

```

void battle(void) {
    my_skill_count();
    your_skill_count();

    while (1) {
        turn = 0;
        if (your_defense == 1) {
            printf("상대방이 방어 기술을 사용하여 다음 턴으로 넘어간다\n");
            scanf("%c", &next1);

            your_defense = 0;
            turn_mory_defense();
            save_mory_defense();
        }
        else {
            pick_my_turn();
            printf("%s의 체력 : %d/250   %s의 체력 : %d/250\n", my_character, my_HP, your_character, your_HP);
            scanf("%c", &next1);
            if (my_HP <= 0) {
                printf("%s가 쓰러졌다.\n %s의 승리!\n", my_character, your_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", your_name);
                break;
            }
            else if (your_HP <= 0) {
                printf("%s가 쓰러졌다.\n %s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                break;
            }
            else if (my_skill1 == 0 && my_skill2 == 0 && my_skill3 == 0 && my_skill4 == 0) //더 이상 기술을 쓸 수 없을 때
            {
                printf("%s는 더 이상 기술을 쓸 수 없다!\n%s의 승리!\n", my_character, your_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", your_name);
                break;
            }
        }
    }
}

```

포켓몬 배틀 게임

SW융합코딩1

김지원, 최서영

ABOUT US

“포켓몬 배틀 게임 ”

『모험하기』

- >> 사용자의 캐릭터는 “피츄” 고정
사용자의 공격은 선택 가능
- >> 상대방(컴퓨터)의 캐릭터는 랜덤 선택
상대방(컴퓨터)의 공격도 랜덤

“상대방이 컴퓨터 ”

“상대방이 다른 USER ”

『다른 사용자와의대결』

- >> 사용자의 캐릭터 선택 가능
사용자의 공격 선택 가능
- >> 상대방(사용자2)의 캐릭터 선택 가능
상대방(사용자2) 공격 선택 가능



C



```
#define _CRT_SECURE_NO_WARNINGS << scanf 관련 오류 무시
#include <stdio.h>
#include <stdlib.h>
#include <string.h> //문자열 관련 함수 쓰기 위한 헤더 파일
#include <time.h>
```

```
char my_name[10]; //사용자의 이름 저장하는 변수
char your_name[10]; //상대방의 이름 저장하는 변수
char my_character[10], your_character[10]; // 나의 캐릭터 이름, 상대방의 캐릭터 이름
char character[4][10] = { "피츨", "이상해씨", "파이리", "교부기" }; //캐릭터 이름 저장
char next1, next2; // enter키 누르면 다음 대화창 뜨도록 만드는 변수
int turn; //내 턴과 상대방 턴을 알려주는 변수
int menu_choice; //menu 선택(1번. 모험하기 or 2번. 다른 사람과 대결)
int my_HP = 250, your_HP = 250; // 나의 체력(HP), 상대방의 체력(HP)-체력이 0이면 지는 것
int HP;
int my_status, your_status; // 나의 상태, 상대방의 상태
int status = 0;
int my_defense, your_defense; //나의 방어, 상대방의 방어
int defense;
char turn_character[10];
int my_skill1, my_skill2, my_skill3, my_skill4, my_heal; //내 기술 횟수 저장
int your_skill1, your_skill2, your_skill3, your_skill4, your_heal; //상대방의 기술 횟수 저장
int skill1, skill2, skill3, skill4, heal;
int pitchu_skill_count[5] = { 5,5,2,1,1 };
int seed_skill_count[5] = { 5,5,2,1,1 };
int piri_skill_count[5] = { 5,1,2,1,1 };
int kobuk_skill_count[5] = { 5,5,2,1,1 };
```

사용할 **헤더 파일**을 선언

사용할 **전역 변수**를 선언



C



```
void my_name_func(void);
void your_name_func(void);
void computer_name_func(void);
void my_character_func(void);
void computer_character_func(void);
void your_character_func(void);
void first_menu(void);
void menu(void);
void adventure(void);
void match(void);
void battle(void);
void my_skill_count(void);
void your_skill_count(void);
void pick_my_turn(void);
void pick_your_turn(void);
void pitchu_turn(void);
void seed_turn(void);
void piri_turn(void);
void kobuk_turn(void);
void turn_mory_skill(void);
void save_mory_skill(void);
void turn_mory_status(void);
void save_mory_status(void);
void current_status(void);
void save_current_status(void);
void turn_mory_defense(void);
void save_mory_defense(void);
void turn_mory_HP(void);
void save_mory_HP(void);
void resave_mory_HP(void);
void turn_mory_character(void);
```

```
void p1();
void p2();
void p3();
void p4();
void s1();
void s2();
void s3();
void s4();
void l1();
void l2();
void l3();
void l4();
void k1();
void k2();
void k3();
void k4();
```

함수 원형
컴파일러에게 함수 알림



C



```
int main(void)
{
    first_menu();
    while (1) {
        printf("1. 모험하기 2. 다른 사용자와의 대결\n");
        printf("어떤 것을 선택하시겠습니까?: ");
        scanf("%d", &menu_choice);
        scanf("%c%c", &next1, &next2);

        if (menu_choice == 1) {
            printf("모험하기를 선택하셨습니다!\n");
            scanf("%c", &next1);
            adventure();
            break;
        }
        else if (menu_choice == 2) {
            printf("다른 사용자와의 대결을 선택하셨습니다!\n");
            scanf("%c", &next1);
            match();
            break;
        }
        else {
            printf("메뉴를 다시 선택하세요!\n");
            scanf("%c", &next1);
        }
    }
    return 0;
}
```

int main(void) 함수

first_menu() 함수 호출



C



first_menu() 함수- “포켓몬 배틀” 프린트

```
void first_menu(void) {
    printf("##### \n");
    printf("##### \n");
    printf("##### # # ##### # # # # # # ##### \n");
    printf(" # # ## # # # # # # # # # # # # \n");
    printf(" # # ##### # # # ## ## ##### ### ##### # # ##### \n");
    printf("##### ## # # ##### ## ## # # # # # # \n");
    printf(" # ## # # # ##### # ##### # # ##### \n");
    printf(" # ## ##### # # # # \n");
    printf("##### ## ## # ##### # # # # ##### \n");
    printf(" ## ## ##### # # # # \n");
    printf("##### \n");
    printf("# \n");
    printf("##### \n");
    printf("\n\n\n");
    printf("#### 텍스트를 넘기려면 엔터키로(엔터키 두 번 눌러야 할 때도 있음)\n");
    scanf("%c", &next1);
}
```



C



```
int main(void)
{
    first_menu();
    while (1) {
        printf("1. 모험하기 2. 다른 사용자와의 대결\n");
        printf("어떤 것을 선택하시겠습니까?: ");
        scanf("%d", &menu_choice);
        scanf("%c%c", &next1, &next2);

        if (menu_choice == 1) {
            printf("모험하기를 선택하셨습니다!\n");
            scanf("%c", &next1);
            adventure();
            break;
        }
        else if (menu_choice == 2) {
            printf("다른 사용자와의 대결을 선택하셨습니다!\n");
            scanf("%c", &next1);
            match();
            break;
        }
        else {
            printf("메뉴를 다시 선택하세요!\n");
            scanf("%c", &next1);
        }
    }
    return 0;
}
```

int main(void) 함수

전역변수 **menu_choice**에 1 or 2입력 받음

1이면 **adventure()** 함수 호출
2이면 **match()** 함수 호출



C



모험, 다른 사용자와의 대결 공통사항

```
void my_name_func(void) {
    int name;
    char no_name[] = "이름 없음";

    while (1) {

        printf("사용자1의 이름을 입력하시겠습니까? [1. YES 2.NO] : ");
        scanf("%d", &name);

        if (name == 1) {
            printf("당신의 이름은 무엇입니까? : ");
            scanf("%s", my_name);
            printf("[%s]님! 환영합니다!\\n", my_name);
            scanf("%c%c", &next1, &next2);
            break;

        }
        else if (name == 2) {
            strncpy(my_name, no_name, sizeof(no_name));
            printf("[%s]님! 환영합니다!\\n", my_name);
            scanf("%c%c", &next1, &next2);
            break;

        }
        else {
            printf("메뉴를 잘못 선택하셨습니다! 다시 골라주세요!\\n");
            scanf("%c%c", &next1, &next2);
        }
    }
}
```

`no_name`을 `sizeof(no_name)`
크기만큼 `my_name` 배열에 저장

```
void adventure(void) {
    my_name_func();
}
```

```
void match(void) {
    my_name_func();
}
```

`my_name_func()` 함수
-사용자의 이름 입력

`name` 변수에 1 or 2 입력 받음

```
char my_name[10]; //사용자의 이름 저장하는 변수
```

1이면 전역변수
`my_name`에 사용자의 이름 저장
2이면 전역변수
`my_name`에 "이름 없음" 저장



C



사용자들의 이름을 입력하고 캐릭터 선택

```
void your_name_func(void) {
    int name;
    char no_name[] = "이름 없음";

    while (1) {

        printf("사용자2의 이름을 입력하시겠습니까? [1. YES 2.NO] : ");
        scanf("%d", &name);

        if (name == 1) {
            printf("당신의 이름은 무엇입니까?: ");
            scanf("%s", your_name);
            printf("[%s]님! 환영합니다!\\n", your_name);
            scanf("%c%c", &next1, &next2);
            break;
        }
        else if (name == 2) {
            strncpy(your_name, no_name, sizeof(no_name));
            printf("[%s]님! 환영합니다!\\n", your_name);
            scanf("%c%c", &next1, &next2);
            break;
        }
        else {
            printf("메뉴를 잘못 선택하셨습니다! 다시 골라주세요!\\n");
            scanf("%c%c", &next1, &next2);
        }
    }
}
```

“computer” 문자열을
your_name 배열에 저장

```
void computer_name_func(void) {
    strncpy(your_name, "computer", sizeof("computer"));
}
```

your_name_func() 함수
-my_name_func과 비슷

```
void match(void) {
    my_name_func();
    my_character_func();
    your_name_func();
}
```

my_name 배열이 아닌
your_name 배열에 이름 저장

```
void adventure(void) {
    my_name_func();
    computer_name_func();
}
```

computer_name_func() 함수



C



```
void adventure(void) {
    my_name_func();
    computer_name_func();
    strncpy(my_character, character[0], sizeof(character[0]));
    computer_character_func();
    printf("야생의 %s가 나타났다!\n", your_character);
    scanf("%c", &next1);
    printf("나와라 %s!\n", my_character);
    scanf("%c", &next1);
    printf("\n%s의 체력 : %d/250    %s의 체력 : %d/250\n\n", my_character, my_HP, your_character, your_HP);
    scanf("%c", &next1);
    battle();
}
```

**adventure() 함수는 사용자의 캐릭터가
피추로 고정 & 상대방이 컴퓨터**

```
void match(void) {
    my_name_func();
    my_character_func();
    your_name_func();
    your_character_func();
    printf("나: %s\tvs\t상대방: %s\t대결!\n", my_character, your_character);
    scanf("%c", &next1);
    printf("나와라 %s!\n", my_character);
    scanf("%c", &next1);
    printf("\n%s의 체력 : %d/250    %s의 체력 : %d/250\n\n", my_character, my_HP, your_character, your_HP);
    scanf("%c", &next1);
    battle();
}
```

match()함수는 상대방이 다른 사용자

모험하기



C



캐릭터 선택 (모험하기)

```
char character[4][10] = { "피츨", "이상해씨", "파이리", "꼬부기" };
```

```
void adventure(void) {  
    my_name_func();  
    computer_name_func();  
    strncpy(my_character, character[0], sizeof(character[0]));  
}
```

character 2중 배열에 저장된 “피츨”
문자열을 my_character에 복사

adventure() 함수
-사용자의 캐릭터가 “피츨”로 고정

전역 변수 my_character 배열에
“피츨” 문자열 저장



C



컴퓨터의 캐릭터 선택

```
void computer_character_func(void) {
    srand((unsigned)time(NULL));
    int x = rand() % 4;

    if (x == 0) {
        strncpy(your_character, character[0], sizeof(character[0]));
    }
    else if (x == 1) {
        strncpy(your_character, character[1], sizeof(character[1]));
    }
    else if (x == 2) {
        strncpy(your_character, character[2], sizeof(character[2]));
    }
    else if (x == 3) {
        strncpy(your_character, character[3], sizeof(character[3]));
    }
}
```

```
void adventure(void) {
    my_name_func();
    computer_name_func();
    strncpy(my_character, character[0], sizeof(character[0]));
    computer_character_func();
}
```

상대방(컴퓨터)의 캐릭터는 rand()
를 이용하여 ¼ 확률로 선택

다른 사용자와의 대결



C



캐릭터 선택 (다른 사용자와의 대결)

```
char character[4][10] = { "피츨", "이상해씨", "파이리", "꼬부기" };
```

```
void my_character_func(void) {
    int choice;

    while (1) {
        printf("1. 피츨 2. 이상해씨 3. 파이리 4. 꼬부기\n");
        printf("[%s]님의 파트너를 선택하세요 : ", my_name);
        scanf("%d", &choice);

        if (choice == 1) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", my_name, character[0]);
            strncpy(my_character, character[0], sizeof(character[0]));
            break;
        }
        else if (choice == 2) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", my_name, character[1]);
            strncpy(my_character, character[1], sizeof(character[1]));
            break;
        }
        else if (choice == 3) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", my_name, character[2]);
            strncpy(my_character, character[2], sizeof(character[2]));
            break;
        }
        else if (choice == 4) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", my_name, character[3]);
            strncpy(my_character, character[3], sizeof(character[3]));
            break;
        }
        else {
            printf("파트너를 다시 선택해주세요!\n");
            scanf("%c%c", &next1, &next2);
        }
    }
    scanf("%c%c", &next1, &next2);
}
```

```
void match(void) {
    my_name_func();
    my_character_func();
}
```

choice 변수로 캐릭터 입력 받음

선택에 따라 character 이중 배열의
각각의 행에 주어진 문자열을
my_character로 복사



C



캐릭터 선택 후 프린트값

```
void your_character_func(void) {
    int choice;

    while (1) {
        printf("1. 피츨 2. 이상해씨 3. 파이리 4. 꼬부기\n");
        printf("[%s]님의 파트너를 선택하세요: ", your_name);
        scanf_s("%d", &choice);

        if (choice == 1) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", your_name, character[0]);
            strncpy(your_character, character[0], sizeof(character[0]));
            break;
        }
        else if (choice == 2) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", your_name, character[1]);
            strncpy(your_character, character[1], sizeof(character[1]));
            break;
        }
        else if (choice == 3) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", your_name, character[2]);
            strncpy(your_character, character[2], sizeof(character[2]));
            break;
        }
        else if (choice == 4) {
            printf("[%s]님의 파트너로 %s를 선택하셨습니다!\n", your_name, character[3]);
            strncpy(your_character, character[3], sizeof(character[3]));
            break;
        }
        else {
            printf("파트너를 다시 선택해주세요!\n");
            scanf("%c%c", &next1, &next2);
        }
    }
    scanf("%c%c", &next1, &next2);
}
```

```
void match(void) {
    my_name_func();
    my_character_func();
    your_name_func();
    your_character_func();
}
```

my_character_func()과 비슷
-문자열이 **your_character**로 복사



C



캐릭터 선택 후 스킬 가능 횟수 저장 (다른 사용자와의 대결)

```
void battle(void) {
    my_skill_count();
    your_skill_count();
}
```

strcmp()함수는 문자열을 비교하여 같으면 0을 반환함

```
void my_skill_count(void) {
    if (strcmp(my_character, "피츨") == 0) {
        my_skill1 = pitchu_skill_count[0];
        my_skill2 = pitchu_skill_count[1];
        my_skill3 = pitchu_skill_count[2];
        my_skill4 = pitchu_skill_count[3];
        my_heal = pitchu_skill_count[4];
    }
    else if (strcmp(my_character, "이상해씨") == 0) {
        my_skill1 = seed_skill_count[0];
        my_skill2 = seed_skill_count[1];
        my_skill3 = seed_skill_count[2];
        my_skill4 = seed_skill_count[3];
        my_heal = seed_skill_count[4];
    }
    else if (strcmp(my_character, "파이리") == 0) {
        my_skill1 = piri_skill_count[0];
        my_skill2 = piri_skill_count[1];
        my_skill3 = piri_skill_count[2];
        my_skill4 = piri_skill_count[3];
        my_heal = piri_skill_count[4];
    }
    else {
        my_skill1 = kobuk_skill_count[0];
        my_skill2 = kobuk_skill_count[1];
        my_skill3 = kobuk_skill_count[2];
        my_skill4 = kobuk_skill_count[3];
        my_heal = kobuk_skill_count[4];
    }
}
```

```
void your_skill_count(void) {
    if (strcmp(your_character, "피츨") == 0) {
        your_skill1 = pitchu_skill_count[0];
        your_skill2 = pitchu_skill_count[1];
        your_skill3 = pitchu_skill_count[2];
        your_skill4 = pitchu_skill_count[3];
        your_heal = pitchu_skill_count[4];
    }
    else if (strcmp(your_character, "이상해씨") == 0) {
        your_skill1 = seed_skill_count[0];
        your_skill2 = seed_skill_count[1];
        your_skill3 = seed_skill_count[2];
        your_skill4 = seed_skill_count[3];
        your_heal = seed_skill_count[4];
    }
    else if (strcmp(your_character, "파이리") == 0) {
        your_skill1 = piri_skill_count[0];
        your_skill2 = piri_skill_count[1];
        your_skill3 = piri_skill_count[2];
        your_skill4 = piri_skill_count[3];
        your_heal = piri_skill_count[4];
    }
    else {
        your_skill1 = kobuk_skill_count[0];
        your_skill2 = kobuk_skill_count[1];
        your_skill3 = kobuk_skill_count[2];
        your_skill4 = kobuk_skill_count[3];
        your_heal = kobuk_skill_count[4];
    }
}
```

battle()함수

--my_skill_count() 함수와
your_skill_count() 함수 호출

캐릭터에 따라 skill 사용 가능 횟수
가 다르기 때문에 조건문을 통해
skill 가능 횟수 저장

```
int pitchu_skill_count[5] = { 5,5,2,1,1 };
int seed_skill_count[5] = { 5,5,2,1,1 };
int piri_skill_count[5] = { 5,1,2,1,1 };
int kobuk_skill_count[5] = { 5,5,2,1,1 };
```



```
while (1) {  
    turn = 0;  
    if (your_defense == 1) {  
        printf("상대방이 방어 기술을 사용하여 다음 턴으로 넘어간다.\n");  
        scanf("%c", &next1);  
        your_defense = 0;  
    }
```

turn= 0이면
사용자1의 공격 차례
turn= 1이면
사용자2의 공격 차례

```
    else {  
        pick_my_turn();  
    }
```

battle()함수

-**while**문으로 무한 반복

누군가의 캐릭터가 죽으면 반복문 끝남

defense는 방어 기술

turn이 0일 때 상대방이 방어를
사용하여 **your_defense**가 1이
면 사용자의 공격 불가하고 상대
방 차례로 넘어감

만약 상대방이 defense 기술을
사용하지 않았을 경우

pick_my_turn()함수 호출



```
void pick_my_turn(void) {  
    if (strcmp(my_character, "피츨") == 0) {  
        pitchu_turn();  
    }  
    else if (strcmp(my_character, "이상해씨") == 0) {  
        seed_turn();  
    }  
    else if (strcmp(my_character, "파이리") == 0) {  
        piri_turn();  
    }  
    else {  
        kobuk_turn();  
    }  
}
```

pick_my_turn() 함수
-나의 캐릭터에 따라 함수 호출

캐릭터에 따른 공격 진행 함수



C



```
printf("%s의 체력 : %d/250   %s의 체력 : %d/250\n", my_character, my_HP, your_character, your_HP);
scanf("%c", &next1);
if (my_HP <= 0) {
    printf("%s가 쓰러졌다.\n %s의 승리!\n", my_character, your_character);
    scanf("%c", &next1);
    printf("%s님이 승리하셨습니다.", your_name);
    break;
}
else if (your_HP <= 0) {
    printf("%s가 쓰러졌다.\n %s의 승리!\n", your_character, my_character);
    scanf("%c", &next1);
    printf("%s님이 승리하셨습니다.", my_name);
    break;
}
else if (my_skill1 == 0 && my_skill2 == 0 && my_skill3 == 0 && my_skill4 == 0) //더 이상 기술을 쓸 수 없을 때
{
    printf("%s는 더 이상 기술을 쓸 수 없다!\n%s의 승리!\n", my_character, your_character);
    scanf("%c", &next1);
    printf("%s님이 승리하셨습니다.", your_name);
    break;
}
```

battle() 함수

나의 캐릭터에 따라 공격 진행한 후
다시 battle()함수로 돌아옴

HP<=0 이거나 기술을 모두 다 썼을
경우 게임에서 패배



C



```
turn = 1;
if (my_defense == 1) {
    printf("나의 방어 사용으로 다음 턴으로 넘어간다\n");
    scanf("%c", &next1);
    my_defense = 0;
}
else {
    pick_your_turn();
    printf("%s의 체력:%d/250   %s의 체력:%d/250\n", my_character, my_HP, your_character, your_HP);
    scanf("%c", &next1);
    if (my_HP <= 0)
    {
        printf("%s가 쓰러졌다\n%s의 승리!\n", my_character, your_character);
        scanf("%c", &next1);
        printf("%s님이 승리하셨습니다.", your_name);
        break;
    }
    else if (your_HP <= 0)
    {
        printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
        scanf("%c", &next1);
        printf("%s님이 승리하셨습니다.", my_name);
        break;
    }
    else if (your_skill1 == 0 && your_skill2 == 0 && your_skill3 == 0 && your_skill4 == 0) //더 이상 기술을 쓸 수 없을 때
    {
        printf("%s는 더 이상 기술을 쓸 수 없다\n%s의 승리!\n", your_character, my_character);
        scanf("%c", &next1);
        printf("%s님이 승리하셨습니다.", my_name);
        break;
    }
}
```

battle() 함수

-turn이 1일 때 상대방의 공격 차례

turn이 0일 때와 비슷



```
void pitchu_turn(void) {  
    int i, k, skill = 0;  
    int random;  
    srand((unsigned)time(NULL));  
    random = rand() % 4;  
  
    turn_mory_skill();  
    turn_mory_HP();  
    current_status();  
}
```

pitchu_turn 함수

-캐릭터가 피츨 때 공격 진행

사용자의 공격인지 상대방의 공격인지 구분하여 스킬 사용 횟수와 HP, 상태 이상을 갱신하기 위한 함수



C



turn에 따라 공격 진행 함수(ex) pitchu_turn)를 두 개 만들지 않기 위해

```
void turn_mory_skill(void) {  
    if (turn == 0) {  
        skill1 = my_skill1;  
        skill2 = my_skill2;  
        skill3 = my_skill3;  
        skill4 = my_skill4;  
        heal = my_heal;  
    }  
    else {  
        skill1 = your_skill1;  
        skill2 = your_skill2;  
        skill3 = your_skill3;  
        skill4 = your_skill4;  
        heal = your_heal;  
    }  
}
```

```
void save_mory_skill(void) {  
    if (turn == 0) {  
        my_skill1 = skill1;  
        my_skill2 = skill2;  
        my_skill3 = skill3;  
        my_skill4 = skill4;  
        my_heal = heal;  
    }  
    else {  
        your_skill1 = skill1;  
        your_skill2 = skill2;  
        your_skill3 = skill3;  
        your_skill4 = skill4;  
        your_heal = heal;  
    }  
}
```

```
void turn_mory_HP(void) {  
    if (turn == 0)  
        HP = your_HP;  
    else  
        HP = my_HP;  
}
```

```
void save_mory_HP(void) {  
    if (turn == 0)  
        your_HP = HP;  
    else  
        my_HP = HP;  
}
```

```
void current_status(void) {  
    if (turn == 0)  
        status = my_status;  
    else  
        status = your_status;  
}
```

```
void save_current_status(void) {  
    if (turn == 0)  
        my_status = status;  
    else  
        your_status = status;  
}
```

turn_mory_skill()
save_mory_skill() 함수
-자신의 skill 갱신

turn_mory_HP()
save_mory_HP() 함수
-사용자가 공격: 상대방의 HP 갱신
-상대방이 공격: 나의 HP 갱신

current_status()
save_current_status() 함수
-자신의 상태 이상을 갱신하기 위
한 함수



```
if (menu_choice == 1 && turn == 1) {  
    if (status == 1) {  
        if (heal == 0) {  
            printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");  
            scanf("%c", &next1);  
            printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);  
            scanf("%c", &next1);  
            printf("%s님이 승리하셨습니다.", my_name);  
            exit(0);  
        }  
        else {  
            printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");  
            scanf("%c", &next1);  
            printf("상태 이상 회복!\n");  
            scanf("%c", &next1);  
            status = 0;  
            heal--;  
            save_current_status();  
        }  
    }  
}
```

상태 이상에 걸렸을 경우 컴퓨터가 **먼저**
치료약을 선택하도록 프로그래밍

pitchu_turn() 함수

-모험하기(상대방이 컴퓨터)일 때 공격
진행 함수를 호출했을 때

상태 이상일 경우 heal(치료약) 사용
치료약이 없을 경우 공격이 불가능해
패배하게 됨



C



```
else {  
    if (random == 0) {  
        if (skill1 == 0) {  
            pitchu_turn();  
        }  
        else {  
            p1();  
            skill1--;  
        }  
    }  
    else if (random == 1) {  
        if (skill2 == 0) {  
            pitchu_turn();  
        }  
        else {  
            p2();  
            skill2--;  
        }  
    }  
    else if (random == 2) {  
        if (skill3 == 0) {  
            pitchu_turn();  
        }  
        else {  
            p3();  
            skill3--;  
        }  
    }  
    else {  
        if (skill4 == 0) {  
            pitchu_turn();  
        }  
        else {  
            p4();  
            skill4--;  
        }  
    }  
}
```

pitchu_turn() 함수

-상대방이 컴퓨터일 때 상태이상에 걸리지 않았을 경우

```
int random;  
srand((unsigned)time(NULL));  
random = rand() % 4;
```

random 변수 이용하여 $\frac{1}{4}$ 확률로 skill 선택
만약 스킬 사용 가능 횟수가 0이면 다시 공격 진행 함수(pitchu_turn()함수)호출



C



```
else {
    printf("1. 기술을 쓴다\n2. 치료약을 쓴다\n3. 기술 정보를 확인한다\n\n피츨는 무엇을 할까?: ");
    scanf("%d", &i);
    printf("\n");
    scanf("%c", &next1);

    if (status == 1) {
        if (i == 1) {
            printf("이상 상태(화상, 마비 등)로 인해 스킬 사용이 불가하다\n");
            printf("치료약을 쓰면 이상 상태에서 벗어날 수 있다\n");
            scanf("%c", &next1);
            pitchu_turn();
        }
        else if (i == 2) {
            if (heal == 0) {
                printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!");
                scanf("%c", &next1);
                if (turn == 0) {
                    printf("%s가 쓰러졌다\n%s의 승리!\n", my_character, your_character);
                    scanf("%c", &next1);
                    printf("%s님이 승리하셨습니다.", your_name);
                    exit(0);
                }
                else {
                    printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                    scanf("%c", &next1);
                    printf("%s님이 승리하셨습니다.", my_name);
                    exit(0);
                }
            }
        }
    }
}
```

i 값 입력 받아 메뉴 선택

pitchu_turn() 함수

-모험하기(상대방이 컴퓨터)가 아니거나 모험하기여도 사용자의 turn일 때

상태 이상에 걸렸을 때는

1번(공격)메뉴 사용 불가능

2번(치료약)으로 상태 이상 치료 후 공격 가능
따라서 만약 **치료약이 0개면 패배**



C



```
else if( i == 3) {  
    printf("기술 정보 확인!\n");  
    printf("1. 박치기: 머리를 내밀고 무작정 돌진하여 공격한다. 가끔 풀죽게 만든다. 나무를 흔들어 포켓몬을 떨어뜨릴 수도 있다.\n\n");  
    printf("2. 번개펀치:전격을 담은 펀치로 상대를 공격한다. 50%% 확률로 상대 포켓몬을 화상입힐 수 있다.\n\n");  
    printf("3. 물 부비부비 :전기가 흐르는 물을 비벼서 공격한다. 랜덤으로 연속 공격이 가능하다. \n\n");  
    printf("4. 10만볼트: 강한 전격을 상대에게 날려서 공격한다. 상대를 90%% 확률로 마비시킬 수 있다.\n\n");  
    printf("메인 화면으로 돌아간다...\n");  
    scanf("%c", &next1);  
    pitchu_turn();  
}  
else {  
    pitchu_turn();  
}
```

pitchu_turn() 함수

-다른 사용자와의 대결 또는 사용자 turn이면서, 상태 이상에 걸렸을 때 3번 메뉴는 이용 가능(공격 기술 정보 확인)



C



pitchu_turn()함수-
다른 사용자와의 대결 또는 사용자 turn
이면서, 상태 이상에 걸리지 않았을 때

skill 변수를 이용해 몇 번째 스킬을
사용할 것인지 입력 받음

스킬을 사용했다면 스킬 사용 가능 횟수 갱신

선택한 스킬의 사용 가능 횟수가 0이라면
다시 공격 진행 함수로 돌아감

```
else {
    if (i == 1) {
        printf("기술 시전!\n");
        scanf("%c", &next1);
        printf("1. 박치기(pp %d/5)\n2. 번개편치(pp %d/5)\n3. 물 부비부비(pp %d/2)\n4. 10만볼트(pp %d/1)\n", skill11, skill12, skill13, skill14);
        printf("뒤로 가시려면 1~4이외의 숫자를 입력하세요\n");
        printf("어떤 기술을 사용할까요? ");
        scanf("%d", &skill1);
        printf("\n");
        scanf("%c", &next1);

        if (skill1 == 1) {
            if (skill11 == 0) {
                printf("이 스킬은 더 이상 쓸 수 없습니다!(스킬 소진)\n");
                scanf("%c", &next1);
                pitchu_turn();
            }
            else {
                p1();
                skill11--;
            }
        }
        else if (skill1 == 2) {
            if (skill12 == 0) {
                printf("이 스킬은 더 이상 쓸 수 없습니다!(스킬 소진)\n");
                scanf("%c", &next1);
                pitchu_turn();
            }
            else {
                p2();
                skill12--;
            }
        }
        else if (skill1 == 3) {
            if (skill13 == 0) {
                printf("이 스킬은 더 이상 쓸 수 없습니다!(스킬 소진)\n");
                scanf("%c", &next1);
                pitchu_turn();
            }
            else {
                p3();
                skill13--;
            }
        }
    }
}
```

i값에 따라 공격 or 치료
약 or 기술 정보 보고

```
else if (i == 2) {
    printf("상처약(%d/1개)\n", heal);
    printf("뒤로 가시려면 1이외의 문자나 숫자를 입력해 잘못된 입력을 하세요\n");
    scanf("%d", &k);
    if (k == 1) {
        if (heal == 0) {
            printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
            pitchu_turn();
        }
        else {
            printf("상처약을 사용했다!\n");
            scanf("%c", &next1);
            status = 0;
            heal--;
            save_current_status();
        }
    }
    else {
        printf("잘못 입력하셨습니다!\n");
        scanf("%c", &next1);
        pitchu_turn();
    }
}
else if (i == 3) {
    printf("기술 정보 확인!\n");
    printf("1. 박치기: 머리를 내밀고 무작정 돌진하여 공격한다. 가끔 플촉게 만든다. 나무를 흔들어 포켓몬을 떨어뜨릴 수도 있다.\n");
    printf("2. 번개편치: 전격을 담은 편치로 상대를 공격한다. 50% 확률로 상대 포켓몬을 확실히할 수 있다.\n");
    printf("3. 물 부비부비: 전기가 흐르는 물을 비벼서 공격한다. 랜덤으로 연속 공격이 가능하다.\n");
    printf("4. 10만볼트: 강한 전격을 상대에게 날려서 공격한다. 상대를 90% 확률로 마비시킬 수 있다.\n");
    printf("메인 화면으로 돌아간다...\n");
    scanf("%c", &next1);
    pitchu_turn();
}
else {
    pitchu_turn();
}

save_wory_skill();
save_wory_hp();
```

변화된 skill값과 HP값 저장



C



seed_turn()함수-캐릭터가 이상해씨일 때 공격 진행,
piri_turn()함수-캐릭터가 파이리일 때 공격 진행,
kobuk_turn()함수-캐릭터가 꼬부기일 때 공격 진행

```
void seed_turn(void) {
    int i, k, skill = 0;
    int random;

    srand((unsigned)time(NULL));
    random = rand() % 4;

    turn_mory_skill();
    turn_mory_HP();
    current_status();

    if (menu_choice == 1 && turn == 1) {
        if (status == 1) {
            if (heal == 0) {
                printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
                scanf("%c", &next1);
                printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                exit(0);
            }
            else {
                printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");
                scanf("%c", &next1);
                printf("상태 이상 회복!\n");
                scanf("%c", &next1);
                status = 0;
                heal--;
                save_current_status();
            }
        }
        else {
            if (random == 0) {
                if (skill11 == 0) {
                    kobuk_turn();
                }
                else {
                    k1();
                    skill11--;
                }
            }
            else if (random == 1) {
                if (skill12 == 0) {
                    kobuk_turn();
                }
            }
        }
    }
}
```

```
void piri_turn(void) {
    int i, k, j, skill = 0;
    int random;

    srand((unsigned)time(NULL));
    random = rand() % 4;

    turn_mory_skill();
    turn_mory_HP();
    current_status();

    if (menu_choice == 1 && turn == 1) {
        if (status == 1) {
            if (skill14 == 0 && heal == 0) {
                printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
                scanf("%c", &next1);
                printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                exit(0);
            }
            else if (skill14 == 0 && heal != 0) {
                printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");
                scanf("%c", &next1);
                printf("상태 이상 회복!\n");
                scanf("%c", &next1);
                status = 0;
                heal--;
                save_current_status();
            }
            else {
                l4();
                skill14--;
            }
        }
        // 컴퓨터 공격 status가 1일 때
        else {
            if (random == 0) {
                if (skill11 == 0) {
                    kobuk_turn();
                }
                else {
                    k1();
                    skill11--;
                }
            }
            else if (random == 1) {
                if (skill12 == 0) {
                    kobuk_turn();
                }
            }
        }
    }
}
```

pitchu_turn()함수와 거의 비슷
-차이점: 캐릭터마다 공격이 다름

```
void kobuk_turn(void) {
    int i, k, skill = 0;
    int random;

    srand((unsigned)time(NULL));
    random = rand() % 4;

    turn_mory_skill();
    turn_mory_HP();
    current_status();

    if (menu_choice == 1 && turn == 1) {
        if (status == 1) {
            if (heal == 0) {
                printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
                scanf("%c", &next1);
                printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                exit(0);
            }
            else {
                printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");
                scanf("%c", &next1);
                printf("상태 이상 회복!\n");
                scanf("%c", &next1);
                status = 0;
                heal--;
                save_current_status();
            }
        }
        else {
            if (random == 0) {
                if (skill11 == 0) {
                    kobuk_turn();
                }
                else {
                    k1();
                    skill11--;
                }
            }
            else if (random == 1) {
                if (skill12 == 0) {
                    kobuk_turn();
                }
            }
        }
    }
}
```




C



```
void seed_turn(void) {
    int i, k, skill = 0;
    int random = 0;

    turn_mory_skill();
    turn_mory_HP();
    current_status();

    if (menu_choice == 1 && turn == 1) {
        srand((unsigned)time(NULL));
        random = rand() % 4;

        if (status == 1) {
            if (heal == 0) {
                printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
                scanf("%c", &next1);
                printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                exit(0);
            }
            else {
                printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");
                scanf("%c", &next1);
                printf("상태 이상 회복!\n");
                scanf("%c", &next1);
                status = 0;
                heal--;
            }
        }
    }
}
```

```
}
save_mory_skill();
if (skill != 4 && random != 3)
    save_mory_HP();
}
```

```
void turn_mory_HP(void) {
    if (turn == 0)
        HP = your_HP;
    else
        HP = my_HP;
}
```

```
void save_mory_HP(void) {
    if (turn == 0)
        your_HP = HP;
    else
        my_HP = HP;
}
```

seed_turn() 함수

-skill 4번 또는 random이 3일 때 수행되는 공격은 s4()[회복]이기 때문

자신의 HP를 HP에 저장하여 회복시켜야 함
turn이 0일 때 my_HP저장
turn이 1일 때 your_HP저장

turn_mory_HP()함수 save_mory_HP()함수

-공격할 때 줄어든 HP를 갱신하기 위한 함수
HP에 자신의 HP가 아닌 상대방의 HP를 저장하여 자신의 turn일 때 상대방의 HP를 곱함



C



piri_turn() 함수-l4()[객기]라는 공격 때문 객기: 자신이 상태 이상에 걸렸을 때 기술의 위력이 2배가 된다

```
void piri_turn(void) {
    int i, k, j, skill = 0;
    int random;

    srand((unsigned)time(NULL));
    random = rand() % 4;

    turn_mory_skill();
    turn_mory_HP();
    current_status();

    if (menu_choice == 1 && turn == 1) {
        if (status == 1) {
            if (skill4 == 0 && heal == 0) {
                printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
                scanf("%c", &next1);
                printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                exit(0);
            }
            else if (skill4 == 0 && heal != 0) {
                printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");
                scanf("%c", &next1);
                printf("상태 이상 회복!\n");
                scanf("%c", &next1);
                status = 0;
                heal--;
                save_current_status();
            }
            else {
                l4();
                skill4--;
            }
        } // 컴퓨터 공격 status가 1일 때
        else {
```

컴퓨터일 때-
치료약과 객기 모두 0이면 패배
치료약과 객기 둘 중 하나만 있으면 존재하는 것 사용
치료약과 객기 둘 다 있으면 객기 사용

```
if (status == 1) {
    if (i == 1) {
        printf("이상 상태(화상, 마비 등)로 인해 객기만 사용 가능하다!\n");
        printf("객기를 사용하려면 1을 입력하라(메뉴창으로 돌아가기: 1 이외의 수 입력)");
        scanf("%d", &j);
        if (j == 1) {
            if (skill4 == 0) {
                printf("객기를 모두 사용하여 스킬 사용이 불가하다!\n");
                scanf("%c", &next1);
                piri_turn();
            }
            else {
                l4();
                skill4--;
            }
        }
        else {
            piri_turn();
        }
    }
    else if (i == 2) {
        if (heal == 0) {
            printf("치료약을 모두 사용해 더 이상 쓸 수 없습니다!\n");
            scanf("%c", &next1);
            if (turn == 0) {
                printf("%s가 쓰러졌다\n%s의 승리!\n", my_character, your_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", your_name);
                exit(0);
            }
            else {
                printf("%s가 쓰러졌다\n%s의 승리!\n", your_character, my_character);
                scanf("%c", &next1);
                printf("%s님이 승리하셨습니다.", my_name);
                exit(0);
            }
        }
        else {
            printf("치료약을 사용하여 이상 상태(화상, 마비)를 치료하였다!\n");
            scanf("%c", &next1);
            printf("상태 이상 회복!\n");
            scanf("%c", &next1);
            status = 0;
            heal--;
            save_current_status();
        }
    }
}
```

사용자의 turn이거나 상대방이 다른 사용자일 때-
치료약과 객기 둘 중 하나 선택 가능



C



캐릭터들의 기본 공격

damage는 일정
percentage(공격 성공률) 일정

```
void pl() //피츨의 박치기
{
    int damage, percentage;

    printf("피츨의 박치기 공격!\n");
    scanf("%c", &next1);

    damage = 30;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 80) //80%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}
```

```
void kl() //꼬부기의 아쿠아테일
{
    int damage, percentage;
    printf("꼬부기의 아쿠아테일 공격!\n");
    scanf("%c", &next1);

    damage = 40;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 90) //90%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}
```

```
void sl() //이상해씨의 풀먹기
{
    int damage, percentage;
    printf("이상해씨의 풀먹기 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));

    damage = 40;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 80) //80%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}
```

```
void s2() //이상해씨의 덩굴 채찍
{
    int damage, percentage;

    printf("이상해씨의 덩굴 채찍 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));

    damage = 30;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 80) //80%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}
```



```
void p1() //피츨의 박치기
{
    int damage, percentage;

    printf("피츨의 박치기 공격!\n");
    scanf("%c", &next1);

    damage = 30;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 80) //80%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}

void turn_mory_character(void) {
    if (turn == 0)
        strncpy(turn_character, your_character, sizeof(your_character));
    else
        strncpy(turn_character, my_character, sizeof(my_character));
}
```

모든 스킬 함수(ex) p1())에서
turn_mory_character()함수가 호출됨

사용자의 공격인지 상대방의 공격인지 구분
하여 damage를 누가 받았는지 알기 위해
turn이 0: 상대방이 damage 받음
turn이 1: 사용자가 damage 받음

turn_mory_character() 함수
turn이 0일 때 turn_character에
your_character 속에 있는 문자열을 넣고
Turn이 1일 땐 my_character 속에 있는
문자열을 넣음



C



캐릭터들의 상태 이상 공격

```
void p2() //피츨의 번개편치
{
    int damage, percentage, burns_percentage;

    printf("피츨의 번개편치 공격!\n");
    scanf("%c", &next1);

    percentage = 1 + rand() % 100;
    burns_percentage = 1 + rand() % 100;
    damage = 40;

    turn_wory_character();
    turn_wory_status();

    if (percentage <= 90) //80%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
        if (burns_percentage <= 50) {
            printf("%s는(은) 화상에 걸렸다!\n", turn_character);
            status = 1;
        }
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
    save_wory_status();
}
```

```
void p4() //10만볼트
{
    int damage, percentage, paralysis_percentage;

    printf("피츨의 10만 볼트 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));

    damage = 50;
    percentage = 1 + rand() % 100;
    paralysis_percentage = 1 + rand() % 100;

    turn_wory_character();
    turn_wory_status();

    if (percentage <= 90) //90%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
        if (paralysis_percentage <= 90) //마비확률 80%
        {
            printf("%s는(은) 마비에 걸렸다!\n", turn_character);
            scanf("%c", &next1);
            status = 1;
        }
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
    save_wory_status();
}
```

```
void l1() //파이어의 불대문자
{
    int damage, percentage, burns_percentage;
    printf("파이어의 불대문자 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));

    damage = 40;
    percentage = 1 + rand() % 100;
    burns_percentage = 1 + rand() % 100;

    turn_wory_character();
    turn_wory_status();

    if (percentage <= 90) //90%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
        if (burns_percentage <= 30) //화상 확률 30%
        {
            printf("%s는(은) 화상에 걸렸다!\n", turn_character);
            scanf("%c", &next1);
            status = 1;
        }
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
    save_wory_status();
}
```

damage는 일정
percentage(공격 성공률) 일정
공격에 성공했을 때 일정 확률로 상대방을 상태 이상(화상, 마비)에 빠지게 함

```
void k3() //열탈
{
    int damage, percentage, burns_percentage;
    printf("꼬부기의 열탈 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));

    damage = 30;
    percentage = 1 + rand() % 100;
    burns_percentage = 1 + rand() % 100;

    turn_wory_character();
    turn_wory_status();

    if (percentage <= 90) //90%
    {
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
        scanf("%c", &next1);
        HP -= damage;
        if (burns_percentage <= 50) //화상 확률 50%
        {
            printf("%s는(은) 화상에 걸렸다!\n", turn_character);
            scanf("%c", &next1);
            status = 1;
        }
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
    save_wory_status();
}
```




C



```
void p3() //물 부비부비
{
    int damage, percentage, num, total_damage;

    printf("피츨의 물 부비부비 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));

    damage = 15;
    percentage = 1 + rand() % 65; //65%의 확률
    num = rand() % 3 + 2; //2~4번 랜덤

    turn_mory_character();

    if (percentage <= 65)
    {
        total_damage = damage * num;
        printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, total_damage);
        scanf("%c", &next1);
        HP -= total_damage;
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}
```

p3()함수

- damage는 30(15×2), 45(15×3), 60(15×4)값을 랜덤 선택
- percentage(공격 성공 확률)은 일정



C



```
void s3() //이상해풀의 연속 앞날가르기
{
    int damage, percentage, i;

    srand((unsigned)time(NULL));
    printf("이상해풀의 연속 앞날가르기!\n");
    scanf("%c", &next1);

    turn_mory_character();

    for (i = 1; i <= 5; i++)
    {
        percentage = 1 + rand() % 100;
        if (percentage <= 55) //55%의 확률
        {
            printf("%d회차 성공!\n", i);
            damage = 10 + rand() % 6; //10~16
            printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);
            HP -= damage;
        }
        else {
            printf("%d회차 실패!\n", i);
        }
        printf("\n");
    }
}
```

s3()함수

- percentage(공격 성공 확률)이 일정한 상태로 5번 공격
- 공격에 성공했을 때의 damage는 10~16의 수를 랜덤 선택
- 총 damage는 성공했을 때의 damage 값들을 모두 더한 값



C



```
void s4() //이상해풀의 광합성
{
    int recover, percentage, weather;

    srand((unsigned)time(NULL));

    printf("이상해씨의 광합성!\n");
    scanf("%c", &next1);
    percentage = 1 + rand() % 100;
    weather = rand() % 3 + 1;

    turn_mory_character();
    recover_mory_HP();

    if (percentage <= 90) //90%
    {
        if (weather == 1)
        {
            printf("날씨는 평상시와 같다!\n");
            recover = 50;
        }
        else if (weather == 2)
        {
            printf("날씨는 쾌청하다! 회복 효과가 더 좋아졌다.\n");
            recover = 100;
        }
        else
        {
            printf("날씨가 흐리다.\n");
            recover = 25;
        }
        printf("이상해씨는 %d만큼의 체력을 회복했다!\n", recover);
        scanf("%c", &next1);
        HP += recover;
        if (HP > 250)
            HP = 250;
    }
    else
    {
        printf("그러나 실패했다!\n");
        scanf("%c", &next1);
    }
    resave_mory_HP();
}
```

HP가 250 이상이면 더 이상
증가하지 않는다

```
void recover_mory_HP(void) {
    if (turn == 0)
        HP = my_HP;
    else
        HP = your_HP;
}
```

```
void resave_mory_HP(void) {
    if (turn == 0)
        my_HP = HP;
    else
        your_HP = HP;
}
```

s4()함수

-기술 성공 확률 일정

-날씨를 랜덤 선택: 날씨에 따라
회복량이 다름

자신의 turn일 때 자신의 HP를
회복해야 하기 때문에
recover_mory_HP()함수와
resave_mory_HP()함수를 통해
HP를 갱신



C



```
void l2() //파이리의 철판방패
```

```
{  
    int damage, percentage;  
    printf("파이리의 철판방패!\n");  
    scanf("%c", &next1);  
}
```

```
    srand((unsigned)time(NULL));
```

```
    damage = 30;  
    percentage = 1 + rand() % 100;
```

```
    turn_mory_character();
```

```
    if (percentage <= 60) //60%
```

```
{  
    printf("%s는(은) %d의 데미지를 입었다!\n", turn_character, damage);  
    scanf("%c", &next1);  
    HP -= damage;  
}
```

```
else
```

```
{  
    printf("그러나 빗나갔다!\n");  
    scanf("%c", &next1);  
}
```

```
    turn_mory_defense();
```

```
    printf("파이리는 바로 다음 상대의 공격을 막을 수 있다!\n"); //철판 방패: damage 값에 실패해도 방어기술은 항상 사용 가능
```

```
    scanf("%c", &next1);
```

```
    defense = 1;
```

```
    save_mory_defense();
```

```
void turn_mory_defense(void) {  
    if (turn == 0)  
        defense = my_defense;  
    else  
        defense = your_defense;  
}
```

```
void save_mory_defense(void) {  
    if (turn == 0)  
        my_defense = defense;  
    else  
        your_defense = defense;  
}
```

l2()함수

-damage 일정

-percentage(공격 성공 확률) 일정

-turn이 0일 때 상대방(컴퓨터 또는 다른 유저)의 공격 1회 막음

-turn이 1일 때 사용자의 공격 1회 막음

turn_mory_defencse()함수와
save_mory_defense()함수를 통해
turn이 0일 때 my_defense갱신
turn이 1일 때 your_defense 갱신
--방어는 자기자신에게 적용되어야 하기 때문



C



캐릭터들의 콤보 공격

void l3() //파이어의 회오리문자

```
int damage[5];
int percentage;
int damage_total = 0;
int i, combo = 0;
```

```
printf("파이어의 회오리문자 공격!\n");
scanf("%c", &next1);
```

```
rand((unsigned)time(NULL));
percentage = 1 + rand() % 100;
```

```
turn_mory_character();
```

```
for (i = 0; i < 5; i++)
```

```
{
    if (percentage <= (85 - i * 5))
    {
        damage[i] = 1;
        combo = 13 + i;
        damage_total += damage[i] * combo;
    }
    else
    {
        damage[i] = 0;
        break;
    }
}
```

```
printf("%d회 콤보 공격 성공!\n", i);
printf("%s는 %d의 데미지를 입었다!\n", turn_character, damage_total);
scanf("%c", &next1);
HP -= damage_total;
```

void k4() //비 바라기

```
int damage[5];
int percentage;
int damage_total = 0;
int i, combo = 0;
```

```
printf("꼬부기의 비 바라기 공격!\n");
scanf("%c", &next1);
```

```
rand((unsigned)time(NULL));
percentage = 1 + rand() % 100;
```

```
turn_mory_character();
```

```
for (i = 0; i < 5; i++)
```

```
{
    if (percentage <= (80 - i * 10))
    {
        damage[i] = 1;
        combo = 15 + i;
        damage_total += damage[i] * combo;
    }
    else
    {
        damage[i] = 0;
        break;
    }
}
```

```
printf("%d회 콤보 공격 성공!\n", i);
printf("%s는(은) 총 %d의 데미지를 입었다!\n", turn_character, damage_total);
scanf("%c", &next1);
HP -= damage_total;
```

l2()함수, k4()함수

-5번 연속으로 공격

BUT 연속 공격에 실패할 경우 그대로 공격
멈춤(콤보 공격)-공격할 때마다 percentage(공격 성공 확
률)은 일정하게 낮아짐-공격할 때마다 damage는 일정하게 높아
짐-총 damage는 실패한 공격이 있기 전까지
damage들의 총합



C



```
void l4() //파이리의 객기(자신의 상태가 마비, 화상 상태일 때 기술의 위력이 2배)
{
    int damage, percentage;
    printf("파이리의 객기 공격!\n");
    scanf("%c", &next1);

    srand((unsigned)time(NULL));
    damage = 40;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 80)
    {
        if (status == 0)
        {
            printf("%s는 %d의 데미지를 입었다!\n", turn_character, damage);
            scanf("%c", &next1);
            HP -= damage;
        }
        else
        {
            printf("%s는 %d의 데미지를 입었다!\n", turn_character, damage * 2);
            scanf("%c", &next1);
            HP -= damage * 2;
        }
    }
    else
    {
        printf("그러나 빛나갔다!\n");
        scanf("%c", &next1);
    }
}
```

percentage(기술 성공 확률)은 일정
damage도 일정

l4()

-자신의 상태가 0(상태 이상 없음)이면
damage

-자신의 상태가 1(상태 이상 있음)이면
damage*2

Piri_turn()함수에서 호출한 함수[l3()]이
기 때문에 다시 current_status()함수와
save_current_status()함수를 호출할
필요가 없음(이미 호출되어 있기 때문)



C



```
void k2() //꼬부기의 소금물
{
    int damage, percentage;
    printf("꼬부기의 소금물 공격!\n");
    scanf("%c", &next1);

    damage = 30;
    percentage = 1 + rand() % 100;

    turn_mory_character();

    if (percentage <= 80) //80%
    {
        if (your_HP <= 125)
        {
            printf("%s는 %d의 데미지를 입었다!\n", turn_character, damage * 2);
            scanf("%c", &next1);
            HP -= damage * 2;
        }
        else
        {
            printf("%s는 %d의 데미지를 입었다!\n", turn_character, damage);
            scanf("%c", &next1);
            HP -= damage;
        }
    }
    else
    {
        printf("그러나 빗나갔다!\n");
        scanf("%c", &next1);
    }
}
```

k2() -

percentage(기술 성공 확률)은 일정

damage도 일정

turn이 0이고 상대방의 HP가 총 HP의 절반 이하 일 때, damage는 2배

turn이 1이고 사용자의 HP가 총 HP의 절반 이하일 때, damage는 2배

kobuk_turn()함수에서 호출한 함수 [k2()]이기 때문에 다시

turn_mory_HP()함수와 save_mory_HP() 함수를 호출할 필요가 없음 (이미 호출되어 있기 때문)