단어 없애기 게임 구현

자유 주제

2019 June

19120450김 유리

대구가톨릭대학교 IT공학부

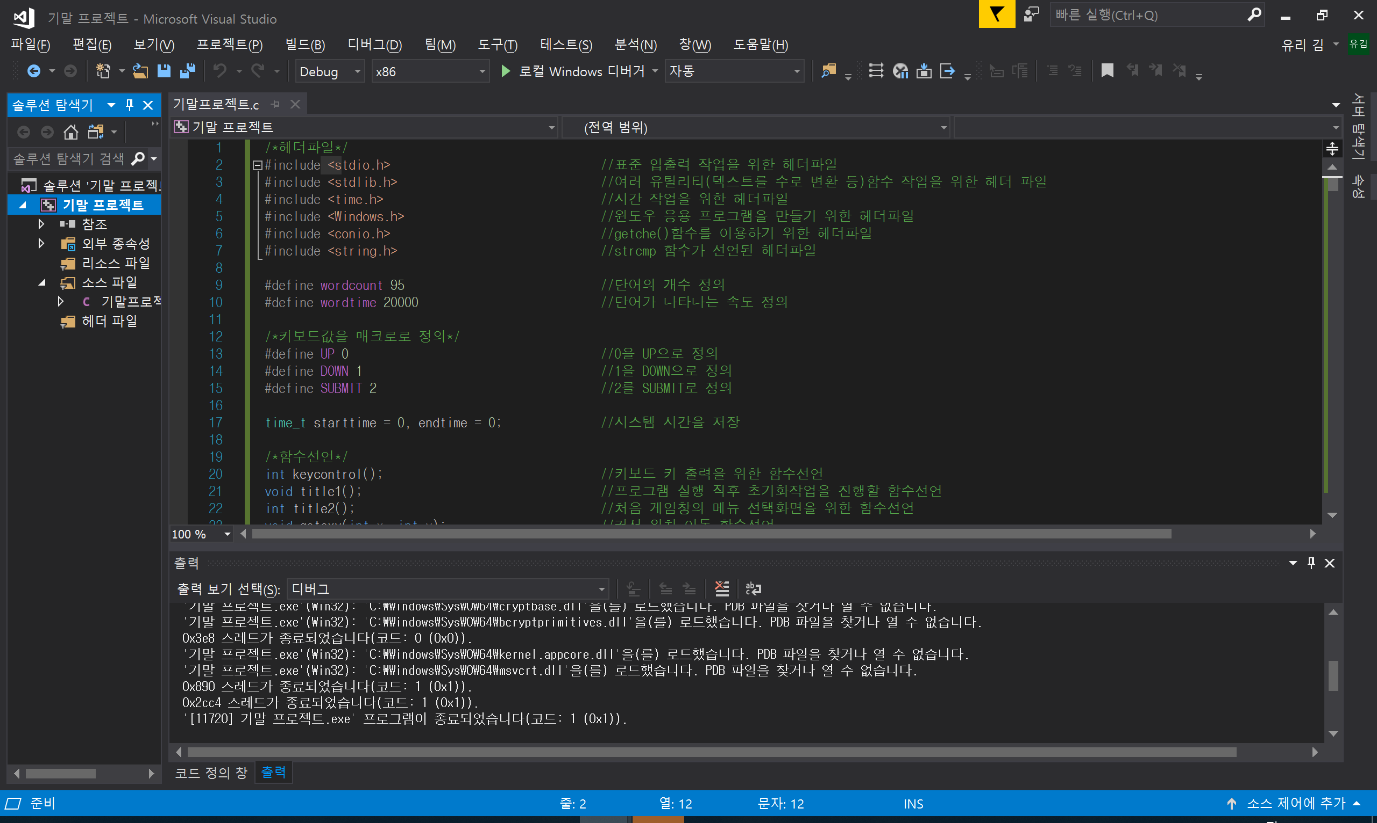
[목 차]

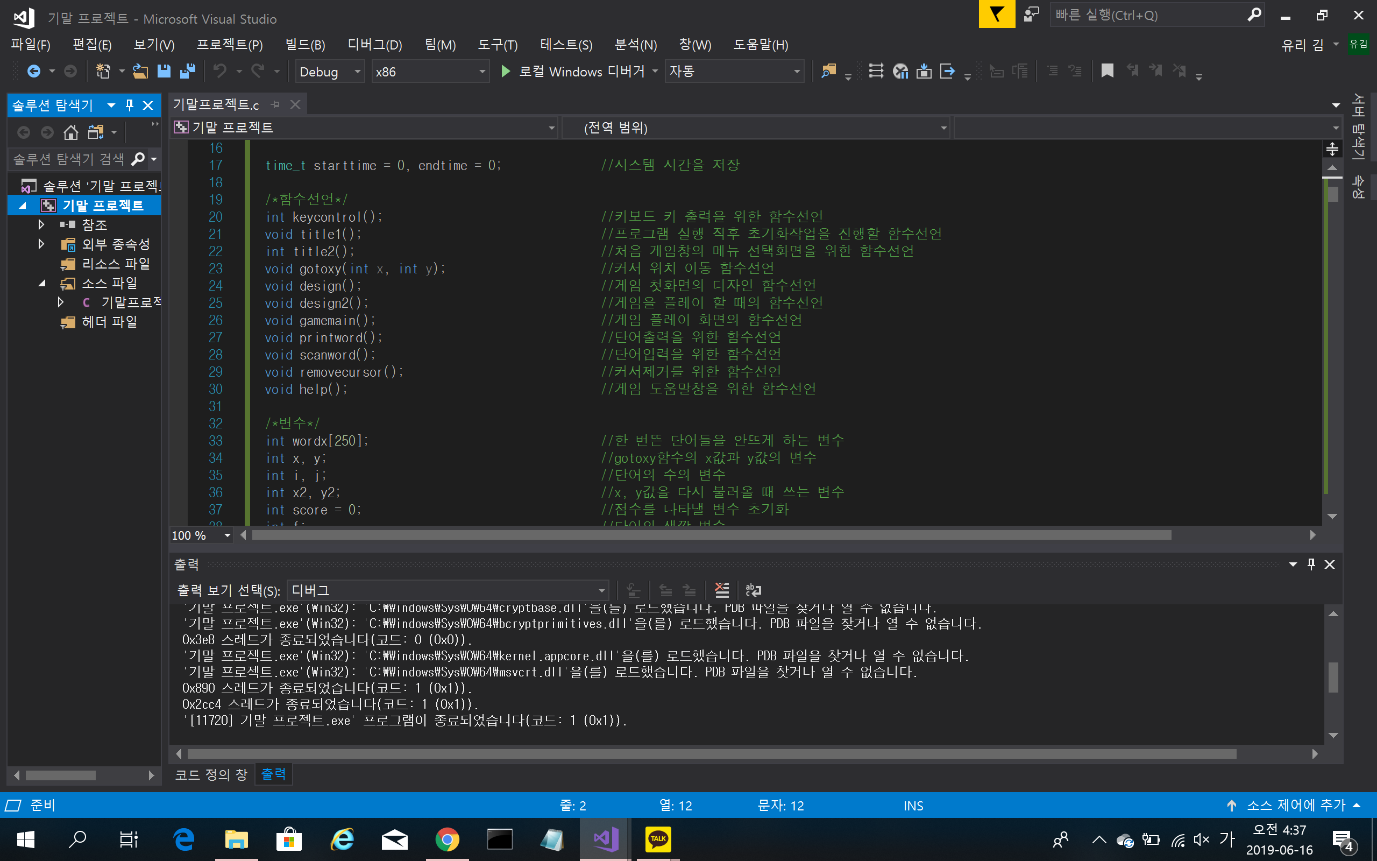
1. 서 론 ------------------------------------------------ 2  
2. 관련 연구 ---------------------------------------------- 3   
3. 프로그램 설계 ----------------------------------------- 4  
4. 프로그램의 구현 -------------------------------------- 5  
5. 프로그램의 동작과 결과 -------------------------- 15  
6. 결과 분석 및 토의 ---------------------------------- 18  
7. 결론 ------------------------------------------------------ 18  
8. 참고 문헌 ---------------------------------------------- 19  
9. 별첨 (소스코드와 주석) --------------------------- 20

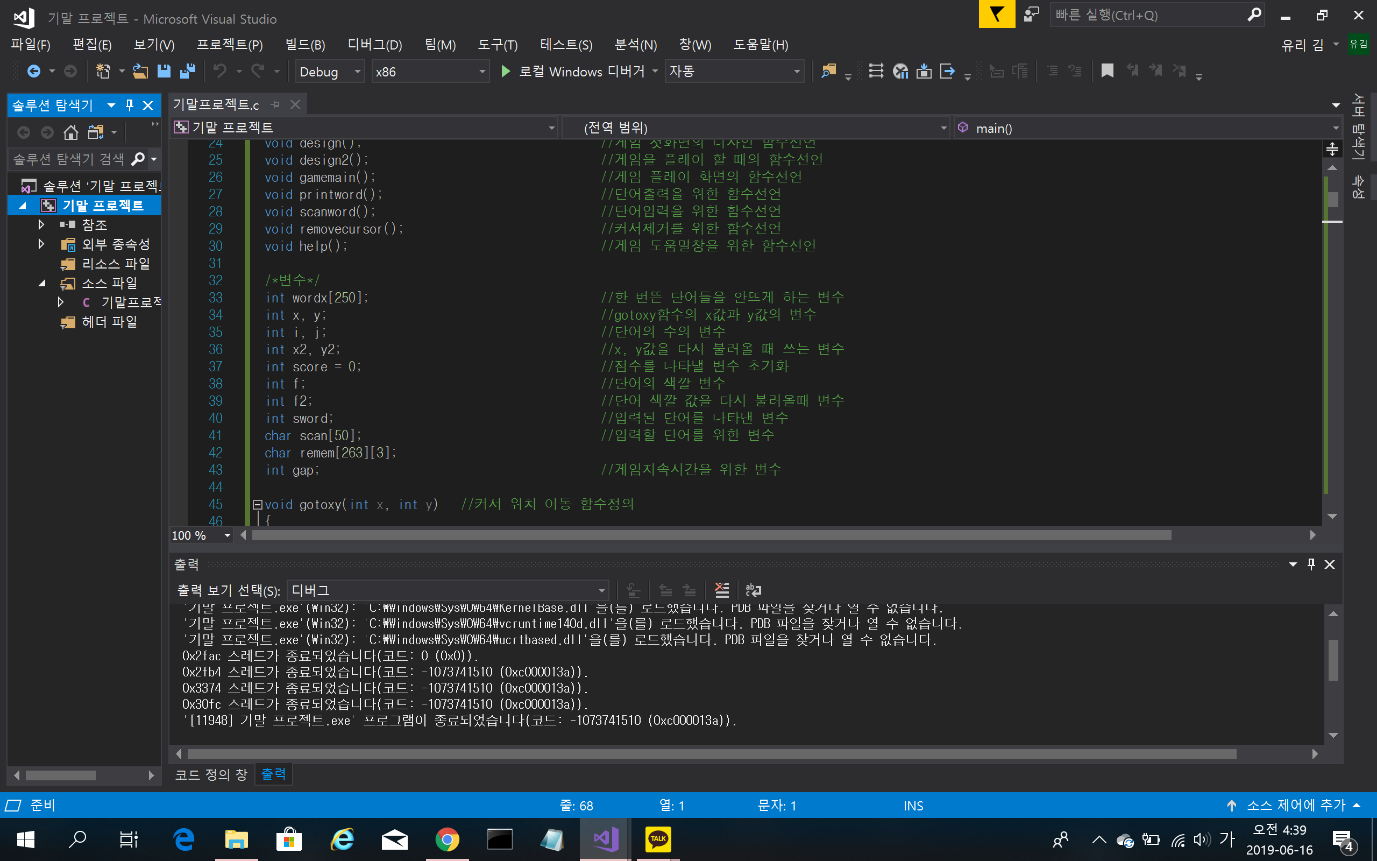
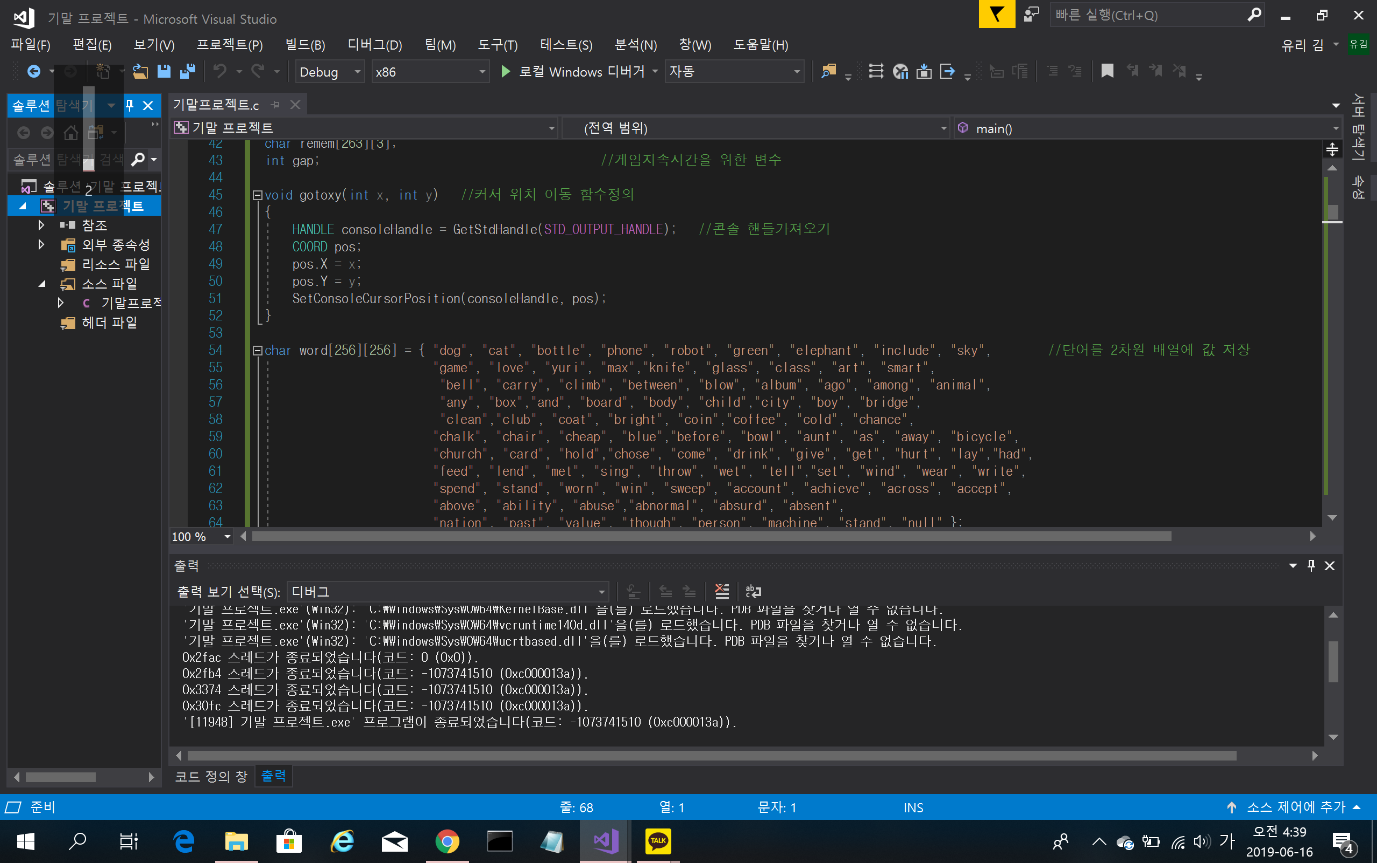
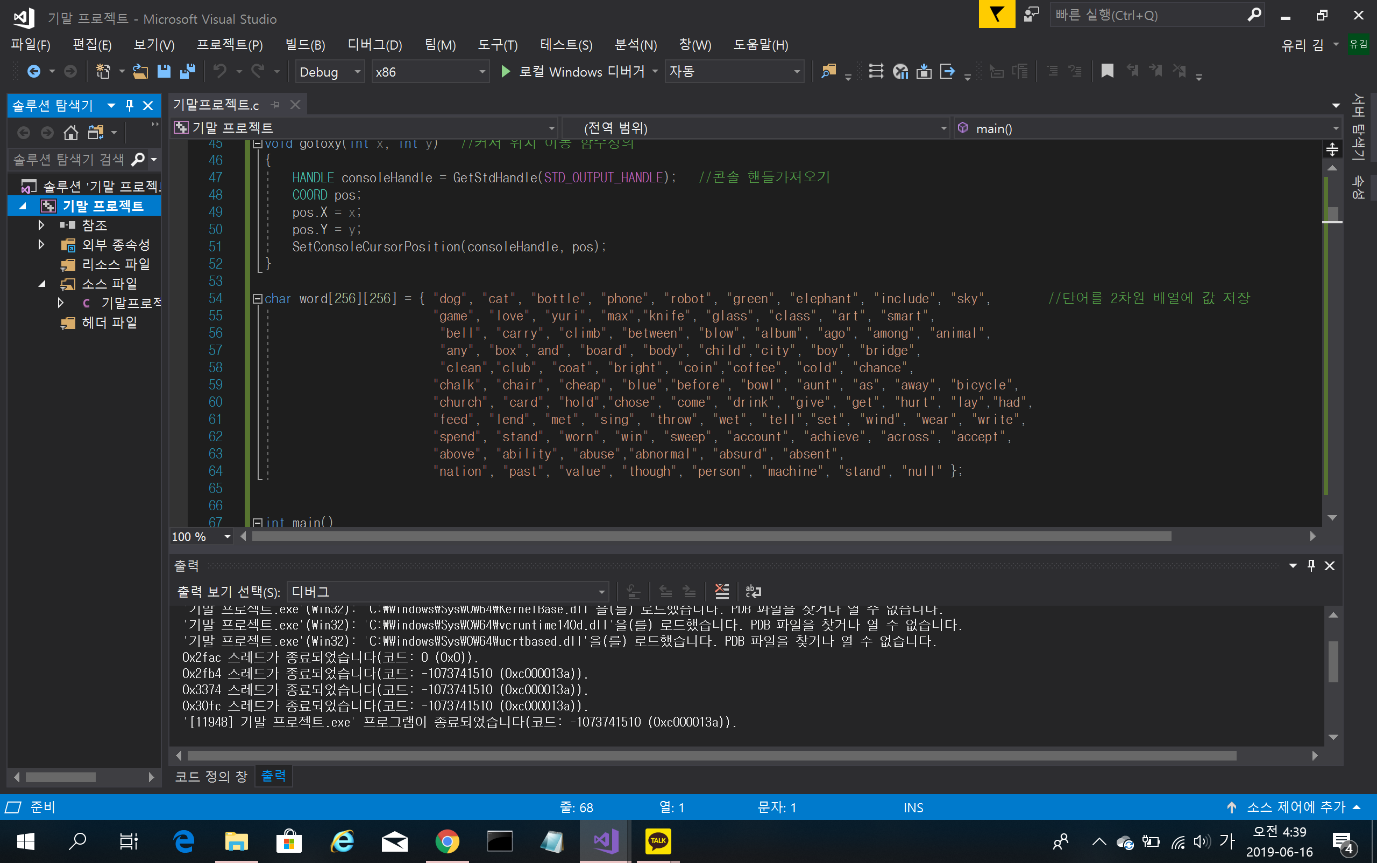
1.서론  
  
이 보고서는 기말 프로젝트 위한 보고서로 전공 수업시간에 배운 C언어를 이용하여 만든 ‘단어 없애기 게임’에 대한 보고서다. 이 게임은 산성비와 비슷한 게임으로 화면에 나타난 단어를 키보드로 입력하여 없애는 게임이다. 이 보고서에서는 게임을 만들기 위해 쓴 헤더 파일, 함수 등에 대해 소개 할 것이며, 어떻게 이 게임을 구현하고 설계 했는지 소개하고 마지막으로 이 게임을 만들면서 느낀 점을 소개 할 것이다.

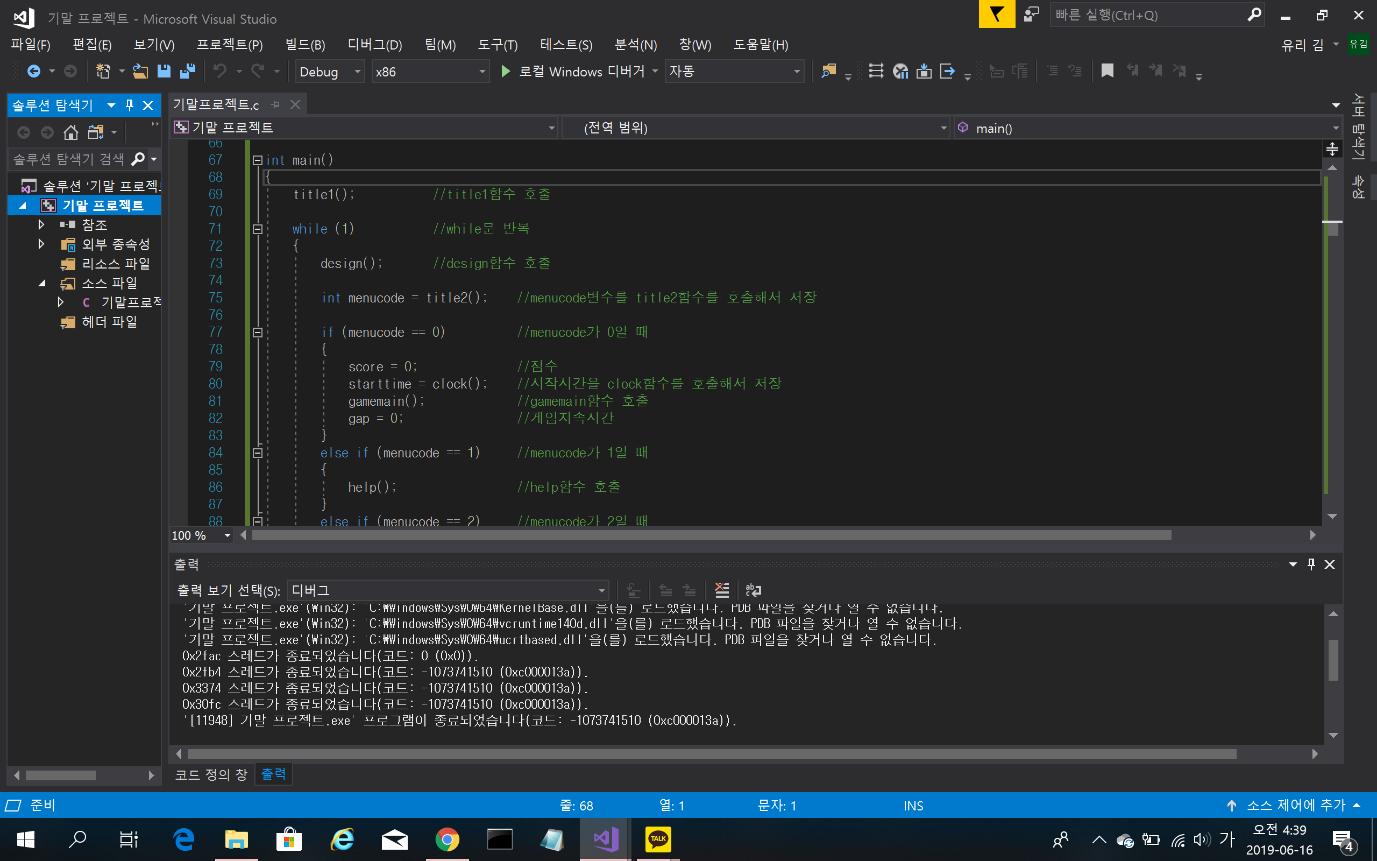
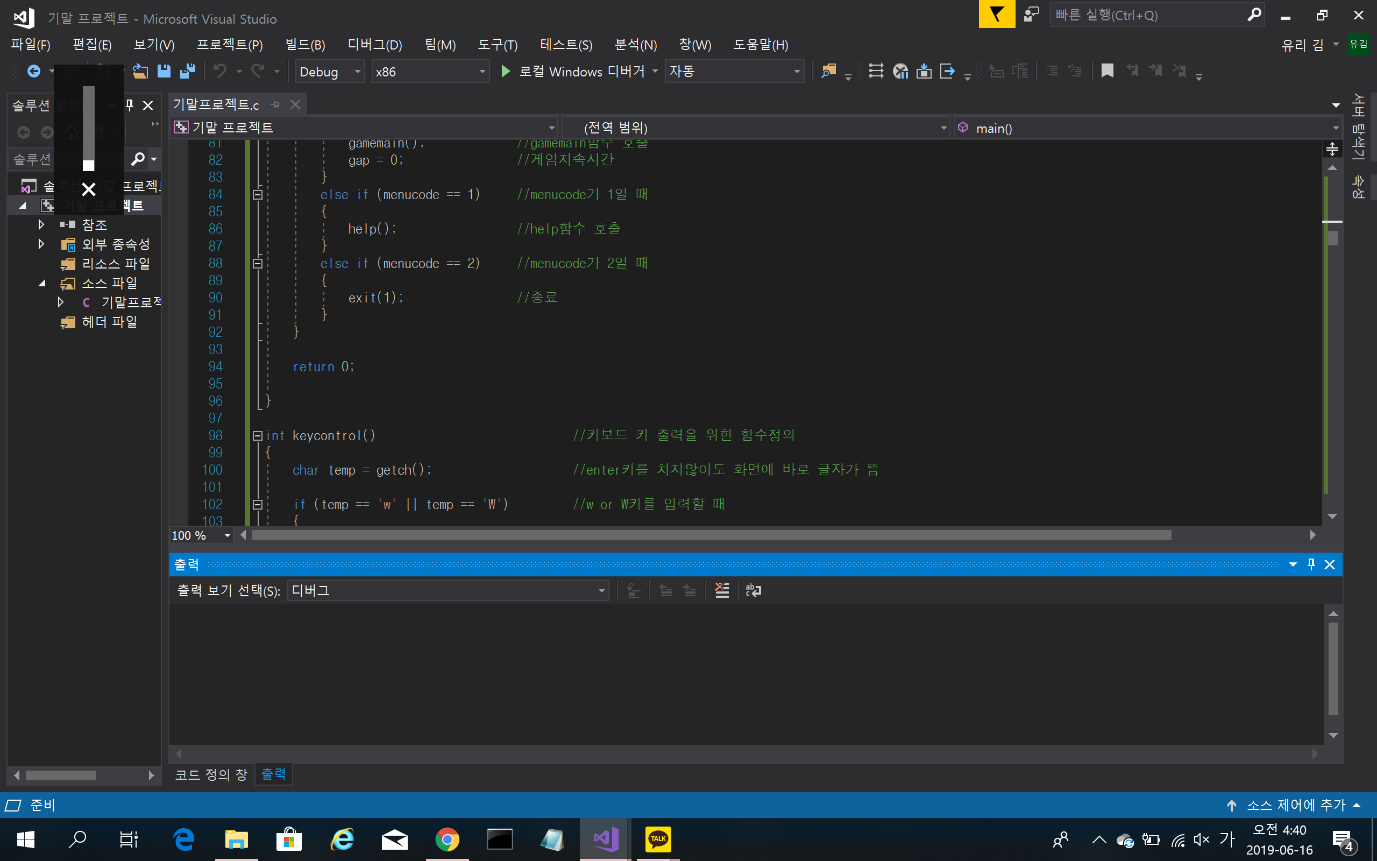
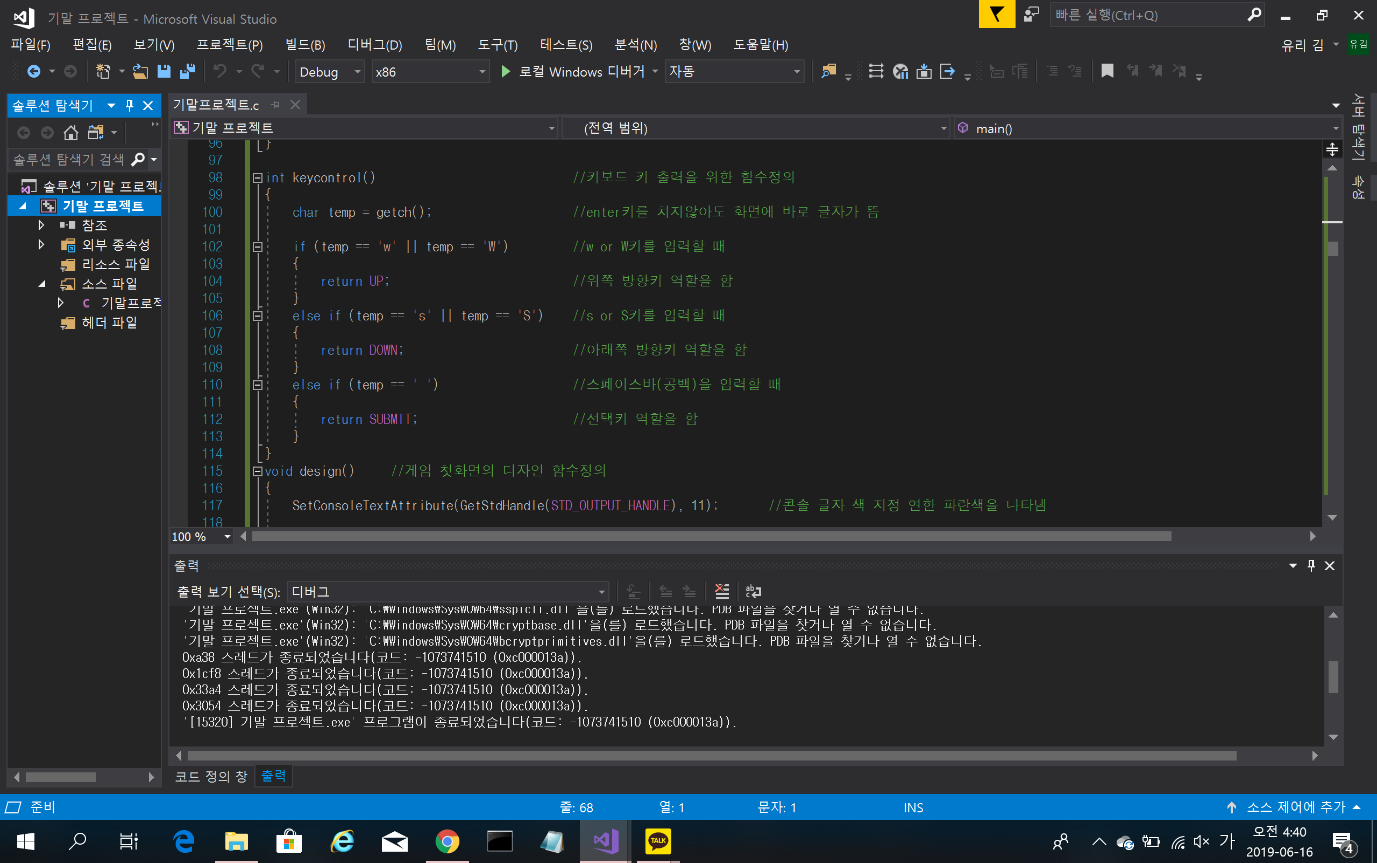
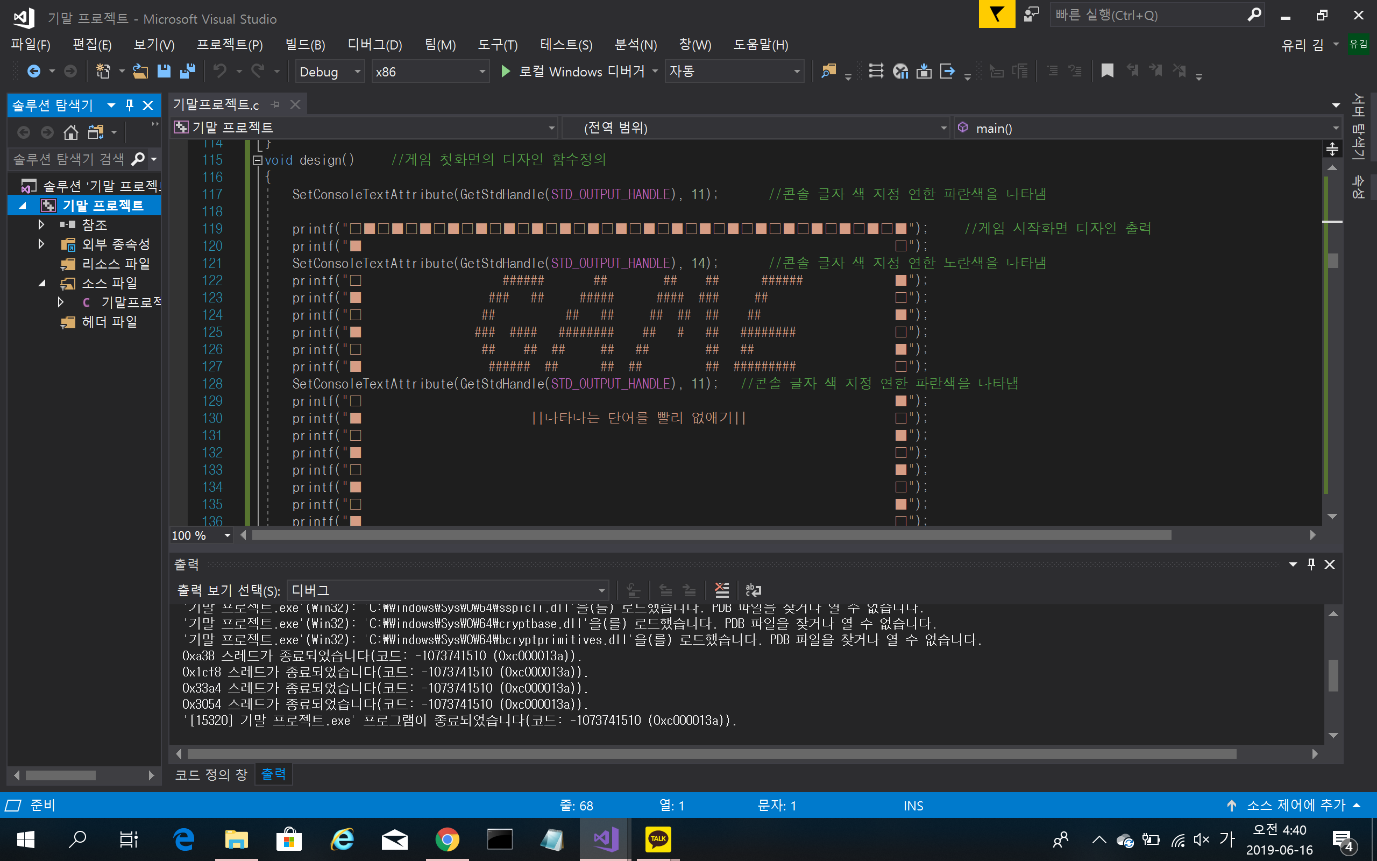
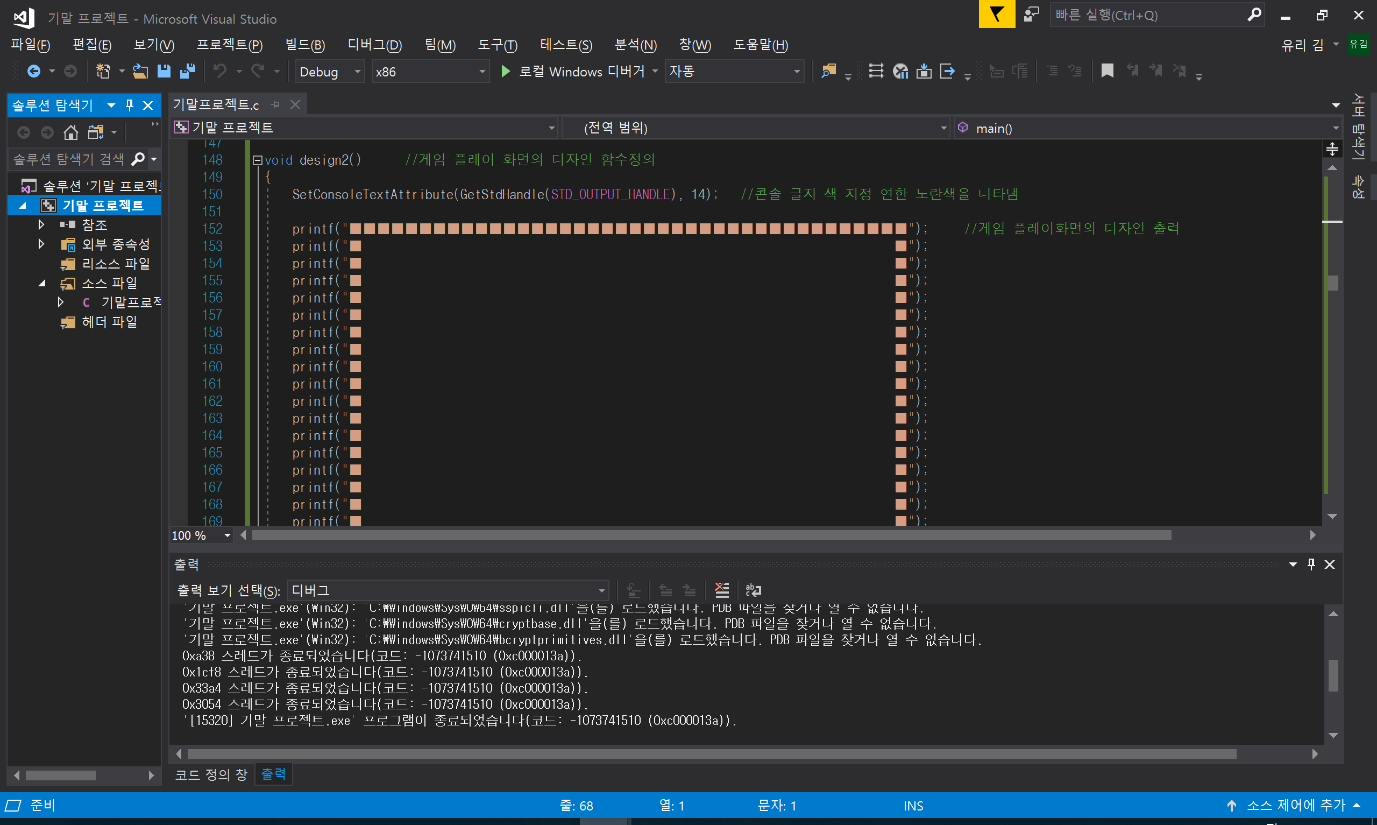
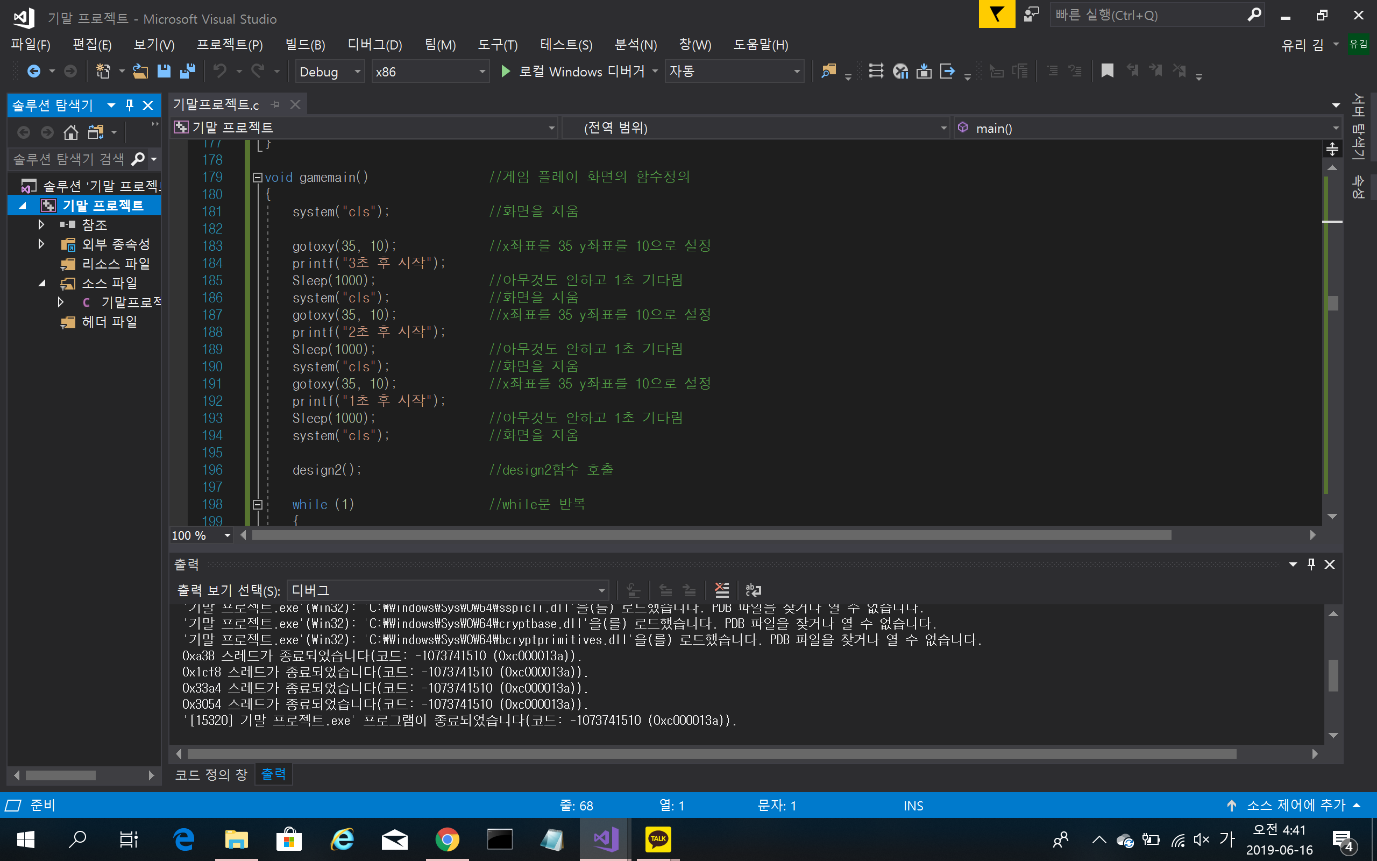
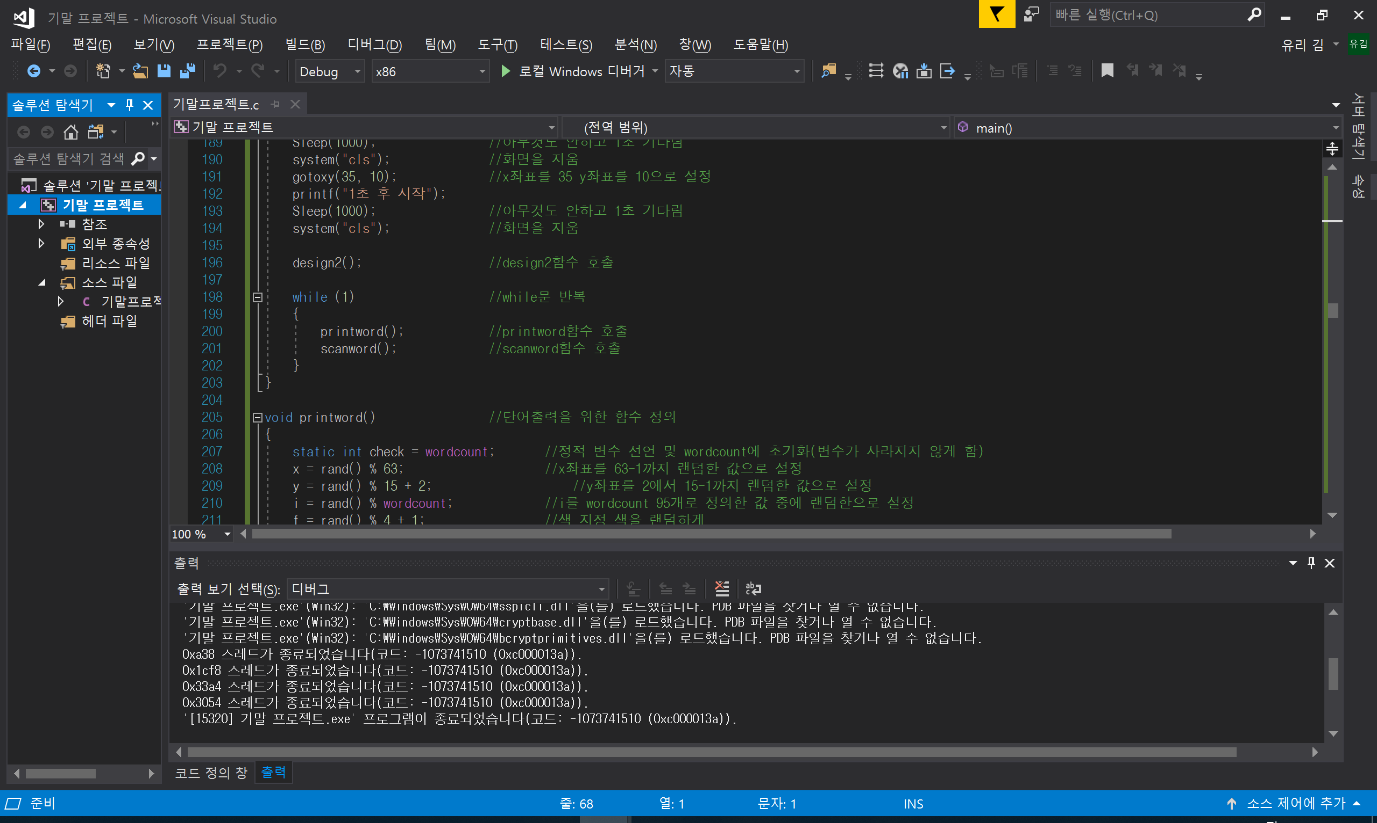
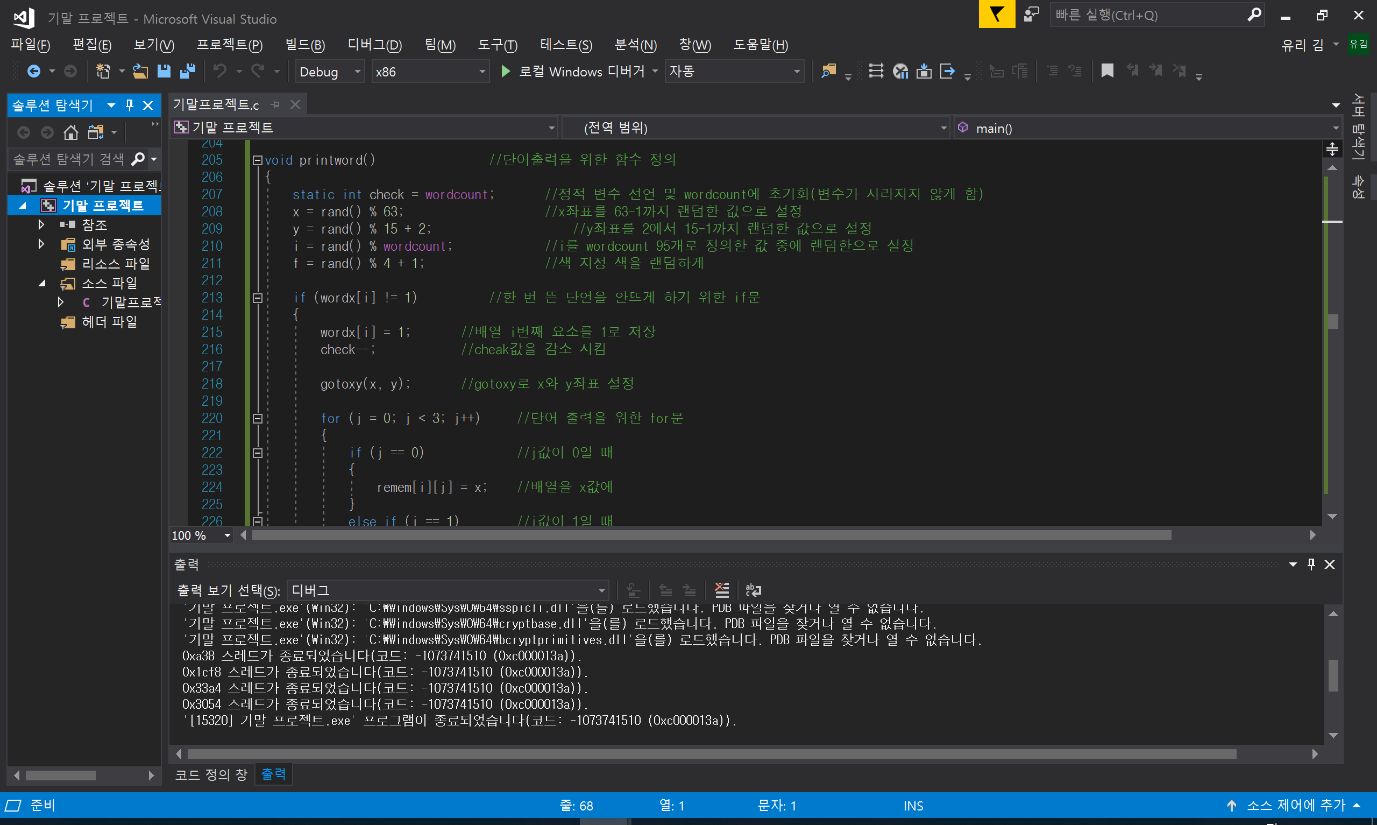
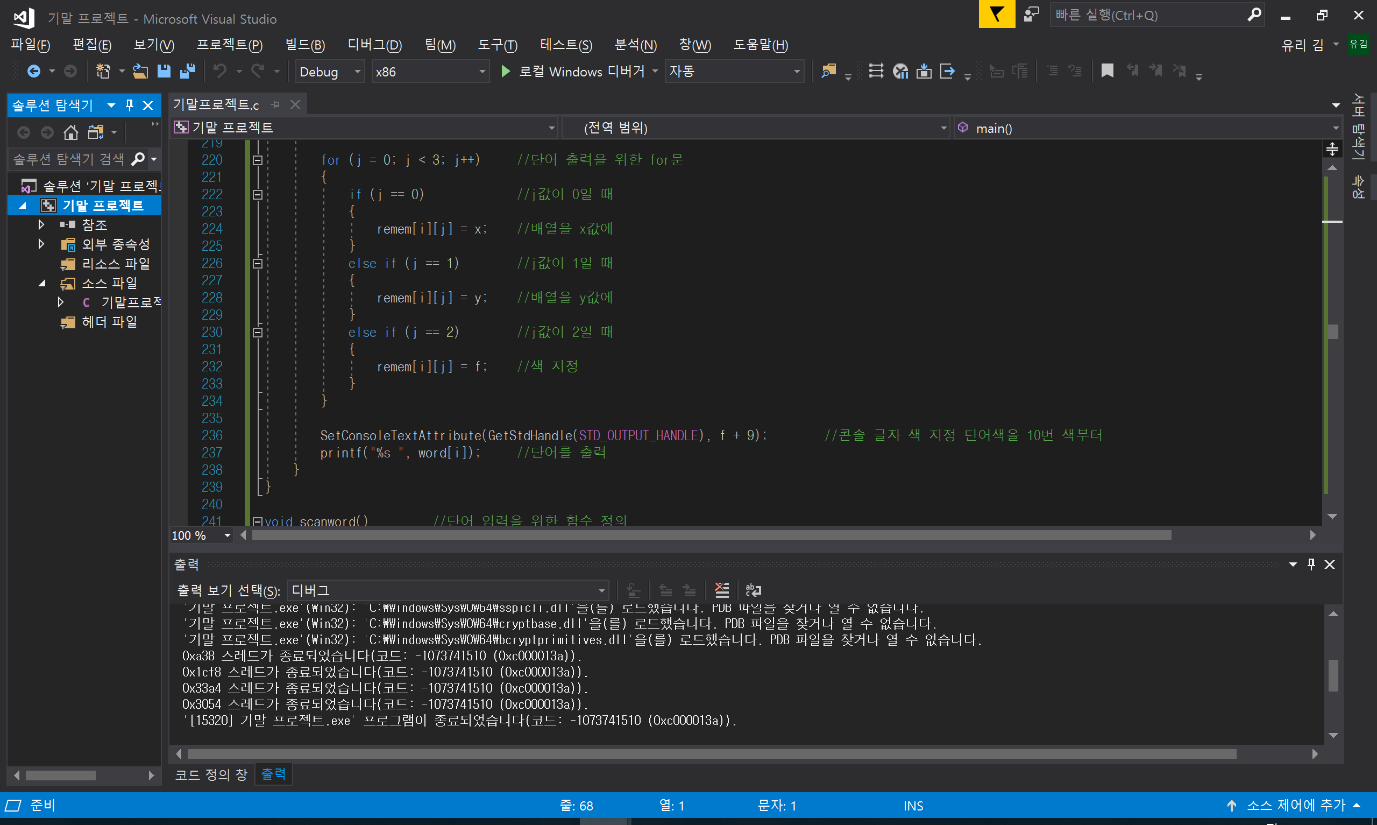
2.관련 연구  
  
-gotoxy함수: 사실상 라이브러리 함수는 아니며, visual c/c++내에서는gotoxy함수의 정의를 찾아볼 수 없다. 그러므로 사용자가 즉 사용자 지정 함수로 gotoxy함수로 선언하여 사용한다. 콘솔 화면을 특정 위치로 커서를 옮긴 후 출력한다.ex) gotoxy(1,2); 이런 식으로 사용. 이 함수를 사용하기 위해서는 Windows.h 헤더 파일을 사용해야 한다.

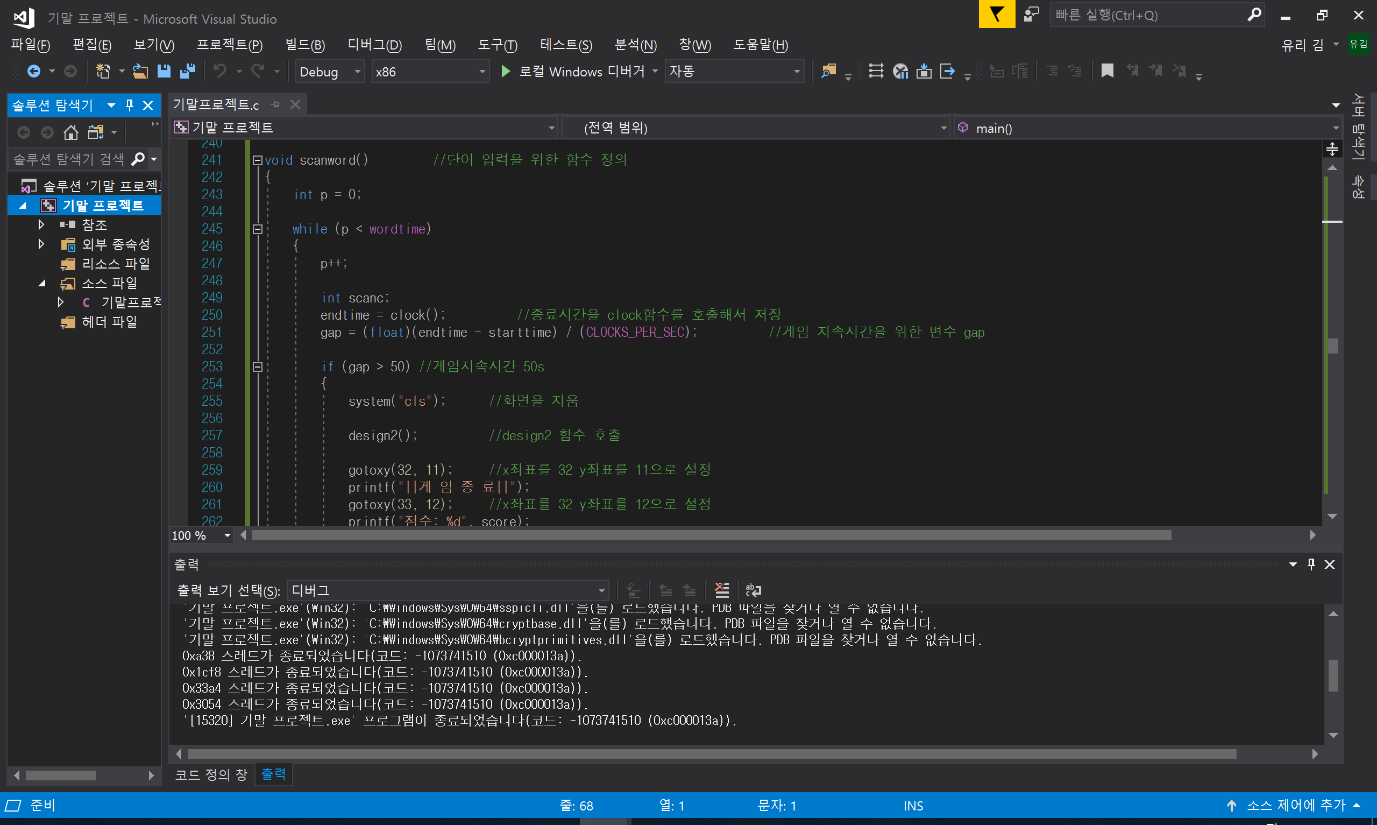
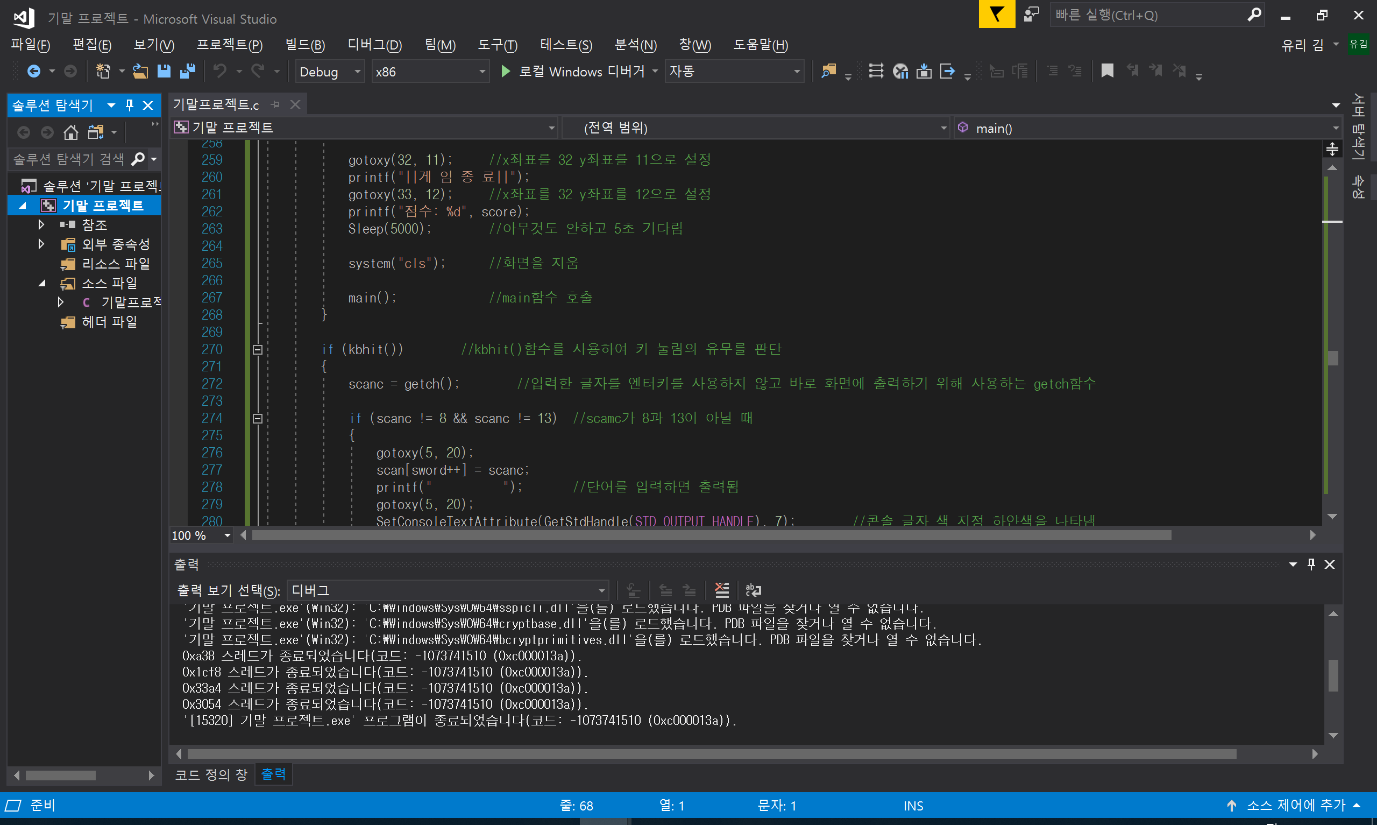
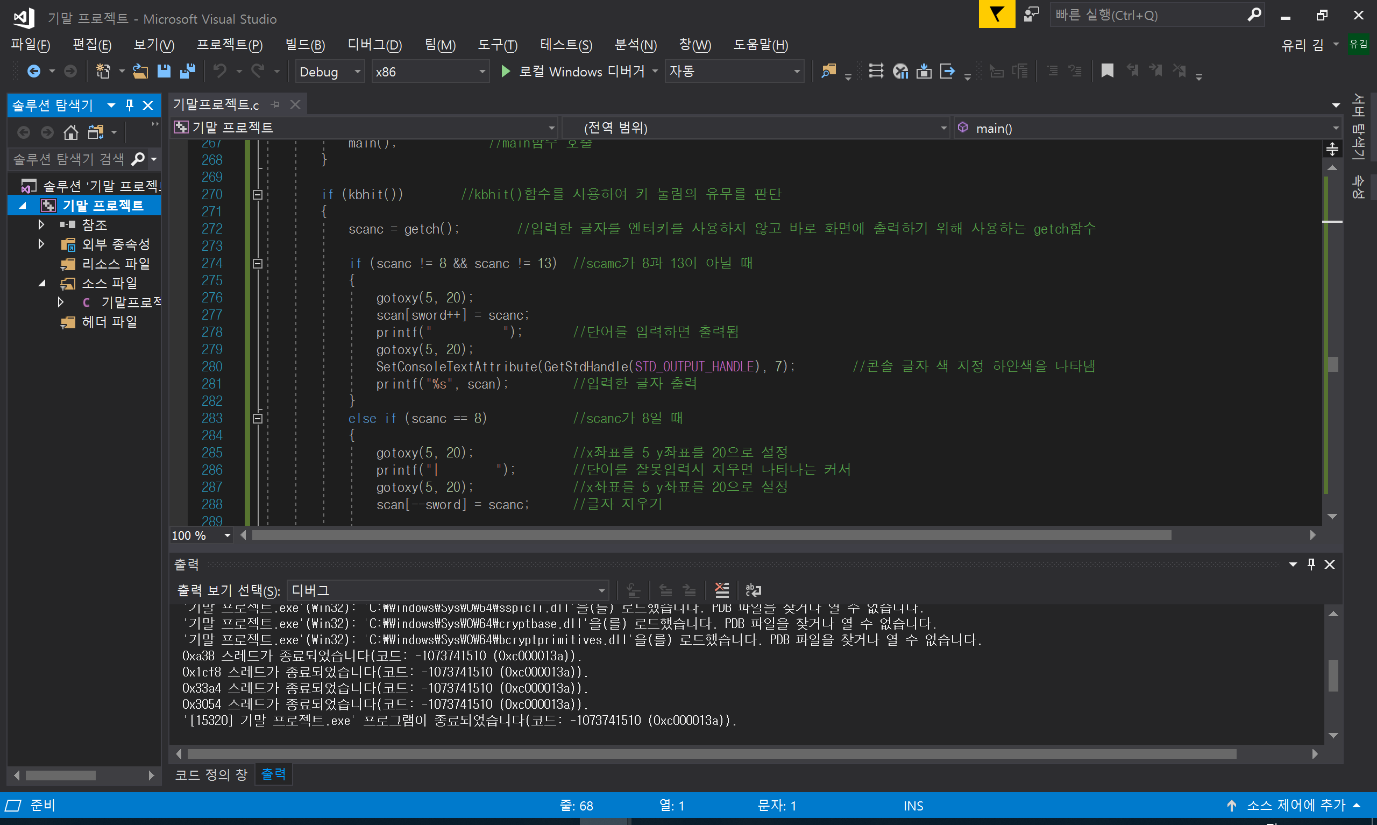
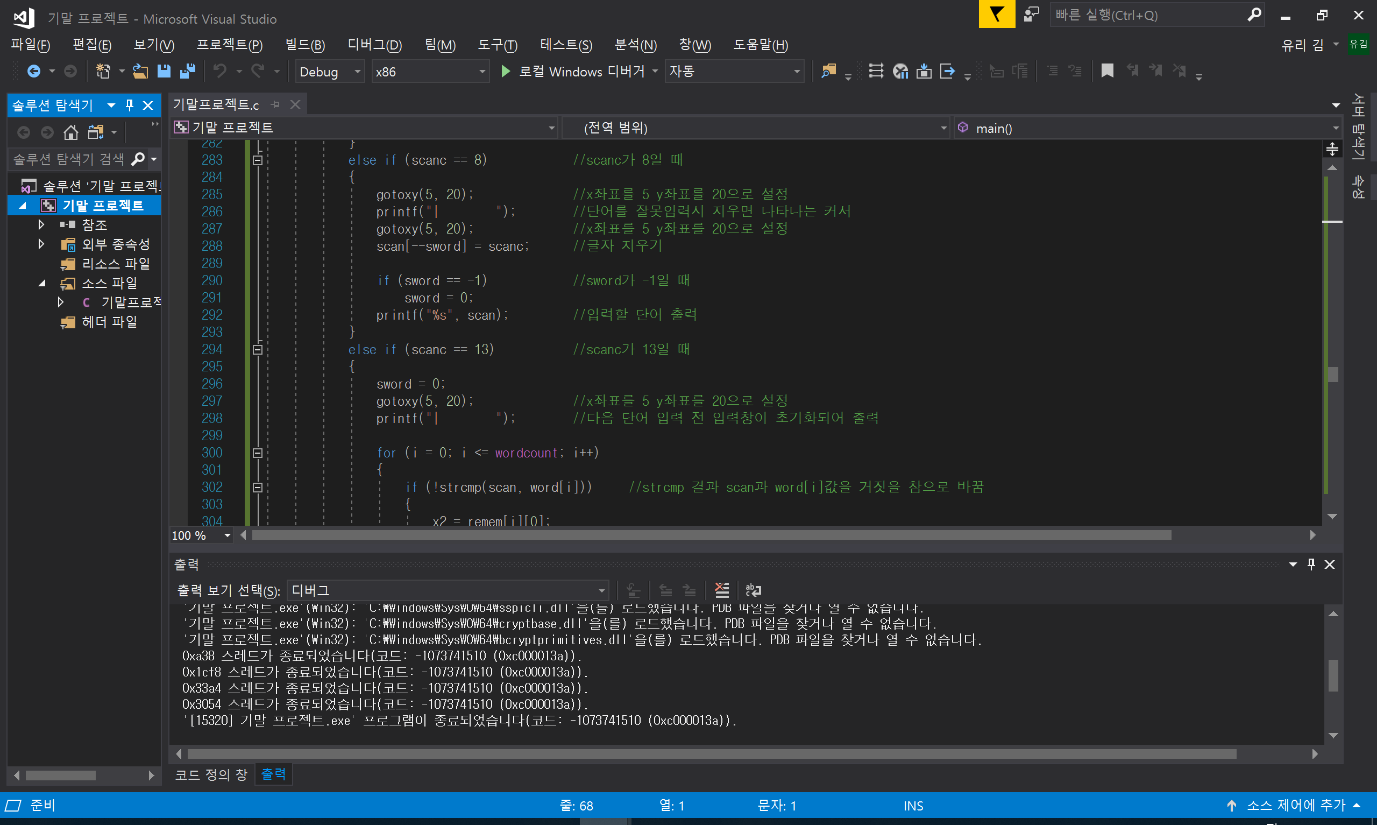
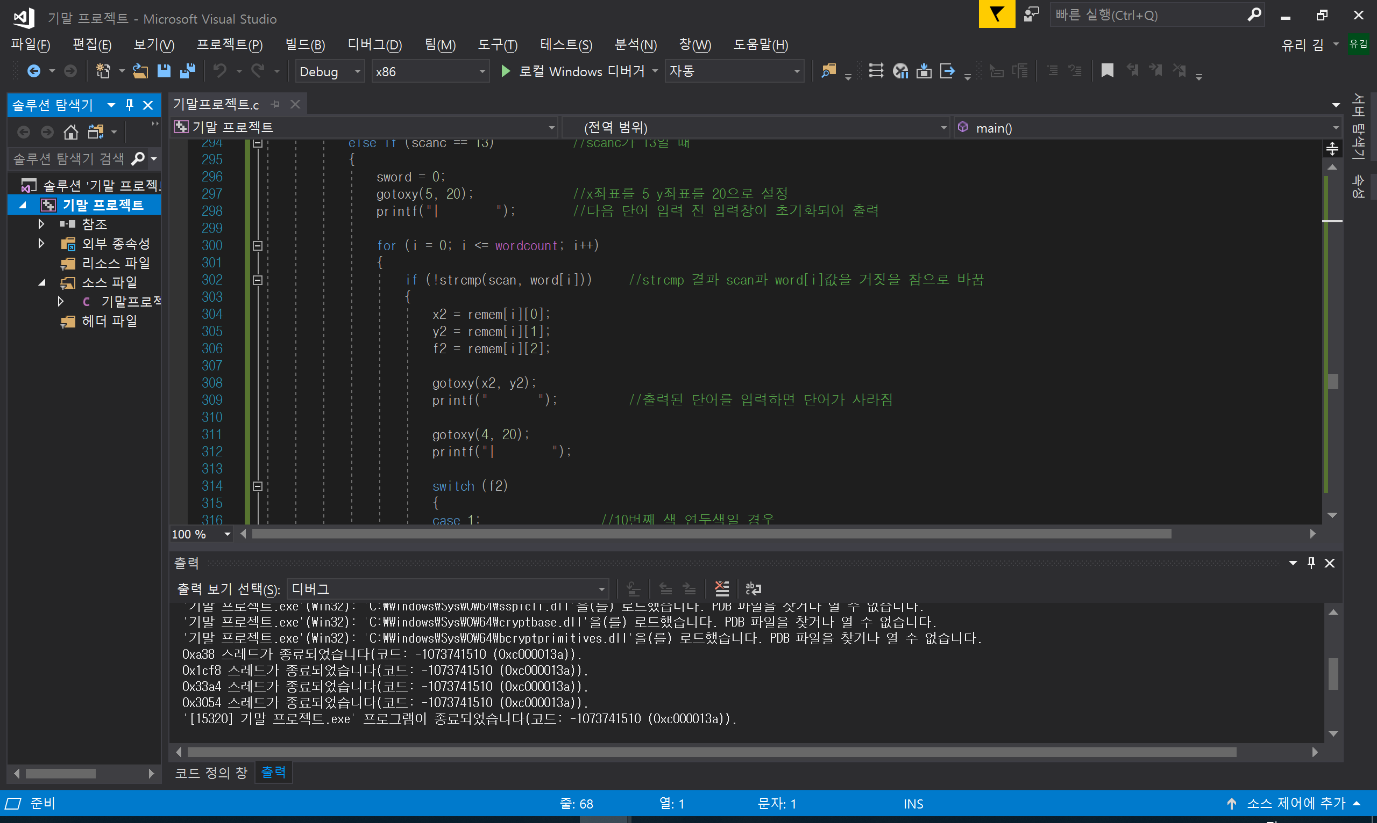
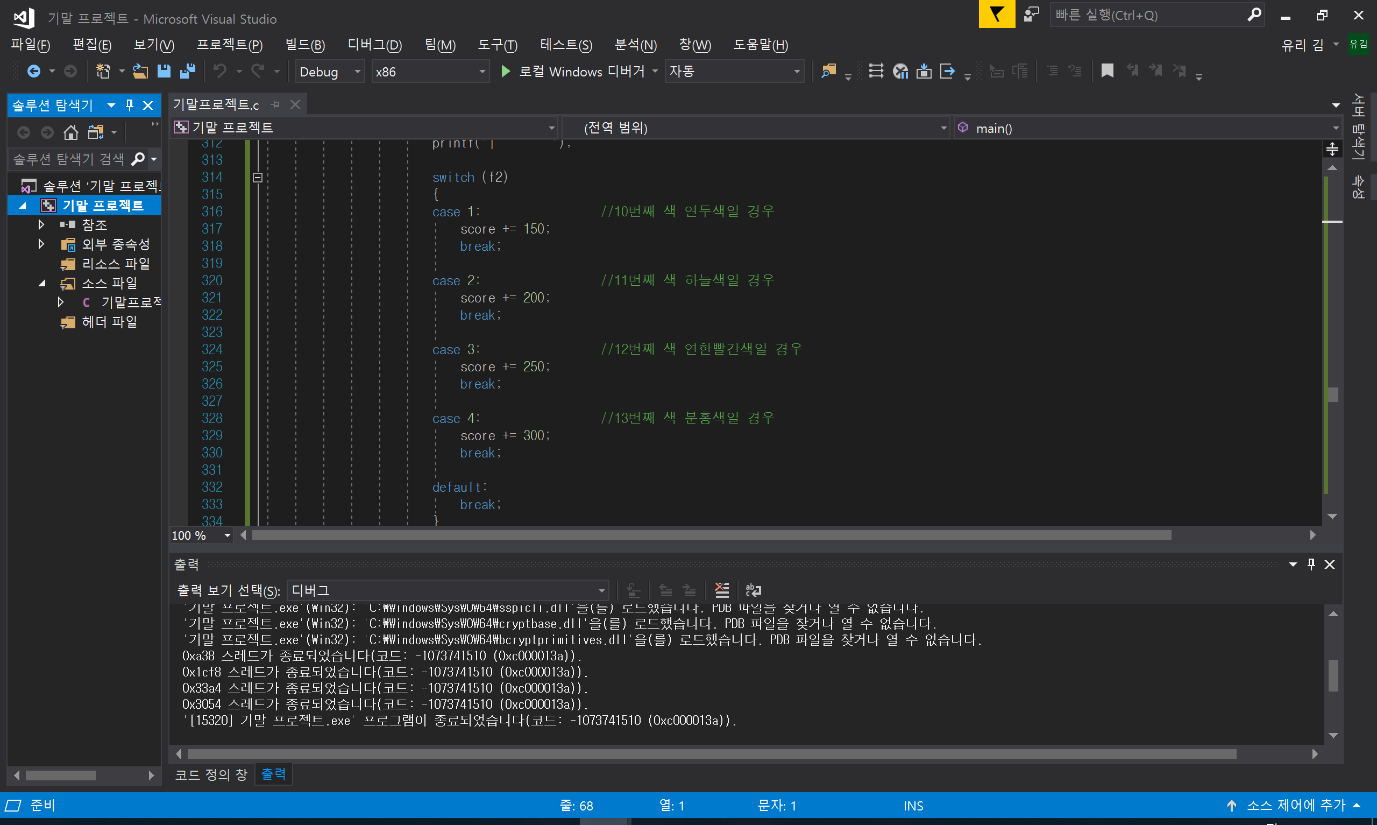
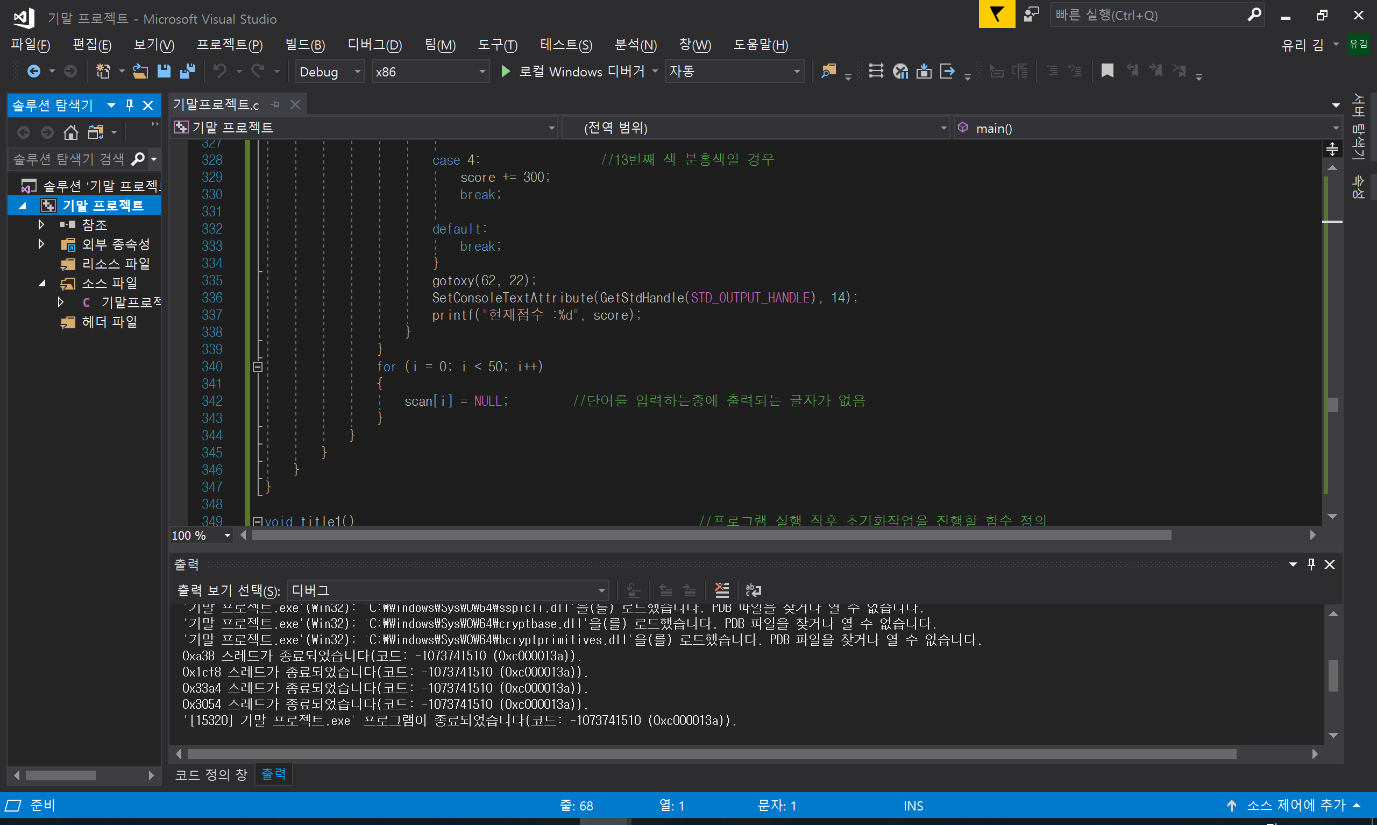
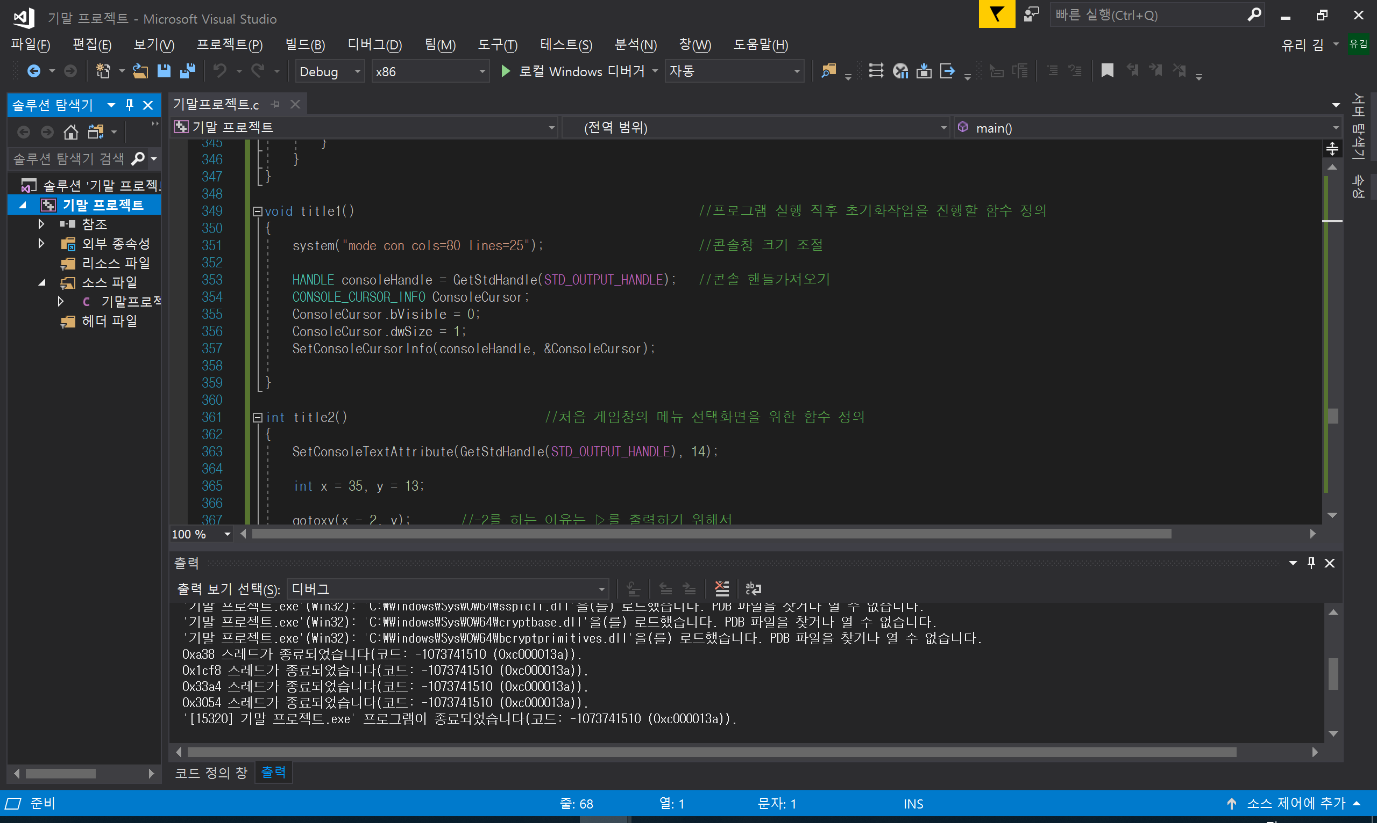
- system("mode con cols=80 lines=25") : 콘솔창의 크기를 조절하기 위한 명령이다. 이 명령을 사용하기 위해서는Windows.h 헤더 파일을 사용해야 된다.  
  
- SetConsoleTextAttribute() : 콘솔창의 글자 색 변경을 위해 사용을 하였으며, 이 함수를 사용하기 위해서는 Windows.h 헤더 파일을 사용해야 한다.  
  
getch() : 이 함수를 사용하면 enter키를 사용하지 않아도 치지 않아도 화면에 글자가 바로 뜬다. 입력 문자 수정이 불가능 하고 이 함수를 사용하기 위해서는 conio.h 헤더 파일을 사용해야 한다.  
  
-system(“cls”) : 출력된 화면을 지우는 명령이다. 이 명령을 사용하려면 Windows.h 헤더 파일을 사용해야 된다.  
  
-Sleep() : 화면을 일정시간동안 멈추는 명령이며, 1초를 멈추려면 Sleep(1000);을 입력해주면 된다. Windows.h 헤더 파일을 사용해야 된다. 이 명령을 사용하려면 Windows.h 헤더 파일을 사용해야 된다.  
  
-exit(1) : 이 함수를 사용하면 창이 즉시 종료된다.  
  
  
  
3.프로그램의 설계  
  
  
이 게임은 원래 산성비 게임을 만들려고 계획하였으나 게임을 구현하던 도중 화면에 출력되는 단어의 색깔을 다르게 하여 단어를 입력하여 없애면 없앤 단어의 색 별로 점수를 얻는 게임을 만들면 어떨까? 라는 생각으로 만들게 되었다. 이 게임을 만들기 위해 맨 처음에 어떤 게임을 만들지 구상을 하였고 구상을 한 결과가 “단어 없애기 게임”이고 게임의 UI를 개성 있게 꾸미고 싶어서 콘솔 창을 조절하는 명령과 콘솔 창 글씨의 색 변경을 위해 필요한 함수를 찾아 공부하였다. 그런 다음 글자를 입력하면 없어지는 원리와 점수를 득점하는 방법을 연구하였고 gotoxy함수를 이용하여 내가 원하는 위치에 문자, 숫자가 출력될 수 있게 연구하였다. 그리고 키보드 값을 입력 받아서 방향키 역할을 할 수 있게 자료를 찾고 소스를 짰다. 그리고 커서의 깜박거림을 없애기 위해 함수를 연구하고 실제로 구현해 보았다.

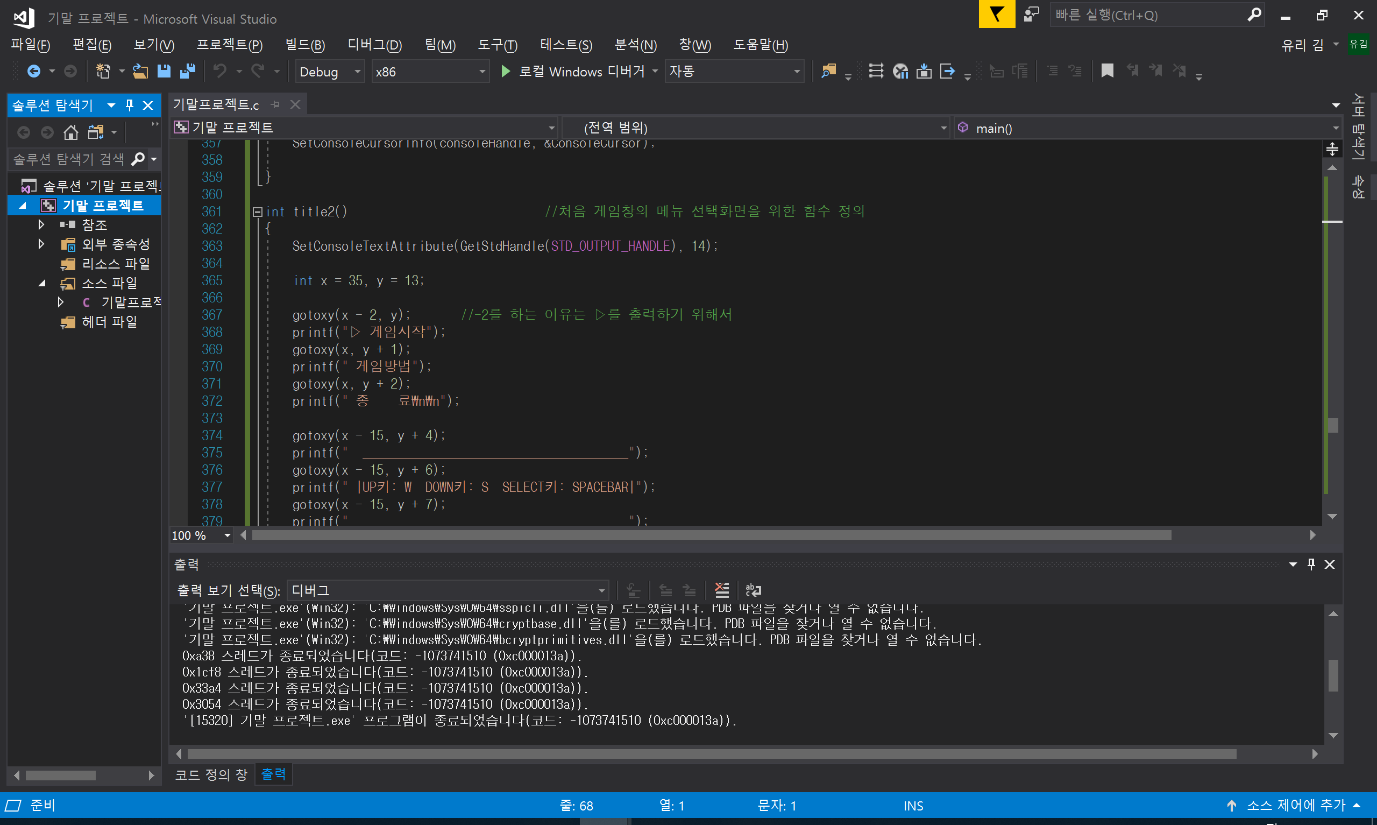
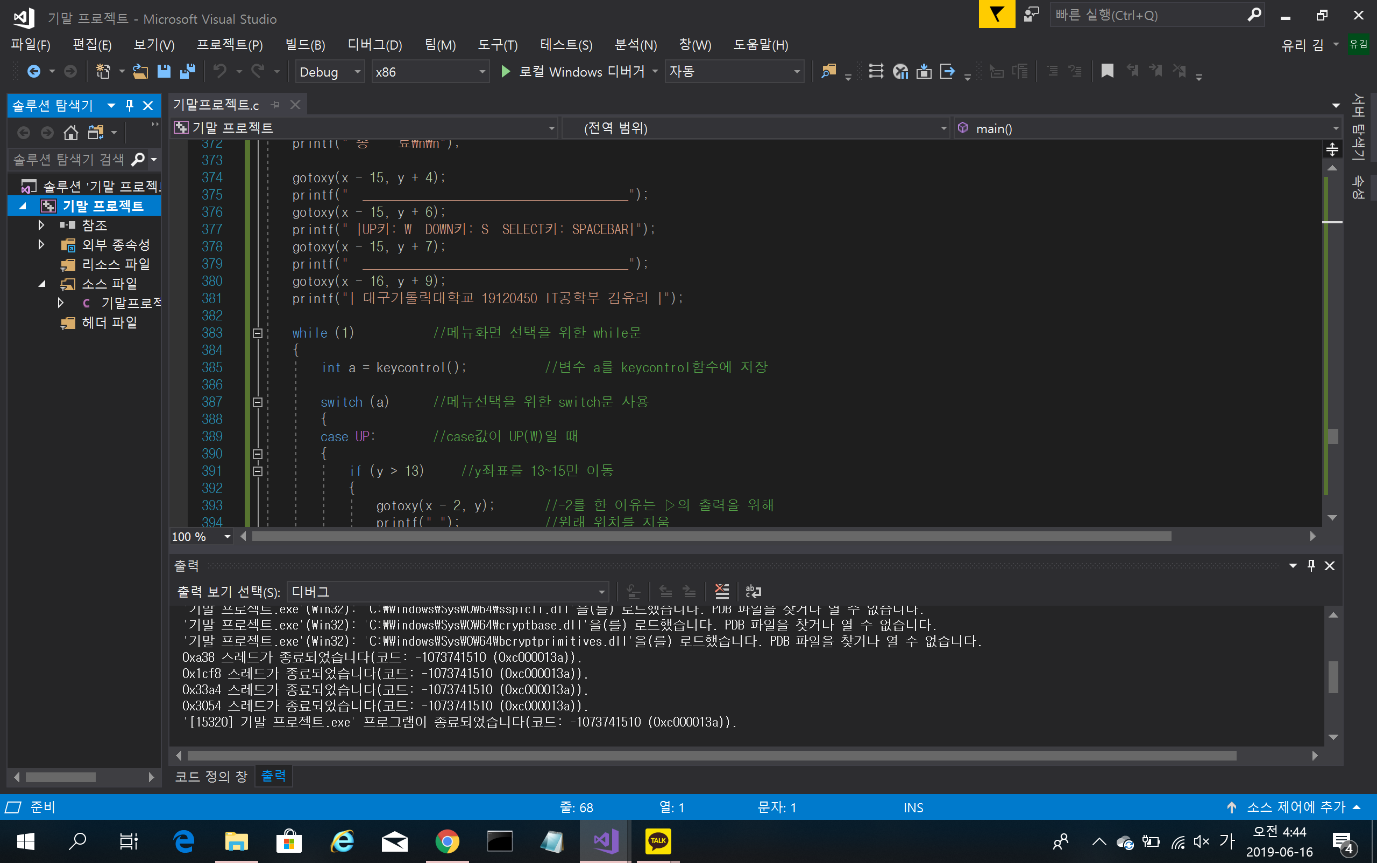
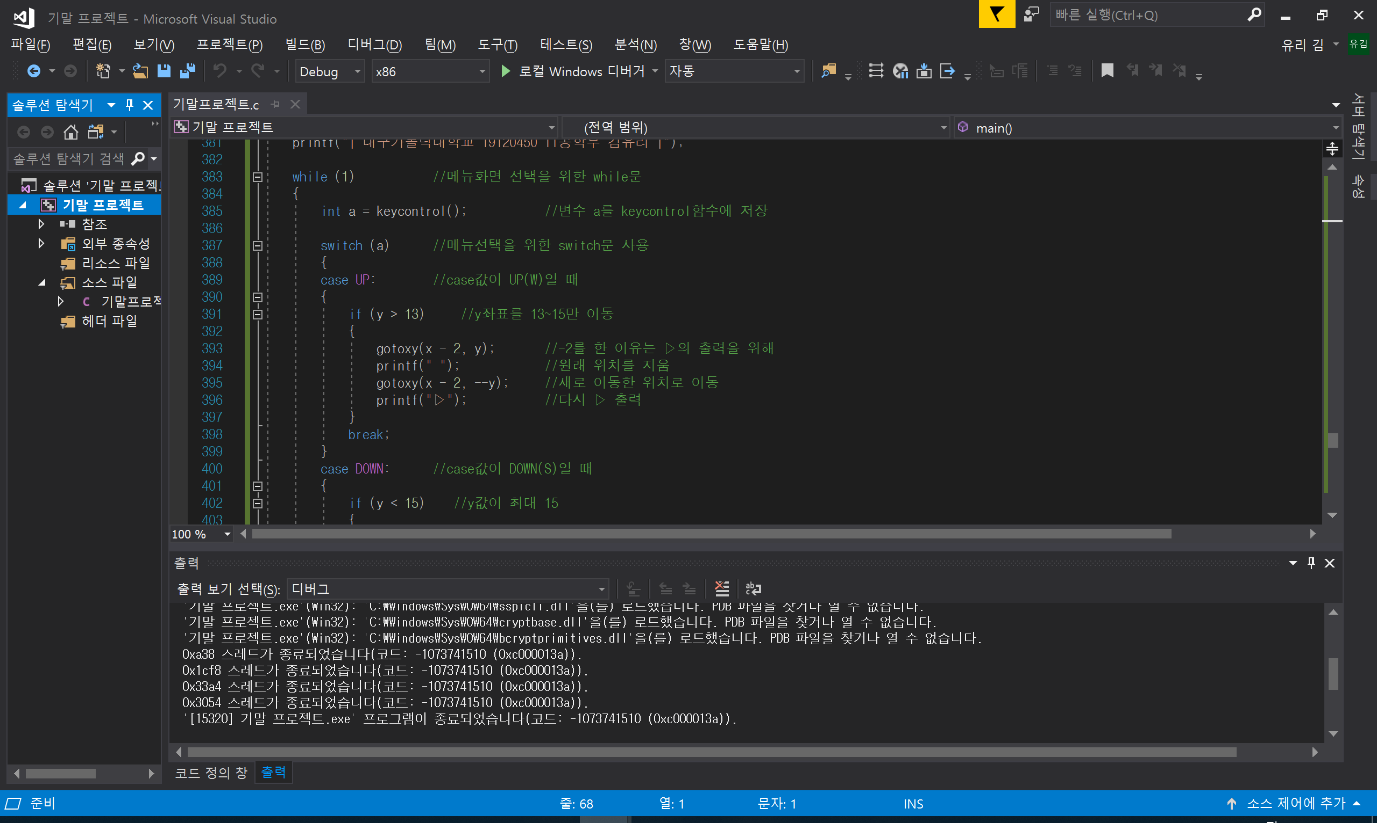
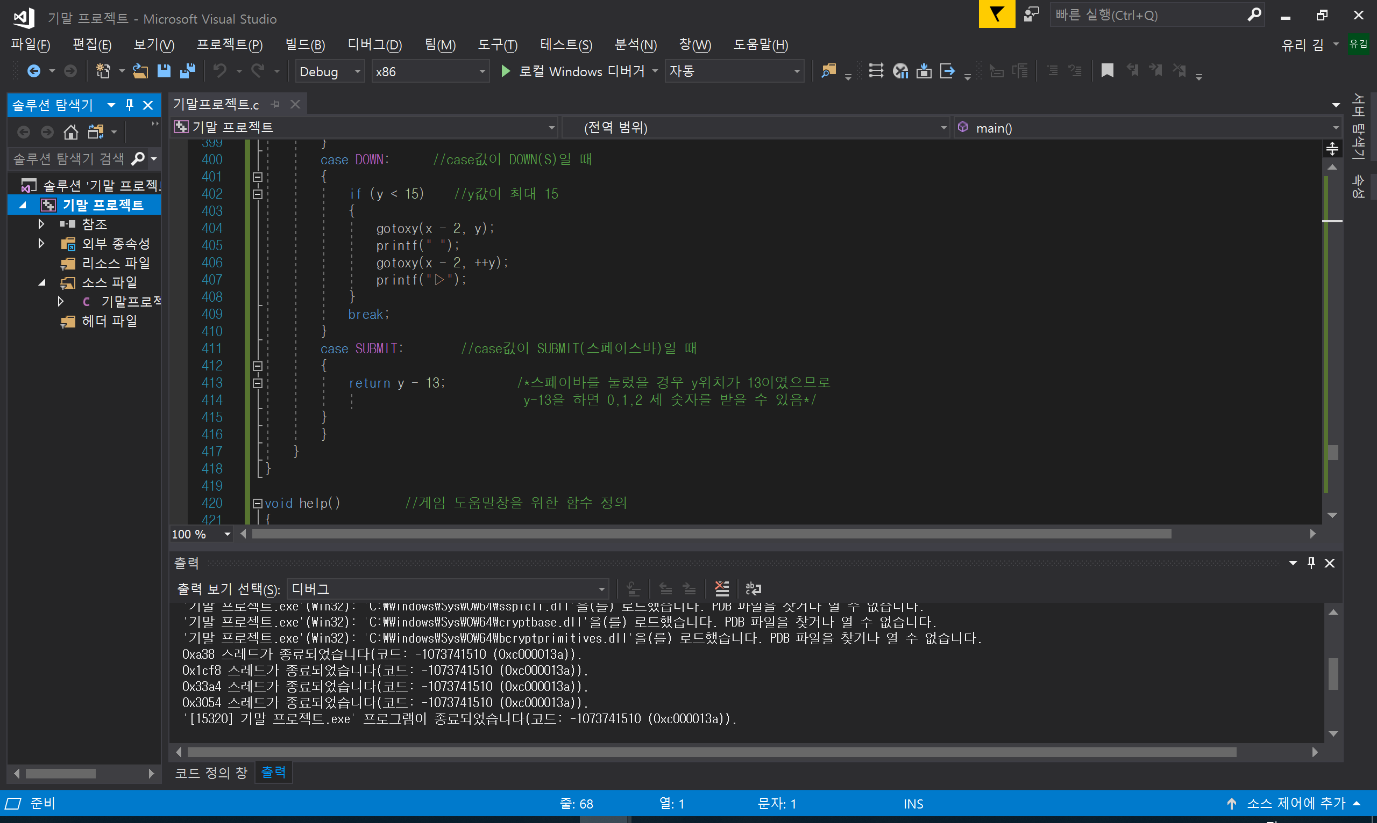
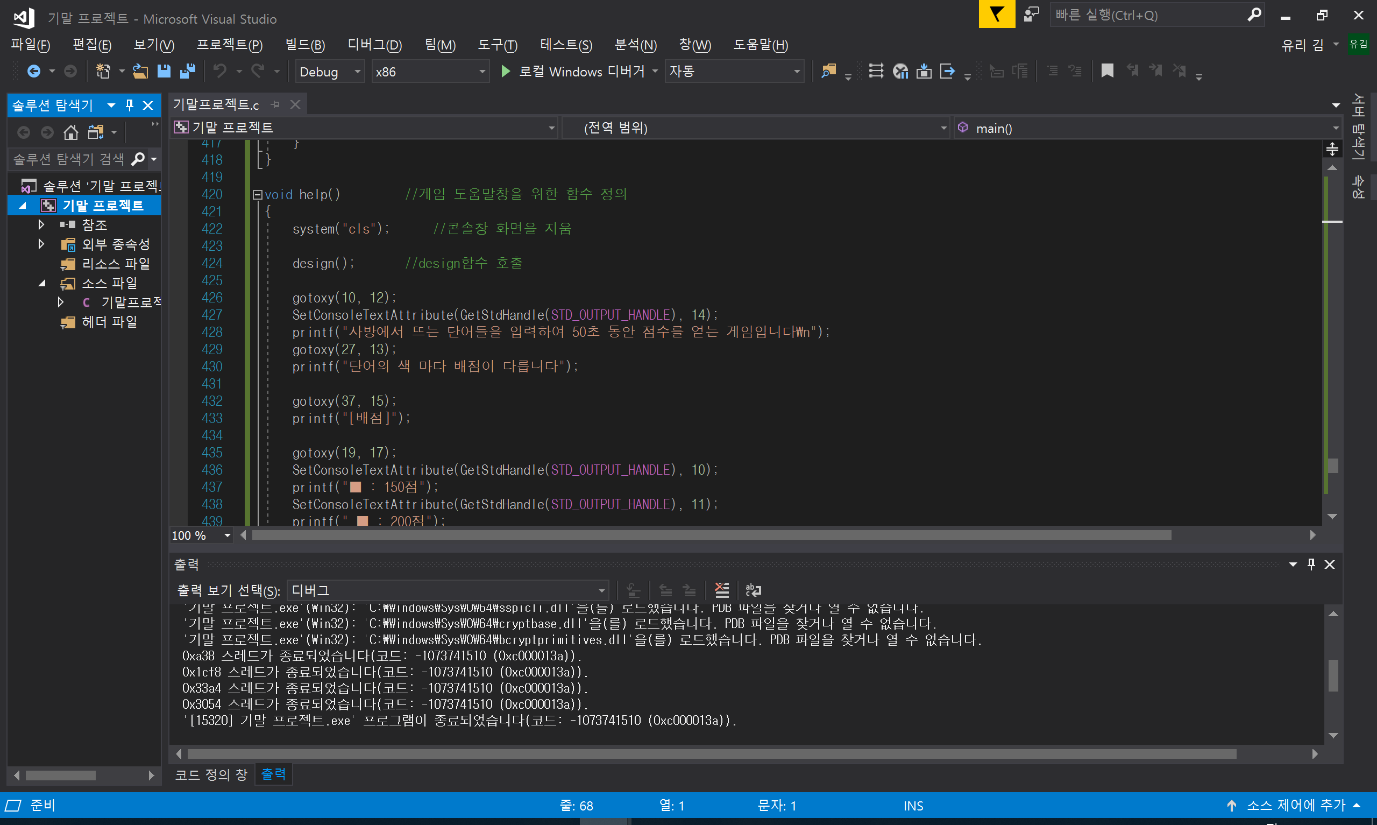
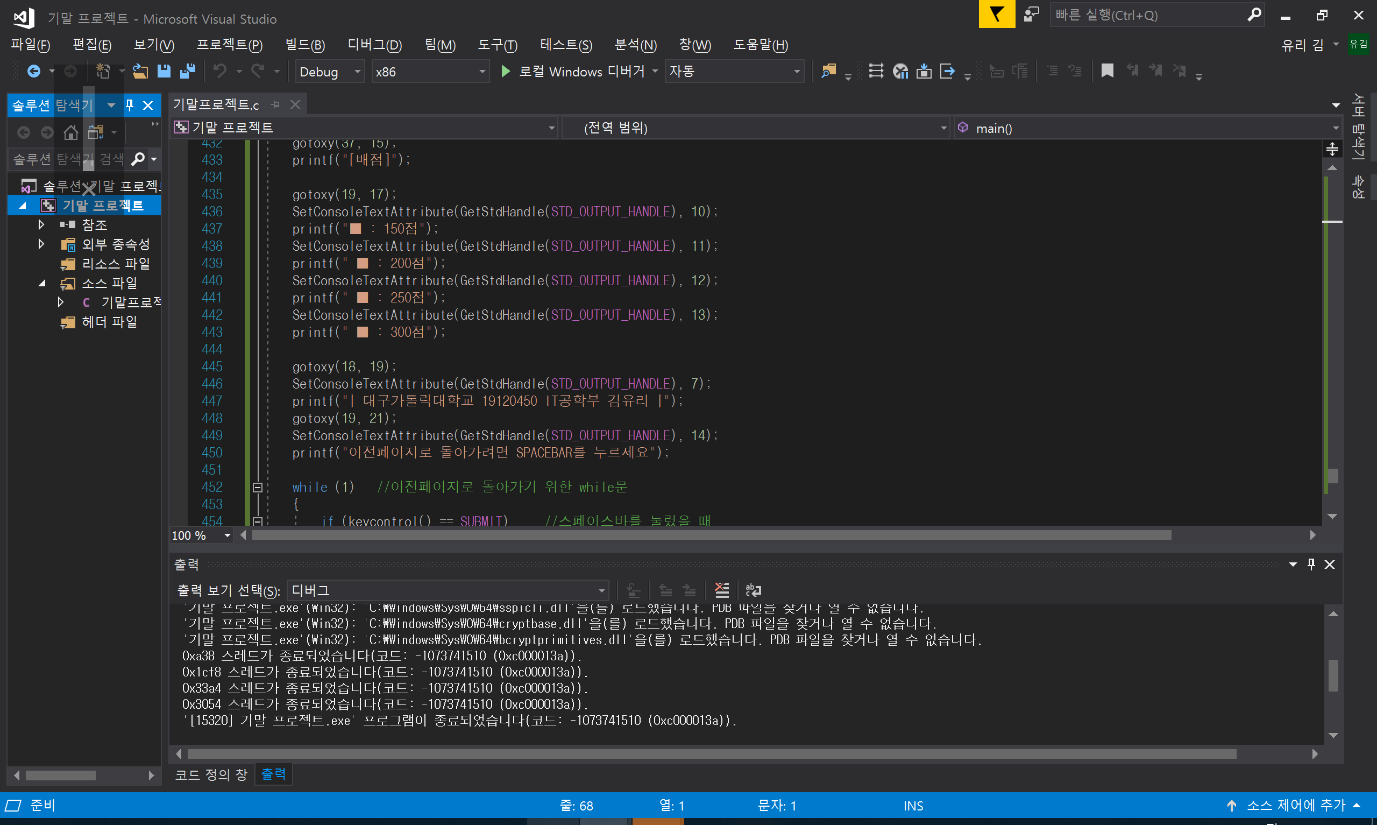
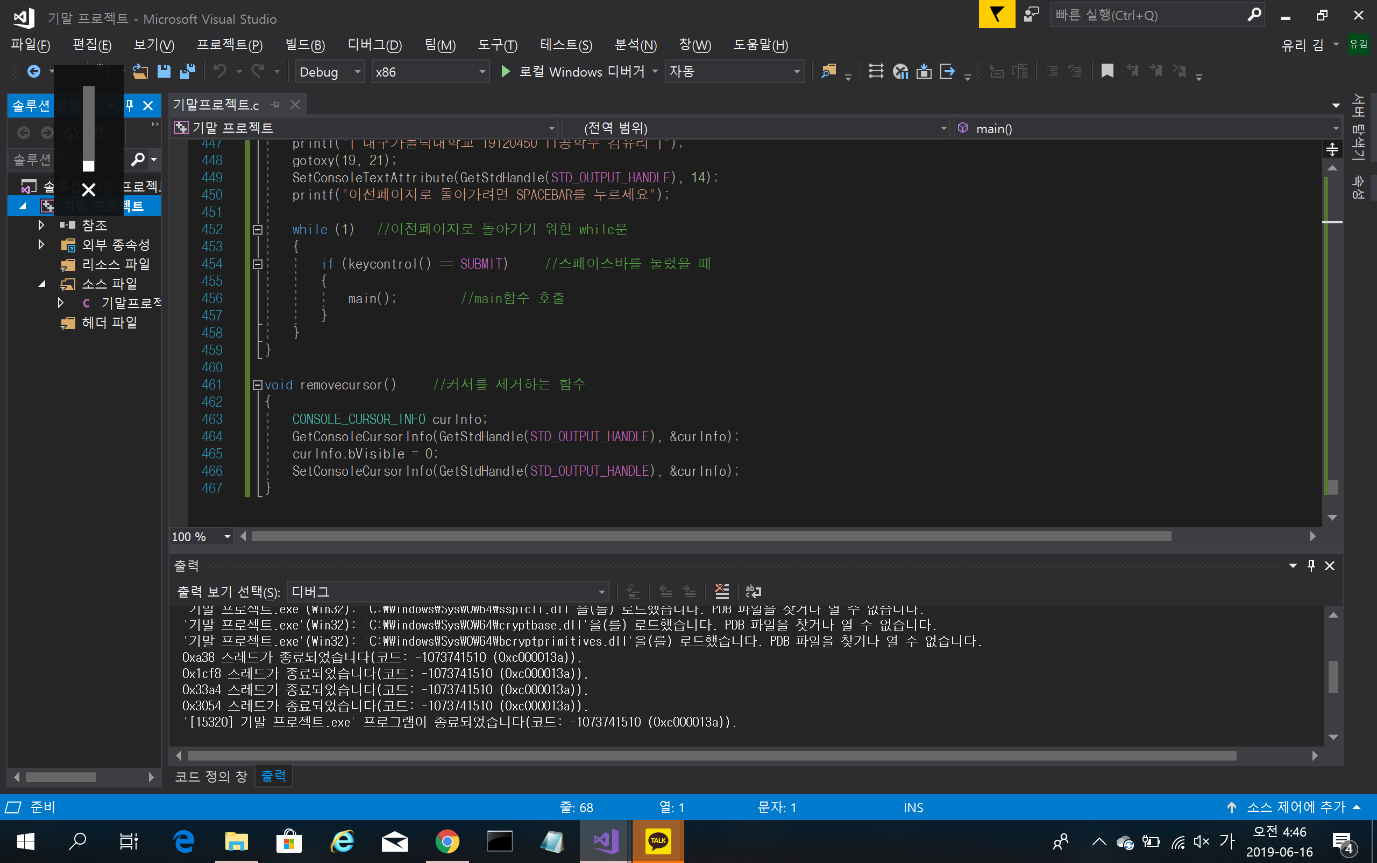
4.프로그램의 구현  
  
  
 [자료 4-1] 헤더 파일  
헤더 파일을 쓰기 위해서는 #include을 써야한다.  
-stdio.h는 표준 입출력 작업을 위한 헤더 파일이다.  
-stdlib.h는 여러 유틸리티 함수 작업을 위한 헤더파일이다. 이 프로그램에서는 rand함수를 위해 쓴 헤더 파일이다.   
-time.h는 시간 작업을 위한 헤더 파일이며, 이 프로그램에서는 time\_t : 시간 타입을 위한 이 변수를 위해 썼다.   
-Windows.h는 윈도우 응용 프로그램을 만들기 위한 헤더 파일이며 이 프로그램에서는gotoxy함수, system, SetConsoleTextAttribute(), Sleep 등을 사용하기 위해 썼다.  
-conio.h는 getche함수를 이용하기 위해서 쓴 헤더파일이다.  
-string.h는 strcmp 함수가 선언된 헤더 파일이다.  
  
스크린샷, 모니터, 검은색이(가) 표시된 사진

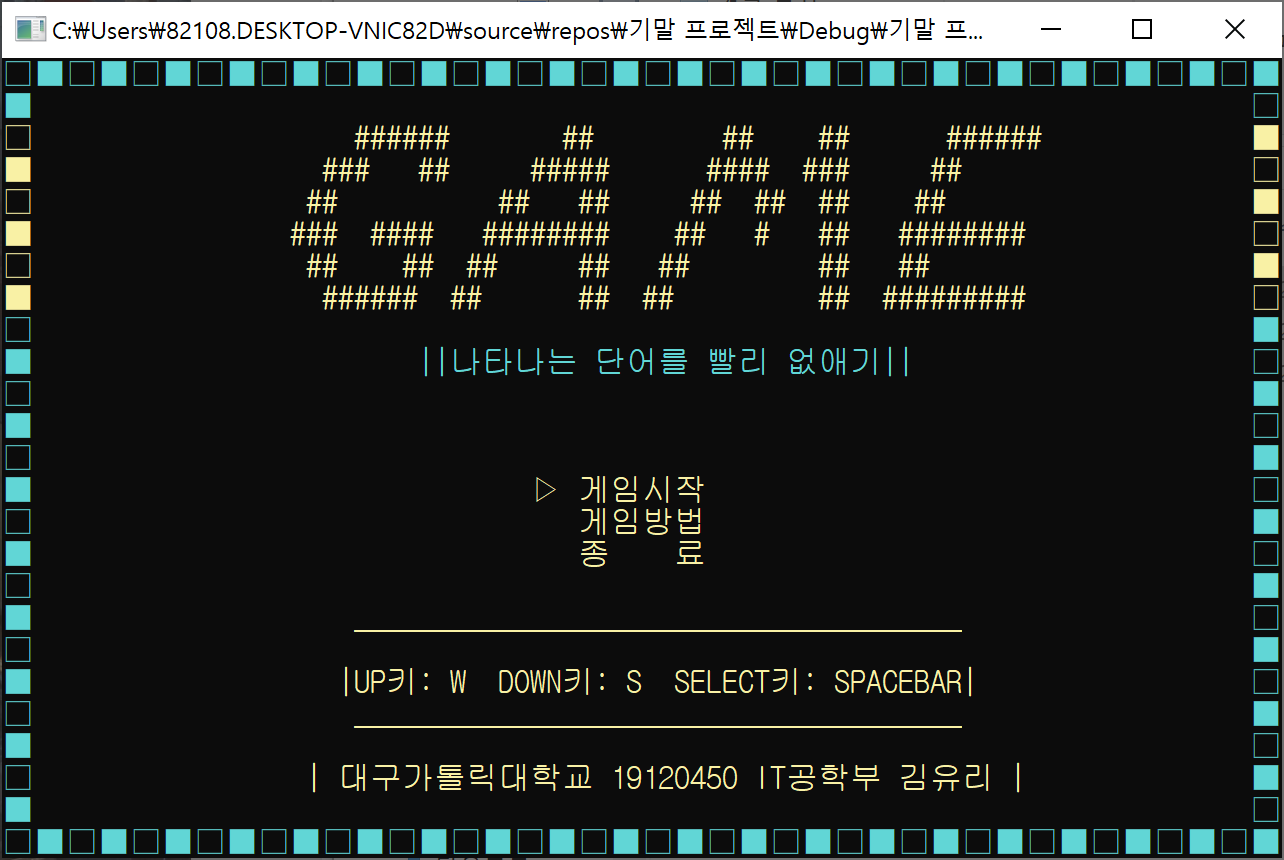
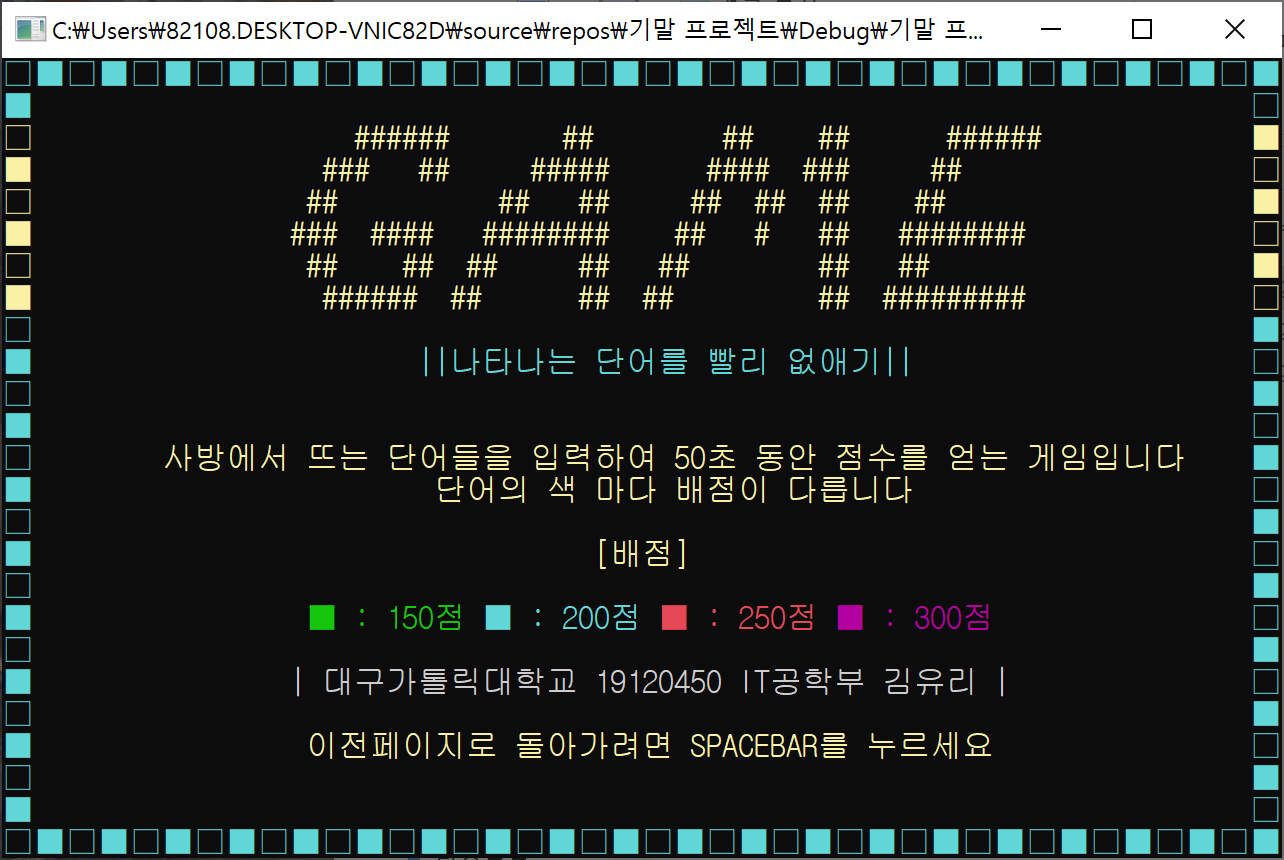
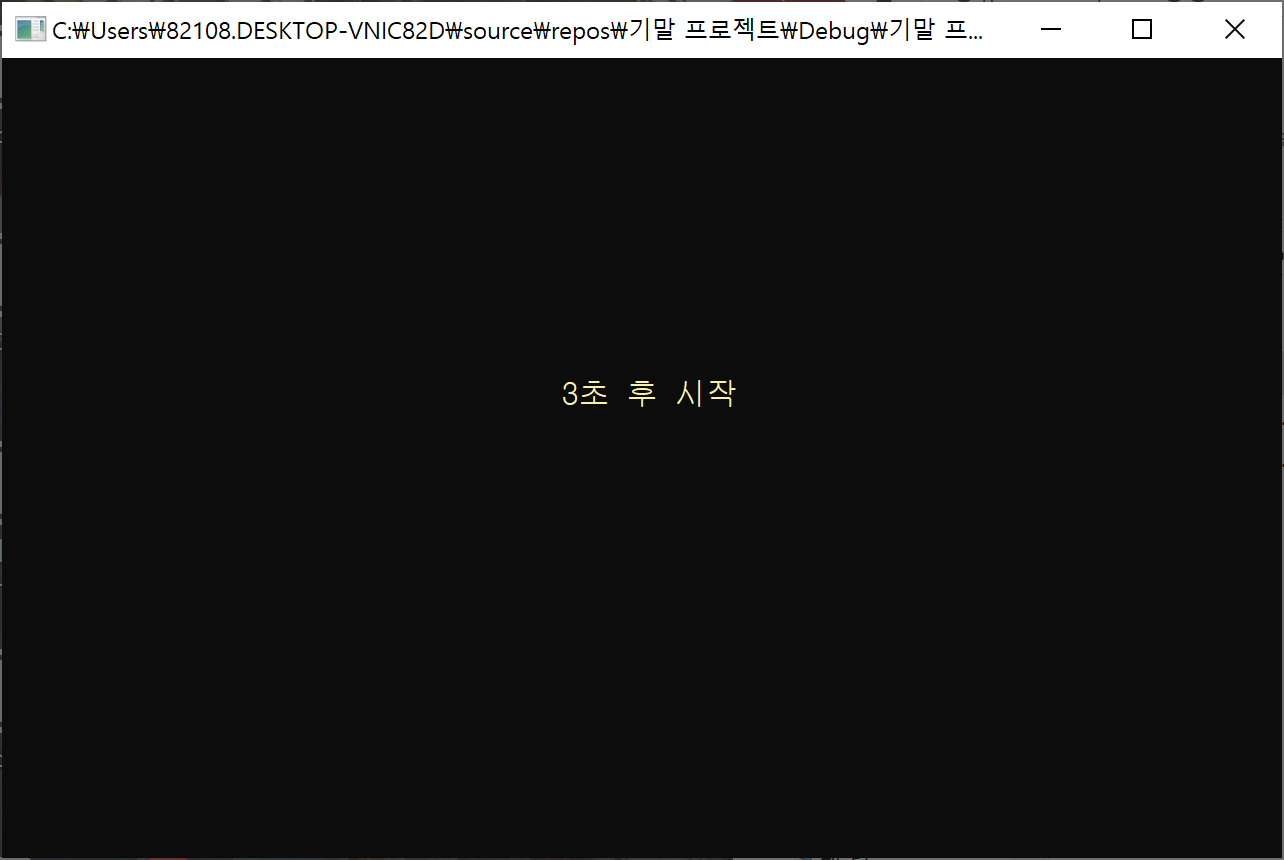
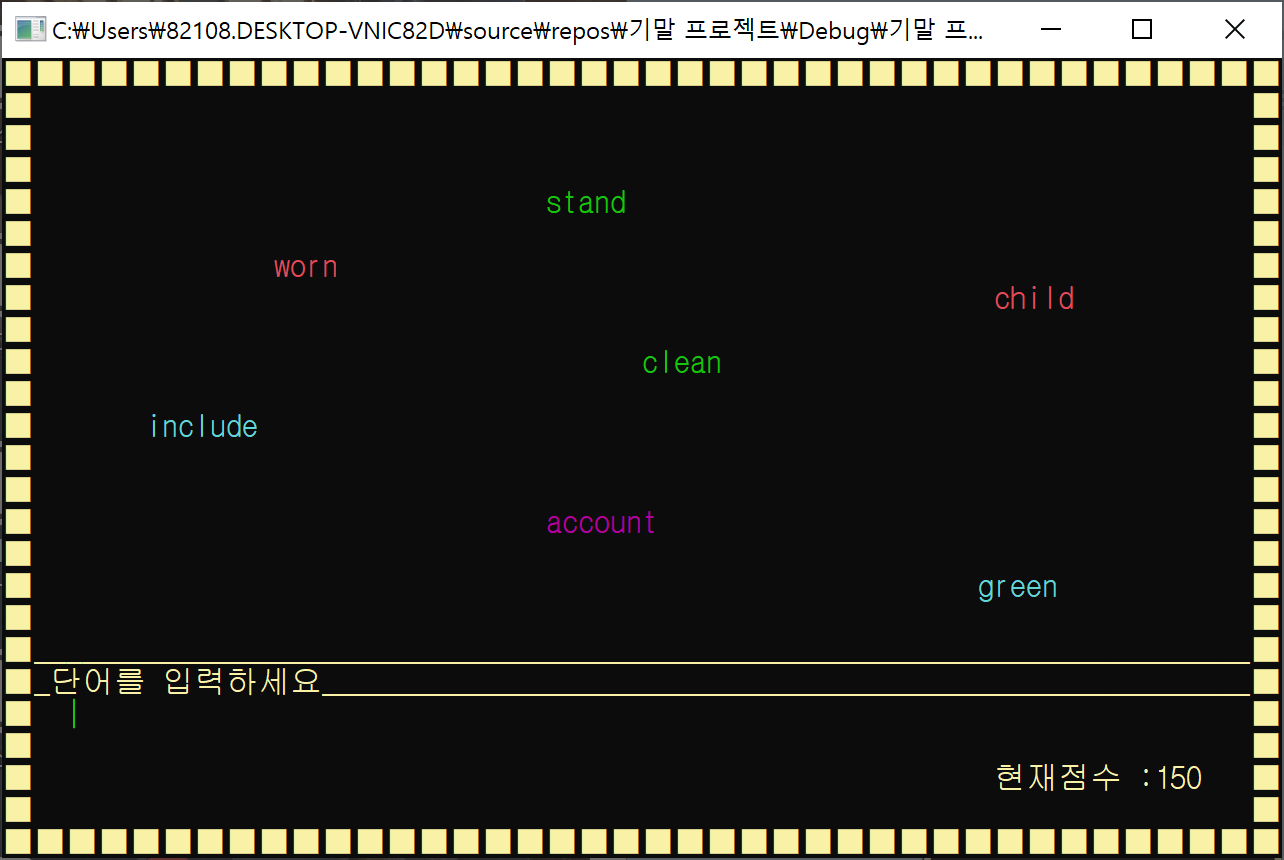
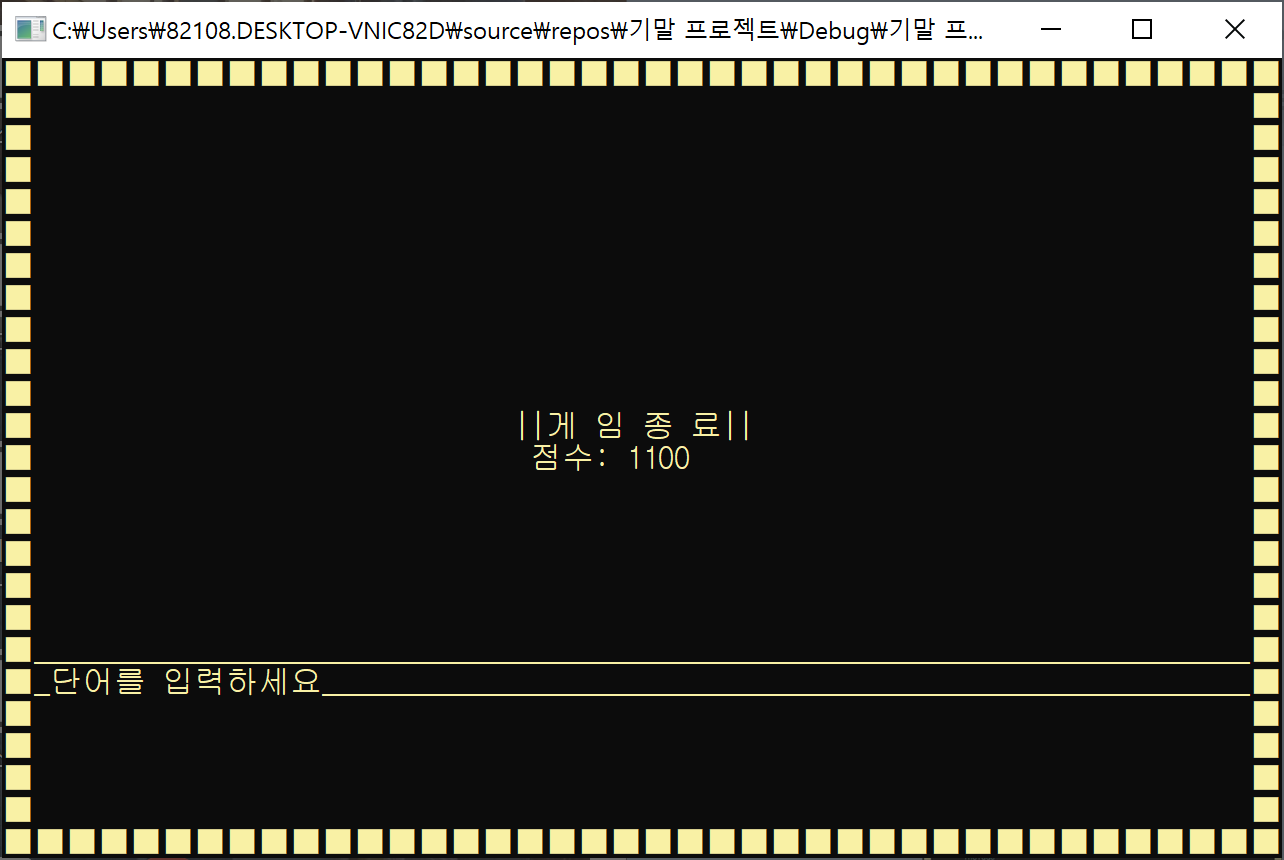
자동 생성된 설명  
 [자료 4-2] define정의  
단어의 개수를 95로 정의 하였고 단어가 나타나는 속도를 20000 즉 2초로 정의하였다.  
키보드 값은 아스키 표를 참고 한 것이 아니라 내가 임의로 0 1 2로 정의하였다.  
  
 [자료 4-3] 함수 선언  
게임에 필요한 함수를 먼저 선언하였다. 어떤 용도로 쓰이는 함수인지는 위의 [자료 4-3]의 주석을 참고한다.

   
 [자료 4-4] 변수  
이 프로그램에서 쓰일 변수를 미리 선언하였다. 어떤 용도로 쓰이는 변수인지는 위의 [자료 4-4]의 주석을 참고한다.  
  
  
  
  
 [자료 4-5] gotoxy함수  
gotoxy함수는 커서 위치 이동을 제어하는 함수로 내가 원하는 위치에 글자와 문자 숫자를 출력하기 위해 사용하였다.  
  
  
 [자료 4-6] 단어를 2차원배열에 저장  
위 자료는 게임플레이 화면에서 나타날 단어들을 이차원 배열에 저장을 하였다.

  
  
 [자료 4-7] main 함수  
위 자료는 main함수로써 이 프로그램에서 실행되어야 할 함수들을 호출하여 썼다.  
콘솔창의 크기를 조절해서 출력하는 함수 title1을 호출하였고, 반복문 while안에 게임 시작 화면의 메뉴 코드를 선택하면 실행되는 함수들을 호출하였다.  
  
  
  
  
 [자료 4-8] keycontrol 함수  
keycontrol함수는 키보드 키의 출력을 위한 함수이며 이 함수를 사용하기 위해 getch함수를 사용하였다. getch함수의 이용은 위의 [자료 4-8]을 참고하며, 위쪽 방향키 역할을 키보드 w, 아래쪽 방향키 역할은 키보드 s, 선택키 역할은 스페이스바로 정의 하였다.  
  
  
  
  
 [자료 4-9] design 함수  
위 자료는 게임 첫 화면의 디자인을 나타내는 함수를 정의 한 것이며, 여기서 SetConsoleTextAttribute()는 콘솔 글자 색을 지정하는 함수이다.  
  
  
  
  
  
 [자료 4-10] design2 함수  
게임 플레이 시 화면의 디자인을 나타내는 함수이다.  
  
  
  
  
 [자료 4-11] gamemain 함수  
게임 플레이 화면이 시작 될 때의 화면을 정의한 함수인데 여기서 gotoxy함수는 gotoxy(x값, y값)을 숫자로 입력해 사용을 한다. Sleep함수는 Sleep(2000); 이런 식으로 사용을 하는데 1000은 1초를 나타내며 2000은 2초이다. System(“cls”)는 화면을 지우는 명령이다.  
  
  
  
  
  
  
 [자료 4-12] printword함수  
[자료 4-12]는 단어 출력을 위한 함수 정의를 나타낸 것이며, 자세한 내용은 위의 자료의 주석을 참고하기 바란다.

  
  
  
  
  
  
  
 [자료 4-13] scanword 함수  
위 자료는 게임 플레이 시 단어를 입력하는 것을 위해 필요한 함수로 이 자료에 대한 설명은 자료에 있는 주석을 참고하기 바란다.  
  
  
  
  
  
 [자료 4-14] title1 함수  
[자료 4-14]는 콘솔창의 크기를 조절하기 위한 함수를 나타낸 것이다.

  
  
  
  
 [자료 4-15] title2 함수  
위 자료는 게임을 실행할 시 처음 나타나는 화면의 메뉴 선택을 나타내는 함수이며 자세한 설명은 위 자료의 주석을 참고하기 바란다.  
  
  
  
  
  
  
  
  
 [자료 4-16] help 함수  
게임 도움말 창을 나타내기 위한 함수이다. System(“cls”)를 써서 콘솔창 화면을 지운 다음 도움말 창을 출력한다. 이 함수에서는 이전 페이지(시작화면)으로 돌아가기 위해 반복문 while을 썼다.  
  
  
  
  
  
스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명  
 [자료 4-17] 커서를 제거하는 함수  
콘솔창에서 커서의 깜빡임을 제거하기 위해서 쓴 함수이다.  
  
5.프로그램의 동작과 결과  
  
  
 [자료 5-1] 게임시작 창  
위 자료는 exe파일을 실행하면 처음 뜨는 창이다. 게임 타이틀과 메뉴 선택창이 나온다.  
  
  
 [자료 5-2] 게임 도움말 창  
게임방법 메뉴에 들어가면 나오는 창이다. 게임에 대한 소개와 게임 방법에 관한 설명이 있다.  
  
 [자료 5-3] 게임시작 시 뜨는 창  
게임시작 메뉴를 선택하면 맨 처음에 뜨는 창이다.  
  
  
 [자료 5-4] 게임 플레이 화면  
게임 플레이 화면이다. 나타나는 단어를 입력해서 없애는 게임이다. 단어의 색 마다 점수가 다르다.  
  
 [자료 5-5] 게임 플레이 시 나타나는 점수  
단어를 입력하면 점수가 표시된다.  
  
  
 [자료 5-6] 게임이 끝났을 때 창  
게임 종료 창이며 게임 점수가 뜬다. 5초뒤 시작화면으로 돌아간다.



6. 결과 분석 및 토의  
  
앞에서 말했지만 처음 이 게임은 산성비 게임을 계획하고 만들었으나 게임을 구현하면서 단어를 없애면 단어의 색깔 별로 점수를 얻는 게임으로 만들어보고 싶어서 이 게임을 만들게 되었다. 산성비 게임처럼 단어가 화면에 뜬 다음 밑으로 떨어지는 건 아니지만 단어를 입력하면 색깔 별로 점수를 얻고 단어가 사라진다. 게임의 오류는 없었고 잘 실행 되었다. 다만 아쉬웠던 점은 게임이 끝난 경우 다음 스테이지로 넘어가게 만들고 싶었지만 만들지 못했다는 점이다. 이 외에는 게임이 잘 작동되고 처음 만든 게임이지만 기대 이상의 결과가 나와서 만족스럽다.

7. 결론  
  
게임을 만드는 것이 쉽지 않다고 느꼈다. 게임 구상부터 게임 디자인, 그리고 게임을 어떤 방향으로 잡아서 알고리즘을 짜볼까 라는 생각들을 실제로 실현하는게 너무 어려웠다. 평소 게임을 즐겨하는데 내가 게임을 할 때는 아무 생각 없이 한 게임들이 막상 내가 직접 만들어 보니 게임을 제작 하는게 얼마나 힘든 일인지 알게 되었다. 내가 만든 게임은 간단한 게임일지라도, 내가 게임을 만들기 위해 구상과 소스코드를 짠 것이 대견스럽다. 이를 계기로 더 다양한 게임을 만들어 보고싶다는 생각을 하게 되었다.  
  
8. 참고문헌  
  
C언어로 배우는 프로그래밍 기초 Perpect C (강환수, 강환일, 이동규)  
  
[커서 이동 함수] <http://blog.naver.com/PostView.nhn?blogId=sharonichoya&logNo=220873844942>  
  
[콘솔 창 조절]  
<http://blog.daum.net/_blog/BlogTypeView.do?blogid=02XGV&articleno=13720680&categoryId=558131&regdt=20190530212352>

[게임 동작 원리 & 게임 참고]  
<https://blog.hyomin.xyz/17>

[메뉴 선택 창 원리]

<https://blog.naver.com/PostView.nhn?blogId=dnpc7848&logNo=220339329524&targetKeyword=&targetRecommendationCode=1>

[콘솔 창 글자 색]  
<http://blog.naver.com/PostView.nhn?blogId=wcwtmt&logNo=10172917569>

9. 별첨 (소스코드와 주석)

/\*헤더파일\*/

#include <stdio.h> //표준 입출력 작업을 위한 헤더파일

#include <stdlib.h> //여러 유틸리티(텍스트를 수로 변환 등)함수 작업을 위한 헤더 파일

#include <time.h> //시간 작업을 위한 헤더파일

#include <Windows.h> //윈도우 응용 프로그램을 만들기 위한 헤더파일

#include <conio.h> //getche()함수를 이용하기 위한 헤더파일

#include <string.h> //strcmp 함수가 선언된 헤더파일

#define wordcount 95 //단어의 개수 정의

#define wordtime 20000 //단어가 나타나는 속도 정의

/\*키보드값을 매크로로 정의\*/

#define UP 0 //0을 UP으로 정의

#define DOWN 1 //1을 DOWN으로 정의

#define SUBMIT 2 //2를 SUBMIT로 정의

time\_t starttime = 0, endtime = 0; //시스템 시간을 저장

/\*함수선언\*/

int keycontrol(); //키보드 키 출력을 위한 함수선언

void title1(); //프로그램 실행 직후 초기화작업을 진행할 함수선언

int title2(); //처음 게임창의 메뉴 선택화면을 위한 함수선언

void gotoxy(int x, int y); //커서 위치 이동 함수선언

void design(); //게임 첫화면의 디자인 함수선언

void design2(); //게임을 플레이 할 때의 함수선언

void gamemain(); //게임 플레이 화면의 함수선언

void printword(); //단어출력을 위한 함수선언

void scanword(); //단어입력을 위한 함수선언

void removecursor(); //커서제거를 위한 함수선언

void help(); //게임 도움말창을 위한 함수선언

/\*변수\*/

int wordx[250]; //한 번뜬 단어들을 안뜨게 하는 변수

int x, y; //gotoxy함수의 x값과 y값의 변수

int i, j; //단어의 수의 변수

int x2, y2; //x, y값을 다시 불러올 때 쓰는 변수

int score = 0; //점수를 나타낼 변수 초기화

int f; //단어의 색깔 변수

int f2; //단어 색깔 값을 다시 불러올때 변수

int sword; //입력된 단어를 나타낸 변수

char scan[50]; //입력할 단어를 위한 변수

char remem[263][3];

int gap; //게임지속시간을 위한 변수

void gotoxy(int x, int y) //커서 위치 이동 함수정의

{

HANDLE consoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE); //콘솔 핸들가져오기

COORD pos;

pos.X = x;

pos.Y = y;

SetConsoleCursorPosition(consoleHandle, pos);

}

char word[256][256] = { "dog", "cat", "bottle", "phone", "robot", "green", "elephant", "include", "sky", //단어를 2차원 배열에 값 저장

"game", "love", "yuri", "max", "knife", "glass", "class", "art", "smart",

"bell", "carry", "climb", "between", "blow", "album", "ago", "among", "animal",

"any", "box", "and", "board", "body", "child", "city", "boy", "bridge",

"clean", "club", "coat", "bright", "coin", "coffee", "cold", "chance",

"chalk", "chair", "cheap", "blue", "before", "bowl", "aunt", "as", "away", "bicycle",

"church", "card", "hold", "chose", "come", "drink", "give", "get", "hurt", "lay", "had",

"feed", "lend", "met", "sing", "throw", "wet", "tell", "set", "wind", "wear", "write",

"spend", "stand", "worn", "win", "sweep", "account", "achieve", "across", "accept",

"above", "ability", "abuse", "abnormal", "absurd", "absent",

"nation", "past", "value", "though", "person", "machine", "stand", "null" };

int main()

{

title1(); //title1함수 호출

while (1) //while문 반복

{

design(); //design함수 호출

int menucode = title2(); //menucode변수를 title2함수를 호출해서 저장

if (menucode == 0) //menucode가 0일 때

{

score = 0; //점수

starttime = clock(); //시작시간을 clock함수를 호출해서 저장

gamemain(); //gamemain함수 호출

gap = 0; //게임지속시간

}

else if (menucode == 1) //menucode가 1일 때

{

help(); //help함수 호출

}

else if (menucode == 2) //menucode가 2일 때

{

exit(1); //종료

}

}

return 0;

}

int keycontrol() //키보드 키 출력을 위한 함수정의

{

char temp = getch(); //enter키를 치지않아도 화면에 바로 글자가 뜸

if (temp == 'w' || temp == 'W') //w or W키를 입력할 때

{

return UP; //위쪽 방향키 역할을 함

}

else if (temp == 's' || temp == 'S') //s or S키를 입력할 때

{

return DOWN; //아래쪽 방향키 역할을 함

}

else if (temp == ' ') //스페이스바(공백)을 입력할 때

{

return SUBMIT; //선택키 역할을 함

}

}

void design() //게임 첫화면의 디자인 함수정의

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 11); //콘솔 글자 색 지정 연한 파란색을 나타냄

printf("□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■"); //게임 시작화면 디자인 출력

printf("■ □");

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14); //콘솔 글자 색 지정 연한 노란색을 나타냄

printf("□ ###### ## ## ## ###### ■");

printf("■ ### ## ##### #### ### ## □");

printf("□ ## ## ## ## ## ## ## ■");

printf("■ ### #### ######## ## # ## ######## □");

printf("□ ## ## ## ## ## ## ## ■");

printf("■ ###### ## ## ## ## ######### □");

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 11); //콘솔 글자 색 지정 연한 파란색을 나타냄

printf("□ ■");

printf("■ ||나타나는 단어를 빨리 없애기|| □");

printf("□ ■");

printf("■ □");

printf("□ ■");

printf("■ □");

printf("□ ■");

printf("■ □");

printf("□ ■");

printf("■ □");

printf("□ ■");

printf("■ □");

printf("□ ■");

printf("■ □");

printf("□ ■");

printf("■ □");

printf("□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■");

}

void design2() //게임 플레이 화면의 디자인 함수정의

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14); //콘솔 글자 색 지정 연한 노란색을 나타냄

printf("■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"); //게임 플레이화면의 디자인 출력

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_■");

printf("■\_단어를 입력하세요\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■ ■");

printf("■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■");

}

void gamemain() //게임 플레이 화면의 함수정의

{

system("cls"); //화면을 지움

gotoxy(35, 10); //x좌표를 35 y좌표를 10으로 설정

printf("3초 후 시작");

Sleep(1000); //아무것도 안하고 1초 기다림

system("cls"); //화면을 지움

gotoxy(35, 10); //x좌표를 35 y좌표를 10으로 설정

printf("2초 후 시작");

Sleep(1000); //아무것도 안하고 1초 기다림

system("cls"); //화면을 지움

gotoxy(35, 10); //x좌표를 35 y좌표를 10으로 설정

printf("1초 후 시작");

Sleep(1000); //아무것도 안하고 1초 기다림

system("cls"); //화면을 지움

design2(); //design2함수 호출

while (1) //while문 반복

{

printword(); //printword함수 호출

scanword(); //scanword함수 호출

}

}

void printword() //단어출력을 위한 함수 정의

{

static int check = wordcount; //정적 변수 선언 및 wordcount에 초기화(변수가 사라지지 않게 함)

x = rand() % 63; //x좌표를 63-1까지 랜덤한 값으로 설정

y = rand() % 15+2; //y좌표를 2에서 15-1까지 랜덤한 값으로 설정

i = rand() % wordcount; //i를 wordcount 95개로 정의한 값 중에 랜덤한으로 설정

f = rand() % 4 + 1; //색 지정 색을 랜덤하게

if(wordx[i]!=1) //한 번 뜬 단언을 안뜨게 하기 위한 if문

{

wordx[i] = 1; //배열 i번째 요소를 1로 저장

check--; //cheak값을 감소 시킴

gotoxy(x, y); //gotoxy로 x와 y좌표 설정

for (j = 0; j < 3; j++) //단어 출력을 위한 for문

{

if (j == 0) //j값이 0일 때

{

remem[i][j] = x; //배열을 x값에

}

else if (j == 1) //j값이 1일 때

{

remem[i][j] = y; //배열을 y값에

}

else if (j == 2) //j값이 2일 때

{

remem[i][j] = f; //색 지정

}

}

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), f+9); //콘솔 글자 색 지정 단어색을 10번 색부터

printf("%s ", word[i]); //단어를 출력

}

}

void scanword() //단어 입력을 위한 함수 정의

{

int p = 0;

while (p < wordtime)

{

p++;

int scanc;

endtime = clock(); //종료시간을 clock함수를 호출해서 저장

gap = (float)(endtime - starttime) / (CLOCKS\_PER\_SEC); //게임 지속시간을 위한 변수 gap

if (gap > 50) //게임지속시간 50s

{

system("cls"); //화면을 지움

design2(); //design2 함수 호출

gotoxy(32, 11); //x좌표를 32 y좌표를 11으로 설정

printf("||게 임 종 료||");

gotoxy(33, 12); //x좌표를 32 y좌표를 12으로 설정

printf("점수: %d", score);

Sleep(5000); //아무것도 안하고 5초 기다림

system("cls"); //화면을 지움

main(); //main함수 호출

}

if (kbhit()) //kbhit()함수를 사용하여 키 눌림의 유무를 판단

{

scanc = getch(); //입력한 글자를 엔터키를 사용하지 않고 바로 화면에 출력하기 위해 사용하는 getch함수

if (scanc != 8 && scanc != 13) //scamc가 8과 13이 아닐 때

{

gotoxy(5, 20);

scan[sword++] = scanc;

printf(" "); //단어를 입력하면 출력됨

gotoxy(5, 20);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 7); //콘솔 글자 색 지정 하얀색을 나타냄

printf("%s", scan); //입력한 글자 출력

}

else if (scanc == 8) //scanc가 8일 때

{

gotoxy(5, 20); //x좌표를 5 y좌표를 20으로 설정

printf("| "); //단어를 잘못입력시 지우면 나타나는 커서

gotoxy(5, 20); //x좌표를 5 y좌표를 20으로 설정

scan[--sword] = scanc; //글자 지우기

if (sword == -1) //sword가 -1일 때

sword = 0;

printf("%s", scan); //입력할 단어 출력

}

else if (scanc == 13) //scanc가 13일 때

{

sword = 0;

gotoxy(5, 20); //x좌표를 5 y좌표를 20으로 설정

printf("| "); //다음 단어 입력 전 입력창이 초기화되어 출력

for (i = 0; i <= wordcount; i++)

{

if (!strcmp(scan, word[i])) //strcmp 결과 scan과 word[i]값을 거짓을 참으로 바꿈

{

x2 = remem[i][0];

y2 = remem[i][1];

f2 = remem[i][2];

gotoxy(x2, y2);

printf(" "); //출력된 단어를 입력하면 단어가 사라짐

gotoxy(4, 20);

printf("| ");

switch (f2)

{

case 1: //10번째 색 연두색일 경우

score += 150;

break;

case 2: //11번째 색 하늘색일 경우

score += 200;

break;

case 3: //12번째 색 연한빨간색일 경우

score += 250;

break;

case 4: //13번째 색 분홍색일 경우

score += 300;

break;

default:

break;

}

gotoxy(62, 22);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

printf("현재점수 :%d", score);

}

}

for (i = 0; i < 50; i++)

{

scan[i] = NULL; //단어를 입력하는중에 출력되는 글자가 없음

}

}

}

}

}

void title1() //프로그램 실행 직후 초기화작업을 진행할 함수 정의

{

system("mode con cols=80 lines=25"); //콘솔창 크기 조절

HANDLE consoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE); //콘솔 핸들가져오기

CONSOLE\_CURSOR\_INFO ConsoleCursor;

ConsoleCursor.bVisible = 0;

ConsoleCursor.dwSize = 1;

SetConsoleCursorInfo(consoleHandle, &ConsoleCursor);

}

int title2() //처음 게임창의 메뉴 선택화면을 위한 함수 정의

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

int x = 35, y = 13;

gotoxy(x - 2, y); //-2를 하는 이유는 ▷를 출력하기 위해서

printf("▷ 게임시작");

gotoxy(x, y + 1);

printf(" 게임방법");

gotoxy(x, y + 2);

printf(" 종 료\n\n");

gotoxy(x - 15, y + 4);

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(x - 15, y + 6);

printf(" |UP키: W DOWN키: S SELECT키: SPACEBAR|");

gotoxy(x - 15, y + 7);

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(x - 16, y + 9);

printf("| 대구가톨릭대학교 19120450 IT공학부 김유리 |");

while (1) //메뉴화면 선택을 위한 while문

{

int a = keycontrol(); //변수 a를 keycontrol함수에 저장

switch (a) //메뉴선택을 위한 switch문 사용

{

case UP: //case값이 UP(W)일 때

{

if (y > 13) //y좌표를 13~15만 이동

{

gotoxy(x - 2, y); //-2를 한 이유는 ▷의 출력을 위해

printf(" "); //원래 위치를 지움

gotoxy(x - 2, --y); //새로 이동한 위치로 이동

printf("▷"); //다시 ▷ 출력

}

break;

}

case DOWN: //case값이 DOWN(S)일 때

{

if (y < 15) //y값이 최대 15

{

gotoxy(x - 2, y);

printf(" ");

gotoxy(x - 2, ++y);

printf("▷");

}

break;

}

case SUBMIT: //case값이 SUBMIT(스페이스바)일 때

{

return y - 13; /\*스페이바를 눌렀을 경우 y위치가 13이였으므로

y-13을 하면 0,1,2 세 숫자를 받을 수 있음\*/

}

}

}

}

void help() //게임 도움말창을 위한 함수 정의

{

system("cls"); //콘솔창 화면을 지움

design(); //design함수 호출

gotoxy(10, 12);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

printf("사방에서 뜨는 단어들을 입력하여 50초 동안 점수를 얻는 게임입니다\n");

gotoxy(27, 13);

printf("단어의 색 마다 배점이 다릅니다");

gotoxy(37, 15);

printf("[배점]");

gotoxy(19, 17);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 10);

printf("■ : 150점");

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 11);

printf(" ■ : 200점");

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 12);

printf(" ■ : 250점");

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 13);

printf(" ■ : 300점");

gotoxy(18, 19);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 7);

printf("| 대구가톨릭대학교 19120450 IT공학부 김유리 |");

gotoxy(19, 21);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

printf("이전페이지로 돌아가려면 SPACEBAR를 누르세요");

while (1) //이전페이지로 돌아가기 위한 while문

{

if (keycontrol() == SUBMIT) //스페이스바를 눌렀을 때

{

main(); //main함수 호출

}

}

}

void removecursor() //커서를 제거하는 함수

{

CONSOLE\_CURSOR\_INFO curInfo;

GetConsoleCursorInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &curInfo);

curInfo.bVisible = 0;

SetConsoleCursorInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &curInfo);

}