

# 게임프로그래밍

## (7\_2\_1 게임 업그레이드)

날짜 : 2023 . 10 . 11

학번 : 2019775045

이름 : 이상배



경성대학교  
KYUNGSUNG UNIVERSITY

# 목 차

1. 업그레이드 전의 코드 한 번 더 소개
2. 전 코드에 업그레이드 한 기능들 소개
3. 코드 실행

## 7\_2\_1 의 출력결과

정방행렬에 나타난 숫자 중에서  
가장 많이 출현한 숫자를 찾아라!

제한시간은 5초입니다.

아무키나 누르면 시작합니다.

```
1 2 1 2 3
4 4 3 4 2
2 4 0 0 2
2 3 1 3 3
3 1 4 4 2
```

출현횟수가 가장 많은 숫자 입력>

```
1 0 1 1 4
4 4 2 2 3
3 1 2 3 0
1 2 2 3 3
0 2 2 2 0
```

출현횟수가 가장 많은 숫자 입력>  
제한시간을 넘었습니다.

```
3 3 3 1 3
4 2 0 2 2
3 0 2 1 3
1 4 2 4 2
3 1 3 3 4
```

출현횟수가 가장 많은 숫자 입력>3  
맞았습니다.  
번호: 3, 횟수 : 9

```
1 3 1 0 4
3 0 1 1 1
3 4 4 3 4
2 0 1 1 3
2 0 0 0 4
```

출현횟수가 가장 많은 숫자 입력>4  
틀렸습니다.  
번호: 1, 횟수 : 7

## 추가한 기능 (1:글꼴 색, 2: 배경 색)

```
int main(void)
{
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hConsole, 0x00 | 0x70);
}
```

- 이 함수는 글꼴과 콘솔의 색을 변경 해주는 함수로써 <windows.h> 헤더파일 내의 함수이다.
- 앞의 0x00은 검은색 글자 색을 나타내고 뒤의 0x70은 흰색 배경을 나타낸다.

## 추가한 기능 (3:gotoxy)

```
void gotoxy(int x, int y) {  
    COORD coord;  
    coord.X = x;  
    coord.Y = y;  
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);  
}
```

- Gotoxy를 사용해 콘솔 내의 텍스트 위치 변경

```

int select_mode(void) {
    int choice;
    int matrixSize;
    int score = 0;

    system("cls");
    gotoxy(10, 5); // 텍스트를 원하는 위치로 이동
    printf("=== 모드 선택 ===\n");
    gotoxy(10, 7);
    printf("1. Time Attack Mode\n");
    gotoxy(10, 8);
    printf("2. Infinity Mode\n");
    gotoxy(10, 10);
    printf("=== 1,2 이외의 키는 게임이 종료됩니다 ===");
    gotoxy(10, 13);
    printf("모드를 선택하세요 (1-2): ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            // Time Attack 모드 선택
            time_intro();
            time_game_control();
            break;
        case 2:
            // Infinity 모드 선택
            matrixSize = Infinity_select_difficulty();
            Infinity_intro();
            while (1) // 무한 루프를 사용하여 게임을 반복 실행
            {
                Infinity_game_control(matrixSize, &score);
            }
            break;
        default:
            printf("게임을 종료합니다.");
            Beep(0, 500);
            // 기본값으로 게임 종료
            exit(0);
    }
}

```

## 추가된 기능 (4: 모드 선택)

- Time attack 모드와 infinity 모드를 선택할 수 있는 화면을 추가함.

```

int num_of_answers = 0; // 정답 숫자의 개수
clock_t start, end;
double pst;
int correct = 0; // 정답 여부를 나타내는 변수 추가

system("cls");
infinity_print_number(matrixSize, cnt);
printf("\n\n");
printf("출현 횟수가 가장 많은 숫자 입력 >");

// 정답 숫자들을 찾아 배열에 저장
int max_num = 0;
for (int i = 0; i < INFINITY_MAX_NUMBER; i++) {
    if (cnt[i] > max_num) {
        max_num = cnt[i];
        num_of_answers = 0;
        answer[num_of_answers++] = i;
    } else if (cnt[i] == max_num) {
        answer[num_of_answers++] = i;
    }
}

```

## 추가된 기능 (5: 중복 정답 처리)

- 전 코드와 달리 출현 숫자가 같은 수가 여러 개 있으면 그 숫자 중에서 큰 값이 정답이 처리가 되었는데 중복 정답 처리로 함



## 추가된 기능 (6: 무한모드 난이도 설정)

- Infinity 모드를 설정하면 각 난이도마다 설정된 행렬 크기를 설정한다.

```
switch (choice)
{
    case 1:
        matrixSize = EASY_SIZE;
        break;
    case 2:
        matrixSize = NORMAL_SIZE;
        break;
    case 3:
        matrixSize = HARD_SIZE;
        break;
    case 4:
        matrixSize = VERY_HARD_SIZE;
        break;
    default:
        matrixSize = NORMAL_SIZE; // 기본은 Normal 난이도로 설정
        break;
}

return matrixSize;
}
```



## 추가된 기능 (7: 무한모드)

```
while (1) // 무한 루프를 사용하여 게임을 반복 실행
{
    Infinity_game_control(matrixSize, &score);
}
break;
```

- 행렬이 무한으로 생성되며 틀릴 때까지 문제를 보여주고 정답이거나 오답일 시 가장 많이 출현한 숫자와 출현 횟수를 표시한다.
- 틀릴 시에는 game over가 되며 정답을 맞춘 횟수(score)가 표시된다.

## 추가된 기능 (8 : 모드 선택으로 돌아가기)

```
if (!correct) {
    printf("\n\n틀렸 습 니 다 T_T \n");
    printf("Game Result 정 답 횟 수 : %d\n", *score);
    printf("입 력 한 숫 자 : %d \n", user);
    printf("번 호 : ");
    for (int i = 0; i < num_of_answers; i++) {
        printf("%d ", answer[i]);
    }
    printf("\n횟 수 : %d\n", cnt[answer[0]]);
    Beep((int)C4, 1000);
    printf("아 무 키 나 누 르 면 모 드 선택 화 면 으 로 도 아 갑 니 다 . \n");
    getch();
    select_mode();
}
```

틀렸을 때에 select\_mode 함수를 호출하여 다시 모드 선택화면으로 돌아가게 한다.

## 추가된 기능 (9 : 사운드)

```
Beep((int)C4, 500);  
Beep((int)E4, 500);  
Beep((int)G4, 500);  
printf("\n\n맞 았 습 니 다 ^.^\n");
```

- 맞았을 때는 도, 미, 솔 소리가 나고 틀렸을 때는 도 소리만 난다.

```
printf("\n횃수 : %d\n", cnt[answer[0]]);  
Beep((int)C4, 1000);  
printf("아무 키나 누르면 모드 선택화면으로 돌아갑니다. \n");
```

```

void time_game_control(void)
{
    double C4 = 261.63; // 도
    double E4 = 329.63; // 미
    double G4 = 392.00; // 솔
    int cnt[TIME_MAX_NUMBER], answer, user, chances = 2;
    clock_t start, end;
    double pst;
    int correct = 0; // 정답 여부를 나타내는 변수 추가

    system("cls");
    time_print_number(TIME_SIZE, cnt);
    answer = find_num(cnt);
    printf("\n\n");
    printf("출현 횟수가 가장 많은 숫자 입력>");
    start = clock();
    while (!kbhit())
    {
        end = clock();
        pst = (double)(end - start) / CLOCKS_PER_SEC;
    }
}

```

// 정답을 입력받을 때까지 기회를 주며, 기회가 남아있을 경우 계속 반복합니다.

```

while (chances > 0)
{
    if (user == answer)
    {
        Beep((int)C4, 500);
        Beep((int)E4, 500);
        Beep((int)G4, 500);
        printf("\n\n맞습니다 ^.^ \n");
        printf("입력한 숫자 : %d \n", user);
        printf("정답 숫자 : %d\n", answer);
        printf("정답 숫자의 출현 횟수 : %d\n", cnt[answer]);
        printf("정답을 입력받는데 걸린 시간 : %.2f 초 \n", pst);
        correct = 1;
        break;
    }
    else
    {
        chances--;
        if (chances > 0)
        {
            printf("입력한 숫자 : %d \n", user);
            printf("\n틀렸습니다 T.T. 다시 입력하세요. 남은 기회 : %d\n", chances);
            Beep((int)C4, 500);
            user = getch() - '0';
        }
    }
}

```

```

    else
    {
        printf("입력한 숫자 : %d \n", user);
        printf("\n틀렸습니다 T.T. 기회를 모두 사용하셨습니다. Game Over\n");
        Beep((int)C4, 500);
        printf("아무 키나 누르면 모드 선택화면으로 돌아갑니다. \n");
        getch();
        select_mode();
    }
}

if (correct)
{
    printf("아무 키나 누르면 모드 선택화면으로 돌아갑니다. \n");
    getch();
    select_mode();
}
}

```

## 추가된 기능 (10 : 타임어택 기능)

- 제한 시간을 없애고 15 \* 15의 행렬에서 찾을 때까지 시간을 재는 모드
- 정답을 맞추면 정답까지 걸린 시간을 출력한다. 틀리면 시간 기록이 출력 되지 않는다.

```
chances = 2;
```

## 추가된 기능 (11 : 재도전 기회 기능)

- 기회를 추가하여 틀릴 시에 재도전을 할 수 있는 기능을 추가하였다.

```
else
{
    chances--;
    if (chances > 0)
    {
        printf("입력한 숫자 : %d \n", user);
        printf("\n틀렸습니다 T_T. 다시 입력하세요. 남은 기회 : %d\n", chances);
        Beep((int)C4, 500);
        user = getch() - '0';
    }
    else
    {
        printf("입력한 숫자 : %d \n", user);
        printf("\n틀렸습니다 T_T. 기회를 모두 사용하셨습니다. Game Over\n");
        Beep((int)C4, 500);
        printf("아무 키나 누르면 모드 선택화면으로 돌아갑니다. \n");
        getch();
        select_mode();
    }
}
```

## 추가된 기능 (12 : 종료 기능)

```
default:
    printf("게임을 종료합니다.");
    Beep(0,500);
    // 기본값으로 게임 종료
    exit(0);
```

- 모드 선택 메뉴에서 1, 2 이외의 키를 입력할 시에 게임이 종료되게 하였다.

# 업그레이드 한 코드 실행

---



# 출처

<https://ko.wikipedia.org/wiki/Stdio.h>

<https://ko.wikipedia.org/wiki/Time.h>

<https://ko.wikipedia.org/wiki/Stdlib.h>

<https://ko.wikipedia.org/wiki/Windows.h>

<https://en.wikipedia.org/wiki/Conio.h>

교수님의 강의자료

chatgpt

30점