

## 1. 보물찾기게임 프로세스 설명

5X5 행렬의 각 배열에 0~8의 랜덤한 값을 저장하고 player는 [2][2], 중앙부터 시작해

1:상 2:하 3:좌 4:우 case로 랜덤하게 이동하여 한번 이동할 때 마다 이동한지점에 X를 출력하고

num : 시행횟수 point : 랜덤으로 이동한 곳의 배열에 저장된 값을 합한 값 direction : 이동한방향 또한 동시에 출력한다.

이 때, 합산한 점수가 10점이 넘으면 player의 승리.

시행횟수가 6회이상이거나 랜덤한 이동중 5X5 행렬의 가장자리를 넘어가는 이동 발생 시 패배.

## 2. 코드설명

### 1) 주어진 코드 설명

```
void mapCreate(int map[][Depth])
{
    srand(time(NULL));
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (rand() % 3)
                map[i][j] = rand() % 9;
        }
    }
    map[2][2] = 9;
}
```

mapCreate 함수의 경우 [0][0] ~ [4][4]의 5X5 행렬을 만들기 위해 for문을 사용했고

각 배열에 0~8까지의 랜덤한 값을 부여하기위해 rand함수를 이용(vareity를 위해 if(rand() % 3)기  
용). 이때 [2][2]는, 게임 룰이 중앙에서 시작이므로 9를 고정값으로 저장.

---

```
void mapPrint(int map[][Depth])
{
    system("cls");
    printf("-----\n");
    for (int i = 0; i < 5; i++)
```

```

{
    for (int j = 0; j < 5; j++)
    {
        if (map[i][j] == 9)
            printf("X ");
        else
            printf("%d ", map[i][j]);
    }
    printf("\n");

    printf("-----\n");
    printf("방향 : %d \n", direction);
    printf("점수 : %d \n", point);
    printf("횟수 : %d \n", num);
    printf("-----\n");
    Sleep(2000);
}

```

mapPrint 함수의 경우 마찬가지로 for문을 이용해 행렬을 생성했으나 mapCreate와는 다르게 실제로 게임진행 포메이션(5X5행렬, direction을 매개변수로한 방향, point를 매개변수로한 점수, num을 매개변수로한 횟수)을 화면에 출력하는 기능까지 수행. 특이사항으로 랜덤으로 이동한경로에 이동 흔적을 남기기 위해 9의값을 저장하는데, 배열값이 9인 경우 그 자리에 문자X를 출력. Sleep(2000)으로 수행횟수 증가시 지연시간 추가.

---

```

void looseGame()
{
    system("cls");
    printf("-----\n");
    printf("게임에서 패배하셨습니다.\n");
    printf("총 점수 : %d\n", point);
    printf("총 횟수 : %d\n", num);
    printf("-----\n");
    exit(0);
}

```

looseGmae 함수의 경우 게임을진행하며 획득한 point를 매개변수로한 점수, num을 매개변수로한 시행횟수를 출력하며 패배를 알림

---

```

void winGame()
{
    system("cls");
    printf("-----\n");
    printf("게임에서 승리하셨습니다.\n");
    printf("총 점수 : %d\n", point);
    printf("총 횟수 : %d\n", num);
    printf("-----\n");
    exit(0);
}

```

winGame 함수의 경우 게임을 진행하며 얻은 point를 매개변수로한 점수와 num을 매개변수로한 시행횟수를 출력하며 승리를 알림.

## 2) 직접구현한 코드 설명

playGame함수 부분)

```
direction = rand() % 4 + 1;
```

방향을 [2,2] 시작점으로부터 1상 2하 3좌 4우 랜덤 이동 하기위한 direction에 rand함수로 1~4값을 대입연산(기본 rand() % 4 포맷은 0~3까지의 4개 숫자중 랜덤출력이므로 +1을 해주어 범위를 1~4로 늘려줌)

---

```
switch (direction)
```

```

{
    case 1:
    {
        Point_y -= 1;
        num++;

        if (map[Point_y][Point_x] != 9)
        {
            point += map[Point_y][Point_x];
            map[Point_y][Point_x] = 9;
            mapPrint(map);
            break;
        }
        else

```

```

        {
            mapPrint(map);
            break;
        }
    }
}

```

switch문으로 랜덤하게 대입된 direction값을 취사선택해 case 1~4에 따라 해당지점으로 이동하고, 해당 배열의 해당값을 point에 합산한 후 배열에 9값을 저장(이동경로에 X출력을 위함)

---

checkLoose 부분)

```

void checkLoose(int Point_x, int Point_y)
{
    if (num > 5)
    {
        looseGame();
    }
    else if (Point_x == 0 && direction == 3)
    {
        looseGame();
    }
}

```

시행횟수가 5회를 초과할 경우, 또 Point\_x= 0, 즉 1열일때 direction3인 좌측으로 넘어가면 패배 (direction 1,2,4의 경우도 이하동문).

---

checkWin 부분)

```

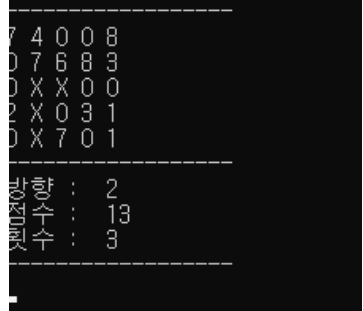
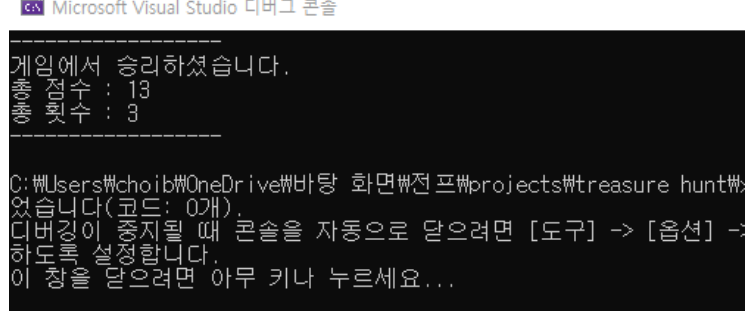
void checkWin(int Point_x, int Point_y)
{
    if (point > 9)
    {
        winGame();
    }
}

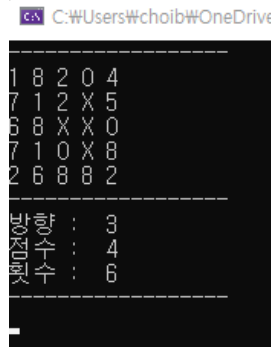
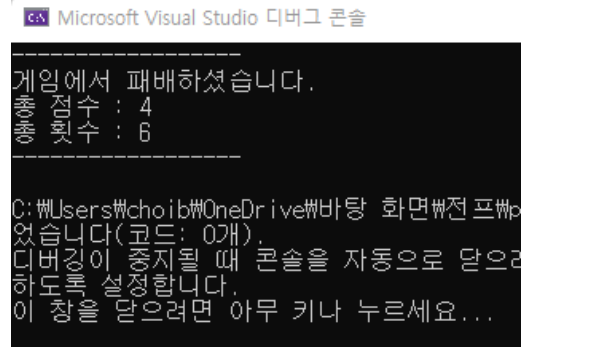
```

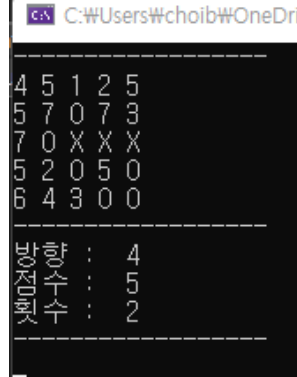
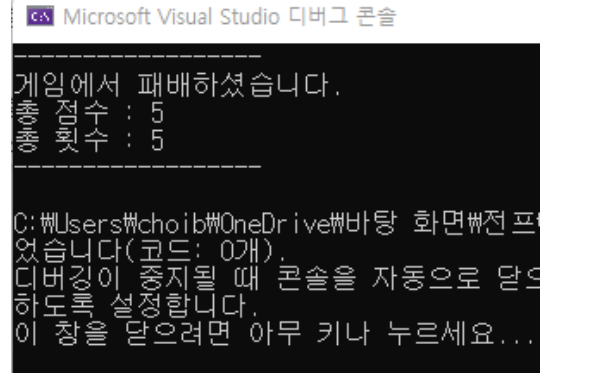
playGame함수에서 loosegame을 항상 먼저 체크하므로 if - else의 개념으로 checkWin함수에서는 시행횟수 num은 고려하지 않고 포인트10점이상일시 승리.

특이사항으로 전역변수 point는 checkWin함수가 호출받는 playGame함수 내에서 point += map[Point\_y][Point\_x];의 대입연산을 수행할 때 Point\_y와 Point\_x를 매개변수로 받으므로 함수의 body내에 Point\_x와 Point\_y를 직접 매개변수로 쓰는 구문이 있지는 않으나, checkWin함수의 헤더부분에 Point\_x, Point\_y를 매개변수로 넣어줌.

### 3. 구현결과

|   |  |
|---|--|
|  |  |
| <p>승리 과정(시행횟수2-&gt;3되며 점수가 10을 넘기며 승리)</p>  | <p>승리문구</p>  |

|  |   |
|--|---|
|  |  |
| <p>패배 과정(패배조건1/시행횟수 6회초과)</p>  | <p>패배문구</p>   |

|   |  |
|---|--|
|  |  |
| <p>패배 과정(패배조건1인 시행횟수 6회가 되지 않았으나 5X5를 벗어난 이동으로인한 패배)</p>                            | <p>패배문구</p>  |