

REPORT



성결대학교
SUNGKYUL UNIVERSITY

지뢰찾기 게임, 개인정보 관리 프로그램



과목명 | C프로그래밍응용

담당교수 | 박미옥

학과 | 컴퓨터공학과

학년 | 1학년

학번 | 20230817, 20230822
20230837, 20230842

이름 | 김도현, 김민철,
김형호, 박성준

제출일 | 2023.12.10

팀명 : 디버깅 오류

팀원

학번

이름

역할

20230822 - 김민철 (팀장, 총괄)

20230817 - 김도현 (알고리즘)

20230837 - 김형호 (레포트)

20230842 - 박성준 (코드)

목차

팀명과 팀원	2
1. 지뢰찾기 게임	
I. 기획 의도	5
II. 협업 방법	6
1. 카카오톡	
2. 디스코드	
3. Replit	
III. 알고리즘	8
1. 자연어	
2. 순서도	
IV. C언어 코드	10
V. 실행 결과	36

2. 개인정보 관리 프로그램

I. 기획 의도 -----39

II. 알고리즘 -----40

1. 자연어

2. 순서도

III. C언어 코드 -----42

IV. 실행 결과 -----54

1. 지뢰찾기 게임

I. 기획 의도

팀원을 모두 모은 후, 각자 게임 코드를 한 개씩 찾아서 모이기로 했다. 하지만 이후 회의에서 인터넷에서 복사해 오는 것 보단, 조금 더 간단한 게임이라도 직접 만들어 보는 것이 C언어 능력과 협업 프로젝트 수행 능력을 더 많이 향상할 수 있다는 의견을 통해 프로그래밍하기 적당한 난이도를 가지고 있는 지뢰찾기 게임을 만들기로 하였다.

지뢰찾기 게임이 원활하게 진행되기 위해서 보드게임의 기능을 하나씩 분석하였다.

- 사용자에게 보이기 위한 보드판을 출력한다.
- 사용자에게 보이지 않는 폭탄이 숨겨진 보드판을 만든다,
- 보드판의 무작위 위치에 폭탄을 생성한다,
- 사용자가 폭탄을 고르면 게임이 끝나고 프로그램이 종료된다,
- 사용자가 입력한 곳 주변에 폭탄이 있으면 폭탄 개수를 표기한다,
- 사용자가 입력한 곳 주변에 폭탄이 없는 빈칸이 연쇄적으로 깨지는 기능
- 폭탄의 위치에는 깃발을 놓아 표기하는 기능
- 깃발을 지우는 기능

간단한 게임이지만 많은 알고리즘과 기능들이 들어있었다.

각 기능들을 알고리즘(순서도, 자연어)으로 짜와서 필요한 부분들은 서로 합치고 보완해서 최종 지뢰찾기 프로그램을 만들기로 하였다. 프로그램을 만들면서 추가하면 좋을 것 같은 기능들을 추가하였다. 지뢰찾기를 처음 접하는 사용자들을 위해 폭탄의 위치를 알 수 있는 정답공개 기능을 추가하였다. 사용자들이 게임을 진행하며 성장하는 재미를 위해서 난이도 선택을 통해 보드판의 크기를 조절할 수 있는 기능을 추가하였다.

무작위로 폭탄이 설치된 보이지 않는 보드판

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	1	1	B	1	0
2	0	1	2	3	B	2	3	2	2	0
3	0	1	B	B	4	B	3	B	1	0
4	0	1	3	3	4	B	4	2	1	0
5	0	0	1	B	2	2	B	1	0	0
6	0	0	1	1	1	1	2	2	1	0
7	0	0	0	0	0	0	1	B	1	0
8	0	0	0	0	0	0	1	1	1	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

II. 협업 방법

팀프로젝트를 진행할 때마다 한 장소에 모여서 진행하는 것은 다 같이 모일 시간을 조율하는 것이 쉽지 않고, 효율적이지 못하다고 판단하였다. 그 때문에 온라인의 다양한 도구들을 이용해서 팀프로젝트를 진행하였다.

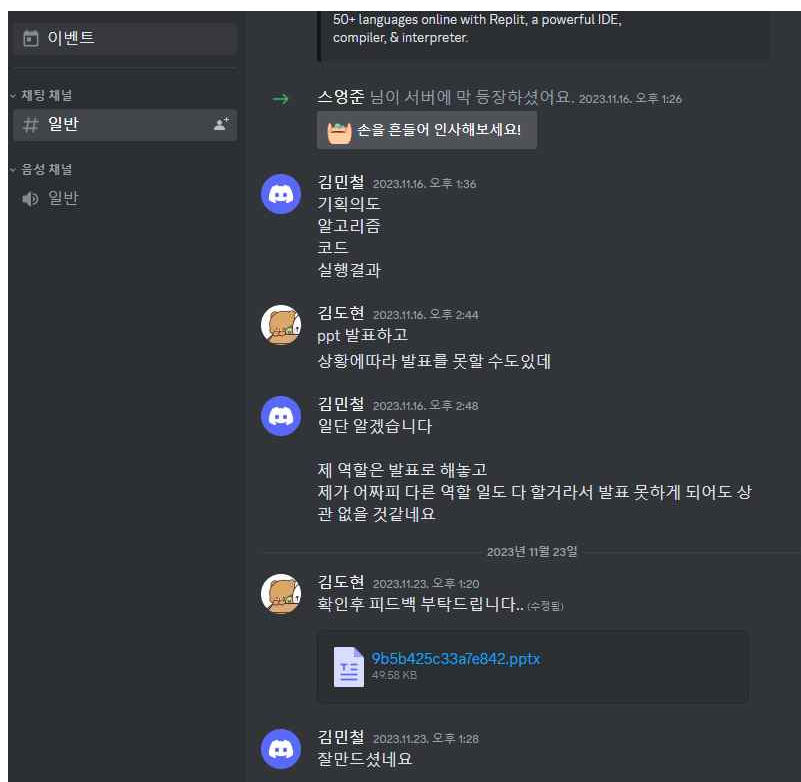
1. 카카오톡

처음 팀이 결성된 후 카카오톡을 통해서 서로 의사소통을 진행하였다. 이후 일정 조율, 프로젝트의 방향성 정하기, 간단한 의견 교환 등을 카카오톡을 통해서 결정하였고 이후 프로젝트 진행 중에도 완료 보고, 피드백, 공지 사항, 파일 공유는 모두 카카오톡을 통해서 진행하였다.

2. 디스코드

원거리에서 다 같이 통화를 하며 음성으로 회의를 진행하기 위해서 디스코드 앱을 이용하였다. 모든 회의는 디스코드를 통해서 진행하였으며 지뢰찾기 게임의 세부적인 기능 조율, 알고리즘 정리, 코딩 등의 채팅으로 진행하기에는 한계가 있는 부분들을 디스코드를 활용해서 진행하였다. 또한 디스코드의 화면공유 기능을 이용해서 각자 준비해 온 순서도, 알고리즘 파일을 이해하기 쉽게 직접 보여주며 설명하면서 작업의 속도를 높일 수 있었다. 음성 회의를 통해서 결정된 내용들은 디스코드의 채팅창에 바로 정리하여 알아보기 쉽게 하였다.

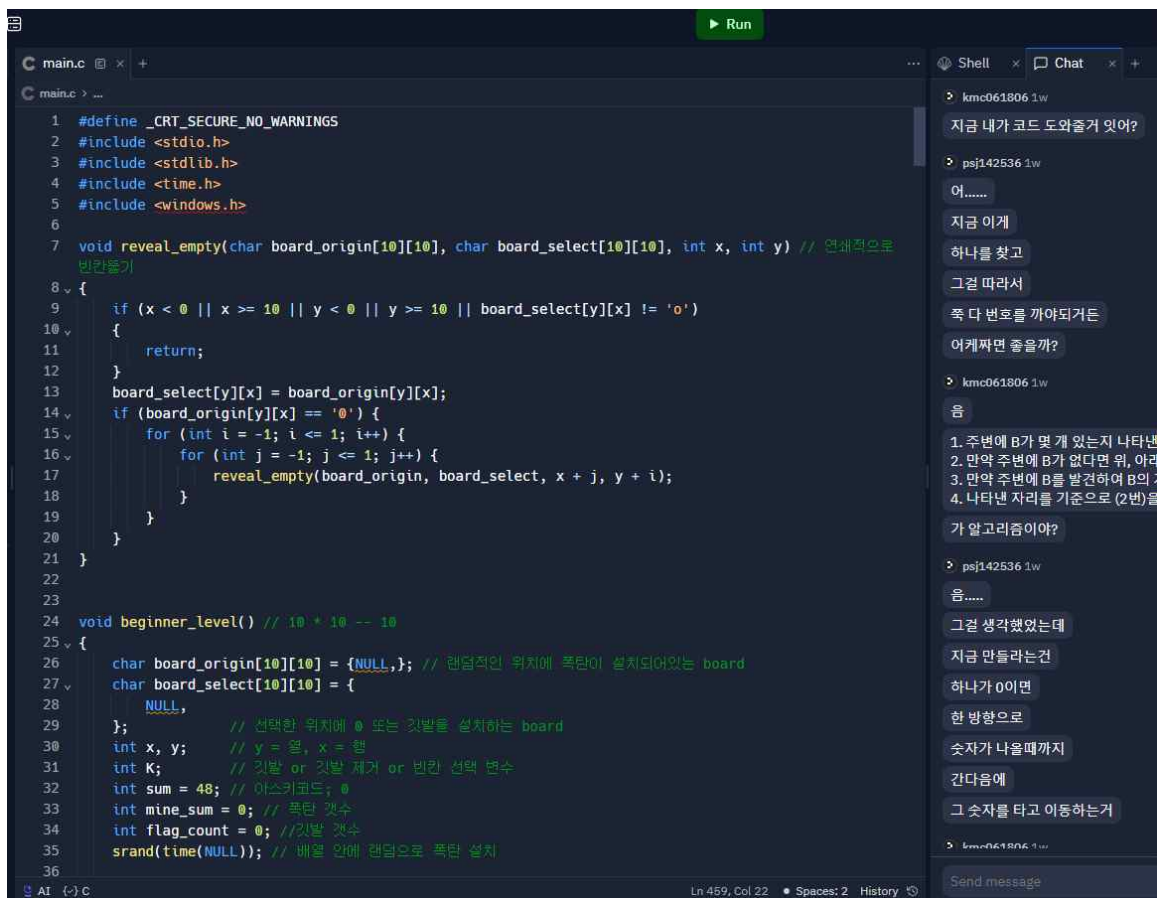
디스코드 활용 사진



3. Replit

프로그램을 짜고 수정할 때마다 서로 파일을 교환하기에는 많은 번거로움이 있었기 때문에 협업 도구를 이용하였다. 처음에는 Visual Studio 2022의 협업 기능을 이용했지만, 프로젝트 책임자만 실행할 수 있다는 단점이 존재하였다. 이후 구성원 모두 자유롭게 편집과 실행이 가능하며 실시간으로 수정 내용이 코드에 적용되고, 필요에 따라서 채팅 기능도 이용할 수 있는 Replit을 이용하였다.

Replit 활용 사진



III. 알고리즘

회의를 통해서 게임의 기능을 결정하고 자연어로 먼저 알고리즘을 작성 후 순서도로 작성하여 기능들을 정리하였다.

1. 자연어

1. 프로그램 시작

2. 난이도를 입력받는다.

2-1. 1번을 입력받으면 초급 난이도 실행

2-2. 2번을 입력받으면 중급 난이도 실행

2-3. 3번을 입력받으면 고급 난이도 실행

2-4. 1, 2, 3 말고 다른 값을 입력받으면 “올바른 숫자를 입력하십시오” 출력 후 2번으로 돌아간다.

3. 다음 행동을 입력받는다

3-1. 1번을 입력받으면 깃발을 놓는다

3-1-1. 빈칸 선택 시 깃발을 놓는다

3-1-2. 깃발이 있는 칸 선택 시 “다시 선택하세요” 출력한다

3-1-3. 넓은 칸 선택 시 “다시 선택하세요” 출력한다

3-2. 2번을 입력받으면 깃발을 취소한다.

3-2-1. 넓은 칸 선택 시 “다시 선택하세요”를 출력한다

3-2-2. 깃발이 놓인 칸 선택 시 깃발을 제거한다

3-2-3. 빈칸 선택 시 “다시 선택하세요”를 출력한다

3-3. 3번을 입력받으면 빈칸을 넓힌다.

3-3-1. 폭탄이 없는 칸을 선택하면 칸을 넓힌다.

3-3-2. 폭탄이 있는 칸을 선택하면 답안지를 보여주고 “폭탄을 선택했습니다.” 출력 후 게임을 종료한다

3-3-3. 깃발이 있는 칸을 선택하면 “다시 선택하세요: 출력하고 창을 정리한다

3-3-4. 모든 칸을 넓히면 “축하합니다. 모든 지뢰의 위치를 알아냈습니다” 출력 후 답지를 보여주고 게임을 종료한다.

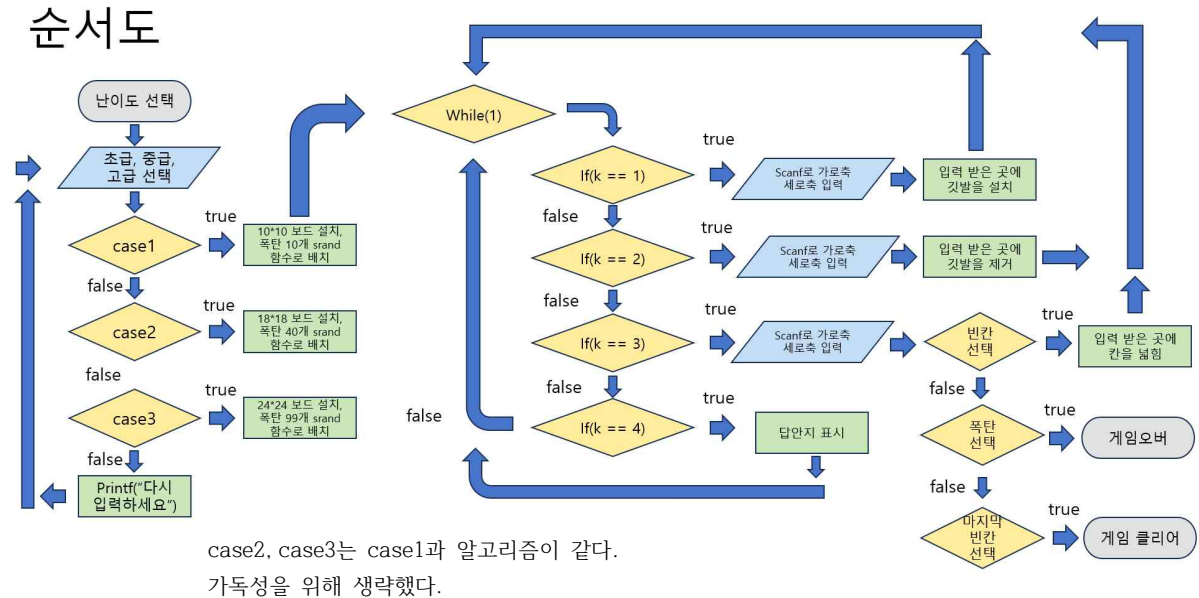
3-4. 4번을 입력받으면 답지를 연다.

3-4-1. 답지를 보여주고 다음 행동 때 답지를 지운다

4. 다시 3번으로 돌아간다.

2. 순서도

순서도는 순서도 체계 시스템을 이용해서 자연어 알고리즘을 먼저 완성한 후에 자연어를 그림으로 나타내는 방식으로 진행하였다.



위와 같이 자연어와 순서도로 프로젝트를 진행하는 팀원 모두가 이해하기 쉽게 정리하고 프로젝트 진행 중에 기억이 안 나거나 애매했던 부분은 자연어, 순서도 알고리즘을 확인하면서 명확한 프로그램을 개발할 수 있었다.

IV. C언어 코드

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <windows.h>

void begin_reveal_empty(char board_origin[10][10], char board_select[10][10], int x, int y) //
초급 난이도 빈칸 연쇄적으로 뚫는 함수
{
    if (x < 0 || x >= 10 || y < 0 || y >= 10 || board_select[y][x] != 'o')
    {
        return;
    }
    board_select[y][x] = board_origin[y][x];
    if (board_origin[y][x] == '0')
    {
        for (int i = -1; i <= 1; i++)
        {
            for (int j = -1; j <= 1; j++)
            {
                begin_reveal_empty(board_origin, board_select, x + j, y + i);
            }
        }
    }
}

void mid_reveal_empty(char board_origin[18][18], char board_select[18][18], int x, int y) // 중
급 난이도 빈칸 연쇄적으로 뚫는 함수
{
    if (x < 0 || x >= 18 || y < 0 || y >= 18 || board_select[y][x] != 'o')
    {
        return;
    }
    board_select[y][x] = board_origin[y][x];
    if (board_origin[y][x] == '0')
    {
```

```

    for (int i = -1; i <= 1; i++)
    {
        for (int j = -1; j <= 1; j++)
        {
            mid_reveal_empty(board_origin, board_select, x + j, y + i);
        }
    }
}

```

`void hard_reveal_empty(char board_origin[24][24], char board_select[24][24], int x, int y) // 고`
 급 난이도 빈칸 연쇄적으로 뚫는 함수

```

{
    if (x < 0 || x >= 24 || y < 0 || y >= 24 || board_select[y][x] != 'o')
    {
        return;
    }
    board_select[y][x] = board_origin[y][x];
    if (board_origin[y][x] == '0')
    {
        for (int i = -1; i <= 1; i++)
        {
            for (int j = -1; j <= 1; j++)
            {
                hard_reveal_empty(board_origin, board_select, x + j, y + i);
            }
        }
    }
}
}

```

`void beginner_level() // 10 * 10 -- 10`

```

{
    char board_origin[10][10] = { NULL, }; // 랜덤적인 위치에 폭탄이 설치되어있는 board
    char board_select[10][10] = { NULL, }; // 사용자가 원하는 기능을 수행하는 board
    int x, y;      // y = 열, x = 행
    int K;         // 깃발 or 깃발 제거 or 빈칸 선택 or 답지 확인 변수
    int sum = 48; // 아스키코드; 0
}

```

```

int mine_sum = 0; // 폭탄 개수
int flag_count = 0; // 깃발 개수
int limit = 0;

srand(time(NULL)); // 배열 안에 랜덤으로 폭탄 설치

for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        board_origin[i][j] = 'o';
    }
}

for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        board_select[i][j] = 'o';
    }
}

while (mine_sum < 10)
{
    x = rand() % 10;
    y = rand() % 10;

    if (board_origin[y][x] == 'B')
    {
        continue;
    }
    else
    {
        board_origin[y][x] = 'B';
        mine_sum++;
    }
}

for (int p = 0; p < 10; p++)

```

```

{
    for (int q = 0; q < 10; q++)
    {
        for (int i = 1; i >= -1; i--)
        {
            for (int j = 1; j >= -1; j--)
            {
                if (board_origin[p - i][q - j] == 'B' && q - j < 10 && q - j > -1 &&
board_origin[p][q] != 'B')
                {
                    sum++;
                }
            }
        }
        if (board_origin[p][q] != 'B')
        {
            board_origin[p][q] = sum;
        }
        sum = 48;
    }
}

printf("  ");

for (int j = 1; j <= 10; j++)
{
    printf("| %2d| ", j);
}

printf("\n");

for (int i = 0; i < 10; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 10; j++)
    {
        printf(" %c ", board_origin[i][j]);
    }

    printf("\n");
}

```

```

    }

printf("\n-----\n\n");

system("cls"); // 답지 날리기

while (1)
{
    printf("   ");

    for (int j = 1; j <= 10; j++)
    {
        printf("| %2d ", j);
    }
    printf("\n");

    for (int i = 0; i < 10; i++)
    {
        printf(" %2d| ", i + 1);

        for (int j = 0; j < 10; j++)
        {
            printf(" %c ", board_select[i][j]);
        }
        printf("\n");
    }

    if (flag_count == 10)
    {
        for (int i = 0; i < 10; i++)
        {
            for (int j = 0; j < 10; j++)
            {
                if (board_select[i][j] == 'P')
                {
                    if (board_origin[i][j] != "B")
                    {

```

```

        break;
    }
}
}

printf("축하합니다. \n모든 지뢰의 위치를 알아냈습니다.\n");
printf(" ");

for (int j = 1; j <= 10; j++)
{
    printf("| %2d| ", j);
}
printf("\n");
for (int i = 0; i < 10; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 10; j++)
    {
        printf(" %c ", board_origin[i][j]);
    }
    printf("\n");
}
break;
}

printf("깃발을 놓을지 빈칸을 뚫을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4.
답지 보기): ");
scanf("%d", &K);

if (K == 1)
{
    if (flag_count == 10)
    {
        printf("깃발의 개수가 모자랍니다.\n");
        continue;
    }
}

```

```

printf("가로축을 입력하세요: ");
scanf("%d", &x);
printf("세로축을 입력하세요: ");
scanf("%d", &y);

if ((x < 1 || x > 10) || (y < 1 || y > 10))
{
    printf("1부터 10사이의 값을 입력하세요.\n");
    continue;
}
x--;
y--;

if (board_select[y][x] >= 48 && board_select[y][x] <= 57 || board_select[y][x]
== 'P')
{
    printf("다시 선택하세요.\n");
}

else
{
    board_select[y][x] = 'P';
    flag_count++;
}
}

if (K == 2)
{
    printf("가로축을 입력하세요: ");
    scanf("%d", &x);
    printf("세로축을 입력하세요: ");
    scanf("%d", &y);

    if ((x < 1 || x > 10) || (y < 1 || y > 10))
    {
        printf("1부터 10사이의 값을 입력하세요.\n");
        continue;
    }

```



```

x--;
y--;

if (board_select[y][x] >= 48 && board_select[y][x] <= 57 || board_select[y][x]
!= 'P')
{
    printf("다시 선택하세요.\n");
}

else
{
    board_select[y][x] = 'o';
    flag_count--;
}
}

else if (K == 3)
{
    printf("가로축을 입력하세요: ");
    scanf("%d", &x);
    printf("세로축을 입력하세요: ");
    scanf("%d", &y);

    if ((x < 1 || x > 10) || (y < 1 || y > 10))
    {
        printf("1부터 10사이의 값을 입력하세요.\n");
        continue;
    }
    x--;
    y--;

    if ((board_select[y][x] >= 48 && board_select[y][x] <= 57) ||
board_select[y][x] == 'P' || (board_select[y][x] >= 0 && board_select[y][x] <= 8))
    {
        printf("다시 선택하세요.\n");
    }

    else if (board_origin[y][x] == 'B') // 폭탄이 있는 칸을 누를 경우

```

```

{
    printf("폭탄을 선택했습니다.\n");
    printf("    ");

    for (int j = 1; j <= 10; j++)
    {
        printf("| %2d| ", j);
    }
    printf("\n");

    for (int i = 0; i < 10; i++)
    {
        printf(" %2d| ", i + 1);

        for (int j = 0; j < 10; j++)
        {
            printf(" %c  ", board_origin[i][j]);
        }
        printf("\n");
    }
    break;
}

else // 연쇄적으로 빈칸 뚫기
{
    begin_reveal_empty(board_origin, board_select, x, y);
}

system("cls"); // 맵 새로 그리기
}

else if (K == 4)
{
    printf("    ");

    for (int j = 1; j <= 10; j++)
    {
        printf("| %2d| ", j);
    }
}

```

```

printf("\n");

for (int i = 0; i < 10; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 10; j++)
    {
        printf(" %c ", board_origin[i][j]);
    }
    printf("\n");
}
printf("\n");
}
}

void intermediate_level() // 18 * 18 --- 40
{
    char board_origin[18][18] = { NULL, }; // 랜덤적인 위치에 폭탄이 설치되어있는 board
    char board_select[18][18] = { NULL, }; // 사용자가 원하는 기능을 수행하는 board
    int x, y; // y = 열, x = 행
    int K; // 깃발 or 깃발 제거 or 빈칸 선택 or 답지 확인 변수
    int sum = 48; // 아스키코드; 0
    int mine_sum = 0; // 폭탄 개수
    int flag_count = 0; // 깃발 개수
    int limit = 0;
    srand(time(NULL)); // 배열 안에 랜덤으로 폭탄 설치

    for (int i = 0; i < 18; i++)
    {
        for (int j = 0; j < 18; j++)
        {
            board_origin[i][j] = 'o';
        }
    }
    for (int i = 0; i < 18; i++)
    {

```

```

        for (int j = 0; j < 18; j++)
        {
            board_select[i][j] = 'o';
        }
    }

while (mine_sum < 40)
{
    x = rand() % 18;
    y = rand() % 18;

    if (board_origin[y][x] == 'B')
    {
        continue;
    }

    else
    {
        board_origin[y][x] = 'B';
        mine_sum++;
    }
}

for (int p = 0; p < 18; p++)
{
    for (int q = 0; q < 18; q++)
    {
        for (int i = 1; i >= -1; i--)
        {
            for (int j = 1; j >= -1; j--)
            {
                if (board_origin[p - i][q - j] == 'B' && q - j < 18 && q - j > -1 &&
board_origin[p][q] != 'B')
                {
                    sum++;
                }
            }
        }
    }
}

```

```

        if (board_origin[p][q] != 'B')
        {
            board_origin[p][q] = sum;
        }
        sum = 48;
    }
}
printf("  ");

for (int j = 1; j <= 18; j++)
{
    printf("| %2d| ", j);
}
printf("\n");

for (int i = 0; i < 18; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 18; j++)
    {
        printf(" %c ", board_origin[i][j]);
    }
    printf("\n");
}

printf("\n-----
-----\n\n");

system("cls"); // 답지 날리기

while (1)
{
    printf("  ");

    for (int j = 1; j <= 18; j++)
    {
        printf("| %2d ", j);
    }
}

```

```

}

printf("\n");

for (int i = 0; i < 18; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 18; j++)
    {
        printf(" %c ", board_select[i][j]);
    }
    printf("\n");
}

if (flag_count == 40)
{
    for (int i = 0; i < 18; i++)
    {
        for (int j = 0; j < 18; j++)
        {
            if (board_select[i][j] == 'P')
            {
                if (board_origin[i][j] != "B")
                {
                    break;
                }
            }
        }
    }

    printf("          축하합니다.          \n모든 지뢰의 위치를 알아냈습니다.\n");
    printf("          ");

    for (int j = 1; j <= 18; j++)
    {
        printf("| %2d| ", j);
    }
    printf("\n");
}

```

```

for (int i = 0; i < 18; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 18; j++)
    {
        printf(" %c  ", board_origin[i][j]);
    }
    printf("\n");
}
break;
}

```

printf("깃발을 놓을지 빈칸을 뚫을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4.
답지 보기): ");

```
scanf("%d", &K);
```

```

if (K == 1)
{
    if (flag_count == 40)
    {
        printf("깃발의 개수가 모자랍니다.\n");
        continue;
    }
}

```

```
printf("가로축을 입력하세요: ");
```

```
scanf("%d", &x);
```

```
printf("세로축을 입력하세요: ");
```

```
scanf("%d", &y);
```

```
if ((x < 1 || x > 18) || (y < 1 || y > 18))
```

```

{
    printf("1부터 18사이의 값을 입력하세요.\n");
    continue;
}

```

```
x--;
```

```
y--;
```

```

        if (board_select[y][x] >= 48 && board_select[y][x] <= 57 || board_select[y][x]
== 'P')
        {
            printf("다시 선택하세요.\n");
        }

        else
        {
            board_select[y][x] = 'P';
            flag_count++;
        }
    }

    if (K == 2)
    {
        printf("가로축을 입력하세요: ");
        scanf("%d", &x);
        printf("세로축을 입력하세요: ");
        scanf("%d", &y);

        if ((x < 1 || x > 18) || (y < 1 || y > 18))
        {
            printf("1부터 10사이의 값을 입력하세요.\n");
            continue;
        }
        x--;
        y--;

        if (board_select[y][x] >= 48 && board_select[y][x] <= 57 || board_select[y][x]
!= 'P')
        {
            printf("다시 선택하세요.\n");
        }

        else
        {
            board_select[y][x] = 'o';
            flag_count--;

```



```

    }
}

else if (K == 3)
{
    printf("가로축을 입력하세요: ");
    scanf("%d", &x);
    printf("세로축을 입력하세요: ");
    scanf("%d", &y);

    if ((x < 1 || x > 18) || (y < 1 || y > 18))
    {
        printf("1부터 18사이의 값을 입력하세요.\n");
        continue;
    }
    x--;
    y--;

    if ((board_select[y][x] >= 48 && board_select[y][x] <= 57) ||
board_select[y][x] == 'P' || (board_select[y][x] >= 0 && board_select[y][x] <= 8))
    {
        printf("다시 선택하세요.\n");
    }

    else if (board_origin[y][x] == 'B') // 폭탄이 있는 칸을 누를 경우
    {
        printf("폭탄을 선택했습니다.\n");
        printf("    ");

        for (int j = 1; j <= 18; j++)
        {
            printf("| %2d| ", j);
        }

        printf("\n");

        for (int i = 0; i < 18; i++)
        {
            printf(" %2d| ", i + 1);

```

```

        for (int j = 0; j < 18; j++)
        {
            printf(" %c ", board_origin[i][j]);
        }
        printf("\n");
    }
    break;
}

else // 연쇄적으로 빈칸 뚫기
{
    mid_reveal_empty(board_origin, board_select, x, y);
}

system("cls"); // 맵 새로 그리기
}

else if (K == 4)
{
    printf(" ");

    for (int j = 1; j <= 18; j++)
    {
        printf("| %2d| ", j);
    }
    printf("\n");

    for (int i = 0; i < 18; i++)
    {
        printf(" %2d| ", i + 1);

        for (int j = 0; j < 18; j++)
        {
            printf(" %c ", board_origin[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}

```

```

    }
}
}

void high_level() // 24 * 24 --- 99
{
    char board_origin[24][24] = { NULL, }; // 랜덤적인 위치에 폭탄이 설치되어있는 board
    char board_select[24][24] = { NULL, }; // 사용자가 원하는 기능을 수행하는 board
    int x, y;      // y = 열, x = 행
    int K;         // 밧발 or 밧발 제거 or 빈칸 선택 or 답지 확인 변수
    int sum = 48; // 아스키코드; 0
    int mine_sum = 0; // 폭탄 개수
    int flag_count = 0; // 밧발 개수
    int limit = 0;
    srand(time(NULL)); // 배열 안에 랜덤으로 폭탄 설치

    for (int i = 0; i < 24; i++)
    {
        for (int j = 0; j < 24; j++)
        {
            board_origin[i][j] = 'o';
        }
    }
    for (int i = 0; i < 24; i++)
    {
        for (int j = 0; j < 24; j++)
        {
            board_select[i][j] = 'o';
        }
    }

    while (mine_sum < 99)
    {
        x = rand() % 24;
        y = rand() % 24;

        if (board_origin[y][x] == 'B')
        {

```

```

        continue;
    }

    else
    {
        board_origin[y][x] = 'B';
        mine_sum++;
    }
}

for (int p = 0; p < 24; p++)
{
    for (int q = 0; q < 24; q++)
    {
        for (int i = 1; i >= -1; i--)
        {
            for (int j = 1; j >= -1; j--)
            {
                if (board_origin[p - i][q - j] == 'B' && q - j < 24 && q - j > -1 &&
board_origin[p][q] != 'B')
                {
                    sum++;
                }
            }
        }
        if (board_origin[p][q] != 'B')
        {
            board_origin[p][q] = sum;
        }
        sum = 48;
    }
}

printf("  ");

for (int j = 1; j <= 24; j++)
{
    printf("| %2d|", j);
}

```

```

printf("\n");

for (int i = 0; i < 24; i++)
{
    printf(" %2d| ", i + 1);

    for (int j = 0; j < 24; j++)
    {
        printf(" %c ", board_origin[i][j]);
    }
    printf("\n");
}

printf("\n-----
-----\n\n");

system("cls"); // 답지 날리기

while (1)
{
    printf(" ");

    for (int j = 1; j <= 24; j++)
    {
        printf("| %2d ", j);
    }
    printf("\n");

    for (int i = 0; i < 24; i++)
    {
        printf(" %2d| ", i + 1);

        for (int j = 0; j < 24; j++)
        {
            printf(" %c ", board_select[i][j]);
        }
        printf("\n");
    }
}

```

```

if (flag_count == 99)
{
    for (int i = 0; i < 24; i++)
    {
        for (int j = 0; j < 24; j++)
        {
            if (board_select[i][j] == 'P')
            {
                if (board_origin[i][j] != "B")
                {
                    break;
                }
            }
        }
    }
    printf("          축하합니다.          \n모든 지뢰의 위치를 알아냈습니다.\n");
    printf("          ");

    for (int j = 1; j <= 24; j++)
    {
        printf("| %2d|", j);
    }
    printf("\n");

    for (int i = 0; i < 24; i++)
    {
        printf(" %2d|", i + 1);

        for (int j = 0; j < 24; j++)
        {
            printf(" %c ", board_origin[i][j]);
        }
        printf("\n");
    }
    break;
}

printf("깃발을 놓을지 빈칸을 뚫을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4.

```

```

    답지 보기): ");
    scanf("%d", &K);

    if (K == 1)
    {
        if (flag_count == 99)
        {
            printf("깃발의 개수가 모자랍니다.\n");
            continue;
        }

        printf("가로축을 입력하세요: ");
        scanf("%d", &x);
        printf("세로축을 입력하세요: ");
        scanf("%d", &y);

        if ((x < 1 || x > 24) || (y < 1 || y > 24))
        {
            printf("1부터 24사이의 값을 입력하세요.\n");
            continue;
        }
        x--;
        y--;

        if (board_select[y][x] >= 48 && board_select[y][x] <= 57 || board_select[y][x]
== 'P')
        {
            printf("다시 선택하세요.\n");
        }

        else
        {
            board_select[y][x] = 'P';
            flag_count++;
        }
    }

    if (K == 2)

```

```

{
    printf("가로축을 입력하세요: ");
    scanf("%d", &x);
    printf("세로축을 입력하세요: ");
    scanf("%d", &y);

    if ((x < 1 || x > 18) || (y < 1 || y > 24))
    {
        printf("1부터 10사이의 값을 입력하세요.\n");
        continue;
    }
    x--;
    y--;

    if (board_select[y][x] >= 48 && board_select[y][x] <= 57 || board_select[y][x]
!= 'P')
    {
        printf("다시 선택하세요.\n");
    }

    else
    {
        board_select[y][x] = 'o';
        flag_count--;
    }
}

else if (K == 3)
{
    printf("가로축을 입력하세요: ");
    scanf("%d", &x);
    printf("세로축을 입력하세요: ");
    scanf("%d", &y);

    if ((x < 1 || x > 24) || (y < 1 || y > 24))
    {
        printf("1부터 24사이의 값을 입력하세요.\n");
        continue;
    }

```



```

    }

    x--;
    y--;

    if ((board_select[y][x] >= 48 && board_select[y][x] <= 57) ||
        board_select[y][x] == 'P' || (board_select[y][x] >= 0 && board_select[y][x] <= 8))
    {
        printf("다시 선택하세요.\n");
    }

    else if (board_origin[y][x] == 'B') // 폭탄이 있는 칸을 누를 경우
    {
        printf("폭탄을 선택했습니다.\n");
        printf("  ");

        for (int j = 1; j <= 24; j++)
        {
            printf("| %2d| ", j);
        }
        printf("\n");

        for (int i = 0; i < 24; i++)
        {
            printf(" %2d| ", i + 1);

            for (int j = 0; j < 24; j++)
            {
                printf(" %c ", board_origin[i][j]);
            }
            printf("\n");
        }
        break;
    }

    else // 연쇄적으로 빈칸 뚫기
    {
        hard_reveal_empty(board_origin, board_select, x, y);
    }

```

```

        system("cls"); // 맵 새로 그리기
    }

    else if (K == 4)
    {
        printf("    ");

        for (int j = 1; j <= 24; j++)
        {
            printf("| %2d| ", j);

        }
        printf("\n");

        for (int i = 0; i < 24; i++)
        {
            printf(" %2d| ", i + 1);

            for (int j = 0; j < 24; j++)
            {
                printf(" %c  ", board_origin[i][j]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

int main()
{
    printf("[지뢰찾기 게임]\n");
    printf("[C프로그래밍응용]\n");
    printf("[팀명: 디버깅 오류]\n");
    printf("[팀원 학과, 학번, 이름, 역할]\n");
    printf("[컴퓨터공학과, 김도현, 20230817, 알고리즘 작성]\n");
    printf("[컴퓨터공학과, 김민철, 20230822, 팀장]\n");
    printf("[컴퓨터공학과, 김형호, 20230837, 레포트 작성]\n");
    printf("[컴퓨터공학과, 박성준, 20230842, 코드 작성]\n");
}

```

```

printf("\n\n");

int level;

printf("난이도를 선택하세요.(1. 초급 // 2. 중급 // 3. 고급): ");
scanf("%d", &level);

switch (level)
{
case 1:
    beginner_level();
    break;

case 2:
    intermediate_level();
    break;

case 3:
    high_level();
    break;

default:
    printf("올바른 숫자를 선택하세요.\n");
}

return 0;
}

```

V. 실행결과

난이도 '초급'으로 실행시킨 결과이다.

난이도를 선택하세요.(1. 초급 // 2. 중급 // 3. 고급): 1

```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
2| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
3| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
4| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
5| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
6| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
7| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
8| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
9| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
10| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
깃발을 놓을지 빈칸을 뿔을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기):

```

1. 깃발 놓기

```

깃발을 놓을지 빈칸을 뿔을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기): 1
가로축을 입력하세요: 4
세로축을 입력하세요: 3
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
2| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
3| 0 | 0 | 0 | P | 0 | 0 | 0 | 0 | 0 | 0 |
4| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
5| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
6| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
7| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
8| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
9| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
10| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
깃발을 놓을지 빈칸을 뿔을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기):

```

2. 깃발 취소

```

깃발을 놓을지 빈칸을 뿔을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기): 2
가로축을 입력하세요: 4
세로축을 입력하세요: 3
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
2| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
3| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
4| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
5| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
6| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
7| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
8| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
9| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
10| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
깃발을 놓을지 빈칸을 뿔을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기):

```

3. 빈칸

```

 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0|
2 | 0 | 1 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0|
3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | 0|
4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0|
5 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0|
6 | 0 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0|
7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
10| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
깃발을 놓을지 빈칸을 뚫을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기):

```

4. 답지 보기

```

깃발을 놓을지 빈칸을 뚫을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기): 4
 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1 | 0 | 1 | 2 | B | 1 | 0 | 0 | 0 | 1 | B|
2 | 0 | 1 | B | 2 | 1 | 0 | 0 | 0 | 1 | 1|
3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | 1|
4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | B | B | 1|
5 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 1|
6 | B | 2 | 1 | 1 | 1 | B | 1 | 1 | 1 | 1|
7 | 1 | 2 | B | 1 | 1 | 1 | 1 | 1 | B | 1|
8 | 0 | 1 | 2 | 2 | 1 | 0 | 0 | 1 | 1 | 1|
9 | 0 | 0 | 1 | B | 1 | 0 | 0 | 0 | 0 | 0|
10| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0|

 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0|
2 | 0 | 1 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0|
3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | 0|
4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0|
5 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0|
6 | 0 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0|
7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
10| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0|
깃발을 놓을지 빈칸을 뚫을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기):

```

5. 게임 클리어

정답 보드판을 보여주며 프로그램이 종료된다.

```

 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1 | 0 | 1 | 2 | P | 1 | 0 | 0 | 0 | 1 | P|
2 | 0 | 1 | P | 2 | 1 | 0 | 0 | 0 | 1 | 1|
3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | 1|
4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | P | P | 1|
5 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 1|
6 | P | 2 | 1 | 1 | 1 | P | 1 | 1 | 1 | 1|
7 | 0 | 0 | P | 1 | 1 | 1 | 1 | 1 | P | 1|
8 | 0 | 0 | 2 | 2 | 1 | 0 | 0 | 1 | 1 | 1|
9 | 0 | 0 | 1 | P | 1 | 0 | 0 | 0 | 0 | 0|
10| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0|
축하합니다
모든 지뢰의 위치를 알아냈습니다
 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|
1 | 0 | 1 | 2 | B | 1 | 0 | 0 | 0 | 1 | B|
2 | 0 | 1 | B | 2 | 1 | 0 | 0 | 0 | 1 | 1|
3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | 1|
4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | B | B | 1|
5 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 1|
6 | B | 2 | 1 | 1 | 1 | B | 1 | 1 | 1 | 1|
7 | 1 | 2 | B | 1 | 1 | 1 | 1 | 1 | B | 1|
8 | 0 | 1 | 2 | 2 | 1 | 0 | 0 | 1 | 1 | 1|
9 | 0 | 0 | 1 | B | 1 | 0 | 0 | 0 | 0 | 0|
10| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0|
C:\Users\User\source\repos\Project5\64\Debug\Pro
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

6. 폭탄 선택 시

정답 보드판을 보여주며 게임이 종료된다.

```

깃발을 놓을지 빈칸을 놓을지 선택하세요.(1. 깃발 // 2. 깃발 취소 // 3. 빈칸 // 4. 답지 보기): 3
가로축을 입력하세요: 1
세로축을 입력하세요: 1
폭탄을 선택했습니다.
 1| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|
1| B| 1| 0| 0| 0| 1| B| 1| 0| 0|
2| 1| 1| 0| 0| 0| 1| 1| 1| 0| 0|
3| 1| 1| 1| 1| 1| 2| 1| 1| 0| 0|
4| 1| B| 1| 1| B| 2| B| 1| 0| 0|
5| 1| 1| 2| 2| 2| 2| 1| 1| 0| 0|
6| 0| 0| 1| B| 1| 0| 0| 0| 0| 0|
7| 0| 0| 2| 2| 3| 1| 1| 0| 0| 0|
8| 0| 0| 2| B| 4| B| 2| 0| 0| 0|
9| 0| 0| 2| B| 4| B| 2| 0| 0| 0|
10| 0| 0| 1| 1| 2| 1| 1| 0| 0| 0|

C:\Users\User\source\repos\Project5\64\Debug\Project5.exe(프로세스 17904개)이(가) 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
  
```

2. 개인정보 관리 프로그램

I. 기획 의도

메뉴가 들어가는 자유 주제의 프로그램을 만들기 전에 회의를 통해서 의견을 수렴하였다. 개인정보 유출이 만연한 사회에서 정보 보호는 중요한 일이기 때문에 비밀번호 입력 기능을 통해 개인정보를 관리할 수 있는 프로그램을 개발하기로 하였다. 큰 주제를 정한 후 프로그램에 추가할 메뉴와 기능들을 정리하였다.

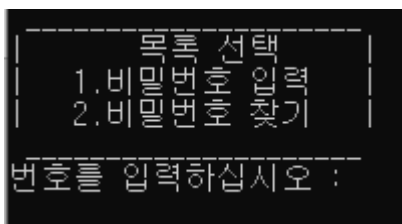
- 1. 추가 2. 삭제 3. 수정 4. 삭제 메뉴를 추가한다.
- 5. 종료 기능을 추가하여 프로그램을 종료한다.
- 각 기능을 수행 후에 N초 동안 수행 결과를 보여주고 메뉴로 돌아간다.
- 프로그램 첫 실행이라면 비밀번호를 생성한다.
- 이후 비밀번호를 입력 시 메뉴로 들어간다.
- 비밀번호를 틀리면 프로그램이 종료된다.

크게 6가지의 기능으로 정리하였다. 이후 프로그램 개발과 회의를 통해서 부족하다고 생각된 기능들을 추가하였다.

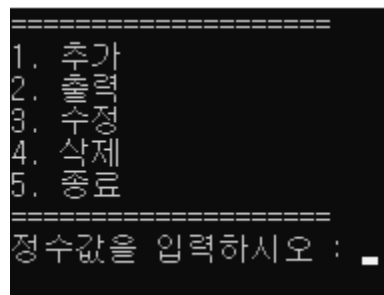
비밀번호를 까먹었을 때를 대비하여 비밀번호 생성 시 비밀번호 힌트를 입력한다. 비밀번호 입력 시 비밀번호 찾기 기능을 추가하여 힌트를 보여준다. 수정과 삭제를 바로 하지 않고 해당 파일이 맞는지 확인하는 절차를 추가한다.

위 기능들을 추가하여서 프로그램 사용 시 실수로 수정, 삭제했을 때 다시 추가해야 하는 번거로움을 해결하였고 비밀번호를 까먹었을 때 아예 개인정보에 접촉할 수 없던 문제를 해결하였다.

비밀번호 입력 출력 화면



메인 메뉴 출력 화면



II. 알고리즘

위에서 소개한 카카오톡, 디스코드 등의 협업 도구를 사용하여 회의를 진행하여 코드의 기능을 결정하고 자연어로 먼저 알고리즘을 작성 후 순서도로 작성하여 기능들을 정리하였다.

1. 자연어

1. 프로그램 시작.

2. 처음인지 입력받는다.

2-1. 처음을 선택하면 비밀번호와 힌트를 새로 저장한다.

2-2 처음이 아니면 저장된 비밀번호와 힌트를 사용한다.

3. 목록을 선택한다.

3-1. 비밀번호를 입력(1)하고 틀리면 종료한다.

3-2. 비밀번호를 찾기(2)를 선택하면 힌트를 3초 동안 보여주고 3번으로 돌아간다.

4. 1부터 5까지의 값을 받는다.

4-1. 추가(1)를 선택하면 이름, 학교, 학과, 학번, 이메일, 전화번호, 혈액형, 국적을 입력받고 파일에 저장한다.

4-2. 출력(2)을 선택하고 이름과 학번을 저장값과 똑같이 입력하면 정보를 10초 동안 보여주고 4번으로 돌아간다.

4-3. 출력(2)에서 잘못된 값을 입력하면 파일을 찾을 수 없다고 10초 동안 보여주고 4번으로 돌아간다.

4-4. 수정(3)을 선택하고 이름과 학번을 저장값과 똑같이 입력하면 정보를 보여주고 파일이 맞는지 확인한다.

4-4-1. 파일이 맞으면 수정값을 입력하고 파일을 수정해 4번으로 돌아간다.

4-4-2. 파일이 아니면 수정하지 않고 4번으로 돌아간다.

4-5. 삭제(4)를 선택하고 이름과 학번을 저장값과 똑같이 입력하면 정보를 보여주고 파일이 맞는지 확인한다.

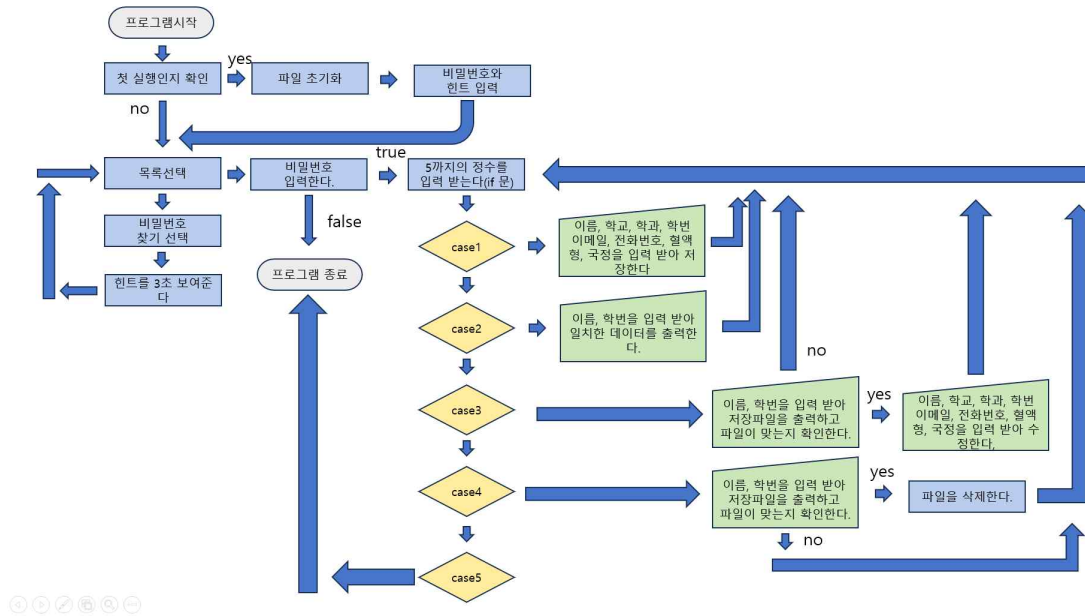
4-5-1. 파일이 맞으면 파일을 삭제하고 4번으로 돌아간다.

4-5-2. 파일이 아니면 삭제하지 않고 4번으로 돌아간다.

4-6. 종료(5)를 선택하면 프로그램을 종료한다.

2. 순서도

순서도는 순서도 체계 시스템을 이용해서 자연어 알고리즘을 먼저 완성한 후에 자연어를 그림으로 나타내는 방식으로 진행하였다.



위와 같이 자연어와 순서도로 프로젝트를 진행하는 팀원 모두가 이해하기 쉽게 정리하고 프로젝트 진행 중에 기억이 안 나가나 애매했던 부분은 자연어, 순서도 알고리즘을 확인하면서 명확한 프로그램을 개발할 수 있었다.

III. C언어 코드

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>

void clearBuffer(void)
{
    while (getchar() != '\n');    //버퍼비우기
}

int search_file() //빈 파일 검색
{
    static char file_name[100] = { NULL, };
    char person_information[1000] = { 0, };
    char clear_sentence[10000] = { 0, };
    int i = 0;

    for (i = 48; i < 53; i++)
    {
        char file_num[2] = { (char)i, 0 };
        strcat(file_name, "text");
        strcat(file_name, file_num);
        strcat(file_name, ".txt");

        FILE* fp = fopen(file_name, "r");
        fgets(person_information, sizeof(person_information), fp);
        fclose(fp);

        if (strcmp(person_information, clear_sentence) == 0)
            break;

        strcpy(file_name, clear_sentence);
        strcpy(person_information, clear_sentence);
    }
}
```

```

strcpy(person_information, clear_sentence);
strcpy(file_name, clear_sentence);

i -= 48;

return i;
}

int searching_person() //사람 찾기
{
    int i = 0;
    char search_name[100] = { 0, };
    char search_ID[100] = { 0, };
    char file[10000] = { 0, };
    char file_name[10000] = { 0, };

    char search_name_perfect[10000] = { 0, };
    char search_ID_perfect[10000] = { 0, };

    //찾아야할 사람 정보 입력받기
    printf("이름과 학번을 입력해주세요.\n");
    clearBuffer();
    printf("이름:");
    scanf("%s", search_name);
    clearBuffer();
    printf("학번:");
    scanf("%s", search_ID);
    printf("Enter키를 눌러주세요.\n");
    clearBuffer();
    system("cls");

    while (1)
    {
        char file_num[2] = { (char)(i + 48), };
        char clear_sentence[10000] = { 0, };

        strcat(file_name, "text");
        strcat(file_name, file_num);
    }
}

```

```

strcat(file_name, ".txt");

FILE* fp = fopen(file_name, "r");
fread(file, 10000, 1, fp);
fclose(fp);

//초성만 쳐도 탐색이 가능해지기에 정보와 이름을 연결하는 기능
strcat(search_name_perfect, "이름: ");
strcat(search_name_perfect, search_name);
strcat(search_name_perfect, "\n");

strcat(search_ID_perfect, "학번: ");
strcat(search_ID_perfect, search_ID);
strcat(search_ID_perfect, "\n");

//맞는지 안맞는지 검색하는 기능
if (strstr(file, search_name_perfect) && strstr(file, search_ID_perfect))
{
    printf("%s", file);
    strcpy(file_name, clear_sentence);
    strcpy(file, clear_sentence);
    break;
}

else if (i == 4)
{
    printf("파일을 찾지 못했습니다.\n");
    strcpy(file_name, clear_sentence);
    strcpy(file, clear_sentence);
    break;
}

//file 변수안에 값이 들어가있기에 값 삭제
strcpy(file, clear_sentence);
strcpy(file_name, clear_sentence);

i++;
}

return (int)i;

```

```
}
```

```
char* cat_info(char information[], char str[]) //항목과 값을 이어주는 함수
```

```
{  
    static char person_information[10000] = { 0, };  
    char clear_sentence[10000] = { 0, };  
    strcpy(person_information, clear_sentence);  
  
    strcat(person_information, str);  
    strcat(person_information, information);  
    strcat(person_information, "\n");  
  
    return person_information;  
}
```

```
void main_program(int neworold)
```

```
{  
    int num = 0;    //목록 선택 값  
    int count = 0;  //반복횟수 저장  
  
    char count_file[100] = { 0, }; //반복문에서의 FILE  
    char person_information[10000] = { 0, };  
    char information[100] = { 0, };  
    char clear_sentence[10000] = { 0, };  
    char answer = 0;  
    char file_name[100] = { NULL, };  
  
    if (neworold == 0)    //만약 새로 만들었다면 파일 모두 초기화  
        for (int i = 48; i < 53; i++)  
        {  
            char file_num[2] = { (char)i, 0 };  
            strcat(file_name, "text");  
            strcat(file_name, file_num);  
            strcat(file_name, ".txt");  
            FILE* fp = fopen(file_name, "w");  
            fclose(fp);  
            strcpy(file_name, clear_sentence);  
        }  
}
```

```

while (1)
{
    system("cls");

    //목록 선택
    for (int i = 0; i < 20; i++)
        printf("=");
    printf("\n");
    printf("1. 추가\n2. 출력\n3. 수정\n4. 삭제\n5. 종료\n");
    for (int i = 0; i < 20; i++)
        printf("=");
    printf("\n");

    printf("정수값을 입력하시오 : ");
    scanf("%d", &num);

    Sleep(1000);
    system("cls");

    //추가를 골랐을때
    if (num == 1)
    {
        int search_file_result = search_file();
        char search_file_function[2] = { (char)(search_file_result + 48), 0 }; //배열로 바
        꾸기

        strcat(count_file, "text");
        strcat(count_file, search_file_function);
        strcat(count_file, ".txt");

        if (search_file_result == 5)
            printf("더 이상 저장할 수 있는 자리가 없습니다.\n"); //5개 이상으로 추가하
            려고 할때 출력

        // 입력받기
    }
    else
    {

```

```

printf("=====\\n");

    strcat(person_information,

"=====\\n");

    printf("이름: ");
    scanf("%s", information);
    strcat(person_information, cat_info(information, "이름: "));

    clearBuffer();
    printf("학교: ");
    scanf("%[^\n]s", information);
    strcat(person_information, cat_info(information, "학교: "));

    clearBuffer();
    printf("학과: ");
    scanf("%[^\n]s", information);
    strcat(person_information, cat_info(information, "학과: "));

    clearBuffer();
    printf("학번: ");
    scanf("%[^\n]s", information);
    strcat(person_information, cat_info(information, "학번: "));

    clearBuffer();
    printf("이메일: ");
    scanf("%[^\n]s", information);
    strcat(person_information, cat_info(information, "이메일: "));

    clearBuffer();
    printf("전화번호: ");
    scanf("%[^\n]s", information);
    strcat(person_information, cat_info(information, "전화번호: "));

    clearBuffer();
    printf("혈액형: ");
    scanf("%[^\n]s", information);
    strcat(person_information, cat_info(information, "혈액형: "));

```

```

        clearBuffer();
        printf("국적: ");
        scanf("%[^\n]s", information);
        strcat(person_information, cat_info(information, "국적: "));

printf("=====\n");

        strcat(person_information,
"=====");

        FILE* fp = fopen(count_file, "w");
        fputs(person_information, fp);
        fclose(fp);
        printf("저장이 완료되었습니다.\n");

        strcpy(count_file, clear_sentence);
        strcpy(person_information, clear_sentence);
        strcpy(information, clear_sentence);
    }
    Sleep(2000);
}
else if (num == 2)
{
    searching_person();
    Sleep(10000);
}
else if (num == 3)
{
    int i = searching_person(); //찾은 파일 번호 받아오기
    if (i == 5)
    {
        printf("삭제를 종료합니다.\n");
        Sleep(5000);
    }
    else
    {
        printf("\n이 파일이 맞습니까?(y or n)");
        scanf("%c", &answer);
        Sleep(2000);
    }
}

```



```

system("cls");
if (answer == 'y')
{

printf("=====\n");

//파일 다시 입력받기 위해 1번 내용과 거의 동일할 예정
printf("이름: ");
scanf("%s", information);
strcat(person_information, cat_info(information, "이름: "));

clearBuffer();
printf("학교: ");
scanf("%[^\n]s", information);
strcat(person_information, cat_info(information, "학교: "));

clearBuffer();
printf("학과: ");
scanf("%[^\n]s", information);
strcat(person_information, cat_info(information, "학과: "));

clearBuffer();
printf("학번: ");
scanf("%[^\n]s", information);
strcat(person_information, cat_info(information, "학번: "));

clearBuffer();
printf("이메일: ");
scanf("%[^\n]s", information);
strcat(person_information, cat_info(information, "이메일: "));

clearBuffer();
printf("전화번호: ");
scanf("%[^\n]s", information);
strcat(person_information, cat_info(information, "전화번호: "));

clearBuffer();
printf("혈액형: ");
scanf("%[^\n]s", information);

```



```

        if (answer == 'y')
        {
            char search_file_function[2] = { (char)(i + 48), 0 };
            strcat(count_file, "text");
            strcat(count_file, search_file_function);
            strcat(count_file, ".txt");
            FILE* fp = fopen(count_file, "w");
            fclose(fp);
            printf("파일이 삭제되었습니다.\n");
            strcpy(count_file, clear_sentence);
            Sleep(5000);
        }
        else
        {
            printf("삭제를 취소합니다.\n");
            Sleep(5000);
        }
    }
}

else if (num == 5)
{
    //프로그램 종료 코드 아마도 break로 예상
    printf("프로그램을 종료하겠습니다.\n");
    break;
}
}

}

int main()
{
    char password[100] = { 0, }; //실제 비밀번호
    char get_password[100] = { 0, }; //입력 비밀번호
    char hint[100] = { 0, }; //힌트
    char answer; //처음인지 아닌지 구분
    int select = 0; //목록선택
    printf("처음이십니까?(y or n)"); //파일을 덮어 씌울지 아니면 새로 만들지 결정
    scanf("%c", &answer);

    int neworold = 0; //새로운 아이디인지 아닌지 판단 저장 변수

```

```

if (answer == 'y')
{
    neworold = 0;
    clearBuffer();
    printf("비밀번호를 설정하십시오 : ");
    fgets(password, sizeof(password), stdin);
    password[strcspn(password, "\n")] = '\0'; //password 입력받을때 띄어쓰기등을 삭제
    FILE* fp = fopen("password.txt", "w"); //password.txt를 있다면 초기화, 없다면 만들어
서 불러옴
    fputs(password, fp); //password.txt에 값 넣기
    fclose(fp); //파일닫기
    printf("비밀번호를 까먹었을시 나올 힌트를 말하십시오(30자 이내) : ");
    fgets(hint, sizeof(hint), stdin);
    hint[strcspn(hint, "\n")] = '\0'; //hint 입력받을때 띄어쓰기, 줄바꿈을 삭제
    fp = fopen("hint.txt", "w"); //hint.txt를 있다면 초기화, 없다면 만들어서 불러옴
    fputs(hint, fp); //hint.txt에 값 넣기
    fclose(fp); //파일닫기
    printf("비밀번호가 설정되었습니다.\n");
}
else
    neworold = 1;
while (1)
{
    system("cls"); //화면 지우기
    printf(" _____\n|          목록 선택          |\n|  1.비밀번호 입력  |\n|  2.비밀번호 찾기  |\n _____\n");
    printf("번호를 입력하십시오 : ");
    scanf("%d", &select);
    if (select == 1)
    {
        clearBuffer();
        printf("비밀번호를 입력하십시오 : ");
        fgets(get_password, sizeof(get_password), stdin); //hint 입력받을때 띄어쓰기,
줄바꿈을 삭제, stdin == 키보드 입력
        get_password[strcspn(get_password, "\n")] = '\0';
        break;
    }
    FILE* fp = fopen("hint.txt", "r");

```

```

    fgets(hint, sizeof(hint), fp);
    printf("%s\n", hint);
    fclose(fp);
    Sleep(3000); //3초동안 코드 실행 지연
}
FILE* fp = fopen("password.txt", "r");
fgets(password, sizeof(password), fp);
fclose(fp);
if (strcmp(password, get_password) == 0)
{
    printf("프로그램을 실행합니다.\n");
    main_program(neworold);
}
else
{
    printf("비밀번호가 틀렸습니다.\n프로그램을 종료합니다.");
}
return 0;
}

```

IV. 실행 결과

1. 첫 실행

```
처음이십니까?(y or n) _
```

2. y 입력

```
처음이십니까?(y or n)y  
비밀번호를 설정하십시오 : 123  
비밀번호를 까먹었을시 나올 힌트를 말하십시오(30자 이내) : 숫자
```

3. 힌트입력 or 1번에서 n 입력 시

```
-----  
|          목록 선택          |  
| 1.비밀번호 입력           |  
| 2.비밀번호 찾기           |  
|-----|  
번호를 입력하십시오 :
```

비밀번호 입력 실패

```
-----  
|          목록 선택          |  
| 1.비밀번호 입력           |  
| 2.비밀번호 찾기           |  
|-----|  
번호를 입력하십시오 : 1  
비밀번호를 입력하십시오 :
```

```
-----  
|          목록 선택          |  
| 1.비밀번호 입력           |  
| 2.비밀번호 찾기           |  
|-----|  
번호를 입력하십시오 : 1  
비밀번호를 입력하십시오 : 1233  
비밀번호가 틀렸습니다.  
프로그램을 종료합니다.
```

5. 메인 메뉴 화면

```
=====
1. 추가
2. 출력
3. 수정
4. 삭제
5. 종료
=====
정수값을 입력하십시오 : _
```

비밀번호 찾기

```
-----  
|          목록 선택          |  
| 1.비밀번호 입력           |  
| 2.비밀번호 찾기           |  
|-----|  
번호를 입력하십시오 : 2  
숫자
```

6. 추가 기능

```
이름: 홍길동  
학교: 성결대학교  
학과: 컴퓨터공학과  
학번: 20230801  
이메일: sku123@gmail.com  
전화번호: 010-1234-5678  
혈액형: A  
국적: 한국  
저장이 완료되었습니다.
```

7. 출력 기능

```
이름과 학번을 입력해주세요  
이름: 홍길동  
학번: 20230801 _
```

```
=====
이름: 홍길동  
학교: 성결대학교  
학과: 컴퓨터공학과  
학번: 20230801  
이메일: sku123@gmail.com  
전화번호: 010-1234-5678  
혈액형: A  
국적: 한국  
=====
```

8. 수정 기능

```
이름과 학번을 입력해주세요  
이름:홍길동  
학번:20230801_
```

```
=====
이름: 홍길동  
학교: 성결대학교  
학과: 컴퓨터공학과  
학번: 20230801  
이메일: sku123@gmail.com  
전화번호: 010-1234-5678  
혈액형: A  
국적: 한국  
=====
이 파일이 맞습니까?(y or n)
```

y 입력 시 1번 추가 기능과 동일하게 추가, n 입력 시 메뉴 화면으로 돌아감

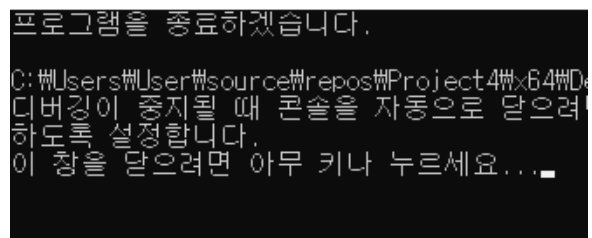
9. 삭제 기능

```
이름과 학번을 입력해주세요  
이름:홍길동  
학번:20230801_
```

```
=====
이름: 홍길동  
학교: 성결대학교  
학과: 컴퓨터공학과  
학번: 20230801  
이메일: sku123@gmail.com  
전화번호: 010-1234-5678  
혈액형: A  
국적: 한국  
=====
이 파일이 맞습니까?(y or n)
```

```
=====
이름: 홍길동  
학교: 성결대학교  
학과: 컴퓨터공학과  
학번: 20230801  
이메일: sku123@gmail.com  
전화번호: 010-1234-5678  
혈액형: A  
국적: 한국  
=====
이 파일이 맞습니까?(y or n)y  
파일이 삭제되었습니다.
```

10. 종료



```
프로그램을 종료하겠습니다.  
C:\Users\User\source\repos\Project4\64\Debug\Project4.exe  
디버깅이 중지될 때 콘솔을 자동으로 닫으려  
하도록 설정합니다.  
이 창을 닫으려면 아무 키나 누르세요...■
```