql_result = mysql_store_result(connection); // select * from login 메모리에 저장
488         while ((sql_row = mysql_fetch_row(sql_result)) != NULL) { // sql_row가 NULL값이 아닐 때 까지 출력
489             printf("Wt  %10s\n", sql_row[0]); // 행 단위로 출력
490         }
491
492         printf("WnWn [ 회원가입(Y) / 로그인(N) ] : "); // 회원가입 여부 출력
493         scanf("%c", &ans); // 키보드로 부터 입력
494         scanf("%*c"); // 입력된 값을 변수에 저장하지 않고 무시
495
496         if (ans == 'y' || ans == 'Y') { // 만약 ans가 'y' 이거나 'Y' 일 경우
497             // 회원가입
498             system("cls"); // 화면 지우기
499
500             gotoxy(12, 3);
501             printf("회 원 가 입"); // 회원가입 화면 출력
502
503             gotoxy(3, 5);
504             printf("아이디 : "); // 아이디 입력창 출력
505             fgets(id, 20, stdin); // 아이디 입력을 20 크기까지 입력을 받음
506             CHOP(id); // 엔터키가 들어오면 공백으로 처리
507
508             gotoxy(3, 8);
509             printf("패스워드 : "); // 패스워드 입력창 출력
510             fgets(pass, 20, stdin); // 패스워드 입력을 20 크기까지 입력을 받음
511             CHOP(pass); // 엔터키가 들어오면 공백으로 처리
512
513             sprintf(query, "insert into login values" "('%s', '%s')", id, pass); //query를 login 테이블에 아이디 패스워드 값
을 저장
514             query_stat = mysql_query(connection, query); // query가 정상적으로 값이 넘어갔는지 확인
515
516             sprintf(query1, "insert into rank values" "('%s', '%d', '%d')", id, 0, 0); // query1을 rank 테이블에 아이디, 점
수, 시간 값을 저장
517             query_stat = mysql_query(connection, query1); // query1이 정상적으로 값이 넘어갔는지 확인
518
519             gotoxy(4, 11);
520             printf("회원가입이 완료되었습니다!"); // 회원가입이 완료되었다는 화면 출력
521             _getch(); // 키보드로 부터 입력대기
522             system("cls"); // 화면 지우기
523         }
524         else if (ans == 'N' || ans == 'n') { // 만약 ans가 'n' 이거나 'N' 일 경우
525             //로그인
526             system("cls"); // 화면 지우기
527
528             gotoxy(12, 3);
529             printf("로 그 인"); // 로 그 인 화면 출력
530
531             gotoxy(3, 5);
532             printf("아이디 : "); // 아이디 입력창 출력
533             fgets(id, 20, stdin); // 아이디 입력을 20 크기까지 입력을 받음
534             CHOP(id); // 엔터키가 들어오면 공백으로 처리
535
536             gotoxy(3, 8);
537             printf("패스워드 : "); // 비밀번호 입력창 출력
538             fgets(pass, 20, stdin); // 패스워드 입력을 20 크기까지 입력을 받음
539             CHOP(pass); // 엔터키가 들어오면 공백으로 처리
540

```



```
541     sprintf(check, "select * from login where id = '%s' and passwd = '%s'", id, pass); // query에 저장된 아이디와 패
    스워드가 맞는지 확인
542     query_stat = mysql_query(connection, check); // query에 정상적으로 값이 넘어갔는지 확인
543     sql_result = mysql_store_result(connection); //select * from login 메모리에 저장
544
545     if ((sql_row = mysql_fetch_row(sql_result)) == NULL) { // 만약 sql_row값이 NULL일 경우
546         system("cls"); // 화면 지우기
547         gotoxy(3, 6);
548         printf("회원정보가 일치하지 않습니다."); // 경고문 출력
549         gotoxy(6, 8);
550         printf("프로그램이 종료됩니다..."); // 프로그램 종료창
551         _getch(); // 키보드로부터 입력 대기
552         exit(1); // 프로그램 종료
553     }
554
555     system("cls"); // 화면 지우기
556     gotoxy(11, 5);
557     printf("로그인 성공!!"); // 로그인 성공 출력
558
559     gotoxy(9, 7);
560     printf("%s님 환영합니다!!\n", id); // 로그인한 아이디를 출력
561
562     gotoxy(2, 9);
563     count++; // 로그인 화면을 한 번만 뜨게 하기 위해 count증감
564     printf("시작하시려면 아무 키나 누르세요."); // 아무키나 누르세요 화면 출력
565     _getch(); // 키 입력 대기
566     mysql_free_result(sql_result); // 메모리 해제
567     break; // While문 빠져나가기
568 }
569 }
570 return 0; // 함수 종료
571 }
```